SCHEDULING AND RESOURCE ALLOCATION FOR CLOUDS: NOVEL ALGORITHMS, STATE SPACE COLLAPSE AND DECAY OF TAILS

BY

QIAOMIN XIE

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Doctoral Committee:

Assistant Professor Yi Lu, Chair
Professor Bruce Hajek
Professor R. Srikant
Professor Pramod Viswanath

ABSTRACT

Scheduling and resource allocation in cloud systems is of fundamental importance to system efficiency. The focus of this thesis is to study the fundamental limits of the scheduling and resource allocation problems in clouds, and design provably high-performance algorithms.

In the first part, we consider data-centric scheduling. Data-intensive applications are posing increasingly significant challenges to scheduling in today's computing clusters. The presence of data induces an extremely heterogeneous cluster where processing speed depends on the task-server pair. The situation is further complicated by ever-changing technologies of networking, memory, and software architecture. As a result, a suboptimal scheduling algorithm causes unnecessary delay in job completion and wastes system capacity. We propose a versatile model featuring a multi-class parallel-server system that readily incorporates different characteristics of a variety of systems. The model has been studied by Harrison, Williams and Stolyar, respectively. However, delay optimality in heavy traffic with unknown arrival rate vectors has remained an open problem. We propose novel algorithms that achieve delay optimality with unknown arrival rates. This enables the application of proposed algorithms to data-centric clusters. New proof techniques are required including construction of an ideal load decomposition. To demonstrate the effectiveness of the proposed algorithms, we implement a Hadoop MapReduce scheduler and show that it achieves more than 10 times improvement over existing schedulers.

The second part studies the resource allocation problem for clouds that provide infrastructure as a service, in the form of virtual machines (VMs). Consolidation of multiple VMs on a single physical machine (PM) has been advocated for improving system utilization. VMs placed on the same PM are subject to resource "packing constraint," leading to stochastic dynamic bin packing models for the real-time assignment of VMs to PMs in a data center.

Due to finite-sized pools of servers, incoming requests might not be fulfilled immediately, and such requests are typically rejected. Hence a meaningful metric in practice is the blocking probability for arriving VM requests. We analyze the power-of-d-choices algorithm, a well-known stateless randomized routing policy with low scheduling overhead. We establish an explicit upper bound on the equilibrium blocking probability, and further demonstrate that the blocking probability exhibits distinct behaviors in different load regions: doubly-exponential decay in the heavy-traffic regime and exponential decay in the critically loaded regime.

To my parents, brothers, sisters-in-law and nieces, for their love and support.

ACKNOWLEDGMENTS

I would like to thank many people for their support over the course of my graduate school. I owe my deepest gratitude to my advisor Yi Lu, for her continuous support and encouragement. By a fortunate coincidence, I got the opportunity to work with her. She has always been ready for discussions and tackling problems together with me. This thesis would not have been completed without her guidance. I also want to thank her for her advice on career paths and life.

I am also grateful to all the faculty members of the Coordinated Science Lab. The collaboration with Professor Srikant led to the second part of this thesis on resource allocation in clouds. I have been fortunate to join Professor Bruce Hajek's group meeting, where I got the opportunity to expand my research interests. I also want to thank him for his thoughtful comments and advice. I would like to thank Professor Pramod Viswanath for serving on my committee. I am also thankful to Professor Grace Gao for her advice on an academic career.

I have been fortunate to interact with many brilliant people at UIUC. Thanks are due to my group members and officemates, past and current, for their useful advice and feedback for my many practice talks and research discussions. I also want to thank my research collaborators, Xiaobo Dong, Mayank Pundir, Cristina Abad and Ali Yekkehkhany.

I would like to thank my many friends who made my graduate life at UIUC better. In particular, I want to thank Peibei Shi, an old friend that I have known since our freshman year at Tsinghua University, for her encouragement, advice and support along the journey. I am also thankful to my dearest friends, Jiangmeng Zhang and Feini Zhang, for their precious friendship.

Last but not least, I wish to thank my parents, brothers, sisters-in-law and nieces, for their love and endless support. They always encourage me to do what I want and pursue a career path I love.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Cloud computing is becoming an essential resource for endeavors in all aspects, including healthcare, education and science. With the large-scale infrastructure provided by data centers, an increasing variety of services are now hosted on clouds, exemplified by search, online social networking, e-commerce, video streaming, database services and storage services [1, 2, 3, 4].

Over the past decade, data centers that host cloud computing services have grown significantly in size, containing tens to hundreds of thousands of commodity servers. Advanced technologies of networking, memory and software architecture have greatly improved the computing capability of these systems. Techniques like containerization and visualization are employed to support resource sharing across multiple tenants. However, despite these efforts, the performance of data centers is far from optimal. For instance, data centers are still operated at quite low utilization, typically in the range of $10 - 30\%$ [5, 6].

Various factors account for low performance of data centers. In particular, as an important component of the software stack, the scheduling and resource allocation mechanism affects the system performance substantially. The scheduling and resource allocation mechanism determines *when* and *which* resources from a shared resource pool (e.g., servers, storage and services) are allocated to each incoming job. Suboptimal resource management could be extremely wasteful with throughput, cause long delays, consume an unnecessarily large amount of power and lack robustness. Hence it is imperative to design an optimal scheduling and resource allocation mechanism.

The metrics used to evaluate the performance of a scheduling algorithm vary a lot across cloud systems that provide different services. In addition, cloud services exhibit diverse characteristics, which pose significantly different challenges on scheduling and resource allocation. The focus of this thesis is to study the fundamental limits of the scheduling problem in various cloud

systems, and design provably high-performance algorithms.

1.1 Two Typical Scheduling Problems in Clouds

A variety of systems have been designed and built to provide cloud services, and most of them fall into one of the following two main types: *data-centric* clusters, and clusters providing *infrastructure* as a service. For the first type, task scheduling is imposed with *data-locality* constraint. The large-scale data stored in the underlying distributed file system results in an extremely heterogeneous cluster, where the processing rate depends on the task-server pair. An example is the map task scheduling in the MapReduce framework [7]. For the second type, incoming jobs are resource requests submitted by customers, in the form of virtual machines. The system must make resource allocation decisions in a way that satisfies the resource requirements of incoming jobs. In this thesis, we study scheduling problems in the two types of systems.

1.1.1 Data-centric Scheduling

Data-parallel applications have become prevalent for processing large data sets from online social networks, search engines, scientific research and healthcare industry. MapReduce [7] pioneered the model, while systems like Dryad [8] and Map-Reduce-Merge [9] generalized the types of data flow.

The key difference between the cloud systems for data-parallel applications and a traditional cluster is the concept of "moving computation to data." Fetching data over a storage network, as in the approach of small data, becomes extremely inefficient for large-scale data processing, since the gigantic size of the data can cause long delay and produce an excessive amount of traffic in the network. In data-intensive clouds, computing tasks are moved to data, which are stored in the distributed file system that co-exists with the computing cluster (e.g., GFS [10] for Google's MapReduce, and HDFS [11] for Hadoop).

To facilitate parallel computation, data files stored on distributed file systems like HDFS are divided into data blocks of fixed size. Each data chunk is replicated on a few nodes to guard against single-machine and rack failures. Accordingly, each job is broken into tasks. For instance, there is one map

task per data block for MapReduce jobs. The distributed nature of data induces an extremely heterogeneous cluster, as data-processing tasks consume different amounts of time and resources at different locations. Even with the increase in the speed of data center networks, there remains a significant variability in average processing rate [12, 13, 14], depending on whether the data reside in memory, on a local disk, in a local rack, in the same cluster or in a different data center.

As the processing speed depends on the task-server pair, scheduling in data-intensive clouds is an affinity scheduling problem [15, 16, 17, 18]. In particular, for data-centric scheduling, the class of tasks is determined by the locations of requested data. The current practice in a MapReduce cluster is to place three replicas of each data chunk in three uniformly sampled servers. This makes the number of types *cubic* in the number of servers in a cluster, which itself can be as large as tens of thousands. On the other hand, each server only provides a few different processing speeds due to the multi-level locality.

Moreover, data placement and skewness of data popularity [13] results in a random load distribution on a system, which makes scheduling in data-centric clouds a fundamentally different problem from that in other large-scale clusters. Many locality-aware scheduling algorithms have been proposed [12, 19, 20, 21, 22]. However, they are not designed to be robust to variation in load and data configurations.

One goal of this thesis is to study the fundamental limits of the scheduling problem in data-intensive clouds, and design provably high-performance algorithms. The two essential criteria used to evaluate the performance of a scheduling algorithm here are throughput and delay. Throughput is equivalent to the efficiency and robustness of the system, and delay is equivalent to the completion time of tasks. Any other criterion, such as data locality or a cost function involving data transfer and waiting time, is meaningful only when translated into long-term throughput and delay in a stochastic environment. We are interested in designing a throughput and delay optimal scheduling algorithm for data-intensive clusters.

1.1.2 Resource Allocation for IaaS Clouds

The popularity of cloud services, particularly infrastructure as a service (IaaS), has increased drastically, due to the two premises offered by cloud computing: resource flexibility and cost efficiency. Customers can scale up and down computing resources they use in real time according to the needs of their applications, and only pay for resources they have actually used. Moreover, customers can access large-scale computing resource at a much lower cost, without setting up and maintaining local infrastructure. A variety of commercial cloud systems are available, like Amazon Web Service [1], Google AppEngine [2], Rackspace [3] and Microsoft Azure [4]. A growing number of enterprise workloads are being moved to clouds, including large-scale services that rely on thousands of servers, such as video streaming for Netflix [23].

For cloud computing systems that provide IaaS, customers submit requests for computing resource in the form of virtual machines (VMs). Each request specifies the amount of physical resources it needs in terms of processor power, memory, I/O bandwidth, disk, etc. The cloud provider allocates computing resource from a large pool of servers according to customers' requirements. In particular, consolidation of multiple VMs on a single physical machine (PM) has been advocated for improving system utilization. An important design issue of such systems is the resource allocation problem: when a user submits a VM request, which physical server(s) should be selected to accommodate the request?

Due to the finite size of server pools, incoming VM requests might not be fulfilled immediately, and such requests are typically rejected [4]. To data center operators, loss of customers means loss of revenue. Hence an important metric in practice is the loss rate of incoming VM requests. There has been a significant amount of work on design issues associated with such systems [24, 25, 26, 27, 28, 29, 30, 31]. In particular, the resource allocation problem has been well studied [26, 32, 28, 33]. However, existing work has not considered the loss rate of incoming VM requests. The goal here is to understand how to route incoming resource requests to PMs in order to minimize the loss rate, i.e., minimize the probability that an arriving VM request does not find the required amount of resources at the selected PM(s).

1.2 Contributions

Scheduling and resource allocation in clouds is of fundamental importance to system efficiency. The objective of this thesis is to explore the fundamental limits of the scheduling and resource allocation problems in clouds, and design provable high-performance algorithms.

To achieve this goal, we take the stochastic model-based approach, assuming there is randomness in the request arrivals and also in the processing time of a request. We use stochastic analysis to characterize the performance of scheduling and resource allocation algorithms rigorously, and also verify high performance of the proposed algorithms via implementation. In this thesis, we focus on two typical scheduling problems in clouds. Below, we provide a brief overview of our contributions towards each problem.

Data-centric scheduling. The presence of data produces an extremely heterogeneous cluster where processing speed depends on the task-server pair. The situation is further complicated by ever-changing technologies of networking, memory, and software architecture. The data-locality problem poses new challenges to scheduling in today's computing clusters. While many locality-aware scheduling algorithms have been proposed in the literature and implemented in practice, most of the existing approaches are not robust to changes in load or skewness of data popularity, and their fundamental throughput and delay properties are unknown.

We propose a versatile model featuring a multi-class parallel-server system that readily incorporates different characteristics of a variety of data-intensive clusters. The model had been studied by Harrison [16, 17], Williams [34, 18] and Stolyar [35], respectively. However, delay optimality in heavy traffic with unknown arrival rate vectors has remained an open problem.

We propose a simple priority algorithm called Pandas for systems with two-level data locality, and a balanced priority algorithm called balanced-Pandas for systems with multi-level locality. We establish throughput and heavy-traffic delay optimality for both algorithms. The main challenge is the construction of a novel *ideal load decomposition* that allows the separate treatment of different subsystems.

We implement Pandas in Hadoop clusters. Trace-driven experiments on Hadoop show that Pandas accelerates the data-processing phase of jobs by 11 times with hot-spots and 2.4 times without hot-spots over existing sched-

ulers. When the difference in processing times due to location is large, such as applicable to the case of memory-locality, the acceleration by Pandas is 22 times. The proposed approach is broadly applicable to all data-parallel applications, and can be integrated with job-level sharing policy (e.g., priority, fairness and capacity), straggler mitigation and other optimization objectives.

Resource allocation for IaaS clouds. VMs placed on the same PM are subject to a resource "packing constraint," leading to stochastic dynamic bin packing models for the real-time assignment of VMs to PMs in a data center. In particular, incoming resource requests that cannot be fulfilled immediately are rejected. This setting motivates us to consider a loss model. We analyze the power-of-d-choices algorithm, a well-known *stateless* randomized routing policy that fits for distributed scheduling. We consider a fluid model that corresponds to large system limit. We establish an explicit upper bound on the equilibrium blocking probability and further demonstrate that the blocking probability exhibits distinct behaviors in different load regions: *doubly-exponential* decay in the heavy-traffic regime and *exponential* decay in the critically loaded regime. The techniques developed may be applicable to other distributed resource allocation mechanisms.

1.3 Thesis Organization

The rest of this thesis is organized as follows. Chapter 2 provides an exposition of the scheduling with data locality problem, and describes a versatile stochastic model that incorporates different characteristics of various data-centric clusters. In Chapter 3, we focus the scheduling problem for the computing cluster without rack structure. We propose Pandas, which is the only known algorithm that is both throughput-optimal and delay-optimal in the heavy-traffic regime without knowing job arrival rates. We investigate the scheduling problem with multi-level locality in Chapter 4. We present balanced-Pandas and show that balanced-Pandas achieves throughput and heavy-traffic optimality simultaneously. In Chapter 5, we study the online assignment of VMs to PMs via a stochastic bin-packing model. Finally, Chapter 6 concludes this thesis.

CHAPTER 2

SCHEDULING WITH DATA LOCALITY

In this chapter we introduce the problem of scheduling with *data locality*, summarize previous work, and describe a versatile model featuring a multi-class parallel-server system that readily incorporates different characteristics of a variety of data-intensive clusters.

2.1 The Problem

Data-parallel applications have become increasingly popular for processing large data sets. A fundamental problem to all data-parallel applications is scheduling with *data locality*, as data-processing tasks consume different amounts of time and resources at different locations. Even with the increase in the speed of data center networks, there remains a significant difference in average processing speed [12, 13, 14] depending on whether the data reside in memory, on a local disk, in a local rack, in the same cluster or in a different data center. As the processing rate depends on the task-server pair, it is an affinity scheduling problem [15, 16, 17], albeit with two unique features:

1. A few different processing rates due to the multi-level locality.

Multi-level locality exists within and across data centers, where the average task processing rate depends on the corresponding data location, whether it is in memory, on local disks, in a local rack, in the same cluster or in different data centers. In particular, running tasks on a server that caches the data in the memory is the most efficient. When the server does not have the data chunk for a particular task in memory or on the local disk, the data needs to be retrieved over the network before processing. Such tasks will be processed at a slower rate, reflecting the potential variation in network transfer when retrieving data from different locations. For instance, it is measured in [12] that running task on a server with data off-rack on average takes twice as

much time as running on a server with data on the local disk.

From the perspective of each task, the cluster is divided into a few subsets, where servers within a subset provide the same processing rate. Therefore, not only is the system not homogeneous, it is not heterogeneous in the traditional sense as there are no fixed sets of fast and slow servers. Instead, each server provides different processing rates for different tasks.

2. An explosive number of task types.

The type of a particular task is determined by the location of its data chunks. To achieve high availability and yet avoid excessive amount of storage, each data chunk is typically replicated on a small number of servers. For instance, the number is 3 for the MapReduce cluster.

When a server breaks down and its disks are replaced, all its local data chunks are restored by copying from their respective replicas. To avoid excessive traffic from any single server, which can disrupt its service, it is desirable to distribute the replicas over a large number of servers. Hence, the current practice in a MapReduce cluster is for each data chunk to uniformly sample three servers. This makes the number of types *cubic* in the number of servers in a cluster, which itself can be as large as tens of thousands. As a result, it is impractical to have one queue for each type of task.

We are interested in designing a throughput and delay optimal scheduling algorithm in a system with multi-level locality. We define the optimality criteria as follows.

Throughput Optimality: An algorithm is throughput optimal if it can stabilize the system when some algorithm can. Throughput optimality means that an algorithm is robust so that whichever load vector strictly within the capacity region is imposed, the system is efficient enough to achieve a finite task completion time. According to Little's law, this implies that the number of tasks queueing in the system is finite, i.e.,

$$\limsup_{t\to\infty} \mathbb{E}\left[\sum_m Q_m(t)\right] < \infty.$$

However, throughput optimality does not guarantee the task completion time to be small.

Heavy-traffic Delay Optimality: It is often difficult to analyze delay, or task completion time, in a stochastic environment. One way to assess whether

an algorithm is competent is to look at the heavy-traffic regime, where the load vector approaches the boundary of the capacity region and the system becomes critically loaded. An algorithm is heavy-traffic optimal if it asymptotically minimizes the average delay as the arrival rate vector approaches the boundary of the capacity region; hence, it is *efficient* under stressed conditions.

In addition, the algorithm should not assume any knowledge of arrival rates in order to be robust with load variations.

2.2 Related Work

Affinity scheduling. There is a large body of work on affinity scheduling for a multi-class parallel server system [15, 16, 17, 35, 34, 18], where the service rate depends on the job class and server node pair. Harrison [16] considered a two-type two-server model, and proposed a discrete review control policy, where system status is reviewed at an interval of fixed length. At the beginning of each review period, the scheduler makes assignments so as to minimize holding cost associated with unallocated jobs at the end of the review period. The discrete-review policy is shown to asymptotically optimize linear holding cost. Harrison and Lopez [17] extended the discrete-review policy to a general parallel-server system, conjecturing its asymptotic optimality in the heavy traffic regime. Bell and Williams [18] established asymptotic optimality of a continuous-review "tree-based" threshold policy with linear holding cost. In particular, it requires knowledge of the arrival rates to solve an optimization problem, which identifies the tree-structure of a graph containing the servers and job classes as nodes.

Stolyar [36] considered a generalized switch model, and showed that the MaxWeight policy asymptotically minimizes the holding costs that are linear combinations of each queue length to the power $\beta+1$ with $\beta > 0$. No knowledge of the arrival rates is needed. Mandelbaum and Stolyar [35] proposed a generalized $c\mu-$rule for the parallel server system. The $c\mu-$rule is shown to achieve asymptotic optimality, for increasing and convex holding cost. Again, it does not require the arrival rate information. However, it is not optimal for linear holding costs in the heavy-traffic regime. Studies [15, 16] have shown that $c\mu-$rule might even result in system instability.

The existing work on affinity scheduling requires a queue for each type of task; hence, it is impractical for this setting as the typical number of task type scales cubically with the number of servers. In addition, previous work either requires the knowledge of the arrival rates of different task types [16, 17, 34], or optimizes a specific function of delay [35, 36], which is *not* delay-optimal in general in the heavy-traffic regime.

Locality-aware scheduling. Among the existing locality-aware scheduling algorithms, the work most closely related to ours is delay scheduling in HFS [12] and JSQ-MaxWeight [37]. HFS focuses on the conflict between data locality and fairness among jobs. While fairness is a job-level priority, delay scheduling is a task-level algorithm that specifies the priority among map tasks based on their data location. However, delay scheduling makes assumptions that may not hold universally: (a) task durations are short and bimodal, and (b) a fixed waiting time parameter works for all loads and skewness of traffic. These assumptions make it difficult for delay scheduling to adapt to changes in workload, network conditions, or node popularity. In contrast, our approach makes no assumption on task durations and is provably robust to the aforementioned changes. Our approach is readily integrated with the fairness part of HFS, as demonstrated in next chapter.

Wang et al. [37] were the first to formulate the scheduling problem with two levels of data-locality from a stochastic network perspective and identified its capacity region. They proposed a scheduling algorithm consisting of the Join the Shortest Queue (JSQ) together with the MaxWeight policy. The JSQ step distributes the load into local and remote queues: a task is preassigned as remote if its local queues are longer than the remote queue. The MaxWeight step stabilizes the queues with a threshold-based priority policy. The JSQ-MaxWeight algorithm was shown to be throughput-optimal. However, it was shown in [37] that it is heavy-traffic optimal only for a very special traffic scenario, where all traffic concentrates on a subset of servers. In particular, some servers receive *zero* local tasks and only provide remote service; and any server with non-zero local tasks is overloaded (with load exceeding 1) and *requires* remote service as a result. Recently they proposed a decentralized scheduling algorithm, based on back-pressure approach, for data-parallel computation on peer-to-peer networks [38]. The proposed algorithm is shown to be throughput optimal, but its delay performance is not

known.

Other locality-aware algorithms include Quincy [19], Bar [20], Maestro [21] and Matchmaking [22]. Like HFS, Quincy [19] has a task-level algorithm that works with the fairness job priority. In particular, at each task arrival and departure, Quincy solves a min-cost flow problem that optimizes a linear combination of data bytes transferred, with a penalty for an unscheduled task and for killing a running task. The optimization is greedy at each step, does not consider the stochastic arrivals of jobs, and does not translate into optimal job completion times over a long horizon. Bar [20] assumes that all jobs have the *same* task execution time in its optimization of makespan. Maestro [21] assumes the knowledge of the number of data blocks on each node *to be processed* in the future, and assumes that each data block is processed exactly once. Matchmaking [22] avoids tuning the waiting time parameter of HFS by making each node wait exactly one heartbeat interval before acquiring a remote task. However, this fixed waiting time still makes it difficult to adapt to skewed node popularity and varying loads.

In addition, there is work focusing on locality and virtual machines (VMs). The ILA scheduler [39] adds a new level of locality due to co-locating VMs on the same node and makes the waiting time of delay scheduling [12] proportional to the data size, which can be smaller than the maximum data block size of 128 MB. However, it is not clear whether ILA's setting of parameter is optimal. Purlieus [40] couples data placement and VM placement, but does not consider task assignment, and hence is complementary to our work.

Data placement. Several data placement techniques have been used to improve locality. Scarlett [13] adopts a proactive replication scheme that periodically replicates files based on predicted data popularity. It focuses on data that receives at least three concurrent accesses. However, it does not consider *node popularity* caused by co-location of moderately popular data, which can be solved by our approach. DARE [41] adopts a reactive approach that probabilistically retains remotely retrieved data and evicts aged replicas. Its reactive nature makes its performance depend on appropriate and timely remote services. As our approach serves the *right* remote tasks, it will be a valuable complement to DARE. PACMan [14] caches data in memory to improve job completion times. Our approach can be readily extended to include memory locality in scheduling to reap the benefit of cached data.

Scheduling MapReduce jobs. There has been much recent work on scheduling MapReduce jobs, including improvement of shuffle phase [42, 43, 44], joint scheduling of shuffle and reduce phase [45, 46], joint scheduling of map and reduce phase [47, 48], straggler mitigation [49, 50, 51], and optimization of job schedules to minimize average response time and makespan [52, 53, 54, 55, 56]. None of these algorithms considers locality of map tasks, and hence they are orthogonal to our approach. Our approach is designed to work with various job-level and phase-level priorities by assigning the *optimal* data-processing task when the job-level algorithm wants *a* data-processing task assigned.

Application-specific scheduling. KMN [57] improves data locality of jobs using a sampled subset of their data. The availability of multiple choices allows it to avoid localized hot-spots in the system. However, as hot-spots caused by node popularity can propagate to a significant fraction of the system due to replicas sharing a workload, our approach will be complementary to KMN during the propagation of hot-spots.

2.3 System Model

We describe our model in the context of MapReduce clusters, but it is applicable to other computing systems, where tasks with data at different locations can be modeled using different processing rates that depend on the locality of the data.

We consider a MapReduce system with a hierarchical network. The system consists of racks, each of which contains multiple servers. Servers within a rack share a common switch. A large data set is divided into blocks, each of which is replicated on a few servers for fault tolerance and performance. A job consists of a number of map tasks, each of which processes a different data chunk. For each task, we call a server a *local server* for the task if the data block to be processed by the task is stored locally, and we call this task a *local task* for the server. Analogously, we call a server a *rack-local server* if the data block to be processed by the task is not stored on the server, but in the same rack as the server, and we call this task a *rack-local task* for the server. A server is a *remote server* if it is neither local nor rack-local for the task and this task is called a *remote task* for the server.

We consider a computing cluster that consists of K racks indexed by $k \in \mathcal{K}$, where $\mathcal{K} = \{1, 2, \cdots, K\}$. There are M parallel servers in the system, indexed by $m \in \mathcal{M}$, where $\mathcal{M} = \{1, 2, \cdots, M\}$. For each server m, we denote by $K(m)$ the index of the corresponding rack where it locates. The cluster is modeled as a time-slotted system, in which tasks arrive at the beginning of each time slot according to some stochastic process. Each data chunk is replicated on a set $\bar{L}$ of servers. As each task processes one data chunk, it has $|\bar{L}|$ local servers. Define the *type* of a task as the set $\bar{L}$ of its local servers. For instance, with $|\bar{L}| = 3$ the task type $\bar{L}$ is defined as:

$$\bar{L} \in \{(m_1, m_2, m_3) \in \mathcal{M}^3, m_1 < m_2 < m_3\},$$

where m_1, m_2, m_3 are the indices of the three local servers.

We use $m \in \bar{L}$ to denote that server m is a local server for type $\bar{L}$ tasks. We use the notation $m \in \bar{L}_k$ if server m is rack-local to type $\bar{L}$ tasks, and similarly, $m \in \bar{L}_r$ if server m is remote to type $\bar{L}$ tasks. Let $\mathcal{L}$ denote the set of task types.

Arrivals. Let $A_{\bar{L}}(t)$ denote the number of type $\bar{L}$ tasks that arrive at the beginning of time slot t. We assume that the arrival process of type $\bar{L}$ tasks is i.i.d. with rate-$\lambda_{\bar{L}}$. We denote the arrival rate vector by $\boldsymbol{\lambda} = (\lambda_{\bar{L}} : \bar{L} \in \mathcal{L})$. The number of total arrivals in one time slot is assumed to be bounded, i.e., $\sum_{\bar{L} \in \mathcal{L}} A_{\bar{L}}(t) \leq C_A$.

Services. For each task, we assume that its service time follows a geometric distribution with mean $1/\alpha$ if processed at a local server, and with mean $1/\beta$ and $1/\gamma$ at a rack-local server and a remote server respectively. On average, a task is processed fastest at a local server, and slowest at a remote server, hence we assume $\alpha > \beta > \gamma$. Each server can process one task at a time and all services are non-preemptive.

CHAPTER 3

PANDAS: PRIORITY ALGORITHM FOR NEAR-DATA SCHEDULING WITH TWO-LEVEL LOCALITY

In this chapter, we consider the scheduling problem for the computing cluster without rack structure. The discrete-time model described in Chapter 2 will be simplified with two levels of locality: within each time slot, a task is completed with probability α at a local server, or with probability γ ($\gamma < \alpha$) at a remote server. The formulation is the same as previous work by Wang et al. [37]. We consider two traffic scenarios that require distinct proof techniques (although our algorithm does not distinguish between them as we assume no knowledge of arrival rates):

Evenly loaded. This is the case where with appropriate load balancing, each server can accommodate its load locally. No remote service is necessary in this scenario.

Locally overloaded (hotspots). More often, the data requested by the incoming traffic are skewed towards a subset of servers [13] and exceed their capacity. We call these servers *beneficiaries* as they require remote service to remain stable, and call the servers with spare capacity *helpers*. This includes the special scenario in [37] for which the JSQ-MaxWeight algorithm is shown to be heavy-traffic optimal, and is more general as it allows non-zero local traffic at helpers, as well as traffic that is local to both a helper and a beneficiary.

We propose Pandas (Priority Algorithm for Near-DAta Scheduling) which is a *task-level* algorithm that specifies the priority among tasks of any data-processing phase by considering data locality. Pandas consists of two main steps:

1. Early detection of hot-spots: While early detection is highly desirable for relieving hot-spots, it is not straightforward. As each data block has multiple replicas, it is incorrect to estimate the traffic at a node by simply summing all workloads whose data reside on this node. Pandas accurately

detects a hot-spot before it causes excessive delay by monitoring a queuing structure with appropriate load balancing.

2. Serve the right remote task: Timely service of remote tasks, that is, tasks whose data need to be fetched over the network, by lightly loaded nodes helps relieving hot-spots. However, not all tasks are equal. Pandas ensures that only tasks contributing to potential hot-spots are served remotely.

We establish the following results:

- We prove that Pandas is throughput optimal, i.e., it can stabilize any arrival rate vector strictly within the capacity region identified in [37]. Since the algorithm has a predetermined priority of "local-tasks first," existing techniques using the L_2 norm Lyapunov drift, such as in [37], do not apply: There exist states with arbitrarily large L_2 norm where the drift remains positive. The main idea is the construction of the *ideal load decomposition* for each arrival vector, which separates the servers into helpers and beneficiaries. The stability of the helper subsystem (which by itself is not Markovian) is established first, and the spare capacity helps stabilize the beneficiary subsystem.

- In addition, we prove that Pandas is heavy-traffic optimal for both the evenly loaded and locally overloaded scenarios, i.e., it asymptotically minimizes the average delay as the arrival rate vector approaches the boundary of the capacity region. Since [37] shows heavy-traffic optimality only for a special traffic scenario, Pandas is so far the only known heavy-traffic optimal algorithm. Further, to the best of our knowledge, this is the only setting of affinity scheduling where a "local-tasks first" algorithm is shown to be heavy-traffic optimal, which can be of separate interest.

  The locally overloaded case is the more challenging of the two. The proof first establishes *state-space collapse*, where we show that the helper subsystem has uniformly bounded moments independent of the heavy-traffic parameter, and the beneficiary subsystem reduces to a single dimension where all queue lengths are equal. We remark that this result depends on our "local-tasks first" policy as the helper queues are drained first, *independent* of the beneficiaries. In contrast, JSQ-MaxWeight results in helper queues growing proportionally with the

beneficiaries. The proof uses construction of *ideal* processes to bound the dependence between helpers and beneficiaries through shared local arrivals and remote services.

- We have integrated Pandas with the Hadoop FIFO scheduler and Fair scheduler (HFS). Each scheduler retains its original job priority. To focus on the performance benefit brought by Pandas to the data-processing phase, we use the SWIM workload [58] to obtain realistic characteristics of data-processing tasks, but with empty reduce phases, as the time taken by the reduce phase can be orthogonally improved by other techniques [45, 46].

  We evaluate Pandas in a variety of environments including Amazon's Elastic Compute Cloud (EC2), a private cluster and via large-scale simulations. Pandas-accelerated FIFO scheduler achieves 11-fold improvement in average job completion time with hot-spot and 2.4-fold improvement without hot-spot over the Hadoop FIFO scheduler. When the difference in processing times due to location is large as in the case of memory-locality, Pandas-accelerated Fair scheduler achieves 22-fold improvement over HFS during a hot-spot.

3.1 Algorithm

Pandas detects hot-spots early by estimating the expected amount of contention at a node. It maintains a queuing structure to identify a subset of nodes that have the potential to become hot-spots via load balancing. Section 3.1.1 describes the details of hot-spot detection. The identification of potential hot-spots allows appropriate decisions of whether to serve a remote task and which remote task to serve. This enables timely relief of hot-spots without assigning too many remote tasks and sacrificing system throughput. Section 3.1.2 describes the details of task assignment.

3.1.1 Early Detection of Hot-spots

While it is easy to detect a highly popular file by simply counting the number of requests for this file, detecting a highly popular *node* is less straightforward.

As each data block has multiple replicas and task processing time varies, it is not known a priori where a task should be processed.

Queuing structure. Pandas maintains a queuing structure within the scheduler as illustrated in Fig. 3.1. The queueing structure contains M queues, where the m-th queue, denoted by Q_m, only receives tasks local to server m. We call it a local queue for tasks of type $\bar{L}$ if $m \in \bar{L}$. Note that there can be tasks local to server m but buffered at Q_n, $n \neq m$, where server n is another local server for the tasks. Let the vector $\mathbf{Q}(t) = (Q_1(t), Q_2(t), \cdots, Q_M(t))$ denote the queue lengths at time t.

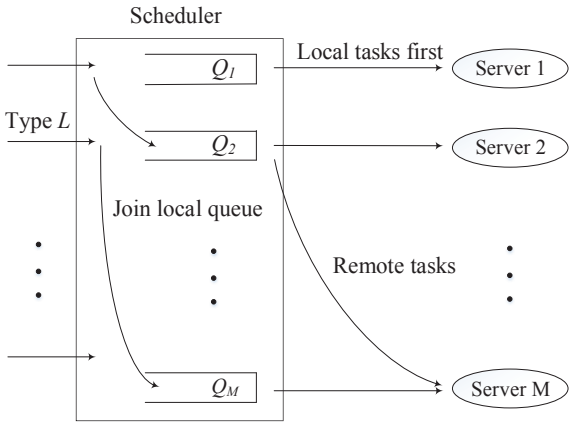


Figure 3.1: The proposed algorithm.

Load balancing: When a task arrives, the scheduler compares the lengths of the task's local queues, $\{Q_m | m \in \bar{L}\}$, and inserts the task into the shortest queue. Ties are broken randomly. Load balancing is an estimate of where each task should be processed based on the *expected* amount of load on each node, since the exact processing time of each task is unknown. The purpose of load balancing is not to permanently assign a task to a node. Rather the purpose is to push away concurrent tasks from a potential hot-spot to other local nodes with less contention, if such nodes exist.

3.1.2 Serve the Right Remote Task

Pandas decides on the sequence of task assignments for idle servers via *prioritized scheduling* as follows:

Local tasks first. When server m becomes idle, the scheduler sends the head-of-line task from Q_m.

Remote tasks. When server m becomes idle and Q_m is empty, the scheduler sends a remote task to server m from the longest queue in the system, if the length of the longest queue, denoted by Q^{max}, exceeds the threshold $T_s = \alpha/\gamma$. The threshold is to ensure that the remote task will experience a smaller completion time in expectation, since the mean processing time at a remote server is $1/\gamma$, and the mean waiting time plus processing time at a local server is Q^{max}/α.

3.1.3 Queue Dynamics

Let $A_{\bar{L},m}(t)$ denote the number of type $\bar{L}$ tasks that are routed to Q_m. The total number of tasks that join queue Q_m, denoted by $A_m(t)$, is given by

$$A_m(t) = \sum_{\bar{L}:m\in\bar{L}} A_{\bar{L},m}(t).$$

We denote the working status of server m at time slot t by $f_m(t)$:

$$f_m(t) = \begin{cases} -1, & \text{if server } m \text{ is idle} \\ n, & \text{if server } m \text{ serves a task from queue } n \end{cases}$$

When server m completes a task at the end of time slot $t-1$, i.e., $f_m(t^-) = -1$, it is available for a new task at time slot t. Note that $f_m(t) = m$ indicates that server m is working on a local task, and $f_m(t) = n$, where $n \neq m$, indicates that server m is working on a remote task. The scheduling decision is based on the working status vector $\mathbf{f}(t) = (f_1(t), f_2(t), \cdots, f_M(t))$ and queue length vector $\mathbf{Q}(t)$.

Let $\eta_m(t)$ denote the scheduling decision for server m at time slot t, which is the index of the queue that server m is scheduled to serve. Note that $\eta_m(t) = f_m(t)$ for all busy servers, and when $f_m(t^-) = -1$, i.e., server m is idle, $\eta_m(t)$ is determined by the scheduler according to the algorithm.

We use $S_m^l(t)$ and $R_m(t)$ to denote the local and remote service provided by server m respectively, where $S_m^l(t) \sim Bern(\alpha I_{\{\eta_m(t)=m\}})$ and $R_m(t) \sim Bern(\gamma I_{\{\eta_m(t)\neq m\}})$ are two Bernoulli random variables with varying probability: $S_m^l(t) \sim Bern(\alpha)$ when server m is scheduled to the local queue, and

Bern(0) otherwise; $R_m(t) \sim Bern(\gamma)$ when server m is scheduled to a remote queue, and Bern(0) otherwise.

Note that the local service *received* by server m is also $S^l_m(t)$, whereas the remote service *received* by server m is $S^r_m(t) \equiv \sum_{n:n\neq m} R_n(t) I_{\{\eta_n(t)=m\}}$, which is the sum of all remote service provided by other servers to server m. Let $S_m(t) \equiv S^l_m(t) + S^r_m(t)$ denote the departure process for queue m. Hence the queue length satisfy the following equation:

$$Q_m(t+1) \;\; = \;\; Q_m(t) + A_m(t) - S_m(t) + U_m(t),$$

where $U_m(t) = \max\{0, S_m(t) - A_m(t) - Q_m(t)\}$ is the unused service.

As the service times follow geometric distributions, $\mathbf{Q}(t)$ together with the working status vector $\mathbf{f}(t)$ form a Markov chain $\{Z(t) = (\mathbf{Q}(t), \mathbf{f}(t)), t \geq 0\}$. We assume that the process is initialized as $(\mathbf{Q}(0), \mathbf{f}(0)) = (\mathbf{0}_{M\times 1}, -\mathbf{1}_{M\times 1})$. Denote the state space by $\mathcal{S} \subset \mathbb{N}^M \times \{-1, 1, 2, \cdots, ..., M\}^M$, which consists of all states that can reached from the initial state. Observe that this Markov chain is irreducible and aperiodic.

The following lemma states a property of the unused service $\mathbf{U}(t)$. It will be used in the proof of throughput and heavy-traffic optimality.

Lemma 3.1. *For any $t \geq 0$,*

$$\langle \mathbf{Q}(t), \mathbf{U}(t) \rangle \leq M^2.$$

Proof. By the definition of $U_m(t)$, $0 \leq U_m(t) \leq M$, and $\sum_{m\in\mathcal{M}} U_m(t) \leq M$. If $U_m(t) = 0$, $Q_m(t)U_m(t) = 0$. We note the fact that $U_m(t) > 0$ only if the number of tasks in Q_m is less than the number of available servers scheduled to Q_m at time t. Since $S_m(t) \leq M$, we have $Q_m(t) < M$. Hence $Q_m(t)U_m(t) < MU_m(t)$. Therefore, $\langle \mathbf{Q}(t), \mathbf{U}(t) \rangle < \sum_{m\in\mathcal{M}} MU_m(t) = M^2$. ∎

3.2 Ideal Load Decomposition

A key component of the proof of both throughput and heavy-traffic optimality is a construction we call the ideal load decomposition. It is ideal in the sense that it *minimizes* the work in the system by locally serving as many

tasks as possible. The construction serves two purposes: 1) The ideal load obtained for each server is used as an intermediary in the proofs of stability and state-space collapse; 2) The construction uniquely identifies two subsystems, helpers and beneficiaries, which have very different behavior and require distinct treatment in the proofs.

Helpers and Beneficiaries

A server is a helper if it is *not overloaded*, *provides* remote service and its local queue does *not receive* remote service under the ideal load decomposition. In contrast, a server is a beneficiary if it is *overloaded*, does *not provide* remote service, and its local queue *receives* remote service from the helpers. We will define an overloaded server in a more precise manner in 3.2.2. While pure helpers and beneficiaries do not exist in a real system, the ideal load decomposition approximately depicts the load distribution in the heavy-traffic regime.

In the rest of the section, we construct the ideal load decomposition. We start from a new definition of the capacity region, which is equivalent to that identified in [37], but uses a more refined decomposition appropriate for our algorithm. The ideal load decomposition is constructed from this refined decomposition in two steps: 1) Identify the overloaded servers; 2) Construct the decomposition that produces helpers and beneficiaries. In particular, we will show that the decomposition constructed at each step can be characterized by a linear program.

3.2.1 An Equivalent Capacity Region

Let Λ be the set of arrival rates such that each element has a decomposition satisfying the following condition:

$$\begin{aligned}\Lambda \;=\; & \big\{\boldsymbol{\lambda} = (\lambda_{\bar{L}} : \bar{L} \in \mathcal{L}) \mid \exists\ (\lambda_{\bar{L},n,m}) \text{ such that} \\ & \lambda_{\bar{L},n,m} \geq 0, \forall \bar{L} \in \mathcal{L}, \forall n \in \bar{L}, m \in \mathcal{M}, \\ & \lambda_{\bar{L}} = \sum_{n:n\in\bar{L}} \sum_{m=1}^{M} \lambda_{\bar{L},n,m}, \forall \bar{L} \in \mathcal{L}, \\ & \sum_{\bar{L}:m\in\bar{L}} \sum_{n:n\in\bar{L}} \frac{\lambda_{\bar{L},n,m}}{\alpha} + \sum_{\bar{L}:m\notin\bar{L}} \sum_{n:n\in\bar{L}} \frac{\lambda_{\bar{L},n,m}}{\gamma} < 1,\ \forall m \in \mathcal{M}\ \big\}, \quad (3.1)\end{aligned}$$

where inequality (3.1) states that the sum of the local and remote load at each server is less than 1.

Lemma 3.2. *The capacity region Λ is equivalent to the capacity region in [37].*

The proof is straightforward. In [37], the rate $\lambda_{\bar{L}}$ is decomposed into $\lambda_{\bar{L},m}$, which is the rate of type-$\bar{L}$ arrival allocated to server m. We further refine the decomposition by simply writing $\lambda_{\bar{L},m} \equiv \sum_n \lambda_{\bar{L},n,m}$, where n is the index of the queue at which a task is *queued* till processed at server m. Observe that $\lambda_{\bar{L},n,m} = 0$ if $n \notin \bar{L}$, since tasks only join their local queues with the proposed algorithm.

3.2.2 Overloaded Servers

Let $\nu_{n,m}$ denote the total rate of arrivals routed to Q_n, and eventually processed at server m, $\nu_{n,m} \equiv \sum_{\bar{L}:n\in\bar{L}} \lambda_{\bar{L},n,m}$.

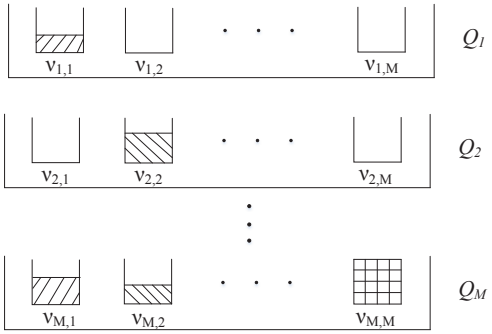


Figure 3.2: Ideal load decomposition.

Figure 3.2 illustrates $(\nu_{n,m})$ where the m-th sub-queue at Q_n denotes the arrivals routed to Q_n but processed at server m. Note that the sub-queues are only a part of the construction, and do not exist in the actual data structure. For each server n, define

$$\lambda_n^l = \nu_{n,n},\ \lambda_n^r = \sum_{m:m\neq n} \nu_{n,m},$$

which give the pseudo-arrival rate of tasks queued at Q_u and served locally by server n, and served remotely by other servers, respectively. Let $\lambda_n \equiv \lambda_n^l + \lambda_n^r$

denote the total pseudo-arrival rate at Q_n. Note that the total rate of remote service *provided* by server m is $\sum_{n:n\neq m} \nu_{n,m}$.

For any subset of servers $\mathcal{N} \subseteq \mathcal{M}$, we denote by $\mathcal{L}_{\mathcal{N}}$ the set of task types *only local* to servers in $\mathcal{N}$.

Lemma 3.3. *For any arrival rate vector $\boldsymbol{\lambda} \in \Lambda$, there exists a decomposition $\{\tilde{\lambda}_{\bar{L},n,m}\}$ which satisfies condition (3.1) and $\forall n \in \mathcal{D} = \{n \in \mathcal{M} : \sum_{\bar{L}:n\in\bar{L}} \sum_m \tilde{\lambda}_{\bar{L},n,m} \geq \alpha\}$, where $\mathcal{D}$ denotes the overloaded set with arrival rate greater than α,*

$$\tilde{\lambda}_{\bar{L},n,m} = 0, \quad \forall \bar{L} \notin \mathcal{L}_{\mathcal{D}}, m \in \mathcal{M}. \tag{3.2}$$

Note that the decomposition $\{\tilde{\lambda}_{\bar{L},n,m}\}$ is such that for the overloaded set $\mathcal{D}$, it only receives non-zero arrivals from task types that are only local to $\mathcal{D}$. In other words, any task type that is also local to some server m not in the overloaded set, will be directed to m. This ensures that the set $\mathcal{D}$ is truly overloaded as no load balancing with the rest of the system will reduce its arrivals. Note that $\mathcal{D}$ is unique for a given arrival vector λ, although the decomposition $\{\tilde{\lambda}_{\bar{L},n,m}\}$ might not be unique. When $\mathcal{D}$ is non-empty, we call the system *locally overloaded*.

The proof takes a decomposition $\left\{\lambda_{\bar{L},n,m}\right\}$ satisfying condition (3.1), and iteratively moves an appropriate amount of local arrivals from overloaded ($\lambda_n \geq \alpha$) queues to underloaded ($\lambda_n < \alpha$) queues. This is possible whenever an overloaded queue receives local arrivals that are also local to some underloaded queue. At the end of each step, either all shared local traffic between the two queues is routed to the underloaded queue, or they have both become underloaded or overloaded. At each step, the amount of arrivals moved is determined such that condition (3.1) is always satisfied.

Linear Programming Characterization. An alternative characterization of the decomposition of Lemma 3.3 is via a linear program. Observe that remote service is required to accommodate the arrivals for a server with $\lambda_n \geq \alpha$, while a server with $\lambda_n < \alpha$ can accommodate its arrivals locally. Given a decomposition $\{\lambda_{\bar{L},n,m}\}$ of $\boldsymbol{\lambda} \in \Lambda$, we define the system *load* as

$$\rho\left(\{\lambda_{\bar{L},n,m}\}\right) = \sum_{n:\lambda_n<\alpha} \frac{\lambda_n}{\alpha} + \sum_{n:\lambda_n\geq\alpha} \left(1 + \frac{\lambda_n - \alpha}{\gamma}\right), \tag{3.3}$$

which is the minimum possible lower bound on the total utilization of all servers, such that the arrivals *routed* to each server according to $\{\lambda_{\bar{L},n,m}\}$ can be accommodated. Consequently, a natural definition of *ideal routing* is a decomposition such that ρ is minimized. The linear program to determine the minimum ρ^* for $\boldsymbol{\lambda}$ is as follows. We refer to it as the *routing optimization problem:*

$$\min_{\{\lambda_{\bar{L},n,m}\}} \rho\left(\{\lambda_{\bar{L},n,m}\}\right)$$

subject to

$$\lambda_{\bar{L},n,m} \geq 0, \forall \bar{L} \in \mathcal{L}, \forall n \in \bar{L}, m \in \mathcal{M}, \tag{3.4}$$

$$\lambda_{\bar{L}} = \sum_{n:n\in\bar{L}} \sum_{m=1}^{M} \lambda_{\bar{L},n,m}, \forall \bar{L} \in \mathcal{L}, \tag{3.5}$$

$$\sum_{\bar{L}:m\in\bar{L}} \sum_{n:n\in\bar{L}} \frac{\lambda_{\bar{L},n,m}}{\alpha} + \sum_{\bar{L}:m\notin\bar{L}} \sum_{n:n\in\bar{L}} \frac{\lambda_{\bar{L},n,m}}{\gamma} < 1, \ \forall m \in \mathcal{M}. \tag{3.6}$$

As a byproduct of the proof of Lemma 3.3, it is easy to see that a decomposition satisfying condition (3.2) gives an optimal solution of the linear program. That is, when all shared type tasks are routed to underloaded queues, the corresponding system load is minimized. In particular, the proof of Lemma 3.3 provides a procedure to construct an optimal solution.

3.2.3 Ideal Load Decomposition

Lemma 3.4. *For any arrival rate vector $\boldsymbol{\lambda} \in \Lambda$, there exists a decomposition $\{\lambda^*_{\bar{L},n,m}\}$ satisfying condition (3.1) and for $\forall m \in \mathcal{M}$, either $m \in \mathcal{H}$ or $m \in \mathcal{B}$, where*

$$\begin{aligned} \mathcal{H} = \{n \in \mathcal{M} \mid \quad & \sum_{\bar{L}:n\in\bar{L}} \sum_{m} \lambda^*_{\bar{L},n,m} < \alpha, \\ & \lambda^*_{\bar{L},n,m} = 0, \quad \forall \bar{L} \in \mathcal{L}, \forall m \neq n\}, \end{aligned} \tag{3.7}$$

$$\begin{aligned} \mathcal{B} = \{n \in \mathcal{M} \mid \quad & \sum_{\bar{L}:n\in\bar{L}} \sum_{m} \lambda^*_{\bar{L},n,m} \geq \alpha, \\ & \lambda^*_{\bar{L},n,m} = 0, \forall \bar{L} \notin \mathcal{L}_{\mathcal{B}}, \forall m \in \mathcal{M} \\ & \lambda^*_{\bar{L},m,n} = 0, \forall \bar{L} \in \mathcal{L}, \forall m \neq n\}. \end{aligned} \tag{3.8}$$

Lemma 3.4 states that for any arrival vector, there exists an ideal load

decomposition, under which a server is either a helper or a beneficiary. A helper server $n \in \mathcal{H}$ receives no remote service, hence $\nu_{n,m} = 0$ for all $m \neq n$. The Q_1 and Q_2 in Fig. 3.2 belong to such servers. Only the local sub-queue has non-zero rate, denoted by $\nu_{n,n}$. A beneficiary server $m \in \mathcal{B}$, provides no remote service, but receives remote service from helpers. The Q_M in Fig. 3.2 illustrates such a situation. Note that Q_M receives remote service from server 1 and 2.

The proof constructs the ideal load decomposition iteratively from $\{\tilde{\lambda}_{\bar{L},n,m}\}$ given in Lemma 3.3. The main idea is that if an underloaded server receives remote service, it can process this work locally while reducing the remote service it provides, until it becomes a helper; if an overloaded server provides remote service, it can instead use this service towards its local load while reducing the remote service it receives, until it becomes a beneficiary.

Linear Programming Characterization. Similarly, we can characterize the ideal load decomposition via a linear program. Given a decomposition $\{\tilde{\lambda}_{\bar{L},n,m}\}$ of $\boldsymbol{\lambda} \in \Lambda$ satisfying condition (3.2), we define the system remaining capacity as

$$C_R\left(\{\tilde{\lambda}_{\bar{L},n,m}\}\right) = \sum_{m\in\mathcal{D}^c} \gamma \left(1 - \sum_{\bar{L}:m\in\bar{L}} \sum_{n:n\in\bar{L}} \frac{\tilde{\lambda}_{\bar{L},n,m}}{\alpha} - \sum_{\bar{L}:m\notin\bar{L}} \sum_{n:n\in\bar{L}} \frac{\tilde{\lambda}_{\bar{L},n,m}}{\gamma}\right) + \sum_{m\in\mathcal{D}} \alpha \left(1 - \sum_{\bar{L}:m\in\bar{L}} \sum_{n:n\in\bar{L}} \frac{\tilde{\lambda}_{\bar{L},n,m}}{\alpha} - \sum_{\bar{L}:m\notin\bar{L}} \sum_{n:n\in\bar{L}} \frac{\tilde{\lambda}_{\bar{L},n,m}}{\gamma}\right), \quad (3.9)$$

which is the maximum amount by which $\sum_{\bar{L}\in\mathcal{L}_\mathcal{D}} \lambda_{\bar{L}}$ can be increased until the boundary of the capacity region is hit. From the proof of Lemma 3.4 we can see that C_R of $\boldsymbol{\lambda}$ is maximized with the ideal load decomposition. It can be determined by the following linear program. We refer to it as the *service optimization problem:*

$$\max_{\{\lambda_{\bar{L},n,m}\}} C_R\left(\{\lambda_{\bar{L},n,m}\}\right)$$

subject to constraints (3.4)-(3.6).

Remark. The two iterative procedures given in the proofs of Lemmas 3.3-3.4 provide a construction of an optimal solution of the corresponding linear program.

3.3 Throughput Optimality

We devote this section to the proof of the following theorem:

Theorem 3.1. (Throughput Optimality) *The proposed algorithm is throughput optimal. That is, it stabilizes any arrival rate vector strictly within the capacity region.*

By Lemma 3.2, it is equivalent to prove that the proposed algorithm stabilizes any arrival rate vector within Λ, defined in 3.2.1. The standard approach using a quadratic Lyapunov function does not apply in our setting, as a remote queue, despite its large queue length, can continue to grow, while a shorter queue receives local service, hence increasing the quadratic drift. Although the remote queue will be served after the local queue is empty, the time taken to obtain a negative drift will depend on the system state.

To address the challenge, we treat the helper and beneficiary subsystems, as defined in Lemma 3.4, separately. The proof has three main steps. First, we show that the helper subsystem is stable using an extension of Lemma 1 in [59]. If the beneficiary subsystem is empty, this alone proves Theorem 3.1. In the case where the *beneficiary* subsystem is non-empty, we show that the beneficiary queues are either all stable or none of them is stable. This allows us to show the stability of the *beneficiary* subsystem by contradiction.

Let M_h and M_b denote the number of helpers and beneficiaries, respectively. For simplicity, assume that $\mathcal{H} = \{1, 2, \cdots, M_h\}$, and $\mathcal{B} = \{M_h + 1, \cdots, ..., M\}$. Let $\mathbf{Q}^{(\mathcal{H})}(t)$ and $\mathbf{Q}^{(\mathcal{B})}(t)$ denote the vector of helper queues and beneficiary queues, respectively.

3.3.1 Stability of Helper Subsystem

We have the following lemma for the stability of the helper subsystem.

Lemma 3.5. *For any arrival rate vector $\boldsymbol{\lambda} \in \Lambda$, the helper queues defined by its ideal load decomposition will be stabilized with the proposed algorithm.*

Throughout this subsection, notations with superscript $^{(\mathcal{H})}$ are used to denote the corresponding vectors for helpers. Since the arrivals and services for helpers depend on the state of beneficiaries, $\{Z^{(\mathcal{H})}(t) = (\mathbf{Q}^{(\mathcal{H})}(t), \mathbf{f}^{(\mathcal{H})}(t)), t \geq 0\}$ itself is not a Markov chain. Here we use an extension of a lemma by

Hajek [60], as presented in [59]. This lemma will be also useful in the heavy-traffic analysis.

Lemma 3.6. *For an irreducible and aperiodic Markov chain* $\{X(t), t \geq 0\}$ *over a countable state space* $\mathcal{X}$*, suppose* $V : \mathcal{X} \to R_+$ *is a nonnegative-valued Lyapunov function. For some positive integer* T*, we define the* T *time slot drift of* V *at* X *as*

$$\Delta V(X) \stackrel{\Delta}{=} [V(X(t_0 + T)) - V(X(t_0))]I(X(t_0) = X),$$

where $I(\cdot)$ *is the indicator function. If the drift satisfies the following conditions:*

(C1). There exist a constant $\delta > 0$ *and* σ *such that*

$$\mathbb{E}\left[\Delta V(X) \mid X(t_0) = X\right] \leq -\delta, \text{ for all } X \in \mathcal{X}, \text{ with } V(X) \geq \sigma.$$

(C2). There exists a constant D *such that*

$$\mathbb{P}(\Delta V(X) \leq D) = 1, \text{ for all } X \in \mathcal{X}.$$

Then there exists a $\theta^* > 0$ *and a* $C^* < \infty$ *such that*

$$\limsup_{t \to \infty} \mathbb{E}\left[e^{\theta^* V(X(t))}\right] \leq C^*.$$

If furthermore the Markov chain $\{X(t), t \geq 0\}$ *is positive recurrent, then* $V(X(t))$ *converges in distribution to a random variable* $\hat{V}$ *for which*

$$\mathbb{E}\left[e^{\theta^* \hat{V}}\right] \leq C^*,$$

which implies that all moments of $\hat{V}$ *exist and are finite.*

By Lemma 3.6, the helper subsystem is stable if there exists a positive inter T and a Lyapunov function V defined on the subsystem only whose T time slot drift satisfies conditions $(C1)$ and $(C2)$. To prove that the T time slot drift satisfies condition $(C1)$, we need the following lemmas. The main idea is to use the ideal load decomposition as a potential set of arrival rates, and show that

1) The actual load arriving at Q_m with the proposed load balancing step is dominated, in an appropriate sense, by the ideal decomposition.

2) The local service at a helper server is sufficient to accommodate all load arriving at its queue according to the ideal decomposition.

We defer the proof of these lemmas to Appendix A. Note that λ_m^* is the pseudo-arrival rate of local tasks for queue Q_m according to the ideal load decomposition. With a slight abuse of notation, let $\boldsymbol{\lambda}^{*(\mathcal{H})}$ denote the corresponding pseudo-arrival rate vector for helpers.

Lemma 3.7. (Arrival.) *Consider any arrival rate vector $\boldsymbol{\lambda} \in \Lambda$ and $\mathcal{H}$ is the corresponding helper set defined in Lemma 3.4. Then under the proposed algorithm, for any $t \geq t_0$,*

$$\mathbb{E}\left[\langle \mathbf{Q}^{(\mathcal{H})}(t), \mathbf{A}^{(\mathcal{H})}(t)\rangle - \langle \mathbf{Q}^{(\mathcal{H})}(t), \boldsymbol{\lambda}^{*(\mathcal{H})}\rangle \mid Z(t_0)\right] \leq 0.$$

The lemma states that the actual arrival rate on the left-hand side of the equation is dominated by the arrival rates in the ideal decomposition, in a dot product with the queue lengths. It indicates that the proposed algorithm keeps the number of tasks at least as balanced as the ideal decomposition. The main idea of the proof is to regroup the arrival rates according to the *types* of tasks, and use the fact that an incoming task always joins the shortest queue.

Lemma 3.8. (Local service.) *Consider any arrival rate vector $\boldsymbol{\lambda} \in \Lambda$ and $\mathcal{H}$ is the corresponding helper set defined in Lemma 3.4. Then under the proposed algorithm, there exists $T_1 > 0$ such that for any $T > T_1$ and any t_0,*

$$\begin{aligned} & \mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} \left(\langle \mathbf{Q}^{(\mathcal{H})}(t), \boldsymbol{\lambda}^{*(\mathcal{H})}\rangle - \langle \mathbf{Q}^{(\mathcal{H})}(t), \mathbf{S}^{(\mathcal{H})}(t)\rangle\right) \mid Z(t_0)\right] \\ \leq\ & -\theta_1 \left\|\mathbf{Q}^{(\mathcal{H})}(t_0)\right\|_1 + C_1, \end{aligned} \tag{3.10}$$

where $\theta_1 > 0$ and C_1 are constants independent of $Z(t_0)$.

Recall that for any $m \in \mathcal{H}$, $\lambda_m^* = \lambda_m^{*l}$, i.e., all arrivals are served locally under the ideal decomposition. Thus Lemma 3.8 indicates that all servers are able to accommodate their local load assigned by the ideal decomposition. The proof uses the fact that the local service rate is always α as long as there

is local tasks present, and the inequality is obtained using the definition of the capacity region.

Proof of Lemma 3.5. Consider the following Lyapunov functions:

$$V_h(Z) = \left\|\mathbf{Q}^{(\mathcal{H})}\right\|, \quad W_h(Z) = \left\|\mathbf{Q}^{(\mathcal{H})}\right\|^2,$$

with the corresponding T-period drifts denoted by:

$$\begin{aligned}\Delta V_h(Z) &:= [V_h(Z(t_0+T)) - V_h(Z(t_0))]I(Z(t_0)=Z),\\ \Delta W_h(Z) &:= [W_h(Z(t_0+T)) - W_h(Z(t_0))]I(Z(t_0)=Z).\end{aligned}$$

Observe that $V_h(Z) = \sqrt{W_h(Z)}$. By the concavity of the square root function, we have

$$\Delta V_h(Z) \le \frac{1}{2\|\mathbf{Q}^{(\mathcal{H})}\|}[W_h(Z(t_0+T)) - W_h(Z(t_0))]I(Z(t_0)=Z) = \frac{\Delta W_h(Z)}{2\|\mathbf{Q}^{(\mathcal{H})}\|}.$$

We first analyze $\Delta W_h(Z))$.

$$\begin{aligned}&\mathbb{E}\left[\Delta W_h(Z) \mid Z(t_0)\right]\\ =\ &\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1}\left(\left\|\mathbf{Q}^{(\mathcal{H})}(t+1)\right\|^2 - \left\|\mathbf{Q}^{(\mathcal{H})}(t)\right\|^2\right) \mid Z(t_0)\right]\\ =\ &\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1}\left(2\langle\mathbf{Q}^{(\mathcal{H})}(t), \mathbf{A}^{(\mathcal{H})}(t) - \mathbf{S}^{(\mathcal{H})}(t)\rangle + 2\langle\mathbf{Q}^{(\mathcal{H})}(t), \mathbf{U}^{(\mathcal{H})}(t)\rangle\right.\right.\\ &\left.\left.+\left\|\mathbf{A}^{(\mathcal{H})}(t) - \mathbf{S}^{(\mathcal{H})}(t) + \mathbf{U}^{(\mathcal{H})}(t)\right\|^2\right) \mid Z(t_0)\right].\end{aligned}$$

By Lemma 3.1, $\langle\mathbf{Q}^{(\mathcal{H})}(t), \mathbf{U}^{(\mathcal{H})}(t)\rangle \le M^2$. Since both the arrival vector $\mathbf{A}^{(\mathcal{H})}(t)$ and the service vector $\mathbf{S}^{(\mathcal{H})}(t)$ are bounded, so as the unused service vector $\mathbf{U}^{(\mathcal{H})}(t)$, we can upper bound $\left\|\mathbf{A}^{(\mathcal{H})}(t) - \mathbf{S}^{(\mathcal{H})}(t) + \mathbf{U}^{(\mathcal{H})}(t)\right\|^2$ by a constant. Thus the T-time slot drift can be bounded as

$$\begin{aligned}&\mathbb{E}\left[\Delta W_h(Z) \mid Z(t_0)\right]\\ \le\ &2\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1}\langle\mathbf{Q}^{(\mathcal{H})}(t), \mathbf{A}^{(\mathcal{H})}(t) - \mathbf{S}^{(\mathcal{H})}(t)\rangle \mid Z(t_0)\right] + C. \qquad (3.11)\end{aligned}$$

We split the expectation term in (3.11) into two terms by adding and

subtracting an term involving an ideal decomposition $\{\lambda^*_{\bar{L},m,n}\}$ of $\boldsymbol{\lambda}$.

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1}\langle \mathbf{Q}^{(\mathcal{H})}(t), \mathbf{A}^{(\mathcal{H})}(t) - \mathbf{S}^{(\mathcal{H})}(t)\rangle \mid Z(t_0)\right] \\
= \;& \mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1}\left(\langle \mathbf{Q}^{(\mathcal{H})}(t), \mathbf{A}^{(\mathcal{H})}(t)\rangle - \langle \mathbf{Q}^{(\mathcal{H})}(t), \boldsymbol{\lambda}^{*(\mathcal{H})}\rangle\right) \mid Z(t_0)\right] \quad (3.12)\\
&+\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1}\left(\langle \mathbf{Q}^{(\mathcal{H})}(t), \boldsymbol{\lambda}^{*(\mathcal{H})}\rangle - \langle \mathbf{Q}^{(\mathcal{H})}(t), \mathbf{S}^{(\mathcal{H})}(t)\rangle\right) \mid Z(t_0)\right]. \quad (3.13)
\end{aligned}$$

By Lemmas 3.7 and 3.8, it follows that

$$\mathbb{E}\left[\Delta W_h(Z) \mid Z(t_0)\right] \leq -2\theta_1 \left\|\mathbf{Q}^{(\mathcal{H})}(t_0)\right\|_1 + C_2,$$

where θ_1 and C_2 are two positive constants independent of $Z(t_0)$. Therefore,

$$\begin{aligned}
\mathbb{E}\left[\Delta V_h(Z) \mid Z(t_0)\right] &\leq \frac{-2\theta_1 \left\|\mathbf{Q}^{(\mathcal{H})}(t_0)\right\|_1 + C_2}{2\|\mathbf{Q}^{(\mathcal{H})}(t_0)\|} \leq -\theta_1 + \frac{C_2}{2\|\mathbf{Q}^{(\mathcal{H})}(t_0)\|} \\
&= -\theta_1 + \frac{C_2}{2V_h(Z)},
\end{aligned}$$

where the second inequality comes from the fact that l_2 norm of a non-negative vector is always less than its l_1 norm. This means that we have a negative drift for sufficiently large $V_h(Z)$.

In addition, by the boundedness of arrivals and service, we have

$$\left|\left\|\mathbf{Q}^{(\mathcal{H})}(t)\right\| - \left\|\mathbf{Q}^{(\mathcal{H})}(t_0)\right\|\right| \leq \left\|\mathbf{Q}^{(\mathcal{H})}(t) - \mathbf{Q}^{(\mathcal{H})}(t_0)\right\| \leq T\sqrt{M_h}\max\{M, C_A\}.$$

Thus the drift of $V_h(Z)$ is finite with probability 1. By Lemma 3.6, $V_h(Z(t))$ converges in distribution to a random variable $\hat{V}_h$, and there exist constants θ^*_h and C^*_h with $\theta^*_h > 0$ such that $\mathbb{E}\left[e^{\theta^*_h \hat{V}_h}\right] \leq C^*_h$, which implies that the helper subsystem is stable. ∎

Consider the helper subsystem in steady state. Observe that the total arrival rate for this subsystem is at most $\sum_{\bar{L}\in\mathcal{L}^*_{\mathcal{H}}} \lambda_{\bar{L}}$, where $\mathcal{L}^*_{\mathcal{H}}$ is the set of task types that have at least one local server in $\mathcal{H}$. Since arrivals at the helper subsystem can be processed remotely, the total amount of local service provided by helper servers is no greater than $\sum_{\bar{L}\in\mathcal{L}^*_{\mathcal{H}}} \lambda_{\bar{L}}$. Hence the total amount of remote service provided by all helpers in steady state, denoted by

$R_{\mathcal{H}}$, can be lower bounded as

$$R_{\mathcal{H}} \geq \gamma \left(M_h - \frac{1}{\alpha} \sum_{\bar{L} \in \mathcal{L}^*_{\mathcal{H}}} \lambda_{\bar{L}} \right).$$

3.3.2 Stability of Beneficiary Subsystem

Assume that the beneficiary subsystem is non-empty. We will prove the following important property of the beneficiary subsystem.

Lemma 3.9. *For any arrival rate vector $\boldsymbol{\lambda} \in \Lambda$, either all queues in $\mathcal{B}$ are stable or none of them is stable.*

Proof. We prove this lemma by contradiction. Let $\mathcal{F}$ and $\mathcal{F}^c$ denote the set of stable and unstable beneficiaries, respectively. Assume that $\mathcal{F} \neq \emptyset$ and $\mathcal{F}^c \neq \emptyset$. By Lemma 3.5, helper queues $\mathcal{H}$ are stable. Consider the system with queues $\mathcal{F} \cup \mathcal{H}$ in steady state. Since queues in $\mathcal{F}^c$ grow with time, the probability that the maximum queue is among $\mathcal{F} \cup \mathcal{H}$ is arbitrarily small. Hence the amount of remote service offered by helpers and devoted to queues $\mathcal{F} \cup \mathcal{H}$ can be arbitrarily small, denoted by $\delta > 0$. Consider the following two arrival scenarios for $\mathcal{F}$:

Case (1): $\mathbb{E}\left[\sum_{m \in \mathcal{F}} A_m(t)\right] > |\mathcal{F}|\alpha$.

Then $\exists k \in \mathcal{F}$ such that $\mathbb{E}\left[A_k(t)\right] \geq \mathbb{E}\left[\sum_{m \in \mathcal{F}} A_m(t)\right] / |\mathcal{F}| > \alpha$. Thus there exists a constant $\theta > 0$ such that $\mathbb{E}\left[A_k(t)\right] \geq \alpha + \theta$. Note that $\forall m \in \mathcal{F}$, the amount of service it receives satisfies $\mathbb{E}\left[S_m(t)\right] \leq \alpha + \delta$. Choosing sufficiently small $\delta < \theta$, we can have $\mathbb{E}\left[S_k(t)\right] < \mathbb{E}\left[A_k(t)\right]$, which contradicts with the assumption that beneficiary k is stable.

Case (2): $\mathbb{E}\left[\sum_{m \in \mathcal{F}} A_m(t)\right] = |\mathcal{F}|\alpha$.

The total arrival rate for $\mathcal{F}^c$ is given by $\mathbb{E}[\sum_{m \in \mathcal{F}^c} A_m(t)] = \sum_{\bar{L} \in \mathcal{L}_{\mathcal{B}}} \lambda_{\bar{L}} - |\mathcal{F}|\alpha + \sigma$, where $\sigma \geq 0$ is the amount of arrivals local to some server in $\mathcal{H}$ but join $\mathcal{F}^c$. It can be made arbitrarily small as queues in $\mathcal{F}^c$ become sufficiently large.

Consider service received by $\mathcal{F}^c$. For any $m \in \mathcal{F}^c$, its instability implies $\mathbb{P}[Q_m(t) = 0] = 0$, hence $\mathbb{E}\left[S^l_m(t)\right] = \alpha$. We have $\mathbb{E}\left[\sum_{m \in \mathcal{F}^c} S_m(t)\right] \geq \alpha|\mathcal{F}^c| + R_{\mathcal{H}} - \delta$. Define $\epsilon = \alpha M_b + \gamma(M_h - \frac{1}{\alpha}\sum_{\bar{L} \in \mathcal{L}^*_{\mathcal{H}}} \lambda_{\bar{L}}) - \sum_{\bar{L} \in \mathcal{L}_{\mathcal{B}}} \lambda_{\bar{L}}$. Since the ideal load decomposition satisfies (3.1), ϵ is a positive constant. Select

small δ and σ such that $\delta < \frac{\epsilon}{4}$, and $\sigma < \frac{\epsilon}{4}$. Then $\exists T > 0$ such that for $\forall t > T$,

$$\mathbb{E}\left[\sum_{m\in\mathcal{F}^c} S_m(t)\right] > \mathbb{E}\left[\sum_{m\in\mathcal{F}^c} A_m(t)\right] + \frac{\epsilon}{2}.$$

This contradicts with the assumption that all queues in $\mathcal{F}^c$ are unstable. This completes the proof. ∎

Next we show the stability of the beneficiary subsystem.

Lemma 3.10. *For any arrival rate vector $\boldsymbol{\lambda} \in \Lambda$, all queues in $\mathcal{B}$ will be stabilized under the proposed algorithm.*

Proof. Again, we prove the statement by contradiction. By Lemma 3.9, we can assume that all beneficiaries are unstable. With a similar argument as the second case in the proof of Lemma 3.9, we can show that this assumption does not hold. Therefore, all beneficiaries are stable. ∎

3.4 Heavy-traffic Optimality

In this section, we show that the proposed algorithm achieves queue length optimality in the heavy-traffic limit. That is, the proposed algorithm asymptotically minimizes the number of backlogged tasks. We consider the two cases separately: locally overloaded and evenly loaded. The proof follows the Lyapunov drift-based approach recently developed in [59]. The main steps include: 1. Obtain a lower bound on the expected queue length as $\epsilon \to 0$; 2. Establish state-space collapse of the system in the heavy-traffic limit; 3. Obtain a matching upper bound on the expected queue length as $\epsilon \to 0$. However, as the "local-tasks first" policy excludes the use of a quadratic Lyapunov function, the main challenge is to prove the state-space collapse and derive a matching upper bound for the locally overloaded case. The main idea is to first show uniform boundedness for the helper queues, and analyze the Lyapunov drift for the beneficiary subsystem with a steady-state helper subsystem, and bound the amount of remote service received by helpers and the amount of helper traffic routed to beneficiaries. The proof for the evenly loaded case is considerably simpler.

3.4.1 Locally Overloaded Traffic

With locally overloaded traffic, there exists a set of beneficiary queues. Without loss of generosity, consider the traffic regime such that $\{1, 2, \cdots, M_h\}$ are helper servers and $\{M_h+1, \cdots, M\}$ are beneficiary servers, where $0 < M_h < M$. Additionally, there exists an ideal decomposition such that the pseudo-arrival rate for any beneficiary is strictly greater than its local processing capacity. That is, for any $n \in \mathcal{B}$,

$$\lambda_n^* = \sum_{\bar{L}:\bar{L}\in\mathcal{L}_\mathcal{B}, n\in\bar{L}} \sum_m \lambda_{\bar{L},n,m}^* > \alpha.$$

It is easy to verify that this condition is equivalent to the following: for any subset $\mathcal{G}$ of $\mathcal{B}$,

$$\sum_{\bar{L}\in\mathcal{L}(\mathcal{G})} \lambda_{\bar{L}} > |\mathcal{G}|\alpha, \tag{3.14}$$

where $\mathcal{L}(\mathcal{G}) = \{\bar{L} \in \mathcal{L}_\mathcal{B} \mid \exists m \in \mathcal{G}, s.t., m \in \bar{L}\}$ is the set of task types in $\mathcal{L}_\mathcal{B}$ that are local to some servers in $\mathcal{G}$. We call this condition *the heavy locally overloaded traffic condition*. Assume that the arrivals local to helpers satisfy

$$\sum_{\bar{L}\in\mathcal{L}_\mathcal{H}^*} \lambda_{\bar{L}} \equiv \Phi\alpha, \tag{3.15}$$

where $0 \le \Phi < M_h$. For any $\boldsymbol{\lambda} \in \Lambda$, it satisfies that $\sum_{\bar{L}\in\mathcal{L}} \lambda_{\bar{L}} < M_b\alpha + \Phi\alpha + \gamma(M_h - \Phi)$. We assume that

$$\sum_{\bar{L}\in\mathcal{L}} \lambda_{\bar{L}} = M_b\alpha + \Phi\alpha + \gamma(M_h - \Phi) - \epsilon, \tag{3.16}$$

i.e.,

$$\sum_{\bar{L}\in\mathcal{L}_\mathcal{B}} \lambda_{\bar{L}} = M_b\alpha + \gamma(M_h - \Phi) - \epsilon,$$

where $\epsilon > 0$ characterizes the distance between the arrival rate vector and the capacity boundary. We will make a further assumption that the $\sum_{\bar{L}\in\mathcal{L}_\mathcal{H}^*} \lambda_{\bar{L}}$ is independent of ϵ. That is, the total local load for helpers is fixed. This assumption can be removed with more care. The heavy traffic condition for the locally overloaded scenario is now articulated as follows.

Assumption 1 (Assumption for the heavy locally overloaded traffic). Consider the arrival processes $\{A_{\bar{L}}^{(\epsilon)}(t), t \geq 0\}_{\bar{L}\in\mathcal{L}}$, parameterized by $\epsilon > 0$, with mean arrival rate vector $\boldsymbol{\lambda}^{(\epsilon)}$ satisfying the heavy locally overloaded traffic condition (3.14), and equations (3.15)-(3.16). Arrivals local to helpers $\{A_{\bar{L}}^{(\epsilon)}(t), t \geq 0\}_{\bar{L}\in\mathcal{L}_{\mathcal{H}}^*}$ are independent of ϵ. The variance of the number of arrivals that are only local to beneficiaries, i.e., $\mathrm{Var}(\sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}}} A_{\bar{L}}^{(\epsilon)}(t))$, is denoted as $(\sigma_b^{(\epsilon)})^2$, which converges to σ_b^2 as $\epsilon \downarrow 0$.

The corresponding Markov chain $\{Z^{(\epsilon)}(t) = (\mathbf{Q}^{(\epsilon)}(t), \mathbf{f}^{(\epsilon)}(t)), t \geq 0\}$ has been shown to be positive recurrent. Hence the queue-length vector process $\mathbf{Q}^{(\epsilon)}(t)$ converges in distribution to a random vector $\bar{\mathbf{Q}}^{(\epsilon)}$ for any $0 < \epsilon < \bar{\epsilon}$, where $\bar{\epsilon}$ is a positive constant. All theorems in this section concern the *steady-state* queueing process $\bar{\mathbf{Q}}^{(\epsilon)}$.

Helper Subsystem

To establish the heavy-traffic optimality of the proposed algorithm, we first show uniform boundedness for the helper subsystem.

Theorem 3.2. *(***Helper queues***) Consider the limiting queueing process* $\bar{\mathbf{Q}}^{(\epsilon)}$ *under the proposed algorithm, with the arrival processes* $\{A_{\bar{L}}^{(\epsilon)}(t), t \geq 0\}_{\bar{L}\in\mathcal{L}}$, *parameterized by* $\epsilon > 0$, *satisfying Assumption 1. Then there exist a sequence of finite numbers* $\{N_r : r \in \mathbb{N}\}$ *such that for each positive integer* r,

$$\mathbb{E}\left[\left\|\bar{\mathbf{Q}}^{(\epsilon,\mathcal{H})}\right\|^r\right] \leq N_r.$$

Therefore,

$$\lim_{\epsilon\downarrow 0} \epsilon\mathbb{E}\left[\sum_{m\in\mathcal{H}} \bar{Q}_m^{(\epsilon)}\right] = 0.$$

Theorem 3.2 states that the expected queue length of $\bar{\mathbf{Q}}^{(\epsilon,\mathcal{H})}$ is bounded and independent of ϵ. The theorem follows from the same Lyapunov function for Lemma 3.5, which, with the positive recurrence of $\{Z(t), t \geq 0\}$, implies that all moments of $\bar{\mathbf{Q}}^{(\epsilon,\mathcal{H})}$ are bounded according to Lemma 3.6. Therefore, we only need to consider the beneficiary queue lengths in the rest of Section 3.4.1.

Lower Bound

To obtain a lower bound on the steady-state expected beneficiary queue lengths, we consider a hypothetical single server system with the arrival process $\{a^{(\epsilon)}(t),\, t \geq 0\}$ and the service process $\{\beta^{(\epsilon)}(t), t \geq 0\}$, where

$$a^{(\epsilon)}(t) = \sum_{\bar{L} \in \mathcal{L}_{\mathcal{B}}} A^{(\epsilon)}_{\bar{L}}(t), \quad \beta^{(\epsilon)}(t) = \sum_{i \in \mathcal{B}} X_i(t) + \sum_{j \in \mathcal{H}} Y_j(t).$$

Assume that $\{A^{(\epsilon)}_{\bar{L}}(t)\}_{\bar{L} \in \mathcal{L}}$ satisfies Assumption 1. Here $\{X_i(t)\}_{i \in \mathcal{B}}$ and $\{Y_j(t)\}_{j \in \mathcal{H}}$ are independent and each process is temporally i.i.d. For any $i \in \mathcal{B}$, let $X_i(t) \sim Bern(\alpha)$. And $\forall j \in \mathcal{H}$, $Y_j(t) \sim Bern(\gamma(1 - \rho^{(\epsilon)}_j))$, where $\rho^{(\epsilon)}_j$ is the proportion of time helper j spends on local tasks in steady state. Hence $\mathbb{E}\left[\sum_{j \in \mathcal{H}} Y_j(t)\right]$ represents the total amount of remote service provided by helpers. We denote the variance of $\beta^{(\epsilon)}(t)$ by $(\nu^{(\epsilon)}_b)^2$, which converges to a constant ν_b^2 as $\epsilon \to 0$. Let $\{\Psi^{(\epsilon)}(t)\}$ denote the corresponding queue-length process. Assume the single server starts with an empty state, i.e., $\Psi^{(\epsilon)}(0) = 0$. Then $\Psi^{(\epsilon)}(t)$ is stochastically smaller than the total beneficiary queue-length process $\sum_{m \in \mathcal{B}} Q^{(\epsilon)}_m(t)$ of the original system.

As $\{\Psi^{(\epsilon)}(t), t \geq 0\}$ is a positive recurrent Markov Chain, it converges in distribution to a random variable $\bar{\Psi}^{(\epsilon)}$. Utilizing Lemma 4 in [59], one can bound $\mathbb{E}\left[\bar{\Psi}^{(\epsilon)}\right]$ as follows:

$$\mathbb{E}\left[\bar{\Psi}^{(\epsilon)}\right] \geq \frac{(\sigma^{(\epsilon)}_b)^2 + (\nu^{(\epsilon)}_b)^2 + \epsilon^2}{2\epsilon} - \frac{M}{2},$$

which gives a lower bound on $\mathbb{E}\left[\sum_{m \in \mathcal{B}} \bar{Q}^{(\epsilon)}_m\right]$. We have the following theorem.

Theorem 3.3. *(***Lower Bound***) Consider the scheduling problem under an arrival process $\{A^{(\epsilon)}_{\bar{L}}(t), t \geq 0\}_{\bar{L} \in \mathcal{L}}$ satisfying Assumption 1. Thus under the proposed algorithm, the queue-length vector process $\mathbf{Q}^{(\epsilon)}(t)$ converges in distribution to a random vector $\bar{\mathbf{Q}}^{(\epsilon)}$ for any ϵ with $0 < \epsilon < \bar{\epsilon}$. Then*

$$\mathbb{E}\left[\sum_{m \in \mathcal{B}} \bar{Q}^{(\epsilon)}_m\right] \geq \frac{(\sigma^{(\epsilon)}_b)^2 + (\nu^{(\epsilon)}_b)^2 + \epsilon^2}{2\epsilon} - \frac{M}{2}.$$

Therefore, in the heavy traffic limit as $\epsilon \downarrow 0$,

$$\liminf_{\epsilon \downarrow 0} \epsilon \mathbb{E}\left[\sum_{m \in \mathcal{B}} \bar{Q}_m^{(\epsilon)}\right] \geq \frac{\sigma_b^2 + \nu_b^2}{2}. \tag{3.17}$$

State Space Collapse

We will show that under the proposed algorithm, the multi-dimensional state of the system collapses to a single dimension. State space collapse is a key step in establishing heavy-traffic optimality. The prioritized service makes it difficult to show state space collapse of the whole queue length vector $\mathbf{Q}$. Instead, we first show the steady-state beneficiary queue-length vector concentrates along a single direction. Then, it follows from Theorem 3.2 that the system state collapses to a particular direction.

Throughout this subsection, we use notations with superscript $^{(\mathcal{B})}$ to denote the corresponding vectors for beneficiaries. Let the queueing and working status process for beneficiaries be represented as $\{Z^{(\mathcal{B})}(t) = (\mathbf{Q}^{(\mathcal{B})}(t), f^{(\mathcal{B})}(t))\}$. Define

$$\mathbf{c_b} = \frac{1}{\sqrt{M_b}}(\underbrace{1, 1, \cdots, 1}_{B}). \tag{3.18}$$

Then the parallel and perpendicular components of the queue length vector $\mathbf{Q}^{(\mathcal{B})}$ with respect to $\mathbf{c_b}$ are given by:

$$\mathbf{Q}_{||}^{(\mathcal{B})} = \langle \mathbf{c_b}, \mathbf{Q}^{(\mathcal{B})} \rangle \mathbf{c_b}, \quad \mathbf{Q}_{\perp}^{(\mathcal{B})} = \mathbf{Q}^{(\mathcal{B})} - \mathbf{Q}_{||}^{(\mathcal{B})}.$$

We will establish state-space collapse of $\mathbf{Q}^{(\mathcal{B})}$ along the direction $\mathbf{c_b}$, by showing that $\mathbf{Q}_{\perp}^{(\mathcal{B})}$ is bounded and independent of the heavy-traffic parameter ϵ.

Remark. With the bounded moments of the helper queue lengths, the whole queue length vector $\mathbf{Q}$ collapses to the following direction $\mathbf{c}$:

$$\mathbf{c} = \frac{1}{\sqrt{M_b}}(\underbrace{0, 0, \cdots, 0}_{M_h}, \underbrace{1, 1, \cdots, 1}_{M_b}). \tag{3.19}$$

To establish state space collapse, we need to show that when the arrival rate vector $\boldsymbol{\lambda}$ satisfies the heavy locally overloaded traffic assumption, there exists an ideal load decomposition $\{\lambda^*_{\bar{L},m,n}\}$ satisfying the following proposition in addition to Lemma 3.4. The proof is provided in Appendix A.

Lemma 3.11. *Consider an arrival rate vector $\boldsymbol{\lambda}$ that satisfies the heavy locally overloaded traffic assumption, with $0 < \epsilon < \bar{\epsilon}$, where $\bar{\epsilon}$ is a positive constant. Then there exists a decomposition $\left\{\lambda^*_{\bar{L},n,m}\right\}$ of $\boldsymbol{\lambda}$ satisfying the following conditions:*

1. $\forall m \in \mathcal{B}$,
$$\sum_{\bar{L}: m \in \bar{L}} \frac{\lambda^*_{\bar{L},m,m}}{\alpha} = 1 - \epsilon_b;$$

2. $\forall m \in \mathcal{B}$, $\exists \bar{L} \in \mathcal{L}_{\mathcal{B}}$, *s.t.* $\sum_{n \in \mathcal{H}} \lambda^*_{\bar{L},m,n} \geq \lambda_0$;

where ϵ_b is a constant satisfying $0 < \epsilon_b < \frac{\epsilon}{\alpha M_b}$, and λ_0 is a positive constant not depending on ϵ.

The following theorem formally states the state space collapse result.

Theorem 3.4. *(***State space collapse***) Consider the scheduling problem under an arrival process $\{A^{(\epsilon)}_{\bar{L}}(t), t \geq 0\}_{\bar{L} \in \mathcal{L}}$ satisfying Assumption 1. Thus under the proposed algorithm, the queue-length vector process $\mathbf{Q}^{(\epsilon)}(t)$ converges in distribution to a random vector $\bar{\mathbf{Q}}^{(\epsilon)}$ for any ϵ with $0 < \epsilon < \bar{\epsilon}$. Then, there exists a sequence of finite numbers $\{\hat{C}_r : r \in \mathbb{N}\}$ such that for each positive integer r,*
$$\mathbb{E}\left[\left\|\bar{\mathbf{Q}}^{(\epsilon,\mathcal{B})}_{\perp}\right\|^r\right] \leq \hat{C}_r,$$
where $\bar{\mathbf{Q}}^{(\epsilon,\mathcal{B})}_{\perp}$ is the component of $\bar{\mathbf{Q}}^{(\epsilon,\mathcal{B})}$ perpendicular to the direction $\mathbf{c_b}$ defined in (3.18).

Moreover, there exists a sequence of finite numbers $\{C_r : r \in \mathbb{N}\}$ such that for each positive integer r,
$$\mathbb{E}\left[\left\|\bar{\mathbf{Q}}^{(\epsilon)}_{\perp}\right\|^r\right] \leq C_r,$$
where $\bar{\mathbf{Q}}^{(\epsilon)}_{\perp}$ is the component of $\bar{\mathbf{Q}}^{(\epsilon)}$ perpendicular to the direction $\mathbf{c}$ defined in (3.19).

We need the following lemmas to prove Theorem 3.4.

Lemma 3.12. *Let $\mathbf{c}$ be a vector with unit norm in $\mathbb{R}^{M_b}$. Then for any $t \geq 0$,*
$$\left\|\mathbf{Q}^{(\mathcal{B})}_{||}(t+1)\right\|^2 - \left\|\mathbf{Q}^{(\mathcal{B})}_{||}(t)\right\|^2 \geq 2\langle \mathbf{c_b}, \mathbf{Q}^{(\mathcal{B})}(t)\rangle\langle \mathbf{c_b}, \mathbf{A}^{(\mathcal{B})}(t) - \mathbf{S}^{(\mathcal{B})}(t)\rangle,$$

where $\mathbf{Q}_{||}^{(\mathcal{B})}$ *is the parallel component of the beneficiary queue length vector* $\mathbf{Q}^{(\mathcal{B})}$ *with respect to the direction* $\mathbf{c}$*.*

Lemma 3.13. *Consider a time slot* t_0 *and a positive integer* T*. Let* $\mathbf{c}$ *be a vector with unit norm in* $\mathbb{R}^{M_b}$*. Then for any* t *with* $t_0 \leq t < t_0 + T$*,*

$$\left| \left\| \mathbf{Q}_{\perp}^{(\mathcal{B})}(t) \right\| - \left\| \mathbf{Q}_{\perp}^{(\mathcal{B})}(t_0) \right\| \right| \leq T\sqrt{M_b} \max\{M, C_A\}, \tag{3.20}$$

where $\mathbf{Q}_{\perp}^{(\mathcal{B})}$ *is the perpendicular component of the beneficiary queue length vector* $\mathbf{Q}^{(\mathcal{B})}$ *with respect to the direction* $\mathbf{c}$*.*

Lemma 3.14. *Consider a time slot* t_0 *and a positive integer* T*. For any* t *with* $t_0 \leq t < t_0+T$*, let* $G(t) = \langle \mathbf{Q}^{(\mathcal{B})}(t), \mathbf{A}^{(\mathcal{B})}(t) - \mathbf{S}^{(\mathcal{B})}(t) \rangle - \langle \mathbf{c_b}, \mathbf{Q}^{(\mathcal{B})}(t) \rangle \langle \mathbf{c_b}, \mathbf{A}^{(\mathcal{B})}(t) - \mathbf{S}^{(\mathcal{B})}(t) \rangle$*. Then* $G(t) \leq h \left\| \mathbf{Q}_{\perp}^{(\mathcal{B})}(t_0) \right\| + F_0$*, where* $h = \sqrt{M_b} \max\{M, C_A\}$ *and* $F_0 = M_b T (\max\{M, C_A\})^2$ *are constants.*

Proof of Theorem 3.4. Consider the following Lyapunov function

$$V(Z^{(\mathcal{B})}) = \left\| \mathbf{Q}_{\perp}^{(\mathcal{B})} \right\|.$$

By Lemma 3.6, it is sufficient to show that the T-period drift of $V(Z^{(\mathcal{B})})$ is always finite and is negative for sufficient large V. Fix an ϵ within the range specified in the theorem. The T-time slot drift of V is given by $\Delta V(Z^{(\mathcal{B})}) = [V(Z^{(\mathcal{B})}(t_0 + T) - V(Z^{(\mathcal{B})}(t_0)))] I(Z^{(\mathcal{B})}(t_0) = Z)$.

First we show that $\Delta V(Z^{(\mathcal{B})})$ satisfies condition $(C2)$. From lemma 3.13, we can see that $\mathbb{P}\left[\Delta V(Z^{(\mathcal{B})}) \leq C\right] = 1$ with $C = T\sqrt{M_b} \max\{M, C_A\}$.

Next we focus on condition $(C1)$. Consider the following Lyapunov functions:

$$W(Z^{(\mathcal{B})}) = \left\| \mathbf{Q}^{(\mathcal{B})} \right\|^2, W_{||}(Z^{(\mathcal{B})}) = \left\| \mathbf{Q}_{||}^{(\mathcal{B})} \right\|^2,$$

with the corresponding T-period drifts denoted by:

$$\Delta W(Z^{(\mathcal{B})}) := [W(Z^{(\mathcal{B})}(t_0 + T)) - W(Z^{(\mathcal{B})}(t_0))] I(Z^{(\mathcal{B})}(t_0) = Z^{(\mathcal{B})}),$$
$$\Delta W_{||}(Z^{(\mathcal{B})}) := [W_{||}(Z^{(\mathcal{B})}(t_0 + T)) - W_{||}(Z^{(\mathcal{B})}(t_0))] I(Z^{(\mathcal{B})}(t_0) = Z^{(\mathcal{B})}).$$

Then $V(Z^{(\mathcal{B})}) = \sqrt{W(Z^{(\mathcal{B})}) - W_{||}(Z^{(\mathcal{B})})}$. Due to the concavity of the square root function, the drift of $V(Z^{(\mathcal{B})})$ satisfies the following inequality

(Lemma 7 in [59]):

$$\Delta V(Z^{(\mathcal{B})}) \leq \frac{1}{2\left\|\mathbf{Q}_{\perp}^{(\mathcal{B})}\right\|}\left(\Delta W(Z^{(\mathcal{B})}) - \Delta W_{||}(Z^{(\mathcal{B})})\right). \tag{3.21}$$

As it is difficult to study the drift of $V(Z^{(\mathcal{B})})$ directly, we will get started with the drifts of $W(Z^{(\mathcal{B})})$ and $W_{||}(Z^{(\mathcal{B})})$.

For the drift $\Delta W(Z^{(\mathcal{B})})$, following the same argument as in the derivation of the bound (3.11) on $\Delta W(Z^{(\mathcal{B})})$, we can obtain

$$\begin{aligned} & \mathbb{E}\left[\Delta W(Z^{(\mathcal{B})}) \mid Z^{(\mathcal{B})}(t_0)\right] \\ \leq \; & 2\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} \langle \mathbf{Q}^{(\mathcal{B})}(t), \mathbf{A}^{(\mathcal{B})}(t) - \mathbf{S}^{(\mathcal{B})}(t)\rangle \mid Z^{(\mathcal{B})}(t_0)\right] + C', \end{aligned}$$

where $C' > 0$ is a constant.

For the drift $\Delta W_{||}(Z^{(\mathcal{B})})$, by Lemma 3.12, we have

$$\begin{aligned} & \mathbb{E}\left[\Delta W_{||}(Z^{(\mathcal{B})})) \mid Z^{(\mathcal{B})}(t_0)\right] \\ \geq \; & 2\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} \langle \mathbf{c_b}, \mathbf{Q}^{(\mathcal{B})}(t)\rangle \langle \mathbf{c_b}, \mathbf{A}^{(\mathcal{B})}(t) - \mathbf{S}^{(\mathcal{B})}(t)\rangle \mid Z^{(\mathcal{B})}(t_0)\right]. \end{aligned}$$

Let $G(t) = \langle \mathbf{Q}^{(\mathcal{B})}(t), \mathbf{A}^{(\mathcal{B})}(t) - \mathbf{S}^{(\mathcal{B})}(t)\rangle - \langle \mathbf{c_b}, \mathbf{Q}^{(\mathcal{B})}(t)\rangle \langle \mathbf{c_b}, \mathbf{A}^{(\mathcal{B})}(t) - \mathbf{S}^{(\mathcal{B})}(t)\rangle$. Combining the above two inequalities yields:

$$\mathbb{E}\left[\Delta W(Z^{(\mathcal{B})}) - \Delta W_{||}(Z^{(\mathcal{B})}) \mid Z^{(\mathcal{B})}(t_0)\right] \leq 2\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} G(t) \mid Z^{(\mathcal{B})}(t_0)\right] + C'.$$

To bound $\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} G(t) \mid Z^{(\mathcal{B})}(t_0)\right]$, we consider the following random variables

$$\begin{aligned} t_m^* &= \min\{\tau : \tau \geq t_0, f_m(\tau^-) = -1\}, m \in \mathcal{M}, \\ t^* &= \max_{m \in \mathcal{M}} t_m^*. \end{aligned}$$

Therefore, by the time slot t^*, all servers have been available at least once. We decompose the probability space into two parts by using t^*: $D_1 = \{t^* \geq t_0 + K \mid Z^{(\mathcal{B})}(t_0)\}$ and $D_2 = \{t^* < t_0 + K \mid Z^{(\mathcal{B})}(t_0)\}$. Let $T = JK$,where J

and K are positive integers. Then

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} G(t) \mid Z^{(\mathcal{B})}(t_0)\right] \\
= \;& \mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} G(t) \mid Z^{(\mathcal{B})}(t_0), t^* \geq t_0+K\right]\mathbb{P}\left[D_1\right] \qquad (3.22)\\
&+\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} G(t) \mid Z^{(\mathcal{B})}(t_0), t^* < t_0+K\right]\mathbb{P}\left[D_2\right]. \qquad (3.23)
\end{aligned}$$

For the term (3.22), by Lemma 3.14 we have

$$\mathbb{E}\left[\sum_t G(t) \mid Z^{(\mathcal{B})}(t_0), t^* \geq t_0+K\right] \leq hT\left\|\mathbf{Q}_\perp^{(\mathcal{B})}(t_0)\right\| + F_0T. \qquad (3.24)$$

For the term (3.23), we divide the summation into two parts: from $t=t_0$ to $t=t^*$ and from $t=t^*+1$ to $t=t_0+T-1$. The first part can be bounded in a similar way as (3.22) by Lemma 3.14:

$$\mathbb{E}\left[\sum_{t=t_0}^{t^*} G(t) \mid Z^{(\mathcal{B})}(t_0), t^* < t_0+K\right] \leq Kh\left\|\mathbf{Q}_\perp^{(\mathcal{B})}(t_0)\right\| + KF_0. \qquad (3.25)$$

The key step is to bound $\mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1} G(t) \mid t^* < t_0+K, Z^{(\mathcal{B})}(t_0)\right]$.

For each $m \in \mathcal{B}$, define $\hat{A}_m(t) = \sum_{\bar{L}\in\mathcal{L}_\mathcal{B}} A_{\bar{L},m}$, i.e., $\hat{A}$ excludes arrivals that are also local to helpers from beneficiaries. Define $\lambda_m^{*l} = \sum_{\bar{L}:m\in\mathcal{L}} \lambda^*_{\bar{L},m,m}$ and $\lambda_m^{*r} = \sum_{\bar{L}:m\in\mathcal{L}}\sum_{n:n\neq m} \lambda^*_{\bar{L},m,n}$ from the ideal load decomposition $\{\lambda^*_{\bar{L},m,n}\}$ given by Lemma 3.11. Let $T_d = t_0+T-t^*$, and $D(Q(t_0)) = M_bQ^{max}(t_0) - \sum_{m\in\mathcal{B}} Q_m(t_0)$, where $Q^{max}(t)$ is the maximum beneficiary queue length at time t. We use F_i, $i\in\mathbb{N}$, to denote a positive constant not depending on ϵ. In the following argument, we temporarily omit the superscript $^{(\mathcal{B})}$ for brevity. Let $\mathbf{e}$ denote a vector of all ones in $\mathbb{R}^{M_b}$, i.e., $\mathbf{e} = \sqrt{M_b}\mathbf{c_b}$. We break $G(t)$ into four groups and obtain an bound for each group.

$$\begin{aligned}
G(t) \;=\; & \langle \mathbf{Q}, \hat{\mathbf{A}}\rangle - \langle \mathbf{Q}, \boldsymbol{\lambda}^{*l}\rangle - \langle Q^{max}\mathbf{e}, \boldsymbol{\lambda}^{*r}\rangle - \langle \mathbf{Q} - Q^{max}\mathbf{e}, \lambda_0\mathbf{e}\rangle \\
&+\langle \mathbf{Q} - Q^{max}\mathbf{e}, \lambda_0\mathbf{e}\rangle + \langle \mathbf{Q}, \boldsymbol{\lambda}^{*l}\rangle - \langle \mathbf{Q}, \mathbf{S}^l\rangle \\
&+\langle Q^{max}\mathbf{e}, \boldsymbol{\lambda}^{*r}\rangle - \langle \mathbf{Q}, \mathbf{S}^r\rangle - \langle \mathbf{c_b}, \mathbf{Q}\rangle\langle \mathbf{c_b}, \hat{\mathbf{A}} - \mathbf{S}\rangle \\
&-\langle \mathbf{c_b}, \mathbf{Q}\rangle\langle \mathbf{c_b}, \mathbf{A} - \hat{\mathbf{A}}\rangle + \langle \mathbf{Q}, \mathbf{A} - \hat{\mathbf{A}}\rangle.
\end{aligned}$$

Bounds for the four groups are established by the following lemmas.

Lemma 3.15. *(**Arrivals**)*

$$\mathbb{E}\Bigg[\sum_{t=t^*+1}^{t_0+T-1}\Big(\langle\mathbf{Q}(t),\hat{\mathbf{A}}(t)\rangle-\langle\mathbf{Q}(t),\boldsymbol{\lambda}^{*l}\rangle-\langle Q^{max}(t)\mathbf{e},\boldsymbol{\lambda}^{*r}\rangle \\ -\langle\mathbf{Q}(t)-Q^{max}(t)\mathbf{e},\lambda_0\mathbf{e}\rangle)\mid t^*<t_0+K,Z^{(\mathcal{B})}(t_0)\Big]\leq 0.$$

Lemma 3.16. *(**Local service**)*

$$\begin{aligned}&\mathbb{E}\Bigg[\sum_{t=t^*+1}^{t_0+T-1}\big(\langle\mathbf{Q}(t)-Q^{max}(t)\mathbf{e},\lambda_0\mathbf{e}\rangle+\langle\mathbf{Q}(t),\boldsymbol{\lambda}^{*l}\rangle\\ &-\langle\mathbf{Q}(t),\mathbf{S}^l(t)\rangle\big)\mid t^*<t_0+K,Z^{(\mathcal{B})}(t_0)\Big]\\ \leq\;& -T_d\left[\lambda_0 D(Q(t_0))+\alpha\epsilon_0\sum_{m\in\mathcal{B}}Q_m(t_0)\right]+F_1.\end{aligned}$$

Lemma 3.17. *(**Remote service**)*

$$\begin{aligned}&\mathbb{E}\Bigg[\sum_{t=t^*+1}^{t_0+T-1}(\langle Q^{max}(t)\mathbf{e},\boldsymbol{\lambda}^{*r}\rangle-\langle\mathbf{Q}(t),\mathbf{S}^r(t)\rangle\\ &-\langle\mathbf{c_b},\mathbf{Q}(t)\rangle\langle\mathbf{c_b},\hat{\mathbf{A}}(t)-\mathbf{S}(t)\rangle\Big)\mid t^*<t_0+K,Z^{(\mathcal{B})}(t_0)\Big]\\ \leq\;& T_d\left[\frac{\lambda_0}{4}D(Q(t_0))+\alpha\epsilon_0\sum_{m\in\mathcal{B}}Q_m(t_0)\right]+F_2.\end{aligned}$$

Lemma 3.18. *(**Extra arrivals to beneficiaries**)*

$$\begin{aligned}&\mathbb{E}\Bigg[\sum_{t=t^*+1}^{t_0+T-1}\Big(-\langle\mathbf{c_b},\mathbf{Q}(t)\rangle\langle\mathbf{c_b},\mathbf{A}(t)-\hat{\mathbf{A}}(t)\rangle\\ &+\langle\mathbf{Q}(t),\mathbf{A}(t)-\hat{\mathbf{A}}(t)\rangle\Big)\mid t^*<t_0+K,Z^{(\mathcal{B})}(t_0)\Big]\\ \leq\;& T_d\frac{\lambda_0}{4}D(Q(t_0))+F_3.\end{aligned}$$

Combining inequalities from Lemma 3.15-3.18 yields

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1} G(t) \mid Z^{(\mathcal{B})}(t_0), t^* < t_0 + K\right] \\
\leq\ & -(t_0+T-t^*)\frac{\lambda_0}{2}\left(M_b Q^{max}(t_0) - \sum_{m\in\mathcal{B}} Q_m(t_0)\right) + F_1 + F_2 + F_3 \\
=\ & -(t_0+T-t^*)\frac{\lambda_0}{2}\left\|Q^{max}(t_0)\mathbf{e} - \mathbf{Q}^{(\mathcal{B})}(t_0)\right\|_1 + F_4, \\
\leq\ & -(t_0+T-t^*)\frac{\lambda_0}{2}\left\|Q^{max}(t_0)\mathbf{e} - \mathbf{Q}^{(\mathcal{B})}(t_0)\right\| + F_4,
\end{aligned}$$

where $F_4 = F_1 + F_2 + F_3$, and $\|\cdot\|_1$ is the l_1 norm. The last inequality follows by the fact that the l_1 norm of a vector is no smaller than its l_2 norm.

As $\langle \mathbf{c_b}, \mathbf{Q}^{(\mathcal{B})}(t)\rangle$ minimizes the convex function $\left\|x\mathbf{c_b} - \mathbf{Q}^{(\mathcal{B})}(t_0)\right\|$ over $x \in \mathbb{R}$, i.e.,

$$\left\|Q^{max}(t_0)\mathbf{e} - \mathbf{Q}^{(\mathcal{B})}(t_0)\right\| \geq \left\|\langle \mathbf{c_b}, \mathbf{Q}^{(\mathcal{B})}(t)\rangle \mathbf{c_b} - \mathbf{Q}^{(\mathcal{B})}(t_0)\right\| = \left\|\mathbf{Q}_{\perp}^{(\mathcal{B})}(t_0)\right\|,$$

it follows that

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1} G(t) \mid Z^{(\mathcal{B})}(t_0), t^* < t_0 + K\right] \\
\leq\ & -(t_0+T-t^*)\frac{\lambda_0}{2}\left\|\mathbf{Q}_{\perp}^{(\mathcal{B})}(t_0)\right\| + F_4 \\
\leq\ & -(J-1)K\frac{\lambda_0}{2}\left\|\mathbf{Q}_{\perp}^{(\mathcal{B})}(t_0)\right\| + F_4. \qquad (3.26)
\end{aligned}$$

Denote $\mathbb{P}[t^* \geq t_0 + K \mid Z(t_0)]$ by Y_K. Substituting (3.24)-(3.26) in (3.22) and (3.23) yields

$$\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} G(t) \mid Z^{(\mathcal{B})}(t_0), t^* < t_0 + K\right] \leq -F_5\left\|\mathbf{Q}_{\perp}^{(\mathcal{B})}(t_0)\right\| + F_6,$$

where $F_5 = -Y_K hT - (1-Y_K)hK + \frac{\lambda_0}{2}(1-Y_K)(J-1)K$ and $F_6 = Y_K F_0 T + (1-Y_K)F_0 K + (1-Y_K)F_4$.

Then from inequality (3.21), we can obtain the following upper bound on

the drift of $V(Z^{(\mathcal{B})})$:

$$\mathbb{E}\left[\Delta V(Z^{(\mathcal{B})}) \mid Z^{(\mathcal{B})}(t_0)\right] \;\leq\; -F_5 + \frac{F_6 + C'/2}{\left\|\mathbf{Q}_{\perp}^{(\mathcal{B})}(t_0)\right\|}. \tag{3.27}$$

Observe that $\lim_{K,J\to\infty} F_5 = +\infty$. Thus for any $\delta > 0$, there exist large enough K and J such that $-F_5 < -\delta$. Pick any θ with $0 < \theta < \delta$ and let $\zeta = \frac{2F_6+C'}{2(\delta-\theta)}$. Then $\mathbb{E}\left[\Delta V(Z^{(\mathcal{B})}) \mid Z^{(\mathcal{B})}(t_0)\right] \leq -\theta$ for all $Z^{(\mathcal{B})}$ with $V(Z^{(\mathcal{B})}) = \left\|\mathbf{Q}_{\perp}^{(\mathcal{B})}(t_0)\right\| \geq \zeta$. This means that the drift of $V(Z^{(\mathcal{B})})$ is negative for sufficiently large $V(Z^{(\mathcal{B})})$, as the constants θ and ζ do not depend on ϵ. Therefore there exists a sequence of constants $\{\hat{C}_r : r \in \mathbb{N}\}$ such that $\mathbb{E}\left[\left\|\mathbf{Q}_{\perp}^{(\epsilon,\mathcal{B})}\right\|^r\right] \leq \hat{C}_r$ for each $r = 1, 2, \cdots$.

The parallel and perpendicular component of the queue length vector $\mathbf{Q}$ with respect to $\mathbf{c} \in \mathbb{R}^M$ defined in (3.19) are given by:

$$\mathbf{Q}_{||} = \langle \mathbf{c}, \mathbf{Q}\rangle \mathbf{c} = \frac{\sum_{k\in\mathcal{B}} Q_k}{\sqrt{M_b}}\mathbf{c}, \quad \mathbf{Q}_{\perp} = \mathbf{Q} - \mathbf{Q}_{||}.$$

We note the fact that

$$\mathbf{Q}_{\perp} = (\mathbf{Q}^{(\mathcal{H})}, \mathbf{Q}_{\perp}^{(\mathcal{B})}).$$

From Theorem 3.2, all moments of $\mathbf{Q}^{(\mathcal{H})}$ are bounded. Together with the result for $\mathbf{Q}_{\perp}^{(\mathcal{B})}$, it follows that all moments of $\mathbf{Q}_{\perp}$ are bounded. That is, there exist a sequence of constants $\{C_r : r \in \mathbb{N}\}$ such that $\mathbb{E}\left[\|\mathbf{Q}_{\perp}\|^r\right] \leq C_r$ for each $r = 1, 2, \cdots$.

■

Upper Bound

We will derive an upper bound on the steady-state beneficiaries queue-length based, using the Lyapunov drift-based moment bounding technique developed in [59]. The main difficulty arises from the fact that the total amount of service received at beneficiary queues, $\sum_{m\in\mathcal{B}} S_m(t)$, depends on the queueing process $\mathbf{Q}(t)$: for any $m \in \mathcal{B}$, the local service provided by server m, $\{S_m^l(t)\}$ is neither i.i.d, nor independent of $Q_m(t)$; the amount of remote service $\mathcal{B}$ received, $\sum_{m\in\mathcal{B}} S_m^r(t)$, relies on the occurrence of system states that the maximum queue is among $\mathcal{B}$. In addition, the existence of tasks

types shared among $\mathcal{H}$ and $\mathcal{B}$, i.e., task types that are local to some helper and some beneficiary, makes total arrivals for $\mathcal{B}$, $\sum_{m\in\mathcal{B}} A_m(t)$, depend on $\mathbf{Q}(t)$ as well. Hence we define the following ideal processes to decouple the dependence.

Ideal local service process $\hat{\mathbf{S}}^l(t)$:

$$\hat{S}^l_m(t) = \begin{cases} X^l_m(t) & \text{if } m \in \mathcal{B} \\ S^l_m(t) & \text{if } m \in \mathcal{H}, \end{cases}$$

where the processes $\{X^l_m(t), t \geq 0\}_{m\in\mathcal{B}}$ is coupled with $\{S^l_m(t), t \geq 0\}_{m\in\mathcal{B}}$ in the following way: If $\eta_m(t) = m$, $X^l_m(t) = S^l_m(t)$; if $\eta_m(t) \neq m$, $X^l_m(t) = 1$ when $R_m(t) = 1$, and $X^l_m(t) \sim Bern(\frac{\alpha-\gamma}{1-\gamma})$ when $R_m(t) = 0$. Hence $\forall m \in \mathcal{B}$, $\{X^l_m(t), t \geq 0\}$ is i.i.d. with $X^l_m(t) \sim Bern(\alpha)$.

Ideal remote service process $\hat{\mathbf{R}}(t)$:

$$\hat{R}_m(t) = \begin{cases} 0 & \text{if } m \in \mathcal{B} \\ X^r_m(t) & \text{if } m \in \mathcal{H}, \end{cases}$$

where the processes $\{X^r_m(t), t \geq 0\}_{m\in\mathcal{H}}$ is coupled with $\{R_m(t), t \geq 0\}_{m\in\mathcal{H}}$ in the following way: If $\eta_m(t) \neq m$, $X^r_m(t) = R_m(t)$; if $\eta_m(t) = m$, $X^r_m(t) \sim Bern(\gamma)$. Hence for $m \in \mathcal{H}$, $\{X^r_m(t), t \geq 0\}$ is i.i.d. with $X^r_m(t) \sim Bern(\gamma)$.

Ideal scheduling decision process $\hat{\eta}(t)$: For any $m \in \mathcal{B}$, $\hat{\eta}_m(t) = m$. For any $m \in \mathcal{H}$, $\hat{\eta}_m(t) = \eta_m(t)$ if $\eta_m(t) = m$; when $f_m(t^-) = -1$ and $Q_m(t) = 0$, $\hat{\eta}_m(t) = argmax_{n\in\mathcal{B}}\{Q_n(t)\}$. That is, idle helper server with empty local queue is scheduled to serve the maximum beneficiary queue under the *ideal scheduling*.

Ideal remote service received $\hat{\mathbf{S}}^r(t)$:

$$\hat{S}^r_n(t) = \begin{cases} \sum\limits_{m\in\mathcal{H}} \hat{R}_m(t) \cdot I_{\{\hat{\eta}_m(t)=n\}} & \text{if } n \in \mathcal{B} \\ 0 & \text{if } n \in \mathcal{H}. \end{cases}$$

Then the *ideal departure* for queue m is given by $\hat{S}_m(t) = \hat{S}^l_m(t) + \hat{S}^r_m(t)$.

Ideal arrival process $\hat{\mathbf{A}}(t)$: all task types local to both some helper and beneficiary are routed to helpers.

For $m \in \mathcal{B}$, let

$$\hat{A}_m(t) = \sum_{\bar{L} \in \mathcal{L}_{\mathcal{B}}: m \in \bar{L}} A_{\bar{L},m} = A_m(t) - \sum_{\bar{L} \notin \mathcal{L}_{\mathcal{B}}: m \in \bar{L}} A_{\bar{L},m}.$$

For $m \in \mathcal{H}$, let

$$\hat{A}_m(t) = A_m(t) + \sum_{\bar{L} \notin \mathcal{L}_{\mathcal{B}}: m \in \bar{L}} \frac{\sum_{n \in \bar{L} \cap \mathcal{B}} A_{\bar{L},n}}{|\{k : k \in \bar{L} \cap \mathcal{H}\}|}.$$

Then we can rewrite the queue dynamics as

$$\mathbf{Q}(t+1) = \mathbf{Q}(t) + \hat{\mathbf{A}}(t) - \hat{\mathbf{S}}(t) + \hat{\mathbf{U}}(t),$$

where $\hat{\mathbf{U}}(t) = \hat{\mathbf{S}}(t) - \mathbf{S}(t) + \mathbf{A}(t) - \hat{\mathbf{A}}(t) + \mathbf{U}(t)$. This queue dynamics will be used to expand the Lyapunov drift.

We will use the following lemma to derive an upper bound on the expected beneficiary queue lengths. The lemma follows from the fact that the mean drift of function $\left\|\mathbf{Q}_{||}\right\|^2$ equals zero when the system is in steady state.

Lemma 3.19. *For the scheduling system, consider any arrival process with an arrival rate vector strictly within the capacity region. Suppose the queueing process is in steady state under the proposed algorithm. Then for any direction* $\mathbf{c} \in \mathbb{R}^M$*, we have*

$$\begin{aligned} & \mathbb{E}\left[\langle \mathbf{c}, \mathbf{Q}(t)\rangle \langle \mathbf{c}, \hat{\mathbf{S}}(t) - \hat{\mathbf{A}}(t)\rangle\right] \\ = \; & \frac{\mathbb{E}\left[\langle \mathbf{c}, \hat{\mathbf{A}}(t) - \hat{\mathbf{S}}(t)\rangle^2\right]}{2} + \frac{\mathbb{E}\left[\langle \mathbf{c}, \hat{\mathbf{U}}(t)\rangle^2\right]}{2} \qquad (3.28) \\ & + \mathbb{E}\left[\langle \mathbf{c}, \mathbf{Q}(t) + \hat{\mathbf{A}}(t) - \hat{\mathbf{S}}(t)\rangle \langle \mathbf{c}, \hat{\mathbf{U}}(t)\rangle\right]. \qquad (3.29) \end{aligned}$$

Considering the direction $\mathbf{c}$ defined in (3.19), we can obtain an upper bound on $\mathbb{E}\left[\langle \mathbf{c}, \mathbf{Q}(t)\rangle\right] = \frac{1}{\sqrt{M_b}} \mathbb{E}\left[\sum_{m \in \mathcal{B}} Q_m(t)\right]$ by bounding terms in (3.28) and (3.29).

Theorem 3.5. *(***Upper Bound***) Consider the scheduling system under the proposed algorithm with the arrival process* $\{A^{(\epsilon)}_{\bar{L}}(t), t \geq 0\}_{\bar{L} \in \mathcal{L}}$ *satisfying Assumption 1. Thus the queue-length vector process* $\mathbf{Q}^{(\epsilon)}(t)$ *converges in distribution to a random vector* $\bar{\mathbf{Q}}^{(\epsilon)}$ *for any* ϵ *with* $0 < \epsilon < \bar{\epsilon}$*. Then the expected*

beneficiary queue lengths in steady-state can be upper bounded as

$$\mathbb{E}\left[\sum_{m\in\mathcal{B}}\bar{Q}_m^{(\epsilon)}\right] \le \frac{(\sigma_b^{(\epsilon)})^2+(\nu_b^{(\epsilon)})^2}{2\epsilon} + D^{(\epsilon)},$$

where $D^{(\epsilon)} = o(\frac{1}{\epsilon})$*, i.e.,* $\lim_{\epsilon\downarrow 0}\epsilon D^{(\epsilon)} = 0$.

Therefore, in the heavy-traffic limit, the upper bound becomes,

$$\limsup_{\epsilon\downarrow 0}\epsilon\mathbb{E}\left[\sum_{m\in\mathcal{B}}\bar{Q}_m^{(\epsilon)}\right] \le \frac{\sigma_b^2+\nu_b^2}{2}.$$

This upper bound under heavy-traffic limit coincides with the lower bound (3.17), which establishes the first moment heavy-traffic optimality of the proposed algorithm.

Proof. For convenience, we temporarily omit the superscript (ϵ). Under the *ideal arrival process*, shared type tasks that join beneficiaries queues are redistributed among its helper local servers evenly. Hence

$$\mathbb{E}\left[\sum_{m\in\mathcal{B}}\hat{A}_m(t)\right] = \mathbb{E}\left[\sum_{\bar{L}\in\mathcal{L}_\mathcal{B}}A_{\bar{L}}(t)\right] = \sum_{\bar{L}\in\mathcal{L}_\mathcal{B}}\lambda_{\bar{L}} = M_b\alpha+\gamma(M_h-\Phi)-\epsilon.$$

Let ρ_m denote the proportion of time server m spends on serving local queue m in steady state. Then

$$\begin{aligned}
&\mathbb{E}\left[\sum_{m\in\mathcal{B}}\hat{S}_m(t)\right] = M_b\alpha+\sum_{m\in\mathcal{H}}\gamma(1-\rho_m) = M_b\alpha+M_h\gamma-\gamma\sum_{m\in\mathcal{H}}\rho_m,\\
&\mathbb{E}\left[\sum_{m\in\mathcal{H}}\hat{S}_m(t)\right] = \sum_{m\in\mathcal{H}}\alpha\rho_m,\\
&\mathbb{E}\left[\sum_{m\in\mathcal{B}}\hat{S}_m(t)-\sum_{m\in\mathcal{B}}\hat{A}_m(t)\right] = \epsilon+\gamma\left(\Phi-\sum_{m\in\mathcal{H}}\rho_m\right) = \epsilon+\delta,
\end{aligned}$$

where $\delta = \gamma(\Phi-\sum_{m\in\mathcal{H}}\rho_m)$. Since the amount of local service provided by helpers $\sum_{m\in\mathcal{H}}\alpha\rho_m$ is not greater than the arrival rate of tasks local to helpers $\sum_{\bar{L}\in\mathcal{L}_\mathcal{H}^*}\lambda_{\bar{L}} \equiv \Phi\alpha$, we have $\delta\ge 0$. We will further show that $\delta\to 0$ as $\epsilon\downarrow 0$ later.

For any time slot t, we analyze each term in Lemma 3.19 with respect to

the collapse direction $\mathbf{c}$ defined in (3.19).

$$\begin{aligned}
& \mathbb{E}\left[\langle \mathbf{c}, \mathbf{Q}(t)\rangle \langle \mathbf{c}, \hat{\mathbf{S}}(t) - \hat{\mathbf{A}}(t)\rangle\right] \\
= & \frac{1}{M_b}\mathbb{E}\left[\left(\sum_{m\in\mathcal{B}} Q_m(t)\right)\left(\sum_{m\in\mathcal{B}} \hat{S}_m(t) - \sum_{m\in\mathcal{B}} \hat{A}_m(t)\right)\right] \\
\overset{(a)}{=} & \frac{1}{M_b}\mathbb{E}\left[\sum_{m\in\mathcal{B}} Q_m(t)\right]\mathbb{E}\left[\sum_{m\in\mathcal{B}} \hat{S}_m(t) - \sum_{m\in\mathcal{B}} \hat{A}_m(t)\right] \\
= & \frac{\epsilon+\delta}{M_b}\mathbb{E}\left[\sum_{m\in\mathcal{B}} Q_m(t)\right], \qquad (3.30)
\end{aligned}$$

where (a) follows from the fact that the total arrivals of tasks that are only local to beneficiaries do not depend on the beneficiary queue-lengths, so as the ideal service process for beneficiary queues.

$$\begin{aligned}
\mathrm{Var}\left(\sum_{m\in\mathcal{B}} \hat{A}_m(t)\right) &= \mathrm{Var}\left(\sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}}} A_{\bar{L}}(t)\right) = (\sigma_b^{(\epsilon)})^2, \\
\mathrm{Var}\left(\sum_{m\in\mathcal{B}} \hat{S}_m(t)\right) &= M_b\alpha(1-\alpha) + \sum_{m\in\mathcal{H}} \gamma(1-\rho_m)[1-\gamma(1-\rho_m)] = (\nu_b^{(\epsilon)})^2,
\end{aligned}$$

where we recall that $(\sigma_b^{(\epsilon)})^2$ and $(\nu_b^{(\epsilon)})^2$ are the variances of the arrival process $a^{(\epsilon)}(t)$ and the service process $\beta^{(\epsilon)}$ for the single server system defined for the lower bound.

As $\{\hat{\mathbf{A}}(t)\}$ and $\{\hat{\mathbf{S}}(t)\}$ are independent, and $\mathbb{E}\left[\langle \mathbf{c}, \hat{\mathbf{S}}\rangle\right] - \mathbb{E}\left[\langle \mathbf{c}, \hat{\mathbf{A}}\rangle\right] = \frac{\epsilon+\delta}{\sqrt{M_b}}$, it follows that

$$\mathbb{E}\left[\langle \mathbf{c}, \hat{\mathbf{A}}(t) - \hat{\mathbf{S}}(t)\rangle^2\right] = \frac{1}{M_b}\left((\sigma_b^{(\epsilon)})^2 + (\nu_b^{(\epsilon)})^2 + (\epsilon+\delta)^2\right). \qquad (3.31)$$

In steady state, $\mathbb{E}\left[\langle \mathbf{c}, \hat{\mathbf{A}}(t) - \hat{\mathbf{S}}(t) + \hat{\mathbf{U}}(t)\rangle\right] = \mathbb{E}\left[\langle \mathbf{c}, \mathbf{Q}(t+1)\rangle - \langle \mathbf{c}, \mathbf{Q}(t)\rangle\right] = 0$. Thus $\mathbb{E}\left[\langle \mathbf{c}, \hat{\mathbf{U}}(t)\rangle\right] = \mathbb{E}\left[\langle \mathbf{c}, \hat{\mathbf{S}}(t) - \hat{\mathbf{A}}(t)\rangle\right] = \frac{\epsilon+\delta}{\sqrt{M_b}}$. Also,

$$\langle \mathbf{c}, \hat{\mathbf{U}}(t)\rangle = \langle \mathbf{c}, \hat{\mathbf{S}}(t) - \mathbf{S}(t) + \mathbf{A}(t) - \hat{\mathbf{A}}(t) + \mathbf{U}(t)\rangle \leq \frac{2M + C_A}{\sqrt{M_b}}.$$

By the coupling of $\hat{\mathbf{S}}(t)$ and $\mathbf{S}(t)$, $\langle \mathbf{c}, \hat{\mathbf{S}}(t) - \mathbf{S}(t)\rangle \geq 0$. In addition, $\langle \mathbf{c}, \mathbf{A}(t) -$

$\hat{\mathbf{A}}(t)\rangle \geq 0$. Hence $\langle \mathbf{c}, \hat{\mathbf{U}}(t)\rangle \geq 0$. Therefore

$$\mathbb{E}\left[\langle \mathbf{c}, \hat{\mathbf{U}}(t)\rangle^2\right] \leq \frac{2M + C_A}{\sqrt{M_b}} \mathbb{E}\left[\langle \mathbf{c}, \hat{\mathbf{U}}(t)\rangle\right] = \frac{(2M + C_A)(\epsilon + \delta)}{M_b}. \tag{3.32}$$

Next we bound (3.29).

$$\begin{aligned} & \mathbb{E}\left[\langle \mathbf{c}, \mathbf{Q}(t) + \hat{\mathbf{A}}(t) - \hat{\mathbf{S}}(t)\rangle \langle \mathbf{c}, \hat{\mathbf{U}}(t)\rangle\right] \\ = \; & \mathbb{E}\left[\langle \mathbf{c}, \mathbf{Q}(t)\rangle \langle \mathbf{c}, \hat{\mathbf{U}}(t)\rangle\right] + \mathbb{E}\left[\langle \mathbf{c}, \hat{\mathbf{A}}(t) - \hat{\mathbf{S}}(t)\rangle \langle \mathbf{c}, \hat{\mathbf{U}}(t)\rangle\right] \\ \leq \; & \mathbb{E}\left[\langle \mathbf{c}, \mathbf{Q}(t)\rangle \langle \mathbf{c}, \hat{\mathbf{U}}(t)\rangle\right] + \frac{C_A(\epsilon + \delta)}{M_b}. \end{aligned}$$

We can expand the expectation term as

$$\begin{aligned} & \langle \mathbf{c}, \mathbf{Q}(t)\rangle \langle \mathbf{c}, \hat{\mathbf{U}}(t)\rangle \\ = \; & \langle \mathbf{Q}(t), \hat{\mathbf{U}}(t)\rangle - \langle \mathbf{Q}_\perp(t), \hat{\mathbf{U}}_\perp(t)\rangle \\ = \; & \langle \mathbf{Q}(t), \hat{\mathbf{S}}(t) - \mathbf{S}(t)\rangle + \langle \mathbf{Q}(t), \mathbf{A}(t) - \hat{\mathbf{A}}(t)\rangle + \langle \mathbf{Q}(t), \mathbf{U}(t)\rangle \quad (3.33) \\ & - \langle \mathbf{Q}_\perp(t), \hat{\mathbf{U}}_\perp(t)\rangle. \quad (3.34) \end{aligned}$$

We need the following lemmas to bound the four terms in (3.33)-(3.34).

Lemma 3.20. $\mathbb{E}\left[\left\|\hat{\mathbf{U}}(t)\right\|^2\right] \leq R\epsilon$, *where R is a constant not depending on ϵ.*

Lemma 3.21. $\mathbb{E}\left[\langle \mathbf{Q}(t), \hat{\mathbf{S}}(t) - \mathbf{S}(t)\rangle\right] \leq R_0\epsilon + R_1\sqrt{M}\mathbb{E}\left[\langle \mathbf{e}, \hat{\mathbf{S}}(t) - \mathbf{S}(t)\rangle\right]$, *where* $\mathbf{e} = \frac{1}{\sqrt{M}}\underbrace{(1, 1, \cdots, 1)}_{M}$, *$R_0 > 0$ and $R_1 > 0$ are constants not depending on ϵ.*

Lemma 3.22. $\langle \mathbf{Q}(t), \mathbf{A}(t) - \hat{\mathbf{A}}(t)\rangle \leq 0$.

Lemma 3.23. $\langle \mathbf{Q}(t), \mathbf{U}(t)\rangle \leq M\sqrt{M}\langle \mathbf{e}, \mathbf{U}(t)\rangle$, *where* $\mathbf{e} = \frac{1}{\sqrt{M}}\underbrace{(1, 1, \cdots, 1)}_{M}$.

Let $R_2 = \max\{R_1, M\}$, then by Lemmas 3.21 and 3.23,

$$\begin{aligned} & \langle \mathbf{Q}(t), \hat{\mathbf{S}}(t) - \mathbf{S}(t)\rangle + \langle \mathbf{Q}(t), \mathbf{U}(t)\rangle \\ \leq \; & R_0\epsilon + R_2\sqrt{M}\mathbb{E}\left[\langle \mathbf{e}, \hat{\mathbf{S}}(t) - \mathbf{S}(t) + \mathbf{U}(t)\rangle\right] \\ = \; & R_0\epsilon + R_2\left(\epsilon - \frac{\alpha - \gamma}{\gamma}\delta\right). \end{aligned}$$

The last equality follows from the fact that $\mathbf{Q}(t)$ is in steady state, which implies that $\mathbb{E}\left[\langle \mathbf{e}, \mathbf{A}(t) - \mathbf{S}(t) + \mathbf{U}(t)\rangle\right] = \mathbb{E}\left[\langle \mathbf{e}, \mathbf{Q}(t+1)\rangle - \langle \mathbf{e}, \mathbf{Q}(t)\rangle\right] = 0$. Thus $\mathbb{E}\left[\langle \mathbf{e}, \hat{\mathbf{S}}(t) - \mathbf{S}(t) + \mathbf{U}(t)\rangle\right] = \mathbb{E}\left[\langle \mathbf{e}, \hat{\mathbf{S}}(t) - \mathbf{A}(t)\rangle\right] = \frac{1}{\sqrt{M}}\left(\epsilon - \frac{\alpha-\gamma}{\gamma}\delta\right)$.

Next we use the state space collapse result to bound $-\langle \hat{\mathbf{U}}_\perp(t), \hat{\mathbf{U}}_\perp(t)\rangle$.

$$\begin{aligned}
\mathbb{E}\left[-\langle \mathbf{Q}_\perp(t), \hat{\mathbf{U}}_\perp(t)\rangle\right] &\overset{(a)}{\leq} \sqrt{\mathbb{E}\left[\|\mathbf{Q}_\perp(t)\|^2\right]\mathbb{E}\left[\left\|\hat{\mathbf{U}}(t)\right\|^2\right]} \\
&\overset{(b)}{\leq} \sqrt{C_2\mathbb{E}\left[\left\|\hat{\mathbf{U}}(t)\right\|^2\right]} \\
&\overset{(c)}{\leq} \sqrt{C_2 R\epsilon},
\end{aligned}$$

where (a) follows from Cauchy-Swartz inequality; (b) comes from the state space collapse result; (c) follows from Lemma 3.20.

We can bound the term (3.29) as

$$\begin{aligned}
&\mathbb{E}\left[\langle \mathbf{c}, \mathbf{Q}(t) + \hat{\mathbf{A}}(t) - \hat{\mathbf{S}}(t)\rangle\langle \mathbf{c}, \hat{\mathbf{U}}(t)\rangle\right] \\
\leq\ & \frac{C_A(\epsilon+\delta)}{M_b} + R_0\epsilon + R_2\left(\epsilon - \frac{\alpha-\gamma}{\gamma}\delta\right) + \sqrt{C_2 R\epsilon}. \qquad (3.35)
\end{aligned}$$

Now we reintroduce the superscript $^{(\epsilon)}$. Substituting (3.30)-(3.32) and (3.35) in (3.28) and (3.29) yields:

$$\begin{aligned}
\frac{\epsilon+\delta}{M_b}\mathbb{E}\left[\sum_{m\in\mathcal{B}} Q_m(t)\right] \leq\ & \frac{1}{2M_b}\left((\sigma_b^{(\epsilon)})^2 + (\nu_b^{(\epsilon)})^2 + (\epsilon+\delta)^2\right) + \sqrt{C_2 R\epsilon} \\
& + \frac{(2M+3C_A)(\epsilon+\delta)}{2M_b} + R_0\epsilon + R_2\left(\epsilon - \frac{\alpha-\gamma}{\gamma}\delta\right).
\end{aligned}$$

Therefore

$$\mathbb{E}\left[\sum_{m\in\mathcal{B}} Q_m(t)\right] \leq \frac{(\sigma_b^{(\epsilon)})^2 + (\nu_b^{(\epsilon)})^2}{2(\epsilon+\delta)} + D^{(\epsilon)} \overset{(a)}{\leq} \frac{(\sigma_b^{(\epsilon)})^2 + (\nu_b^{(\epsilon)})^2}{2\epsilon} + D^{(\epsilon)},$$

where (a) follows from the fact that $\delta \geq 0$, and

$$\begin{aligned}
D^{(\epsilon)} =\ & \frac{\epsilon+\delta}{2} + M_b\sqrt{C_2 R}\frac{\sqrt{\epsilon}}{\epsilon+\delta} + M + \frac{3C_A}{2} \\
& + M_b(R_0+R_2)\frac{\epsilon}{\epsilon+\delta} - \frac{M_b R_2(\alpha-\gamma)}{\gamma}\frac{\delta}{\epsilon+\delta}.
\end{aligned}$$

As $\epsilon \downarrow 0$, $\delta \downarrow 0$, we have $\limsup_{\epsilon \downarrow 0} \epsilon D^{(\epsilon)} = 0$. Thus $D^{(\epsilon)} = o(\frac{1}{\epsilon})$. ∎

3.4.2 Evenly Loaded Traffic

In this subsection, we establish the heavy-traffic delay optimality of the proposed algorithm in the regime with only helper servers. The proof for the evenly loaded traffic case follows exactly the same three steps for the locally overloaded traffic case. The symmetry brought about by the uniform ideal load for all queues will significantly simplify the proof.

Heavy Evenly Loaded Traffic Regime

We consider the arrival rate vector $\boldsymbol{\lambda}^{(\epsilon)} \in \Lambda$, parameterized by $\epsilon > 0$, $\boldsymbol{\lambda}^{(\epsilon)} = (1-\epsilon_0)\bar{\boldsymbol{\lambda}}$, where $\epsilon_0 = \frac{\epsilon}{M\alpha}$, and $\bar{\boldsymbol{\lambda}}$ is an arrival rate vector on the boundary of the capacity region Λ such that all servers are fully utilized to handle its local load. That is, it lies in the set $\mathcal{F}$,

$$\begin{aligned}\mathcal{F} = \big\{\boldsymbol{\lambda} = (\lambda_{\bar{L}} : \bar{L} \in \mathcal{L}) \mid \quad & \exists\, (\lambda_{\bar{L},n,m}) \text{ such that} \\ & \lambda_{\bar{L},n,m} \geq 0, \forall \bar{L} \in \mathcal{L}, \forall n \in \bar{L}, \forall m \in \mathcal{M}, \\ & \lambda_{\bar{L},n,m} = 0, \forall \bar{L} \in \mathcal{L}, \forall n \in \bar{L}, \forall m \neq n, \\ & \lambda_{\bar{L}} = \sum_{n:n\in\bar{L}} \sum_{m=1}^{M} \lambda_{\bar{L},n,m}, \forall \bar{L} \in \mathcal{L}, \\ & \sum_{\bar{L}:m\in\bar{L}} \lambda_{\bar{L},m,m} = \alpha,\ \forall m \in \mathcal{M}\ \big\}. \end{aligned} \tag{3.36}$$

Thus the heavy-traffic limit corresponds to the scenario $\epsilon \downarrow 0$. It is easy to see that with $\boldsymbol{\lambda}^{(\epsilon)}$, all servers in the system are helpers, i.e., $\mathcal{H} = \mathcal{M}$.

Additionally, the limiting arrival rate vector $\bar{\boldsymbol{\lambda}} \in \mathcal{F}$ satisfies a condition called *resource pooling,* under which there is a one-dimensional state-space collapse in the heavy-traffic limit. The resource polling condition means that all servers are *connected* in the following sense: Consider a decomposition of $\bar{\boldsymbol{\lambda}} \in \mathcal{F}$ satisfying (3.36). Server m *connects directly* with server m' if there exists a task type $\bar{L} \in \mathcal{L}$ local to both m and m', such that $\lambda_{\bar{L},m,m} > 0$ and $\lambda_{\bar{L},m',m'} > 0$. Server m is *connected* with server m' if there exists a sequence of servers $m_1, \cdots, m_k$, such that $m_1 = m$, $m_k = m'$, and m_i connects directly

with m_{i+1} for all $i = 1, 2, \cdots, k-1$.

Definition (Resource pooling). An arrival rate vector $\bar{\boldsymbol{\lambda}} \in \mathcal{F}$ satisfies the resource pooling condition if there exists a decomposition of $\bar{\boldsymbol{\lambda}}$ such that all servers are connected.

Assumption 2 (Assumption for the heavy evenly loaded traffic). Consider the arrival processes $\{A_{\bar{L}}^{(\epsilon)}(t), t \geq 0\}_{\bar{L} \in \mathcal{L}}$, parameterized by $\epsilon > 0$, with mean arrival rate vector $\boldsymbol{\lambda}^{(\epsilon)} = (1 - \epsilon_0)\bar{\boldsymbol{\lambda}}$, where $\epsilon_0 = \frac{\epsilon}{M\alpha}$, and $\bar{\boldsymbol{\lambda}} \in \mathcal{F}$ satisfies the resource pooling condition. The variance of the number of arrivals, $\mathrm{Var}(\sum_{\bar{L} \in \mathcal{L}} A_{\bar{L}}^{(\epsilon)}(t))$, is denoted as $(\sigma^{(\epsilon)})^2$, which converges to σ^2 as $\epsilon \downarrow 0$.

For any arrival $\{A_{\bar{L}}^{(\epsilon)}(t), t \geq 0\}_{\bar{L} \in \mathcal{L}}$ satisfying Assumption 2, as the mean arrival rate vector $\boldsymbol{\lambda}^{(\epsilon)} \in \Lambda$, the proposed algorithm stabilizes the system. Therefore, the corresponding Markov chain $\{Z^{(\epsilon)}(t) = (\mathbf{Q}^{(\epsilon)}(t), \mathbf{f}^{(\epsilon)}(t)), t \geq 0\}$ is positive recurrent. Hence the queue-length vector process $\mathbf{Q}^{(\epsilon)}(t)$ converges in distribution to a random vector $\bar{\mathbf{Q}}^{(\epsilon)}$ for any $0 < \epsilon < \bar{\epsilon}$, where $\bar{\epsilon}$ is a positive constant. All theorems in this subsection concern the *steady-state* queueing process $\bar{\mathbf{Q}}^{(\epsilon)}$. We obtain the three theorems analogous to the locally overloaded case.

Remark: If the arrival rate vector $\bar{\boldsymbol{\lambda}}$ makes some server pairs $(\hat{m}, m)$ isolated from each other, we can always decompose servers into disjoint groups, such that servers within each group are connected, while isolated from servers outside. For each connected group $\mathcal{H}_i$, we can establish state-space collapse and obtain an upper bound on $\mathbb{E}\left[\sum_{m \in \mathcal{H}_i} \bar{Q}_m^{(\epsilon)}\right]$. Together they give an upper bound on $\mathbb{E}\left[\sum_m \bar{Q}_m^{(\epsilon)}\right]$, which coincides with the lower bound in the heavy traffic limit.

Lower Bound

Consider a single server system with the arrival process $\{a^{(\epsilon)}(t) = \sum_{\bar{L} \in \mathcal{L}} A_{\bar{L}}^{(\epsilon)}(t), t \geq 0\}$ and the service process $\{\beta^{(\epsilon)}(t) = \sum_{m=1}^{M} X_m(t), t \geq 0\}$, where $\{X_m(t), t \geq 0\}_{m \in \mathcal{M}}$ are independent, and each process is temporally i.i.d. with $X_m(t) \sim Bern(\alpha)$. Assume that the mean of $\{A_{\bar{L}}^{(\epsilon)}(t)\}_{\bar{L} \in \mathcal{L}}$ satisfies $\boldsymbol{\lambda}^{(\epsilon)} \in \Lambda$ and $\mathcal{H} = \mathcal{M}$. We denote the variances of the arrival and service processes as $(\sigma^{(\epsilon)})^2 = \mathrm{Var}(a^{(\epsilon)}(t))$, $\nu^2 = \mathrm{Var}(\beta^{(\epsilon)}(t))$. Then the corresponding queue-

length process $\{\Psi'(t), t \geq 0\}$ is stochastically smaller than the total queue lengths process $\{\sum_{m=1}^{M} Q_m^{(\epsilon)}(t)\}$ of the original system. Again, utilizing the lower bound from Lemma 4 in [59], we can establish an lower bound on the performance of the proposed algorithm.

Theorem 3.6. *(***Lower bound***)*

$$\mathbb{E}\left[\sum_{m\in\mathcal{M}} \bar{Q}_m^{(\epsilon)}\right] \geq \frac{(\sigma^{(\epsilon)})^2 + \nu^2 + \epsilon^2}{2\epsilon} - \frac{M}{2}.$$

Therefore, in the heavy traffic limit,

$$\liminf_{\epsilon\downarrow 0} \epsilon\mathbb{E}\left[\sum_{m=1}^{M} \bar{Q}_m^{(\epsilon)}\right] \geq \frac{\sigma^2 + \nu^2}{2}. \tag{3.37}$$

State Space Collapse

Recall that with the arrival process $\{A_{\bar{L}}^{(\epsilon)}(t), t \geq 0\}_{\bar{L}\in\mathcal{L}}$ satisfying Assumption 2, the queue-length process under the proposed algorithm $\mathbf{Q}^{(\epsilon)}(t)$ converges in distribution to a random vector $\bar{\mathbf{Q}}^{(\epsilon)}$ for any $0 < \epsilon < \bar{\epsilon}$. We will show that the queue length vector $\bar{\mathbf{Q}}^{(\epsilon)}$ collapses to the direction of a unit vector, i.e., the vector

$$\mathbf{c_e} = \frac{1}{\sqrt{M}}(\underbrace{1, 1, \cdots, 1}_{M}). \tag{3.38}$$

Then the parallel and the perpendicular component of any $\mathbf{Q} \in \mathbb{R}^M$ with respect to the direction $\mathbf{c_e}$ become:

$$\mathbf{Q}_{||} = \frac{\sum_m Q_m}{\sqrt{M}}\mathbf{c_e}, \quad \mathbf{Q}_{\perp} = \left[Q_k - \frac{\sum_m Q_m}{M}\right]_{k=1}^{M}.$$

Theorem 3.7. *(***State space collapse***) There exists a sequence of finite numbers* $\{C_r' : r \in \mathcal{N}\}$ *such that for each positive integer* r*,*

$$\mathbb{E}\left[\left\|\bar{\mathbf{Q}}_{\perp}^{(\epsilon)}\right\|^r\right] \leq C_r',$$

where $\mathbf{Q}_{\perp}^{(\epsilon)}$ *is the component of* $\mathbf{Q}^{(\epsilon)}$ *perpendicular to* $\mathbf{c_e}$ *defined in (3.38).*

Analogue to Lemma 3.11, we have the following lemma for the ideal load

decomposition of any arrival rate vector satisfying the resource pooling condition, which is essential for showing state space collapse.

Lemma 3.24. *Consider any arrival rate vector* $\boldsymbol{\lambda} = (1-\epsilon_0)\bar{\boldsymbol{\lambda}}$, *where* $\epsilon_0 = \frac{\epsilon}{M\alpha}$, *and* $\bar{\boldsymbol{\lambda}} \in \mathcal{F}$ *satisfies the resource pooling condition. Consider any* $0 < \epsilon < \bar{\epsilon}$, *where* $\bar{\epsilon}$ *is a positive constant. Then there exists a decomposition* $\left\{\lambda^*_{\bar{L},n,m}\right\}$ *of* λ *satisfying Lemma 3.4 and the following conditions:*

1. $\forall m \in \mathcal{M}$,
$$\sum_{\bar{L}:m\in\bar{L}} \lambda^*_{\bar{L},m,m} = \alpha(1-\epsilon_0);$$
2. *there exists a positive constant* λ_{min} *not depending on* ϵ, *such that for any two servers* m *and* m' *that are connected directly, there exists a task type* $\bar{L} \in \mathcal{L}$, *such that* $\lambda^*_{\bar{L},m,m} \geq \lambda_{min}$, $\lambda^*_{\bar{L},m',m'} \geq \lambda_{min}$.

Additionally, we need lemmas analogue to Lemmas 3.12-3.14, with beneficiary queue length vector $\mathbf{Q}^{(\mathcal{B})}$ replaced by all queue length vector $\mathbf{Q}$. The proof of these lemmas is similar to that of Lemma 3.12-3.14, so we skip it here. Through the following argument, we omits the superscript $^{(\epsilon)}$ for ease of exposition.

Lemma 3.25. *Let* $\mathbf{c}$ *be a vector with unit norm in* $\mathbb{R}^M$. *Then for any* $t \geq 0$,

$$\left\|\mathbf{Q}_{||}(t+1)\right\|^2 - \left\|\mathbf{Q}_{||}(t)\right\|^2 \geq 2\langle \mathbf{c}, \mathbf{Q}(t)\rangle\langle \mathbf{c}, \mathbf{A}(t) - \mathbf{S}(t)\rangle,$$

where $\mathbf{Q}_{||}$ *is the parallel component of the queue length vector* $\mathbf{Q}$ *with respect to the direction* $\mathbf{c}$.

Lemma 3.26. *Consider a time slot* t_0 *and a positive integer* T. *Let* $\mathbf{c}$ *be a vector with unit norm in* $\mathbb{R}^M$. *Then for any* t *with* $t_0 \leq t < t_0 + T$,

$$|\|\mathbf{Q}_\perp(t)\| - \|\mathbf{Q}_\perp(t_0)\|| \leq T\sqrt{M}\max\{M, C_A\},$$

where $\mathbf{Q}_\perp$ *is the perpendicular component of the queue length vector* $\mathbf{Q}$ *with respect to the direction* $\mathbf{c}$.

Lemma 3.27. *Consider a time slot* t_0 *and a positive integer* T. *For any* t *with* $t_0 \leq t < t_0 + T$, *let* $G_e(t) = \langle \mathbf{Q}(t), \mathbf{A}(t) - \mathbf{S}(t)\rangle - \langle \mathbf{c}_{\mathbf{b}e}, \mathbf{Q}(t)\rangle\langle \mathbf{c}_{\mathbf{b}e}, \mathbf{A}(t) - \mathbf{S}(t)\rangle$. *Then* $G_e(t) \leq h'\|\mathbf{Q}_\perp(t_0)\| + F'_0$, *where* $h' = \sqrt{M}\max\{M, C_A\}$ *and* $F'_0 = MT(\max\{M, C_A\})^2$ *are constants.*

Proof of Theorem 3.7. Consider the Lyapunov function $V_e(Z) = \|\mathbf{Q}_\perp\|$. We again show that the T-period drift of $V_e(Z)$, given by $\Delta V_e(Z) = [V_e(Z(t_0 + T) - V_e(Z(t_0)))]I(Z(t_0) = Z)$, satisfies the two conditions in Lemma 3.6. The proof follows exactly the same steps for the locally overloaded traffic case.

By employing the same analysis of the conditional drifts of $W(Z) = \|Q\|^2$ and $W_{||}(Z) = \left\|\mathbf{Q}_{||}\right\|^2$ in Section 3.4.1, and using the bound (3.21) on $\Delta V_e(Z)$, we have

$$\mathbb{E}\left[\Delta V_e(Z) \mid Z(t_0)\right] \leq \frac{\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} G_e(t) \mid Z(t_0)\right] + C}{\|\mathbf{Q}_\perp\|},$$

where C is a constant and $G_e(t) = \langle \mathbf{Q}(t), \mathbf{A}(t) - \mathbf{S}(t)\rangle - \langle \mathbf{c_e}\mathbf{Q}(t)\rangle\langle \mathbf{c_e}, \mathbf{A}(t) - \mathbf{S}(t)\rangle$.

A key different step is to bound $\mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1} G_e(t) \mid Z(t_0), t^* < t_0 + K\right]$. Since we consider the entire system, we do not have the shared arrival issue here. In addition, as the local load for each server approaches 1, each server devotes to serving its local queue. Hence the remote service vanishes as $\boldsymbol{\lambda}^{(\epsilon)}$ is close to the capacity boundary, which enable us to get rid of the remote service terms in bounding $G_e(t)$. We have the following inequalities analogue to Lemmas 3.15-3.17.

Lemma 3.28. *For any $t^* < t < t_0 + T$,*

$$\mathbb{E}\left[\langle \mathbf{Q}(t), \mathbf{A}(t)\rangle - \langle \mathbf{Q}(t), \boldsymbol{\lambda}^*\rangle \mid t^*, Z(t_0)\right] \leq -\lambda_{min}\|\mathbf{Q}_\perp(t_0)\| + F_1',$$

where F_1' is a positive constant not depending on ϵ.

Lemma 3.29.

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1} \left(\langle \mathbf{Q}(t), \boldsymbol{\lambda}^*\rangle - \langle \mathbf{Q}(t), \mathbf{S}(t)\rangle\right) \mid t^*, Z(t_0)\right] \\ &\leq -(t_0 + T - t^*)\frac{\epsilon}{M}\sum_m Q_m(t_0) + F_2',\end{aligned}$$

where F_2' is a positive constant not depending on ϵ.

Lemma 3.30. *For any $t^* < t < t_0 + T$,*

$$\mathbb{E}\left[\langle \mathbf{c_b}_e, \mathbf{Q}(t)\rangle\langle \mathbf{c_b}_e, \mathbf{A}(t) - \mathbf{S}(t)\rangle \mid t^*, Z(t_0)\right] \geq -\frac{\epsilon}{M}\sum_m Q_m(t_0) - F_3',$$

where F_3' is a positive constant not depending on ϵ.

It follows from these lemmas that

$$\begin{aligned}\mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1} G_e(t) \mid Z(t_0), t^* < t_0 + K\right] &\leq -(t_0 + T - t^*)\lambda_{min}\|\mathbf{Q}_\perp(t_0)\| + F_4' \\ &\leq -(J-1)K\lambda_{min}\|\mathbf{Q}_\perp(t_0)\| + F_4',\end{aligned}$$

where $F_4' = (F_1' + F_3')(J-1)K + F_2'$ is constant independent of ϵ.

Utilizing Lemma 3.27 and the above inequality yields an upper bound on the drift of $V_e(Z)$, which is similar to (3.27). Hence the drift of $V_e(Z)$ is negative for sufficiently large $V_e(Z)$. Moreover, Lemma 3.26 implies finite drift of $V_e(Z)$. Note that the Markov chain $\{Z^{(\epsilon)}(t) = (\mathbf{Q}^{(\epsilon)}(t), \mathbf{f}^{(\epsilon)}(t)), t \geq 0\}$ is positive recurrent. Therefore, by Lemma 3.6, all moments of $V_e(Z)$ are finite and independent of ϵ. State space collapse of $\mathbf{Q}$ along the direction $\mathbf{c_e}$ follows.

■

Upper Bound

Again we construct an *ideal service process* $\{\hat{\mathbf{S}}(t), t \geq 0\}$ that makes $\sum_m \hat{S}_m(t)$ independent of $\sum_m Q_m(t)$. In particular, $\forall m \in \mathcal{M}$, its *ideal local service process* $\hat{S}_m^l(t)$ is defined in the same way as that for beneficiaries in the locally overloaded traffic case.

Ideal local service process $\hat{\mathbf{S}}^l(t)$:

$$\hat{S}_m^l(t) = X_m^l(t), \forall m \in \mathcal{M},$$

where the processes $\{X_m^l(t), t \geq 0\}_{m\in\mathcal{M}}$ is coupled with $\{S_m(t), t \geq 0\}_{m\in\mathcal{M}}$ in the following way: If $\eta_m(t) = m$, $X_m^l(t) = S_m^l(t)$; if $\eta_m(t) \neq m$, $X_m^l(t) = 1$ when $R_m(t) = 1$, and $X_m^l(t) \sim Bern(\frac{\alpha-\gamma}{1-\gamma})$ when $R_m(t) = 0$. Hence $\forall m \in \mathcal{M}$, $\{X_m^l(t), t \geq 0\}$ is i.i.d. with $X_m^l(t) \sim Bern(\alpha)$.

Ideal remote service process $\hat{\mathbf{R}}(t)$: For any $m \in \mathcal{M}$, $\hat{R}_m(t) = 0$.

Ideal scheduling decision process $\hat{\eta}(t)$: For any $m \in \mathcal{M}$, $\hat{\eta}_m(t) = m$.

Since the total amount of arrivals for the system is independent of queue-length process, we do not need to define *ideal arrival process* here.

We can rewrite the queue dynamics as

$$\mathbf{Q}(t+1) = \mathbf{Q}(t) + \mathbf{A}(t) - \hat{\mathbf{S}}(t) + \hat{\mathbf{U}}(t),$$

where $\hat{\mathbf{U}}(t) = \hat{\mathbf{S}}(t) - \mathbf{S}(t) + \mathbf{U}(t)$. Again setting the drift of $W_{||}(Z) = \left\|\mathbf{Q}_{||}\right\|^2$ to zero gives the following equation, which is similar to that in Lemma 3.19.

$$\mathbb{E}\left[\langle \mathbf{c_e}, \mathbf{Q}\rangle \langle \mathbf{c_e}, \hat{\mathbf{S}} - \mathbf{A}\rangle\right] = \frac{\mathbb{E}\left[\langle \mathbf{c_e}, \mathbf{A} - \hat{\mathbf{S}}\rangle^2\right]}{2} + \frac{\mathbb{E}\left[\langle \mathbf{c_e}, \hat{\mathbf{U}}\rangle^2\right]}{2} \quad (3.39)$$

$$+\mathbb{E}\left[\langle \mathbf{c_e}, \mathbf{Q} + \mathbf{A} - \hat{\mathbf{S}}\rangle \langle \mathbf{c_e}, \hat{\mathbf{U}}\rangle\right]. \quad (3.40)$$

We obtain the upper bound on $\mathbb{E}\left[\sum_{m\in\mathcal{M}} Q_m\right]$ by bounding each term in (3.39)-(3.40). We omit the standard calculation here.

Theorem 3.8. *(***Upper bound***)*

$$\mathbb{E}\left[\sum_{m=1}^{M} \bar{Q}_m^{(\epsilon)}\right] \leq \frac{(\sigma^{(\epsilon)})^2 + \nu^2}{2\epsilon} + D_e^{(\epsilon)},$$

where $D_e^{(\epsilon)} = o(\frac{1}{\epsilon})$*, i.e.,* $\lim_{\epsilon\downarrow 0} \epsilon D_e^{(\epsilon)} = 0$.

Therefore, in the heavy-traffic limit, we have

$$\limsup_{\epsilon\downarrow 0} \epsilon\mathbb{E}\left[\sum_{m=1}^{M} \bar{Q}_m^{(\epsilon)}\right] \leq \frac{\sigma^2 + \nu^2}{2}.$$

The heavy-traffic optimality of the proposed algorithm follows by the coincidence of lower and upper bounds.

3.5 Evaluation

We evaluate the performance of Pandas in this section. In particular, we have integrated Pandas with the Hadoop FIFO scheduler and Fair scheduler (HFS). Each scheduler retains its original job priority. To focus on the performance benefit brought by Pandas to the data-processing phase, we use the SWIM workload [58] to obtain realistic characteristics of data-processing tasks, but with empty reduce phases, as the time taken by the reduce phase

can be orthogonally improved by other techniques [45, 46]. The workload study [61] that SWIM is based on also shows that 75% of jobs in the Facebook trace have no shuffle stage and the map outputs are directly written to the file system.

We describe the evaluation setup in 3.5.1, evaluate Pandas-accelerated FIFO scheduler against Hadoop FIFO scheduler in 3.5.2, and Pandas-accelerated Fair scheduler against HFS in 3.5.3. The overhead of Pandas is discussed in 3.5.4.

3.5.1 Evaluation Setup

We discuss the environment, trace characteristics, evaluation metrics and Pandas' thresholds setting in this section.

Environment

For both the Elastic Compute Cloud (EC2) [1] and a private cluster, we run a modified version of Hadoop-1.2.1, configured with a block size of 256 MB and a replication factor 3. We use 100 "m3.xlarge" instances on EC2 and 28 "m1.xlarge" instances on OpenStack [62] in the private cluster. Table 3.1 shows details of the instances.

Table 3.1: Types of instances used in the experiments.

	Nodes	Memory (GB)	VCPU	Map Slots
EC2	100	15	4	4
Private	28	16	8	4

1. EC2. We use EC2 for evaluation with a long trace characterized by hot-spots occurring and disappearing. As rack structures are not available in EC2, every other node is regarded as a remote node. Only the average remote slowdown σ_s/σ_l is computed, where σ_l is the average processing time (not including waiting time) of a local task, and σ_s is that of a remote task. The average is taken over all nodes in the system. Slowdown is measured by collecting local and remote task completion times under the Hadoop schedulers.

The remote slowdown was measured to be 2 on EC2 instances [12]. In our experiments, the slowdown varies with the placement of assigned VMs, and tends to increase with load and hot-spots, which lead to a higher level of network congestion and disk contention. The largest average slowdown we measured on EC2 is 6. Figure 3.3 shows the CDF of slowdown, defined as (processing time of a remote task / σ_l), where σ_l is the average processing time of local tasks. Although 75% of the remote tasks experience a slowdown less than 5, some of the remote tasks experience as much as 45x slowdown.

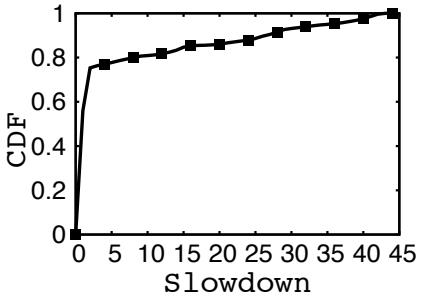


Figure 3.3: Slowdown distribution on EC2.

2. Private cluster. We use the private cluster for evaluation under stressed conditions, sensitivity analysis and short traces with fixed load. As all nodes in our private cluster are in the same rack and are exclusively used for the experiment, there is neither traffic from other VMs co-located on the same physical node as in a multi-tenant environment, nor background traffic replicating, redistributing and importing files. As a result, there is virtually no slowdown in the system. For our experiments, we send background traffic from each VM to its neighboring VM to create a slowdown.

3. Large-scale simulation. We use simulation with 500 nodes over a long time horizon to evaluate the performance of Pandas in a large cluster across all loads up to system capacity. It allows us to gain insight into the transient behavior observed with the long trace where hot-spot occurs and disappears. It also allows us to explore the performance of Pandas at load regions where experiments are not possible, as the FIFO scheduler and HFS crash due to excessive queuing of jobs. Pandas is able to avoid excessive queuing due to its throughput optimality.

Our simulation models task processing times as heavy-tailed random variables. We use a truncated Pareto distribution with shape parameter 1.9 to generate the number of tasks for each job. The parameter of HFS is tuned

according to HFS evaluation [12] such that 95% data locality is achieved. At each task arrival, a set of three nodes are chosen to be its local nodes. Uniform load is simulated by sampling the nodes uniformly at random. Skewed load is simulated by sampling from half the cluster with probability 0.8, and from the remaining half with probability 0.2. This mimics hot-spots in the presence of uniform traffic. The slowdown of the system is set to 2.

Trace characteristics

We generate traces by sampling jobs from the SWIM benchmark [58] so that 1) we preserve the Pareto job size distribution [12]; 2) The length of the trace and the number of files are appropriately scaled for the capacity of the different clusters by SWIM. We leave the reduce phases empty to focus on the improvement of the data-processing phase only, as the time taken by the reduce phase can be orthogonally improved by other techniques [45, 46].

We did not run the SWIM benchmark directly because SWIM does not model data popularity. It assumes that the file system is only populated with data accessed in this experiment and each task independently chooses its data location. This is different from what happens in a real cluster, where different tasks can access the same data block, hence creating correlated patterns in accesses. SWIM is essentially assuming that each job process a completely different file, which is not the case in practice. Therefore, the data popularity, as well as the node popularity, is always uniform. Moreover, unlike hot-spots caused by skewed data popularity, hot-spots caused by skewed node popularity occur randomly and are hard to reproduce as they depend on the random placement of data blocks by HDFS. A trace constructed with the same distribution of data popularity might cause a hot-spot in one experiment, but not in another.

In view of the above, we generate hot-spots by making a subset of nodes more popular than others. For all experiments on EC2 and the private cluster, hot-spots are generated by increasing the popularity of 40% of nodes in the system, resulting in a skewed node popularity. The actual node popularity varies with the load, which will be specified for each experiment.

Table 3.2 shows the job size distribution used for the long trace on EC2, and Table 3.3 shows the job size distribution used on the private cluster. They have the same Pareto distribution, but the trace for the private cluster

Table 3.2: Job size distribution used on EC2.

Bins	1	2	3	4	5	6	7
Job Count	570	240	210	120	90	90	60
Map Count per Job	1	2	10	50	100	200	400

Table 3.3: Job size distribution used on private cluster.

Bins	1	2	3	4	5	6	7	8
Job Count	237	95	77	55	42	37	30	27
Map Count per Job	1	2	4	10	25	50	100	200

has jobs of smaller sizes as the capacity of the private cluster is smaller.

Table 3.4: Trace characteristics on EC2.

Job Range	Node popularity, load
1-230	Uniform, 0.24
231-460	Uniform, 0.48
461-690	Uniform, 0.72
691-920	Skewed, 0.48
921-1150	Uniform, 0.48
1151-1380	Skewed, 0.24

Table 3.4 shows the long trace containing 6 stages with varying load and node popularity. Like the experiments on HFS [12], the inter-arrival time of jobs is generated from an exponential distribution. The uniform node popularity is obtained by using the default data placement of SWIM. The skewed node popularity, or hot-spots, are generated by directing all traffic to 40% of the nodes in the system.

Evaluation metrics

We use the following metrics for performance evaluation:

Map completion time. It is the average completion time of all map tasks in a trace or in a sliding window. Completion time of a map task is defined as the time interval between task arrival and the moment the task finishes processing. For the long trace, the average is computed in a sliding window of k jobs, i.e., the value at point i on x-axis shows the average computed over jobs $[i-k+1, i-k+2, \ldots i]$ for $i \geq k$ and over jobs $[1, 2, \ldots i]$ for $1 < i < k$.

This gives higher resolution to changes in map completion time as hot-spots occur and disappear.

Job completion time. It is the average completion time of all jobs in a trace or in a sliding window. Completion time of a job is defined as the time interval between job arrival and the moment the job finishes processing. The sliding window is defined in the same way as for map completion time above.

Data locality. It is measured as the percentage of map tasks processed at a local node.

Speed-up. Speed-up in job completion time is defined as

$$\text{Speed-up} = \frac{T - T_P}{T}, \tag{3.41}$$

where T is the original job completion time under FIFO or HFS, and T_P is that with Pandas acceleration. A negative speed-up indicates that the job is slowed down with Pandas. Note that, with this definition, the maximum possible speed-up is 100%, corresponding to $T_P = 0$, but the maximum possible slowdown is infinite. A 100% slowdown indicates that the job completion time has doubled.

Jobs in bins. For the detailed analysis on the private cluster, we divide jobs into three classes based on their sizes in Table 3.3: 1) Small jobs (placed in bin $1-3$, having less than 5 map tasks), 2) medium jobs (placed in bin $4-6$, having $10-50$ map tasks) and 3) large jobs (placed in bin $7-8$, having at least 100 map tasks).

Pandas thresholds

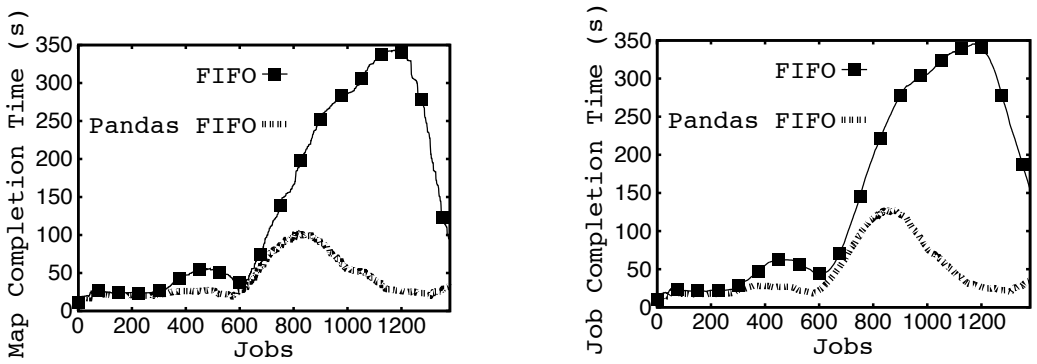
On both EC2 and the private cluster, only the remote threshold T_s is used due to the absence of rack structure. We set T_s based on the slowdown measured with the default Hadoop schedulers. In all experiments, the thresholds are set to fixed values for each trace, even if the average slowdown varies with changing loads and node popularity.

3.5.2 Pandas-accelerated FIFO

Long trace on EC2

We run a trace of 1380 jobs on 5000 files on EC2. The job size distribution is given in Table 3.2. The average slowdown measured with this set of VMs under FIFO is 1.5, hence we set the Pandas remote threshold T_s to 2.

Figure 3.4 shows the average task completion time and job completion time in a sliding window of 230 jobs. Before hot-spots occur, Pandas-accelerated FIFO outperforms FIFO at all times and the largest improvement of 2.3-fold reduction occurs for jobs $231 - 460$ at load 0.48.



(a) Task completion time

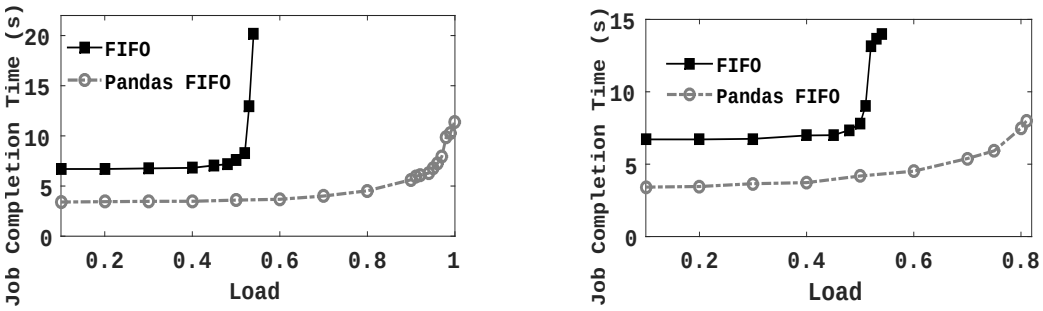
(b) Job completion time

Figure 3.4: Pandas-accelerated FIFO achieves up to 11-fold improvement over FIFO on EC2.

As soon as the hot-spots occur, the performance of FIFO degrades drastically. With 0.48 skewed load, the completion time with the FIFO Scheduler severely degrades, while Pandas-accelerated FIFO consistently produces low completion times. Even when the hot-spot disappears, its lingering effect makes the completion time with FIFO continue to increase sharply, while Pandas-accelerated FIFO produces much lower completion time under uniform load. For instance, with jobs $921 - 1150$, Pandas-accelerated FIFO achieves 11-fold improvement over FIFO. The completion time with FIFO improves gradually with the skewed load of 0.24. In this scenario, the load is skewed but low, hence contention occurs less frequently. The last set of jobs $1151 - 1380$ receives a 4.5-fold improvement in job completion time with Pandas-accelerated FIFO.

Large-scale simulation

The behavior of FIFO is clearer with simulation over the entire range of loads within system capacity. Figure 3.5a shows that with uniform data locality, FIFO incurs very large delay beyond 0.6 load. Figure 3.5b shows that with hot-spots, FIFO incurs very large delay beyond 0.52 load. FIFO's aggressive assignment of remote tasks and waste of throughput have resulted in instability. This explains the drastic increase in completion time under FIFO in Figure 3.4. In contrast, Pandas is throughput-optimal and heavy-traffic optimal, hence producing low delays for all loads up to capacity. For both scenarios, the corresponding standard variations are dramatically small (of order 10^{-3}), thus we do not show the error bars on the graph.



(a) Uniform load

(b) Skewed load

Figure 3.5: Average job completion time.

Detailed performance on private cluster

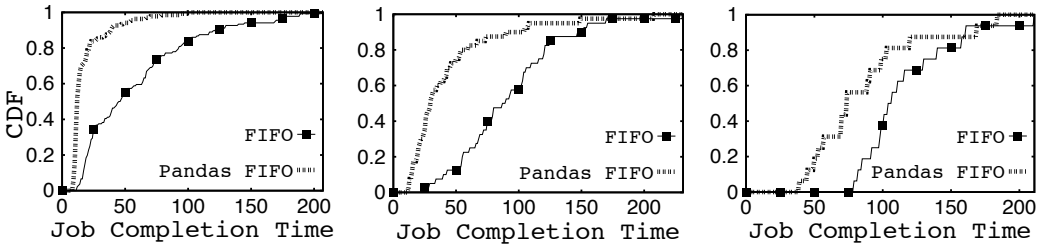
We run a short trace with the job size distribution in Table 3.3, but scaled to a total of 192 jobs and 3700 map tasks. The trace accesses 1000 files at 0.2 load. The threshold T_s is set to 2. Table 3.5 shows the average map task and job completion time for uniform and skewed loads. Even at such a low load, Pandas-accelerated FIFO achieves 2.38-fold and 2-fold improvements in average job completion time for uniform and skewed loads respectively.

We focus on uniform load. Similar behavior is observed under skewed load.

Figure 3.6 shows the CDFs of job completion time for the three classes of jobs. Pandas reduces almost all job completion times and only the largest jobs experience a slight increase in completion time. This is a result of the improvement in overall system efficiency by Pandas.

Table 3.5: Pandas-accelerated FIFO outperforms FIFO at 0.2 load.

	Average Map Completion Time (s)		Average Job Completion Time (s)	
Workload Behavior	FIFO	Pandas FIFO	FIFO	Pandas FIFO
Uniform	43.37	30.88	70.25	29.48
Skewed	73.44	58.12	143.3	71.86



(a) Bin 1-3 (b) Bin 4-6 (c) Bin 7-8

Figure 3.6: Average job completion time at 0.2 uniform load.

Figure 3.7 shows the data locality of FIFO and Pandas-accelerated FIFO for each bin. Pandas-accelerated FIFO achieves almost 100% data locality for all bins. Not surprisingly, the largest improvement is observed for the small jobs as they have the fewest choice of local nodes and are frequently assigned to remote nodes under FIFO. The improvement in system throughput by Pandas also leads to a larger number of idle nodes, hence higher data locality.

Figure 3.8 shows the speed-up in job completion times. We observe that most jobs experience a speed-up, with 50.4% of jobs experiencing at least a 60% speed-up, corresponding to 2.5-fold reduction in completion time, and 19.17% of jobs experiencing at least a 80% speed-up, corresponding to 5-fold reduction. Only 3.1% of jobs experience a slowdown, with the largest slowdown being 140%, corresponding to 2.4 times the completion time under FIFO.

Sensitivity analysis

We evaluate the impact of the variation in threshold values on performance. A remote threshold lower than the corresponding slowdown makes the scheduler assign remote tasks too aggressively. On the other hand, a remote threshold higher than the slowdown makes the scheduler too conservative, hence not relieving the hot-spots fast enough.

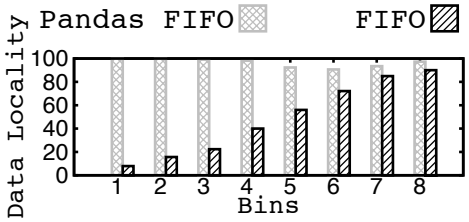


Figure 3.7: Data locality at 0.2 uniform load.

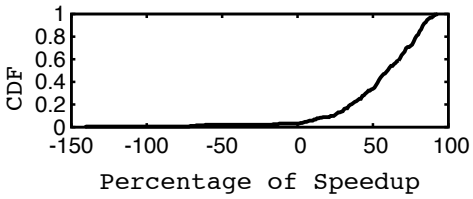


Figure 3.8: Speed-up of jobs at 0.2 uniform load.

Table 3.6 shows the average map and job completion time of Pandas-accelerated FIFO with threshold values 3, 5 and 7 when the average slowdown observed is 5. Even with an inaccurate threshold, we observe 9.25-fold improvement over FIFO in average job completion time while we achieve 11-fold improvement with the correct threshold.

3.5.3 Pandas-accelerated Fair

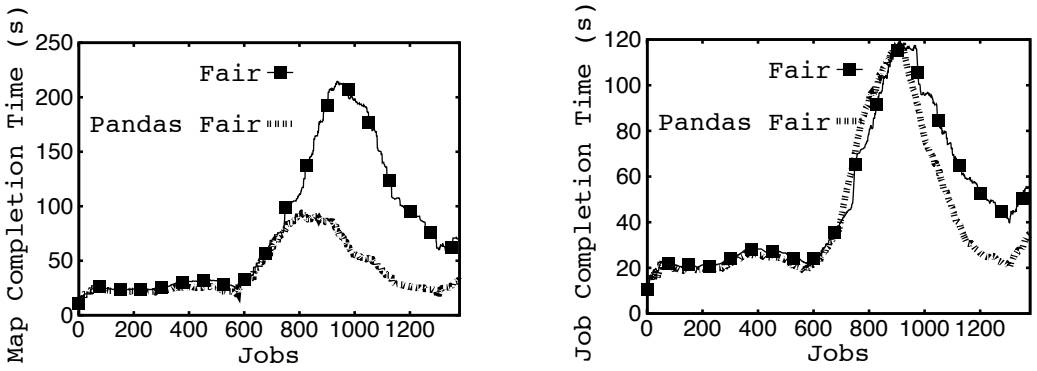
Long trace on EC2

We run the same trace as for FIFO in Section 3.5.2 with the same threshold $T_s = 2$. Figure 3.9 shows the average task completion time and job completion time in a sliding window of 230 jobs. Pandas' performance is comparable to HFS under uniform load and during the first hot-spot. However, after the first hot-spot, Pandas accelerates tasks by 45% and jobs by 47%.

With the default waiting time parameter, HFS is sufficiently aggressive in relieving hot-spots at this load although at the cost of wasting throughput and affecting later jobs. The acceleration of HFS is not as large as that of FIFO as HFS has a different capacity region and can accommodate a higher load than 0.48. Our experiment at a higher load crashed due to excessive queuing of jobs in the HFS scheduler, although Pandas-accelerated Fair did not suffer from queuing due to its throughput optimality. We explore the

Table 3.6: Pandas-accelerated FIFO with different threshold values at 0.3 skewed load.

	FIFO	Threshold = 3	Threshold = 5	Threshold = 7
Map (s)	159.23	42.67	38.17	43.72
Job (s)	381.26	41.2	34.75	38.96



(a) Task completion time

(b) Job completion time

Figure 3.9: Pandas-accelerated Fair achieves up to 47% improvement over HFS.

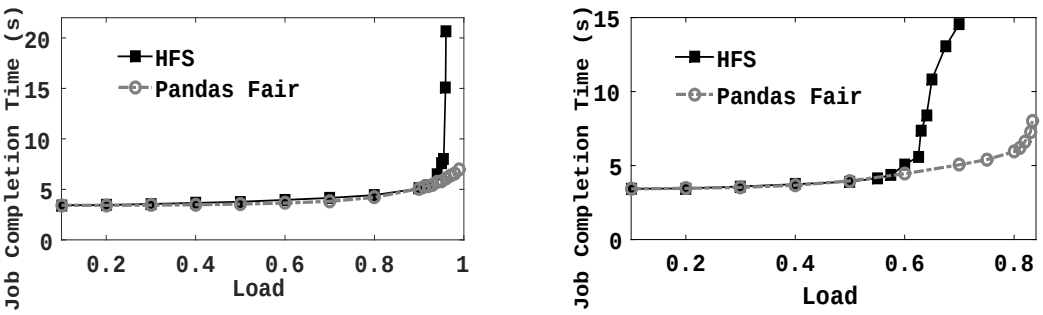
behavior of HFS and Pandas acceleration at higher loads using simulation in Section 3.5.3.

Large-scale simulation

Since the performance of HFS depends on the waiting time parameter, we tune the parameter according to HFS evaluation [12] so that HFS achieves 95% data locality. Figure 3.10a shows that with uniform node popularity, HFS incurs drastic delay only beyond 0.95 load. Figure 3.10b shows that with hot-spots, HFS incurs high delay beyond 0.65 load. At lower loads, Pandas achieves negligible acceleration, while beyond 0.95 and 0.65 loads respectively, Pandas achieves very large acceleration.

We observe that a high data locality of 95% favors uniform node popularity. When the waiting time parameter is tuned for lower data locality, HFS incurs large delays at a smaller load for uniform, but at a larger load for hot-spots, as in Figure 3.9. The waiting time parameter yields a trade-off of performance between the two scenarios. But in both cases, HFS has a larger capacity region than FIFO, and large improvement by Pandas will occur only at loads

beyond 0.48.



(a) Uniform load (b) Skewed load

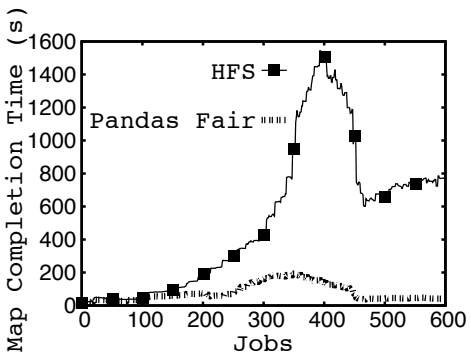
Figure 3.10: Average job completion time.

A stressed test on private cluster

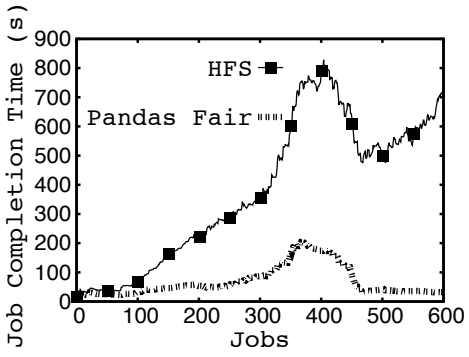
We run a trace of 600 jobs on 1000 files on the private cluster. Table 3.3 shows the job size distribution while Table 3.4 shows load and node popularity with 100 jobs per segment. The peak slowdown under HFS is 30, which is a stressed scenario with a large amount of network contention, or in an environment where the difference in processing time due to location is large such as with memory-locality. We set $T_s = 30$.

Figure 3.11 shows the average task completion time and job completion time in a sliding window of 100 jobs. Before hot-spots occur, Pandas-accelerated Fair outperforms HFS at all times and the largest improvement of 4.1-fold reduction in average job completion time occurs for jobs $201-300$, even if the average task completion time experiences only a 2.7-fold reduction.

As the first hot-spot occurs, the improvement in task completion time reaches 11-fold, while that in job completion time remains around 4.5-fold. Unlike FIFO in Figure 3.4, HFS recovers from the hot-spot much faster, leaving a conspicuous peak in the curve for jobs $301-400$. Another peak appears towards the end of the curve as the second hot-spot occurs. The improvements in task and job completion times reach 15 and 12-fold respectively for jobs $401-500$, as Pandas-accelerated Fair recovers from the hot-spot. The improvement reaches its maximum of 18 and 22-fold respectively at the second hot-spot.



(a) Task completion time



(b) Job completion time

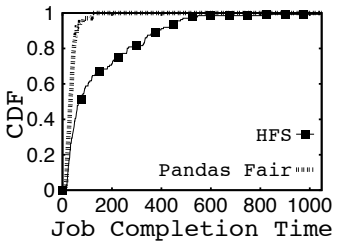
Figure 3.11: Pandas-accelerated Fair achieves up to 22-fold improvement over HFS on the private cluster.

Detailed performance on private cluster

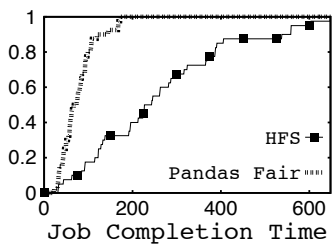
We run the same trace as in Section 3.5.2 at 0.68 load. We set T_s to 8 for uniform load and 27 for skewed load based on the average slowdown measured with HFS. Table 3.7 shows the average map and job completion times. Pandas-accelerated Fair achieves more than 3.3-fold improvement in average job completion time for both uniform and skewed loads.

Table 3.7: Performance at 0.68 load.

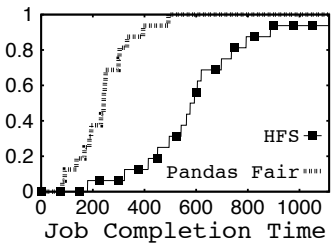
	Average Map Completion Time (s)		Average Job Completion Time (s)	
Workload Behavior	HFS	Pandas Fair	HFS	Pandas Fair
Uniform	194.9	85.05	209.71	61.93
Skewed	913.99	194.41	610.71	182.4



(a) Bin 1-3



(b) Bin 4-6



(c) Bin 7-8

Figure 3.12: Average job completion time at 0.68 uniform load.

We focus on uniform load. Similar behavior is observed under skewed load.

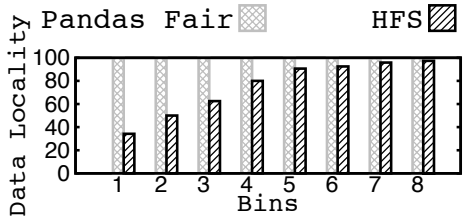


Figure 3.13: Data locality at 0.68 uniform load.

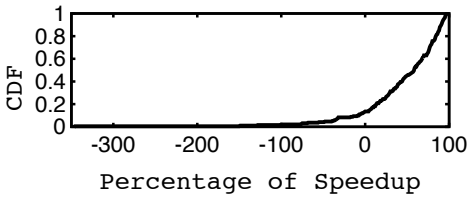


Figure 3.14: Speed-up at 0.68 uniform load.

Figure 3.12 shows the CDF of job completion time. Pandas produces significant improvement for all three classes of jobs. Figure 3.13 shows the data locality of HFS and Pandas-accelerated Fair for each bin. Pandas achieves close to 100% data locality whereas HFS, with its default configuration, achieves only $30 - 60\%$ data locality for the small jobs. We also plot the CDF of speedup for each job in Figure 3.14. Most jobs experience a speed-up, with 46.87% of jobs experiencing at least a 60% speed-up, corresponding to at least 2.5-fold reduction in completion time, and 24.48% of jobs experiencing at least a 80% speed-up, corresponding to at least 5-fold reduction. Only 13.02% of jobs experience a slowdown, with the largest slowdown being 346%, corresponding to 4.46 times the completion time under HFS.

Sensitivity analysis

Table 3.8 shows the average map and job completion time of Pandas-accelerated Fair with threshold values 5, 8 and 10 when the average slowdown observed is 8. Even with an inaccurate threshold, we observe 2.8-fold improvement over HFS in average job completion time while we achieve 3.38-fold improvement with the correct threshold.

Table 3.8: Pandas-accelerated Fair with different threshold values at 0.6 uniform load.

	HFS	Threshold = 5	Threshold = 8	Threshold = 10
Map (s)	194.9	103.34	85.05	115.59
Job (s)	209.71	71.43	61.93	73.74

3.5.4 Scheduler Overhead

The space overhead of Pandas is negligible as our data structures maintain pointers to tasks, rather than keeping multiple copies. Table 3.9 shows the average scheduling delay for the trace with 192 jobs on the private cluster. We observe that the delay is comparable across the schedulers, with Pandas-accelerated FIFO being the fastest.

Table 3.9: Scheduling delay.

Scheduler	FIFO	Pandas FIFO	Fair	Pandas Fair
Delay (ms)	0.96	0.75	0.81	1.15

3.6 Conclusion

In this chapter, we proposed a novel priority algorithm for near-data scheduling. We have shown that the proposed algorithm achieves throughput optimality and heavy-traffic delay optimality for *all* traffic scenarios. The prioritized service imposes challenges to the state space collapse analysis and makes the proof of heavy-traffic optimality go beyond applying the existing drift-based analysis. A novel ideal load decomposition is used to separate the system into subsystems that require distinct treatments. The algorithm is also shown to have superior performance in trace-driven experiments.

CHAPTER 4

BALANCED-PANDAS: SCHEDULING WITH MULTI-LEVEL DATA LOCALITY

In the previous chapter, we have studied scheduling with two-level data locality. However, multiple locality levels exist within and across data centers. In this chapter, we will focus on the scheduling with multi-level data locality. We found that going from two to three levels of locality changes the problem drastically, as a tradeoff between performance and throughput emerges. The priority algorithm presented in Chapter 3, which is both throughput and heavy-traffic optimal for two locality levels, is not even throughput-optimal with three locality levels. We defer detailed explanation to Section 4.1.

The JSQ-MaxWeight algorithm proposed by Wang et al. [37] solved the problem of per-task-type queue with MaxWeight when there are *two* locality levels. Like MaxWeight, JSQ-MaxWeight is throughput-optimal. However, it was shown to be heavy-traffic optimal only for a special traffic scenario where a server is either locally overloaded or receives zero local traffic. We explain in Section 4.2 that an extension of the JSQ-MaxWeight algorithm to three locality levels preserves its throughput optimality, but suffers from the same lack of heavy-traffic optimality in all but a special set of scenarios.

We propose balanced-Pandas, a novel algorithm that uses weighted-workload routing and priority service. The key insight is that throughput optimality requires the *workload* to be kept at the correct ratio at different queues, but the *composition* of the workload can be designed appropriately such that it is delay-optimal in the heavy-traffic regime. We note that this is the only known delay-optimal algorithm in the heavy-traffic regime when the arrival rates are unknown.

We state our results in the rest of this chapter for three-level locality. We consider a discrete-time model for the system, as described in Chapter 2. Within each time slot, a task is completed with probability α at a local server, β at a rack-local server, or γ at a remote server, with $\alpha > \beta > \gamma$. Our main contributions are as follows:

- We identify the capacity region of a system with three locality levels. The capacity is defined to be the set of arrival rate vectors under which the system can be stabilized by some scheduling algorithm.
- We extend the JSQ-MaxWeight algorithm [37] and show that it is throughput-optimal. It is heavy-traffic optimal for special traffic scenarios analogous to that with two-level locality [37].
- We establish the throughput optimality of our proposed algorithm.
- We establish the heavy-traffic optimality of our proposed algorithm. The priority service precludes the use of the L_2 norm Lyapunov drift. The main idea is the construction of a *multi-level* ideal load decomposition for each arrival rate vector, which resolves the problem encountered by Pandas.

4.1 A Performance-versus-Throughput Dilemma

Under Pandas presented in Chapter 3, each server maintains a queue that only receives tasks local to this server. The load balancing step balances tasks across their local queues. Each server serves a local task if its queue is not empty; otherwise it serves a remote task from the longest queue in the system. With two levels of locality, the priority algorithm achieves good delay performance as it maximizes the number of tasks served locally. The system is also throughput-optimal as any remaining capacity of an underloaded server is devoted to remote service.

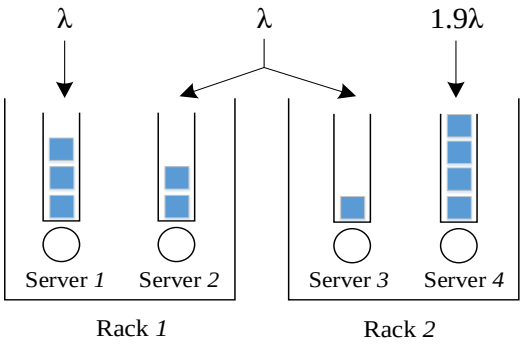


Figure 4.1: A simple system with two racks.

However, for a system with three levels of locality, the priority algorithm is not throughput-optimal. Consider a system with two racks, each consisting of two servers, as illustrated in Fig. 4.1. There are three types of tasks: one type of task is only local to server 1 and has rate λ, one is only local to server 4 and has rate 1.9λ, and the third type is local to both servers 2 and 3 and has rate λ. Assume a local task is served at rate $\alpha = 1$, a rack-local task is served at rate $\beta = 0.9$, and a remote task is served at rate $\gamma = 0.5$. With Pandas, the system is stable only if

$$1.9\lambda < \alpha + \beta(1 - \frac{0.5\lambda}{\alpha}) + \gamma(1 - \frac{\lambda}{\alpha}) + \gamma(1 - \frac{0.5\lambda}{\alpha}),$$

Thus the achievable throughput is $\lambda < 0.9355$, while the system is clearly stabilizable for $\lambda < 1$. The problem with the priority algorithm is that the shared local traffic of rate λ is split evenly among servers 2 and 3. However, the throughput will increase if server 3 serves no local tasks, but instead devotes its capacity to rack-local service for server 4.

While Pandas achieves good delay performance at low load when most tasks can be served locally, it sacrifices throughput at high load. This example raises the question as to whether there exist scheduling algorithms that can *simultaneously* achieve throughput and delay optimality.

4.1.1 Outer Bound of the Capacity Region

We consider a *decomposition* of the arrival rate vector $\boldsymbol{\lambda} = (\lambda_{\bar{L}} : \bar{L} \in \mathcal{L})$. For any task type $\bar{L} \in \mathcal{L}$, $\lambda_{\bar{L}}$ is decomposed into $(\lambda_{\bar{L},m}, m \in \mathcal{M})$, where $\lambda_{\bar{L},m}$ is assumed to be the arrival rate of type $\bar{L}$ tasks for server m. To ensure the arrival rate vector $\boldsymbol{\lambda}$ supportable, a necessary condition is that the sum of local, rack-local and remote load on any server is strictly less than 1, i.e.,

$$\sum_{\bar{L}:m\in\bar{L}} \frac{\lambda_{\bar{L},m}}{\alpha} + \sum_{\bar{L}:m\in\bar{L}_k} \frac{\lambda_{\bar{L},m}}{\beta} + \sum_{\bar{L}:m\in\bar{L}_r} \frac{\lambda_{\bar{L},m}}{\gamma} < 1. \tag{4.1}$$

Let Λ be the set of arrival rates such that each element has a decomposition satisfying condition (4.1):

$$\begin{aligned}\Lambda \;=\; & \big\{\boldsymbol{\lambda} = \big(\lambda_{\bar{L}} : \bar{L} \in \mathcal{L}\big) \,|\exists\ \lambda_{\bar{L},m} \geq 0,\ \forall \bar{L} \in \mathcal{L}, \forall m \in \mathcal{M},\ s.t. \\ & \lambda_{\bar{L}} = \sum_{m=1}^{M} \lambda_{\bar{L},m}, \forall \bar{L} \in \mathcal{L}, \\ & \sum_{\bar{L}:m\in\bar{L}} \frac{\lambda_{\bar{L},m}}{\alpha} + \sum_{\bar{L}:m\in\bar{L}_k} \frac{\lambda_{\bar{L},m}}{\beta} + \sum_{\bar{L}:m\in\bar{L}_r} \frac{\lambda_{\bar{L},m}}{\gamma} < 1, \forall m \in \mathcal{M}\}.\end{aligned}$$

Therefore Λ gives an outer bound of the capacity region.

4.2 Results on JSQ-MaxWeight

In this section, we summarize the results on an extension of the JSQ-MaxWeight algorithm proposed by Wang et al. [37].

We extend the JSQ-MaxWeight algorithm to a system with three levels of locality: local, rack-local and remote. The central scheduler maintains a set of M queues, where the m-th queue, denoted by Q_m, receives tasks local to server m. Let $\mathbf{Q} = (Q_1, Q_2, \cdots, Q_M)$ denote the vector of these queue lengths. The algorithm consists of JSQ routing and MaxWeight scheduling:

JSQ routing: When a task of type $\bar{L}$ arrives, the scheduler compares the lengths of the task's local queues, $\{Q_m | m \in \bar{L}\}$, and inserts the task into the shortest queue. Ties are broken randomly.

MaxWeight scheduling: When server m becomes idle, its scheduling decision $\eta_m(t)$ is chosen from the following set:

$$\arg\max_{n\in\mathcal{M}}\{\alpha Q_n(t) I_{\{n=m\}},\quad \beta Q_n(t) I_{\{K(n)=K(m)\}},\quad \gamma Q_n(t) I_{\{K(n)\neq K(m)\}}\}.$$

Ties are broken randomly.

Let $f_m(t)$ denote the working status of server m at time slot t,

$$f_m(t) = \begin{cases} -1 & \text{if server } m \text{ is idle} \\ n & \text{if server } m \text{ serves a task from queue } n. \end{cases}$$

Note that $f_m(t) = m$ indicates server m working on a local task. If $f_m(t) = n$, where $n \neq m$ and $K(n) = K(m)$, i.e., server n and server m are in the same

rack, server m is working on a rack-local task. Otherwise it is processing a remote task.

The arrivals to Q_m in time slot t are given by

$$A_m(t) = \sum_{\bar{L}:m\in\bar{L}} A_{\bar{L},m}(t),$$

where $A_{\bar{L},m}(t)$ is the number of type $\bar{L}$ tasks that are routed to Q_m.

Let $S_m^l(t)$, $R_m^k(t)$ and $R_m^r(t)$ denote the local, rack-local and remote service provided by server m respectively, where $S_m^l(t) \sim Bern(\alpha I_{\{\eta_m(t)=m\}})$, $R_m^k(t) \sim Bern(\beta I_{\{K(\eta_m(t))=K(m),\eta_m(t)\neq m\}})$ and $R_m^r(t) \sim Bern(\gamma I_{\{K(\eta_m(t))\neq K(m)\}})$ are Bernoulli random variables with varying probability.

Note that the local service *received* by queue Q_m is $S_m^l(t)$, whereas the rack-local service *received* by queue Q_m is $S_m^k(t) \equiv \sum_{n:K(n)=K(m),n\neq m} R_n^k(t) I_{\{\eta_n(t)=m\}}$, which is the sum of all rack-local service provided by other servers within the same rack as m to queue Q_m. Similarly, the remote service *received* by queue Q_m is given by $S_m^r(t) \equiv \sum_{n:K(n)\neq K(m)} R_n^r(t) I_{\{\eta_n(t)=m\}}$. Let $S_m(t) \equiv S_m^l(t) + S_m^k(t) + S_m^r(t)$ denote the departure process for queue m. Hence the queue length satisfy the following equation:

$$Q_m(t+1) \;\; = \;\; Q_m(t) + A_m(t) - S_m(t) + U_m(t),$$

where $U_m(t) = \max\{0, S_m(t) - A_m(t) - Q_m(t)\}$ is the unused service. As the service times follow geometric distributions, $\mathbf{Q}(t)$ together with the working status vector $\mathbf{f}(t)$ form an irreducible and aperiodic Markov chain $\{Z(t) = (\mathbf{Q}(t), \mathbf{f}(t)), t \geq 0\}$.

Theorem 4.1. *Any arrival rate vector strictly within Λ is supportable by JSQ-MaxWeight. Thus Λ is the capacity region of the system and JSQ-MaxWeight is throughput optimal.*

We use $V(t) = \|\mathbf{Q}(t)\|^2$ as the Lyapunov function. We show that there exists a positive integer T such that the T time slots drift of $V(t)$ is bounded within a finite subset of the state space and negative outside this subset. Then the result follows by the extension of the Foster-Lyapunov theorem.

For a system with only two levels of data locality, JSQ-MaxWeight algorithm has been shown to be heavy-traffic optimal for a special traffic scenario [37], where a server is either locally overloaded or receives zero local

traffic. For a system with the rack structure, hence three levels of locality, we consider the following traffic scenario:

All traffic concentrates on a subset of racks, and any rack with non-zero local tasks is overloaded. Moreover, any server in an overloaded rack either receives zero local traffic or is locally overloaded. Denote the set of racks that can have local tasks as $\mathcal{O}$, the set of servers that receives non-zero local traffic as $\mathcal{M}_l$, the set of servers that receives zero local traffic but belongs to racks $\mathcal{O}$ as $\mathcal{M}_k$, the set of servers in racks that receive zero local traffic as $\mathcal{M}_r$. For any subset of servers $\mathcal{S} \subset \mathcal{M}_l$, we denote by $\mathcal{N}(\mathcal{S}) = \{\bar{L} \in \mathcal{L} | \exists m \in \mathcal{S}, \text{ s.t. } m \in \bar{L}\}$ the set of task types with local servers in $\mathcal{S}$. Analogously, for any subset of racks $\mathcal{R} \subset \mathcal{O}$, denote by $\mathcal{N}(\mathcal{R}) = \{\bar{L} \in \mathcal{L} | \exists m, \text{ s.t. } m \in \bar{L}, \text{ and } K(m) \in \mathcal{R}\}$ the set of task types with local servers in racks $\mathcal{R}$. Let $\mathcal{M}_l^{(\mathcal{R})} = \{m \in \mathcal{M}_l | K(m) \in \mathcal{R}\}$ be the set of servers having local traffic and belonging to racks $\mathcal{R}$, and $\mathcal{M}_k^{(\mathcal{R})} = \{m \in \mathcal{M}_k | K(m) \in \mathcal{R}\}$ the set of servers without any local traffic and belonging to racks $\mathcal{R}$. Formally, the heavy-traffic regime assumes that for any $\mathcal{S} \subset \mathcal{M}_l$,

$$\sum_{\bar{L} \in \mathcal{N}(\mathcal{S})} \lambda_{\bar{L}} > |\mathcal{S}|\alpha,$$

and for any $\mathcal{R} \subset \mathcal{O}$,

$$\sum_{\bar{L} \in \mathcal{N}(\mathcal{R})} \lambda_{\bar{L}} > |\mathcal{M}_l^{(\mathcal{R})}|\alpha + |\mathcal{M}_k^{(\mathcal{R})}|\beta.$$

It is easy to see that in a stable system, $\sum_{\bar{L} \in \mathcal{L}} \lambda_{\bar{L}} < |\mathcal{M}_l|\alpha + |\mathcal{M}_k|\beta + |\mathcal{M}_r|\gamma$. We assume that

$$\sum_{\bar{L} \in \mathcal{L}} \lambda_{\bar{L}} = |\mathcal{M}_l|\alpha + |\mathcal{M}_k|\beta + |\mathcal{M}_r|\gamma - \epsilon, \tag{4.2}$$

where $\epsilon > 0$ characterizes the distance of the arrival rate vector from the capacity boundary.

Theorem 4.2. *Consider arrival processes $\{A_{\bar{L}}^{(\epsilon)}(t), t \geq 0\}_{\bar{L} \in \mathcal{L}}$ with arrival rate $\boldsymbol{\lambda}^{(\epsilon)}$ satisfying the above condition, then JSQ-MaxWeight is heavy-traffic optimal.*

Note that JSQ-MaxWeight is not heavy-traffic optimal in other traffic scenarios, when the underloaded racks, and the underloaded servers in over-

loaded racks, receive local traffic, for the same reason as with two levels of locality [37]. One problem is the growth of queues of local tasks at the servers that have zero local queue lengths in the special scenario. The growing queues of local tasks at the underloaded servers and racks result in non-optimal delay. Our balanced-Pandas solves this problem.

4.3 Algorithm

The balanced-Pandas algorithm is illustrated in Fig. 4.2. The central scheduler maintains a set of M queues, where the m-th queue consists of 3 subqueues denoted by Q_m^l, Q_m^k and Q_m^r, which receive tasks local, rack-local and remote to server m respectively. We denote by $\mathbf{Q}(t) = (\mathbf{Q}_1(t), \mathbf{Q}_2(t), \cdots, \mathbf{Q}_M(t))$ the queue lengths at time t, where $\mathbf{Q}_m(t) = (Q_m^l(t), Q_m^k(t), Q_m^r(t))$. We define the expected workload of the m-th queue, $W_m(t)$, as

$$W_m(t) = \frac{Q_m^l(t)}{\alpha} + \frac{Q_m^k(t)}{\beta} + \frac{Q_m^r(t)}{\gamma}.$$

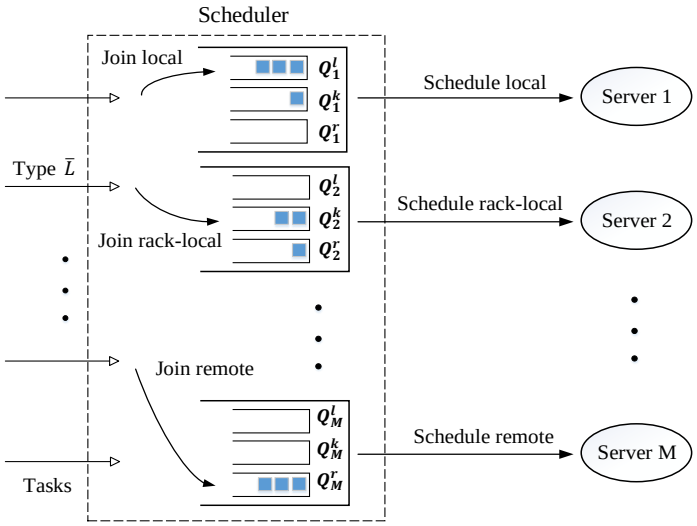


Figure 4.2: The balanced-Pandas algorithm.

At the beginning of each time slot t, the central scheduler routes new arrivals to one of the queues and schedules a new task for an idle server as follows:

Weighted-Workload queueing: When a task of type $\bar{L}$ arrives, the sched-

uler selects three queues that have the least workload among its local servers, rack-local servers and remote servers respectively. They are further weighted by $1/\alpha$, $1/\beta$, $1/\gamma$ respectively, and the task joins the queue with the minimum weighted workload. Ties are broken randomly. The task then joins the corresponding sub-queue depending on whether it is local, rack-local or remote for the selected server. Formally, the final selected queue $m^*(t)$ is in the set

$$\arg\min_{m\in\mathcal{M}} \left\{ \frac{W_m(t)}{\alpha} I_{\{m\in\bar{L}\}}, \frac{W_m(t)}{\beta} I_{\{m\in\bar{L}_k\}}, \frac{W_m(t)}{\gamma} I_{\{m\in\bar{L}_r\}} \right\}.$$

Prioritized scheduling: When a server becomes idle, it serves tasks from its queue in the order of local, rack-local and remote. For instance, both the local and rack-local sub-queues need to be empty before a remote task is served. When all its sub-queues are empty, the server remains idle.

4.3.1 Queue Dynamics

Let $A_{\bar{L},m}(t)$ denote the number of type $\bar{L}$ tasks that are routed to Q_m. The total number of tasks that join local sub-queue Q_m^l, rack-local sub-queue Q_m^k, and remote sub-queue Q_m^r, denoted by $A_m^l(t)$, $A_m^k(t)$ and $A_m^r(t)$, respectively, are given by $A_m^l(t) = \sum_{\bar{L}:m\in\bar{L}} A_{\bar{L},m}(t)$, $A_m^k(t) = \sum_{\bar{L}:m\in\bar{L}_k} A_{\bar{L},m}(t)$, $A_m^r(t) = \sum_{\bar{L}:m\in\bar{L}_r} A_{\bar{L},m}(t)$.

We denote the working status of server m at time slot t by $f_m(t)$:

$$f_m(t) = \begin{cases} -1, & \text{if server } m \text{ is idle} \\ 0, & \text{if server } m \text{ serves a local task from } Q_m^l \\ 1, & \text{if server } m \text{ serves a rack-local task from } Q_m^k \\ 2, & \text{if server } m \text{ serves a remote task from } Q_m^r \end{cases}$$

When server m completes a task at the end of time slot $t-1$, i.e., $f_m(t^-) = -1$, it is available for a new task at time slot t. The scheduling decision is based on the working status vector $\mathbf{f}(t) = (f_1(t), f_2(t), \cdots, f_M(t))$ and the queue length vector $\mathbf{Q}(t)$. Let $\eta_m(t)$ denote the scheduling decision for server m at time slot t. Note that $\eta_m(t) = f_m(t)$ for all busy servers, and when $f_m(t^-) = -1$, i.e., server m is idle, $\eta_m(t)$ is determined by the scheduler according to the algorithm.

Let $S_m^l(t)$, $S_m^k(t)$ and $S_m^r(t)$ be the local, rack-local and remote service provided by server m respectively, where $S_m^l(t) \sim Bern(\alpha I_{\{\eta_m(t)=0\}})$, $S_m^k(t) \sim Bern(\beta I_{\{\eta_m(t)=1\}})$ and $S_m^r(t) \sim Bern(\gamma I_{\{\eta_m(t)=2\}})$ are Bernoulli random variables with varying probability. For instance, $S_m^l(t) \sim Bern(\alpha)$ when server m is scheduled to its local sub-queue, and Bern(0) otherwise. The same applies to $S_m^k(t)$ and $S_m^r(t)$. Then the dynamics of three sub-queues at server m can be described as

$$\begin{aligned} Q_m^l(t+1) &= Q_m^l(t) + A_m^l(t) - S_m^l(t), \\ Q_m^k(t+1) &= Q_m^k(t) + A_m^k(t) - S_m^k(t), \\ Q_m^r(t+1) &= Q_m^r(t) + A_m^r(t) - S_m^r(t) + U_m(t), \end{aligned}$$

where $U_m(t) = \max\{0, S_m^r(t) - A_m^r(t) - Q_m^r(t)\}$ is the unused service. As the service times follow geometric distributions, $\mathbf{Q}(t)$ together with the working status vector $\mathbf{f}(t)$ form an irreducible and aperiodic Markov chain $\{Z(t) = (\mathbf{Q}(t), \mathbf{f}(t)), t \geq 0\}$.

4.3.2 Throughput Optimality

Theorem 4.3. *Balanced-Pandas is throughput optimal. That is, it stabilizes any arrival rate vector strictly within the capacity region.*

To prove Theorem 4.3, we use a Lyapunov function that is quadratic in the *expected workload* in each queue:

$$V(t) = \|\mathbf{W}(t)\|^2 = \sum_m \left(\frac{Q_m^l(t)}{\alpha} + \frac{Q_m^k(t)}{\beta} + \frac{Q_m^r(t)}{\gamma} \right)^2.$$

Note that the service discipline does not affect the proof as the expected *workload* is reduced at the same rate regardless of which sub-queue is served. The proof is similar to that for the throughput-optimality of JSQ-MaxWeight. The weighted-workload queueing effectively replaces the role of MaxWeight services, but leaves the choice of service discipline free for potential achievement of delay optimality.

4.4 Ideal Load Decomposition

A key component of the proof of heavy-traffic optimality of Balanced-Pandas is the construction of an *ideal load decomposition*, analogous to the method used for two levels of locality in the Chapter 3. However, the construction method with three levels of locality is more involved. Instead, we will use the alternative characterization of ideal load decomposition via two linear programs. The decomposition serves two purposes: 1) The ideal load obtained for each server is used as an intermediary in the proofs of state-space collapse; 2) The construction uniquely identifies four types of servers, helpers and beneficiaries in underloaded and overloaded racks respectively, which have very different traffic compositions and require distinct treatment in the proofs.

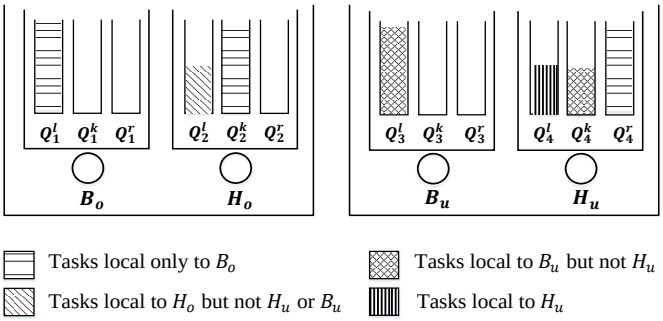


Figure 4.3: The queue compositions of the four types of servers.

Figure 4.3 illustrates the different sub-queue compositions of the four subsystems under the ideal load decomposition:

Helpers in underloaded racks, $\mathcal{H}_u$: A server belongs to $\mathcal{H}_u$ if it is *not overloaded*, *provides* rack-local service *and remote* service, and all tasks local to this server are served locally in the system.

Beneficiaries in underloaded racks, $\mathcal{B}_u$: A server belongs to $\mathcal{B}_u$ if it is *overloaded*, does *not provide* rack-local or remote service, and tasks local to this server receive *rack-local* service but *not remote* service.

Helpers in overloaded racks, $\mathcal{H}_o$: A server belongs to $\mathcal{H}_o$ if it is *not overloaded*, *provides* rack-local service but not remote service, and all tasks local to this server are served locally in the system.

Beneficiaries in overloaded racks, $\mathcal{B}_o$: A server belongs to $\mathcal{B}_o$ if it is *overloaded*, does *not provide* rack-local or remote service, and tasks local to this server receive *rack-local* service *and remote* service.

We will define overloaded servers and racks in a more precise manner in 4.4.2. While pure helpers and beneficiaries in underloaded or overloaded racks do not exist in a real system, the ideal load decomposition approximately depicts the load distribution in the heavy-traffic regime.

We characterize the ideal load decomposition in the rest of the section. Analogous to the two-level locality case, we will construct the ideal load decomposition via two linear programs: 1) Identify the overloaded servers and racks via routing optimization problem; 2) Construct the decomposition that produces $\mathcal{H}_u$, $\mathcal{B}_u$, $\mathcal{H}_o$, $\mathcal{B}_o$ via service optimization problem. In order to define the overloaded set, we will need an equivalent capacity region with a more refined decomposition similar to (3.1) for two-level locality.

4.4.1 An Equivalent Capacity Region

We define the following equivalent capacity region:

$$\begin{aligned}\bar{\Lambda} \;=\; & \big\{\boldsymbol{\lambda} = \big(\lambda_{\bar{L}} : \bar{L} \in \mathcal{L}\big) \,|\exists\; \lambda_{\bar{L},n,m} \geq 0,\; \forall \bar{L} \in \mathcal{L}, \forall n \in \bar{L}, \forall m \in \mathcal{M},\; s.t. \\ & \lambda_{\bar{L}} = \sum_{n:n\in\bar{L}} \sum_{m=1}^{M} \lambda_{\bar{L},n,m}, \forall \bar{L} \in \mathcal{L}, \\ & \sum_{\bar{L}:m\in\bar{L}} \sum_{n:n\in\bar{L}} \frac{\lambda_{\bar{L},n,m}}{\alpha} + \sum_{\bar{L}:m\in\bar{L}_k} \sum_{n:n\in\bar{L}} \frac{\lambda_{\bar{L},n,m}}{\beta} + \sum_{\bar{L}:m\in\bar{L}_r} \sum_{n:n\in\bar{L}} \frac{\lambda_{\bar{L},n,m}}{\gamma} < 1, \forall m \in \mathcal{M}\}.\end{aligned}$$

Each $\lambda_{\bar{L},m}$ is further decomposed into $\sum_n \lambda_{\bar{L},n,m}$, where $\lambda_{\bar{L},n,m}$ denotes the rate of type $\bar{L}$ tasks local to the server n but processed at server m. The additional index n provides a pseudo-distribution of tasks across their local servers only. It does not affect where they are processed. The information is used for identifying the overloaded set, which only depends on the types and rates of local tasks to a server.

The equivalence of the capacity region is established in similar ways as Lemma 3.2.

4.4.2 Overloaded Servers and Racks

We use the same notation as in Chapter 3. For any subset of servers $\mathcal{S} \subseteq \mathcal{M}$, we denote by $\mathcal{L}_{\mathcal{S}}$ the set of task types *local only* to servers in $\mathcal{S}$, and by $\mathcal{L}^*_{\mathcal{S}}$

the set of task types that have *at least* one local server in $\mathcal{S}$. With a slight abuse of notation, for any subset of racks $\mathcal{R} \subset \mathcal{K}$, we denote by $\mathcal{L}_{\mathcal{R}}$ the set of task types that are *local only* to servers in racks $\mathcal{R}$.

Given a decomposition $\{\lambda_{\bar{L},n,m}\}$ of $\boldsymbol{\lambda}$, let

$$\psi_n = \sum_{\bar{L}:n\in\bar{L}} \sum_{m=1}^{M} \lambda_{\bar{L},n,m}, \ \forall n$$

denote the pseudo-arrival rate of local tasks to server n. We define the *overloaded racks* in a similar way as *overloaded servers* with two-level locality. A rack k is overloaded under a decomposition $\{\lambda_{\bar{L},n,m}\}$ if

$$\sum_{m:K(m)=k,\psi_m\geq\alpha} (\psi_m - \alpha) \geq \beta \sum_{i:K(i)=k,\psi_i<\alpha} (1 - \frac{\psi_i}{\alpha}). \tag{4.3}$$

Note that the LHS of (4.3) gives the amount of local traffic for overloaded servers in rack k that could not be served locally. The RHS of (4.3) is the maximum rack-local service that can be provided by underloaded servers. Hence rack k requires remote service if Eq. (4.3) holds.

We define the rack load $\rho_k\left(\{\lambda_{\bar{L},n,m}\}\right)$ in the same way as (3.3) for two-level locality, which is the minimum possible lower bound on the total utilization of servers needed in order to accommodate arrivals *routed* to servers in rack-k according to $\{\lambda_{\bar{L},n,m}\}$: for any underloaded rack-k,

$$\rho_k = \sum_{\substack{m:K(m)=k\\ \psi_m<\alpha}} \frac{\psi_m}{\alpha} + \sum_{\substack{m:K(m)=k\\ \psi_m\geq\alpha}} (1 + \frac{\psi_m - \alpha}{\beta});$$

for any overloaded rack-k,

$$\rho_k = |\{m : K(m) = k\}| + \frac{1}{\gamma}\left[\sum_{\substack{m:K(m)=k\\ \psi_m\geq\alpha}} (\psi_m - \alpha) - \beta \sum_{\substack{m:K(m)=k\\ \psi_m<\alpha}} (1 - \frac{\psi_m}{\alpha})\right].$$

We then define the system load as

$$\rho\left(\{\lambda_{\bar{L},n,m}\}\right) = \sum_{k\in\mathcal{K}} \rho_k\left(\{\lambda_{\bar{L},n,m}\}\right).$$

Consider the *routing optimization problem:*

$$\min_{\{\lambda_{\bar{L},n,m}\}} \rho\left(\{\lambda_{\bar{L},n,m}\}\right)$$

subject to

$$\lambda_{\bar{L},n,m} \geq 0,\ \forall \bar{L} \in \mathcal{L}, \forall n \in \bar{L}, \forall m \in \mathcal{M}, \tag{4.4}$$

$$\lambda_{\bar{L}} = \sum_{n:n\in\bar{L}} \sum_{m=1}^{M} \lambda_{\bar{L},n,m}, \forall \bar{L} \in \mathcal{L}, \tag{4.5}$$

$$\sum_{\bar{L}:m\in\bar{L}} \sum_{n:n\in\bar{L}} \frac{\lambda_{\bar{L},n,m}}{\alpha} + \sum_{\bar{L}:m\in\bar{L}_k} \sum_{n:n\in\bar{L}} \frac{\lambda_{\bar{L},n,m}}{\beta} + \sum_{\bar{L}:m\in\bar{L}_r} \sum_{n:n\in\bar{L}} \frac{\lambda_{\bar{L},n,m}}{\gamma} < 1, \forall m \in \mathcal{M}. \tag{4.6}$$

Let $\left(\{\tilde{\lambda}_{\bar{L},n,m}\}, \rho^*\right)$ be any fixed optimal solution of this linear program. Then $\{\tilde{\lambda}_{\bar{L},n,m}\}$ gives a set of *overloaded racks* $\mathcal{O}$ satisfying (4.3), and a set of *overloaded servers* $\mathcal{D}$, both of which are unique for the given $\boldsymbol{\lambda}$.

4.4.3 Ideal Load Decomposition

Next we will formally define the four types of servers. Given a decomposition $\{\tilde{\lambda}_{\bar{L},n,m}\}$ of $\boldsymbol{\lambda} \in \Lambda$ that minimizes ρ, we denote the utilization of each server m by

$$w_m = \sum_{\bar{L}:m\in\bar{L}} \sum_{n:n\in\bar{L}} \frac{\tilde{\lambda}_{\bar{L},n,m}}{\alpha} + \sum_{\bar{L}:m\in\bar{L}_k} \sum_{n:n\in\bar{L}} \frac{\tilde{\lambda}_{\bar{L},n,m}}{\beta} + \sum_{\bar{L}:m\in\bar{L}_r} \sum_{n:n\in\bar{L}} \frac{\tilde{\lambda}_{\bar{L},n,m}}{\gamma}.$$

Let $\mathcal{O}$ and $\mathcal{U}$ denote the set of overloaded and underloaded racks determined by the routing optimization problem, respectively. We denote the set of overloaded servers in racks $\mathcal{O}$ by $\mathcal{D}_o$. We define the system remaining capacity as

$$\begin{aligned} C_R\left(\{\tilde{\lambda}_{\bar{L},n,m}\}\right) \ = \ & \sum_{k\in\mathcal{U}} \sum_{\substack{m:K(m)=k \\ \psi_m<\alpha}} \gamma(1-w_m) + \sum_{k\in\mathcal{O}} \sum_{\substack{m:K(m)=k \\ \psi_m<\alpha}} \beta(1-w_m) \\ & + \sum_{k\in\mathcal{O}} \sum_{\substack{m:K(m)=k \\ \psi_m\geq\alpha}} \alpha(1-w_m), \end{aligned} \tag{4.7}$$

which is the maximum amount by which $\sum_{\bar{L}\in\mathcal{L}_{\mathcal{D}_o}} \lambda_{\bar{L}}$ can be increased until the boundary of the capacity region is hit.

Analogously, the ideal load decomposition for the multi-level locality is determined by the following linear program, which we refer to as the *service optimization problem:*

$$\max_{\{\lambda_{\bar{L},n,m}\}} C_R\left(\{\lambda_{\bar{L},n,m}\}\right)$$

subject to constraints (4.4)-(4.6).

Let $\left(\{\lambda^*_{\bar{L},n,m}\}, C^*_R\right)$ be any fixed optimal solution of this linear program. Then under this optimal decomposition, all servers are classified into the following four subsystems:

$$\begin{aligned}
\mathcal{H}_o &= \{n : K(n) \in \mathcal{O} | \psi_n < \alpha\},\\
\mathcal{B}_o &= \{n : K(n) \in \mathcal{O} | \psi_n \geq \alpha\},\\
\mathcal{H}_u &= \{n : K(n) \in \mathcal{U} | \psi_n < \alpha\},\\
\mathcal{B}_u &= \{n : K(n) \in \mathcal{U} | \psi_n \geq \alpha\}.
\end{aligned}$$

Remark. Recall that the decomposition that identifies overloaded servers $\mathcal{D}$ for systems with two-level locality, satisfies the following property: all shared local traffic between overloaded and underloaded servers are routed to underloaded servers. Similarly, the decomposition that achieves ρ^* have some special property, which depends on the relationship between α, β and γ.

An interesting case is $\beta^2 > \alpha\gamma$, which implies that the rack-local rate is significantly larger than the remote rate. This condition holds in MapReduce cluster.

Consider an overloaded rack and an underloaded rack. Suppose there exists traffic that are local to both racks. The condition $\beta^2 > \alpha\gamma$ dictates that all such traffic should be moved to the underloaded rack in the ideal load decomposition *regardless of the load* on the servers. For instance, moving Δ amount of traffic from $\mathcal{H}_o$ to $\mathcal{H}_u$ creates new capacity for $\mathcal{H}_o$ so that it can serve an additional $\beta\frac{\Delta}{\alpha}$ amount of rack-local traffic in the overloaded rack. On the other hand, when a server in $\mathcal{H}_u$ becomes overloaded (and hence becomes $\mathcal{B}_u$), the movement creates new rack-local traffic in the underloaded rack and as a result reduces a $\gamma\frac{\Delta}{\beta}$ amount of remote traffic served in this rack. The condition $\beta^2 > \alpha\gamma$ implies that $\beta\frac{\Delta}{\alpha} > \gamma\frac{\Delta}{\beta}$, i.e., the increase in rack-local

capacity outweighs the decrease in remote capacity. Hence the movement of shared local traffic continues even if a server changes from $\mathcal{H}_u$ to $\mathcal{B}_u$. And in the ideal load decomposition constructed, no shared local traffic between underloaded racks and overloaded racks is routed to overloaded racks.

In other words, the condition $\beta^2 > \alpha\gamma$ ensures that the sacrifice of local service for rack-local service benefits the system capacity by reducing the amount of traffic that should be served remotely. And the final load decomposition is ideal in the sense that it *minimizes* the amount of remote traffic.

4.5 Heavy-traffic Optimality

In this section, we establish the heavy-traffic optimality of balanced-Pandas. The proof follows the framework developed in [59], which we have used to prove the heavy-traffic optimality of Pandas in Chapter 3. However, the Lyapunov drift analysis developed cannot be applied directly to our algorithm due to the prioritized service and a more complicated state-space collapse.

Traffic distributions

The traffic distribution $(\boldsymbol{\lambda} = (\lambda_{\bar{L}} : \bar{L} \in \mathcal{L}))$ on the system can be classified into two categories: the set of overloaded racks $\mathcal{O} = \emptyset$, or $\mathcal{O} \neq \emptyset$. In the first case, each rack can accommodate its load, and the system in the heavy-traffic regime decomposes into independent racks, each of which has two levels of locality. We focus on the second case in this chapter, which is more challenging of the two, and defer the proof for the first case to Appendix B. In particular, we consider the heavy-traffic regime such that $\mathcal{B}_u = \emptyset$, i.e., all servers in underloaded racks can accommodate their arrivals locally.

Figure 4.4 illustrates the one-dimensional state-space vector the system collapses to in the heavy-traffic regime when $\mathcal{B}_u = \emptyset$. There are two key ideas. First, the prioritized service allows us to have a uniformly bounded *helper subsystem* in the heavy-traffic regime, which corresponds to the disappearance of the rack-local and local queues for $\mathcal{H}_u$ and that of the local queue for $\mathcal{H}_o$ in Figure 4.4. Second, the weighted-workload routing distributes the tasks local only to $\mathcal{B}_o$ in the ratio of $\alpha : \beta : \gamma$ in terms of server workload across $\mathcal{B}_o$, $\mathcal{H}_o$ and $\mathcal{H}_u$.

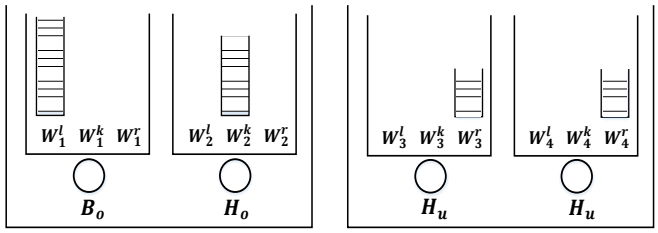


Figure 4.4: The queue compositions of the three types of servers in the heavy-traffic regime with $\alpha : \beta : \gamma = 1 : 0.8 : 0.5$. The workload at the three types of servers maintain the ratio $\alpha : \beta : \gamma = 1 : 0.8 : 0.5$.

4.5.1 Formal Statement of Results

We formally state the main theorems in this subsection and provide the outline of proofs in Section 4.5.2.

Consider the traffic regime such that there exist a set of overloaded racks. Moreover, these racks are truly overloaded in the sense that remote service is required for each rack. Formally, there exists an ideal load decomposition such that the pseudo-arrival rates for any overloaded rack is strictly greater than its capacity. That is, for any rack $k \in \mathcal{O}$,

$$\sum_{m:K(m)=k,\psi_m \geq \alpha} (\psi_m - \alpha) > \beta \sum_{i:K(i)=k,\psi_i<\alpha} (1 - \frac{\psi_i}{\alpha}), \tag{4.8}$$

where $\psi_m = \sum_{\bar{L}:n\in\bar{L}} \sum_{m=1}^{M} \lambda^*_{\bar{L},n,m}$. We refer to this condition as the *heavy rack overloaded traffic assumption.*

Assume that

$$\mathcal{B}_u = \emptyset. \tag{4.9}$$

The local traffic on $\mathcal{H}_u$ and $\mathcal{H}_o$ is assumed to satisfy

$$\sum_{\bar{L}\in\mathcal{L}^*_{\mathcal{H}_u}} \lambda_{\bar{L}} \equiv \Phi_u \alpha, \sum_{\bar{L}\in\mathcal{L}_{\mathcal{H}_o}} \lambda_{\bar{L}} \equiv \Phi_o \alpha, \tag{4.10}$$

where $\mathcal{L}^*_{\mathcal{H}_u} = \{\bar{L} : \exists m \in \mathcal{H}_u \text{ s.t. } m \in \bar{L}\}$, $\mathcal{L}_{\mathcal{H}_o} = \{\bar{L} : \forall m \in \bar{L}, m \in \mathcal{H}_o \cup \mathcal{B}_o, \text{ and } \exists n \in \mathcal{H}_o \text{ s.t. } n \in \bar{L}\}$, $0 \leq \Phi_o < |\mathcal{H}_o|$, and $0 \leq \Phi_u < |\mathcal{H}_u|$. In

addition, we assume that

$$\sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}_o}} \lambda_{\bar{L}} = |\mathcal{B}_o|\alpha + \beta(|\mathcal{H}_o| - \Phi_o) + \gamma(|\mathcal{H}_u| - \Phi_u) - \epsilon, \tag{4.11}$$

where $\epsilon > 0$ characterizes the distance of the arrival rate vector from the capacity boundary. We make the additional assumption that the $\{\lambda_{\bar{L}} : \bar{L} \in \mathcal{L}^*_{\mathcal{H}_u} \cup \mathcal{L}_{\mathcal{H}_o}\}$ are independent of ϵ. That is, the total local load for helpers is fixed. This assumption can be removed with more care. We now state the heavy traffic assumption as follows.

Assumption 3 (Assumption for the heavy rack overloaded traffic). Consider the arrival processes $\{A^{(\epsilon)}_{\bar{L}}(t), t \geq 0\}_{\bar{L}\in\mathcal{L}}$, parameterized by $\epsilon > 0$, with mean arrival rate vector $\boldsymbol{\lambda}^{(\epsilon)}$ satisfying conditions (4.8)-(4.11). Arrivals local to helpers $\{A^{(\epsilon)}_{\bar{L}}(t), t \geq 0\}_{\bar{L}\in\mathcal{L}^*_{\mathcal{H}}}$ are independent of ϵ. We denote by $(\sigma^{(\epsilon)})^2$ the variance of the number of arrivals that are only local to beneficiaries in overloaded racks, i.e., $\mathrm{Var}\Big(\sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}_o}} A^{(\epsilon)}_{\bar{L}}(t)\Big) = (\sigma^{(\epsilon)})^2$, which converges to σ^2 as $\epsilon \downarrow 0$.

Let $\{Z^{(\epsilon)}(t) = (\mathbf{Q}^{(\epsilon)}(t), \mathbf{f}^{(\epsilon)}(t)), t \geq 0\}$ be the system state under balanced-Pandas when the arrival rate is $\boldsymbol{\lambda}^{(\epsilon)}$. Since $\boldsymbol{\lambda}^{(\epsilon)} \in \Lambda$, the Markov chain $Z^{(\epsilon)}(t)$ is positive recurrent and has a steady state distribution. We denote the steady state queue-length vector by $\bar{\mathbf{Q}}^{(\epsilon)}$. All theorems in this section concern the *steady-state* queueing process $\bar{\mathbf{Q}}^{(\epsilon)}$ under balanced-Pandas, with the arrival processes $\{A^{(\epsilon)}_{\bar{L}}(t), t \geq 0\}_{\bar{L}\in\mathcal{L}}$, parameterized by $\epsilon > 0$, satisfying Assumption 3.

Theorem 4.4. *(***Helper queues***) There exist two sequence of finite numbers* $\{N_r : r \in \mathbb{N}\}$ *and* $\{N'_r : r \in \mathbb{N}\}$ *such that for each positive integer* r*,*

$$\mathbb{E}\left[\sum_{m\in\mathcal{H}_u} \left(\bar{Q}^{l(\epsilon)}_m + \bar{Q}^{k(\epsilon)}_m\right)\right] \leq N_r, \quad \mathbb{E}\left[\sum_{m\in\mathcal{H}_o} \bar{Q}^{l(\epsilon)}_m\right] \leq N'_r.$$

Therefore,

$$\lim_{\epsilon\downarrow 0} \epsilon\mathbb{E}\left[\sum_{m\in\mathcal{H}_u} \left(\bar{Q}^{l(\epsilon)}_m + \bar{Q}^{k(\epsilon)}_m\right)\right] = 0, \quad \lim_{\epsilon\downarrow 0} \epsilon\mathbb{E}\left[\sum_{m\in\mathcal{H}_o} \bar{Q}^{l(\epsilon)}_m\right] = 0.$$

Theorem 4.4 states that the helper subsystem is uniformly bounded and independent of ϵ. As the arrival rate approaches the capacity boundary, i.e., $\epsilon \downarrow$

0, the steady state mean queue length $\mathbb{E}\left[\left\|\bar{\mathbf{Q}}\right\|\right] = \mathbb{E}\left[\sum_m (\bar{Q}_m^l + \bar{Q}_m^k + \bar{Q}_m^r)\right] \to \infty$. In order to characterize the scaling order of $\mathbb{E}\left[\left\|\bar{\mathbf{Q}}\right\|\right]$, by Theorem 4.4, we only need to consider

$$\Psi = \sum_{m\in\mathcal{H}_u} Q_m^r + \sum_{m\in\mathcal{H}_o} \left(Q_m^k + Q_m^r\right) + \sum_{m\in\mathcal{B}_o} \left(Q_m^l + Q_m^k + Q_m^r\right).$$

The following theorem gives an lower bound on $\mathbb{E}\left[\Psi^{(\epsilon)}\right]$.

Theorem 4.5. *(***Lower bound***)*

$$\mathbb{E}\left[\Psi^{(\epsilon)}\right] \geq \frac{(\sigma^{(\epsilon)})^2 + (\nu^{(\epsilon)})^2 + \epsilon^2}{2\epsilon} - \frac{M}{2}.$$

Therefore, in the heavy traffic limit as $\epsilon \downarrow 0$*,*

$$\liminf_{\epsilon\downarrow 0} \epsilon\mathbb{E}\left[\Psi^{(\epsilon)}\right] \geq \frac{\sigma^2 + \nu^2}{2}. \tag{4.12}$$

In order to obtain an upper bound on $\mathbb{E}\left[\Psi^{(\epsilon)}\right]$, we first need to show that the steady-state weighted queue-length vector $\mathbf{W}$ collapses to a particular direction. Define $\mathbf{c} = \frac{\tilde{\mathbf{c}}}{\|\tilde{\mathbf{c}}\|} \in \mathbb{R}_+^M$ as a vector with unit l_2 norm, where

$$\tilde{c}_m = \begin{cases} \gamma, & \forall m \in \mathcal{H}_u \\ \beta, & \forall m \in \mathcal{H}_o \\ \alpha, & \forall m \in \mathcal{B}_o \end{cases}$$

The parallel and perpendicular components of the steady-state weighted queue-length vector $\mathbf{W}$ with respect to $\mathbf{c}$ are

$$\mathbf{W}_{||} = \langle \mathbf{c}, \mathbf{W}\rangle \mathbf{c}, \quad \mathbf{W}_{\perp} = \mathbf{W} - \mathbf{W}_{||}.$$

The following theorem states that $\mathbf{W}$ collapses to the direction $\mathbf{c}$ in the sense that its parallel component with respect to $\mathbf{c}$ is bounded, independent of heavy-traffic parameter ϵ.

Theorem 4.6. *(***State space collapse***) There exists a sequence of finite numbers* $\{C_r : r \in \mathbb{N}\}$ *such that for each positive integer* r*,*

$$\mathbb{E}\left[\|\mathbf{W}_{\perp}\|^r\right] \leq C_r,$$

that is, the deviation of $\mathbf{W}$ *from the direction* $\mathbf{c}$ *are bounded and independent of the heavy-traffic parameter* ϵ.

Theorem 4.7. *(***Upper bound***)*

$$\mathbb{E}\left[\Psi^{(\epsilon)}\right] \leq \frac{(\sigma^{(\epsilon)})^2 + (\nu^{(\epsilon)})^2}{2\epsilon} + B^{(\epsilon)},$$

where $B^{(\epsilon)} = o(\frac{1}{\epsilon})$, *i.e.,* $\lim_{\epsilon \downarrow 0} \epsilon B^{(\epsilon)} = 0$. *Therefore, in the heavy-traffic limit, we have*

$$\limsup_{\epsilon \downarrow 0} \epsilon \mathbb{E}\left[\Psi^{(\epsilon)}\right] \leq \frac{\sigma^2 + \nu^2}{2},$$

which coincides with the lower bound (4.12).

4.5.2 Outline of Proofs

(Theorem 4.4.) We first show that in steady state, the expected local load on any helper is upper bounded by a constant $\bar{\rho}_h < 1$ which is independent of ϵ. As shown in Chapter 3, with upper-bounded local load and priority scheduling for local tasks, the expected local queue length is bounded and independent of ϵ. Therefore the local sub-queue lengths of $\mathcal{H}_u$ and $\mathcal{H}_o$ are bounded and independent of ϵ. Under the ideal load decomposition, all tasks of types $\mathcal{L}^*_{\mathcal{H}_u}$ are served locally by $\mathcal{H}_u$ in order to achieve maximum remote capacity for overloaded racks. We can show that in the absence of $\mathcal{B}_u$, the number of tasks in $\mathcal{L}^*_{\mathcal{H}_u}$ that are served rack-locally or remotely vanishes as $\epsilon \downarrow 0$. Therefore we can also show the uniform boundedness of the rack-local sub-queue lengths of $\mathcal{H}_u$.

(Theorem 4.5.) In order to obtain a lower bound on $\mathbb{E}\left[\Phi^{(\epsilon)}\right]$, we construct a single server system $\Psi^{(\epsilon)}(t)$ with an arrival process $\left\{\sum_{\bar{L} \in \mathcal{L}_{\mathcal{B}_o}} A^{(\epsilon)}_{\bar{L}}(t), t \geq 0\right\}$ and a service process $\{b^{(\epsilon)}(t), t \geq 0\}$, which defined as follows:

$$b^{(\epsilon)}(t) \;=\; \sum_{i \in \mathcal{B}_o} X_i(t) + \sum_{j \in \mathcal{H}_o} Y_j(t) + \sum_{n \in \mathcal{H}_u} V_n(t), \tag{4.13}$$

where $\{X_i(t)\}_{i \in \mathcal{B}_o}$, $\{Y_j(t)\}_{j \in \mathcal{H}_o}$ and $\{V_n(t)\}_{n \in \mathcal{H}_u}$ are independent and each process is i.i.d. For all $i \in \mathcal{B}_o$, $X_i(t) \sim \text{Bern}(\alpha)$. For all $j \in \mathcal{H}_o$, $Y_j(t) \sim \text{Bern}(\beta(1 - \rho^l_j))$, where ρ^l_j is the proportion of time helper j spends on local

tasks in steady state. For all $n \in \mathcal{H}_u$, $V_n(t) \sim \text{Bern}(\gamma(1-\rho_n))$, where ρ_n is the proportion of time helper n spends on local and rack-local tasks in steady state. We denote $\text{Var}(b^{(\epsilon)}(t))$ by $(\nu^{(\epsilon)})^2$, which converges to a constant ν^2 as $\epsilon \downarrow 0$. The definition of X_i, Y_j and V_n is such that $\mathbb{E}\left[\sum_{i\in\mathcal{B}_o} X_i(t)\right]$, $\mathbb{E}\left[\sum_{j\in\mathcal{H}_o} Y_j(t)\right]$ and $\mathbb{E}\left[\sum_{n\in\mathcal{H}_u} V_n(t)\right]$ are the maximum amount of local, rack-local and remote services that can be provided for $\sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}_o}} A^{(\epsilon)}_{\bar{L}}(t)$. Then in steady state, $\Phi^{(\epsilon)}(t)$ is stochastically smaller than $\Psi^{(\epsilon)}(t)$. Using Lemma 4 in [59], we can obtain a lower bound on $\mathbb{E}\left[\Psi^{(\epsilon)}\right]$.

(Theorem 4.6.) We consider the Lyapunov function

$$F_\perp(Z) = \|\mathbf{W}_\perp\|.$$

We can show that the drift of $F_\perp(Z)$ is always finite and becomes negative for sufficiently large $F_\perp$. According to Lemma 3.6, all moments of $F_\perp(Z)$ exist and are finite. The main challenge is to show that the ideal load decomposition $\left\{\lambda^*_{\bar{L},n,m}\right\}$ satisfies: $\forall \bar{L} \in \mathcal{L}_{\mathcal{B}_o}$, $\forall m \in \{i \in \mathcal{M} | i \in \bar{L},$ or $i \in \mathcal{H}_u,$ or $i \in \bar{L}_k \cap \mathcal{H}_o\}$, $\sum_{n\in\bar{L}} \lambda^*_{\bar{L},n,m} \geq \kappa$, where κ is a positive constant independent of ϵ. That is, each task type only local to $\mathcal{B}_o$ receives service from all of its local servers, rack-local servers in $\mathcal{H}_o$ and remote servers in $\mathcal{H}_u$. A crucial step to bound the drift of $F_\perp(Z)$ is to use the ideal load decomposition as an intermediary.

(Theorem 4.7.) We obtain an upper bound on $\mathbb{E}\left[\Psi^{(\epsilon)}\right]$ by bounding $\mathbb{E}\left[\|\tilde{\mathbf{c}}\|\langle \mathbf{c}, \tilde{\mathbf{Q}}\rangle\right]$, where $\tilde{\mathbf{Q}} = (\tilde{Q}_1, \tilde{Q}_2, \cdots, \tilde{Q}_M)$,

$$\tilde{Q}_m = \begin{cases} \frac{Q^r_m}{\gamma}, & \forall m \in \mathcal{H}_u \\ \frac{Q^k_m}{\beta} + \frac{Q^r_m}{\gamma}, & \forall m \in \mathcal{H}_o \\ \frac{Q^l_m}{\alpha} + \frac{Q^k_m}{\beta} + \frac{Q^r_m}{\gamma}, & \forall m \in \mathcal{B}_o \end{cases}$$

The corresponding dynamics is given by

$$\tilde{\mathbf{Q}}(t+1) = \tilde{\mathbf{Q}}(t) + \tilde{\mathbf{A}}(t) - \tilde{\mathbf{S}}(t) + \tilde{\mathbf{U}}(t),$$

where $\tilde{\mathbf{A}}$, $\tilde{\mathbf{S}}$ and $\tilde{\mathbf{U}}$ are defined in the same way as $\tilde{\mathbf{Q}}$.

We consider the Lyapunov function $G_{||}(Z) = \left\|\tilde{\mathbf{Q}}_{||}\right\|^2$, where $\tilde{\mathbf{Q}}_{||}$ is the parallel component of the vector $\tilde{\mathbf{Q}}$ with respect to the direction $\mathbf{c}$. Note that the drift of $G_{||}(Z)$ is zero in steady state. However, since the service

rate of each server varies with the task type and depends on the status of its three sub-queues, the terms related to service in the drift of $G_{||}(Z)$ cannot be bounded directly. In addition, tasks arrivals of types such as $\mathcal{L}^*_{\mathcal{H}_u} \cup \mathcal{L}_{\mathcal{H}_o}$ also depend on $\tilde{\mathbf{Q}}$, which makes the terms difficult to bound. Similar to the proof of Theorem 3.5 on upper bound in Chapter 3, we construct a series of ideal arrival and service processes to solve this problem. This allows us to rewrite the dynamics of $\tilde{\mathbf{Q}}$, and bound the terms using Lemma 8 in [59].

4.6 Evaluation

We compare the performance of balanced-Pandas with the JSQ-MaxWeight algorithm and Pandas presented in Chapter 3 via simulation. We consider a continuous-time system of 10 racks, where each rack consists of 50 servers. Tasks arrive at the system according to a Poisson process. The service rates for local, rack-local and remote tasks are $\alpha = 1$, $\beta = 0.9$ and $\gamma = 0.5$, respectively. So the mean slowdown of remote tasks is 2, which is consistent to the measurements in [12]. We consider exponential service time distribution for each task.

The task type is designated at arrival. For each task, a set of three servers are chosen to be its local servers according to the distribution of requested data in the system. We consider two cases:

1. Distribution-1. All the datasets requested by the incoming traffic are distributed uniformly in a subset of B servers, which co-locate at a subset of R racks. This simulates the special traffic scenario where the JSQ-MaxWeight algorithm achieves heavy-traffic optimality. Here we report the results for $R = 5$ and $B = 50 * 5$. That is, the set of three local servers for each task are sampled uniformly randomly from all servers.

2. Distribution-2. At each task arrival, with probability σ_1, the task samples a set of three servers uniformly randomly from a subset of N_1 servers in the first rack; with probability σ_2, it samples uniformly from a subset of N_2 servers in the second rack; with probability $1 - \sigma_1 - \sigma_2$, it samples from all other $M - N_1 - N_2$ servers. We choose $\sigma_1 = 0.2$, $N_1 = 10$, $\sigma_2 = 0.06$, $N_2 = 25$. This simulates the traffic with four types of servers when the mean arrival rate is large. In particular, the first rack becomes overloaded with the N_1 servers as $\mathcal{B}_o$, the other $50 - N_1$ servers as $\mathcal{H}_o$; the N_2 servers in the second

rack become $\mathcal{B}_u$; all the other servers in the system become $\mathcal{H}_u$.

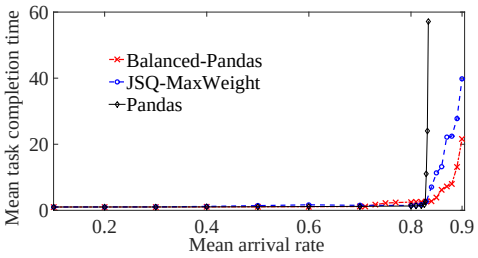
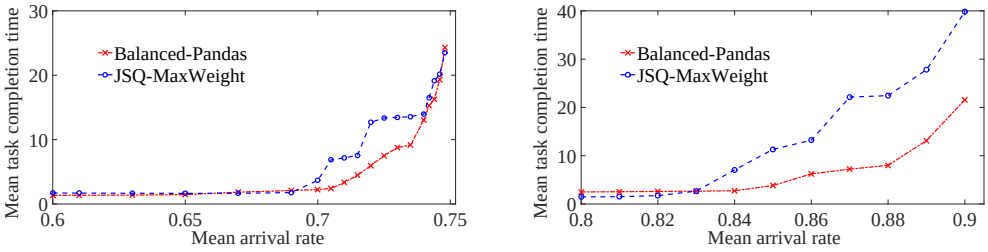


Figure 4.5: Capacity regions with distribution-2.

Figure 4.5 compares the stability regions for JSQ-MaxWeight, the Weighted-Workload algorithm and the priority algorithm. The x-axis shows the mean arrival rate, $\lambda \equiv \sum_{\bar{L}} \lambda_{\bar{L}}/M$, and the y-axis shows the mean completion time for all tasks. A drastic increase in completion time indicates that an algorithm is close to its critical load. For distribution-2, we can compute the capacity region $\lambda < 0.9027$. Observe that both the balanced-Pandas algorithm and JSQ-MaxWeight are stable for $\lambda < 0.9027$, hence are throughput-optimal. However, Pandas becomes unstable at $\lambda \simeq 0.83$. This shows that maximizing the amount of tasks served locally can lead to instability at a much lower load than the full capacity.



(a) Distribution-1 (b) Distribution-2

Figure 4.6: Mean task completion time.

Figure 4.6 compares the delay performance of JSQ-MaxWeight and balanced-Pandas and Pandas at high load. With distribution-1, both algorithms achieve heavy-traffic optimality. Figure 4.6(a) shows that balanced-Pandas has similar performance as JSQ-MaxWeight. With distribution-2, however, balanced-Pandas achieves up to 4-fold improvement over JSQ-MaxWeight algorithm at high load. The balanced-Pandas algorithm is shown to be heavy-traffic optimal for all traffic scenarios. The significant improvement

of balanced-Pandas over JSQ-MaxWeight at high load in Fig. 4.6(b) shows that JSQ-MaxWeight is not heavy-traffic optimal for all traffic scenarios.

4.7 Conclusion

In this chapter, we studied the scheduling problem with multi-level data-locality. We studied an extension of the JSQ-MaxWeight algorithm to three locality levels. We have shown that the JSQ-MaxWeight is throughput optimal but only heavy-traffic optimal for a special traffic scenario. We proposed an algorithm called balanced-Pandas that uses weighted workload routing and priority service. Balanced-Pandas is shown to be throughput optimal, and achieves heavy-traffic optimality for general traffic scenario.

CHAPTER 5

RESOURCE ALLOCATION FOR VMS

In this chapter, we consider cloud computing systems that provide infrastructure as a service (IaaS). Cloud users submit requests for computing resource in the form of virtual machines (VMs). The resource allocation problem for VMs is a stochastic bin-packing problem [63, 64], but with VMs terminating after an application has completed. This motivates the model with jobs arriving and departing the system, which was first considered in [26] and is referred to as a *service* model in [28]. Some recent work along this line focuses on improving resource utilization with different packing algorithms [65, 31]. Some other recent work studies this problem with different performance objectives, including maximizing system throughput [26], minimizing heavy-traffic queue lengths [32], and minimizing the total energy consumption [66].

In this chapter, we are interested in zero-delay service, i.e., a VM is served immediately upon arrival. This model is motivated by the fact that when users submit VM requests to a cloud computing system, any request that is not immediately fulfilled is typically rejected [4]. Therefore, we consider a loss model and focus on the blocking probability, i.e., the probability that an arriving job does not find the required amount of resource at the server, in contrast to the models in [26, 28]. Some recent work [33, 28, 67] also studies zero-delay service. However, their performance objective is to minimize the number of servers occupied, which is different from ours. In particular, they consider the case of infinite number of servers, while we consider finite number of servers and study the blocking probability in the limit as the number of servers goes to infinity.

In our model, we consider one-dimensional packing constraint for the requests of resources. While VM requests can be modelled as multi-dimensional bin-packing, it has been observed that memory is the dominating bottleneck [65]. Due to the large size of a cloud computing system, we consider

asymptotic blocking probability as $N \to \infty$.

We consider the power-of-d-choices routing algorithm for this system. An arriving job is routed to the server with the largest amount of available resource among $d \geq 2$ randomly chosen servers. When none of the chosen servers has enough resource to accommodate the job, it is rejected. Our goal here is to study the asymptotic blocking probability of the power-of-d-choices routing algorithm.

With respect to the power-of-d-choices algorithm, Azar et al. [68] were the first to analyze randomized load balancing schemes using a balls-and-bins model. Another line of work focuses on the queueing systems [69, 70, 71, 72, 73, 74, 75, 76]. In particular, a supermarket model has been used widely to analyze the randomized load balancing schemes. Vvedenskaya et al. [71] and Mitzenmacher [69] showed that when each arriving job is assigned to the shortest $d \geq 2$ randomly chosen queues, the equilibrium queue sizes decay doubly exponentially in the limit as the number of servers goes to infinity. This is a substantial improvement over the $d = 1$ case, where the queue size decays exponentially. While the work in [77] does not address power-of-d choices routing directly, similar analytical techniques have been used there to study the impact of resource pooling in large server farms. However, to the best of our knowledge, the performance of the power-of-d-choices algorithm ($d \geq 2$) for a loss model has not been studied previously. Related work has also been done in parallel with our work in [78].

The rest of this chapter is organized as follows. We first state the precise model and main results. The proofs of these main results will be deferred to later sections. We study the loss model under the power-of-d-choices algorithm ($d \geq 2$) for the case when jobs are homogeneous, i.e., all jobs are of the same type. In particular, we justify the use of fluid approximation of sufficiently large finite systems. We then develop an upper bound for the stationary point of the fluid model and analyze the blocking probability in two different limiting regimes. We then extend our analysis to the case with heterogeneous jobs based on an independence ansatz.

Note on notation: We will use bold letters to denote vectors in $\mathbb{R}^B$ or $\mathbb{N}^J$ or $\mathbb{N}^{J\times N}$, and ordinary letters for scalars. Dot product in the vector spaces $\mathbb{R}^J$ is denoted by $\langle \mathbf{x}, \mathbf{y} \rangle$.

Let $\mathbb{N}_+$ be the set of non-negative integers. The following notations will

be used throughout this chapter:

$$\begin{aligned}
\mathcal{C} &\triangleq \left\{ \mathbf{n} \in \mathbb{N}_+^J : \sum_{j=1}^{J} n_j b_j \leq B \right\}, \\
\mathcal{Q}^{(N)} &\triangleq \left\{ \mathbf{Q} = \{\mathbf{n}_1, \mathbf{n}_2, ..., \mathbf{n}_N\} : \mathbf{n}_m \in \mathcal{C}, \forall m = 1, 2, ..., N \right\}, \\
\mathcal{S} &\triangleq \left\{ \mathbf{s} \in [0,1]^{B+1} : 1 = s_0 \geq s_1 \geq ... \geq s_B \geq 0 \right\}, \\
\mathcal{S}^{(N)} &\triangleq \left\{ \mathbf{s} \in \mathcal{S} : s_i = \frac{K_i}{N}, \text{ for some } K_i \in \mathbb{N}_+, \forall i \right\}, \\
\mathcal{P} &\triangleq \left\{ \mathbf{p} \in \mathbb{R}^{|\mathcal{C}|} : \sum_{i=1}^{|\mathcal{C}|} p_i = 1, \ p_i \geq 0, \ \forall i \right\}.
\end{aligned}$$

And we will use the following notation for asymptotic comparisons; here f and g are positive functions:

1. $f(x) \lesssim g(x)$ for $f(x) = \mathcal{O}(g(x))$, and $f(x) \gtrsim g(x)$ for $f(x) = \Omega(g(x))$.
2. $f(x) \sim g(x)$ for $\lim_{x\to\infty} \frac{f(x)}{g(x)} = 1$.

5.1 Problem Statement and Main Results

We consider a system with N servers, each of which has B units of a resource, such as CPU, memory, etc. This system is accessed by J different types of jobs, where each type of job is characterized by the number of units of resource that it demands. Jobs of type j arrive according to a Poisson process of rate $N\lambda_j$, each type-j job requests b_j units of the resource, and each job stays in the system for an exponentially distributed amount of time with mean 1. We use $\mathbf{b} = (b_1, b_2, ..., b_J)$ to denote the vector of resource units required by different job types. The arrival processes of the different job types and the job holding times are all independent of each other. Let $\lambda = \sum_{j=1}^{J} \lambda_j b_j$ denote the total traffic intensity.

Each arriving job is routed to a server according to a routing policy and requires zero-delay service. If the selected server has sufficient resource to accommodate the arriving job, the job will be processed immediately. Otherwise the job is blocked, i.e., it leaves the system immediately without being served.

For each server m, let $n_{j,m}(t)$ denote the number of type-j jobs that the server is serving at time t. We use

$$\mathbf{n}_m(t) = (n_{1,m}(t), n_{2,m}(t), ..., n_{J,m}(t))$$

to denote the state of server m. Note that $\mathbf{n}_m$ is feasible only if server m has enough resource to accommodate all these jobs. That is,

$$\sum_{j=1}^{J} n_{j,m} b_j \leq B.$$

We consider two cases separately: $J = 1$ which we call the *homogeneous job* case and $J > 1$ which we call the *heterogeneous job* case. In the homogeneous case, we assume without loss of generality that $b_1 = 1$, i.e., all jobs require one unit of resource.

Our goal is to study the blocking probability of the power-of-d-choices routing: under this routing scheme, upon each job arrival, d servers are selected uniformly at random and the job is routed to the least loaded of the servers (the one with the least amount of resource used). If none of the selected servers has sufficient amount of resource, then the arriving job is blocked and lost. The performance in the case $d = 1$ is fundamentally different from the cases where $d > 1$. Therefore, we study these two cases separately. In the case of $d = 1$, since we are routing an arrival to a randomly selected server, we will call this scheme *the random routing* scheme. We will reserve the use of the term *power-of-d-choices* routing to the case where $d > 1$.

Next, we present the main results of this chapter, for the homogeneous job case first followed by the heterogeneous job case.

5.1.1 Homogeneous Jobs

Before we present our main results, we introduce some notation. Consider a system with N servers. Let $S_k^{(N)}(t)$ denote the fraction of servers with at least k jobs in service at time t. The Markov process $\{\boldsymbol{S}^{(N)}(t), t \geq 0\}$ is positive recurrent, and then has a unique equilibrium distribution, denoted by $\boldsymbol{\pi}^{(N)}$. We will approximate $\boldsymbol{\pi}^{(N)}$ by the unique invariant measure of the

following fluid model in a manner which will be made precise later.

Definition 5.1. *(Fluid Model). Given any initial condition $\mathbf{s}^0 \in \mathcal{S}$, a function $\mathbf{s}(t) : [0, \infty) \to \mathcal{S}$ is said to be a solution to the fluid model if:*

1. $\mathbf{s}(0) = \mathbf{s}^0$*;*
2. $s_0(t) = 1$ *for any* $t \geq 0$*;*
3. $\mathbf{s}(t)$ *satisfies the following differential equations for any* $t \geq 0$*:*

$$\frac{ds_k(t)}{dt} = \begin{cases} \lambda(s_{k-1}^d - s_k^d) - k(s_k - s_{k+1}), & 1 \leq k \leq B-1 \\ \lambda(s_{B-1}^d - s_B^d) - Bs_B, & k = B. \end{cases} \tag{5.1}$$

Equation (5.1) can be written as

$$\dot{\mathbf{s}}(t) = \mathbf{F}(\mathbf{s}),$$

where

$$F_k(\mathbf{s}) = \begin{cases} \lambda(s_{k-1}^d - s_k^d) - k(s_k - s_{k+1}), & 1 \leq k \leq B-1 \\ \lambda(s_{B-1}^d - s_B^d) - Bs_B, & k = B. \end{cases}$$

The k-th function $F_k(\mathbf{s})$ is the drift of s_k at point $\mathbf{s}(t)$. The stationary point of the differential equation (5.1), denoted by $\boldsymbol{\pi}$, satisfies

$$\mathbf{F}(\boldsymbol{\pi}) = \mathbf{0}. \tag{5.2}$$

The following theorem presents the main convergence result (in the limit $N \to \infty$) for the homogeneous job case.

The Markov process $\{\boldsymbol{S}^{(N)}(t), t \geq 0\}$ is positive recurrent, and then has a unique stationary probability measure. We use

Theorem 5.1. *For any N, the Markov process $\mathbf{S}^{(N)}(t)$ is positive recurrent, thus it has a unique equilibrium distribution $\boldsymbol{\pi}^{(N)}$. Then the sequence $\boldsymbol{\pi}^{(N)}$ converges weakly to $\delta_{\boldsymbol{\pi}}$, where $\boldsymbol{\pi}$ is the unique stationary point of the fluid model (i.e. $\mathbf{F}(\boldsymbol{\pi}) = \mathbf{0}$), and $\delta_{\boldsymbol{\pi}}$ is the Dirac measure concentrated on $\boldsymbol{\pi}$. That is,*

$$\lim_{N \to \infty} \boldsymbol{\pi}^{(N)} = \delta_{\boldsymbol{\pi}}, \text{ in distribution.}$$

Due to the convergence result above and due to the Poisson nature of the arrival process, π_B^d is a good approximation to the blocking probability experienced by arriving jobs, denoted by $P_b^{(N)}$, in a system with N servers. This is due to the fact that

$$P_b^{(N)} = \mathbb{E}\left[(S_B^{(N)})^d\right].$$

From Theorem 5.1, we can approximate $P_b^{(N)}$ by π_B^d when N is sufficiently large.

While $\boldsymbol{\pi}$ can be computed recursively from Eq. (5.2), we provide a closed-form expression which provides an upper bound on π_B for all values of λ and B for the case $d \geq 2$. This upper bound is useful later to understand the striking performance difference between the cases $d = 1$ and $d > 1$.

Theorem 5.2. *(Upper bound) Let $\boldsymbol{\pi}$ denote the stationary point of the fluid model. Define $\{\bar{\pi}_k\}_{k=0}^B$ as follows:*

$$\bar{\pi}_k = \begin{cases} 1, & 0 \leq k \leq i_0 + 1 \\ \frac{\lambda^{\frac{d^{k-i_0-1}-1}{d-1}}}{(k-1)(k-2)^{d^1}\cdots(i_0+1)^{d^{k-i_0-2}}}, & i_0 + 1 < k \leq B. \end{cases} \tag{5.3}$$

where $i_0 = \lfloor \lambda \rfloor$.

Then $\bar{\boldsymbol{\pi}}$ is an upper bound for $\boldsymbol{\pi}$, i.e., for any $0 \leq k \leq B$,

$$\bar{\pi}_k \geq \pi_k.$$

Note that in the case $d = 1$, since we are randomly selecting a server, by the property of Poisson processes, the blocking probability is given by the well-known Erlang-B formula for $M/M/B/B$ systems:

$$\mathcal{B}(B, \lambda) = \frac{\lambda^B/B!}{\sum_{k=0}^B (\lambda^k/k!)}. \tag{5.4}$$

Comparing equations (5.3) and (5.4), we can see that the blocking probability goes to zero faster in the case of $d \geq 2$, compared to that for $d = 1$.

To provide further insight into the blocking probability P_b in the case of $d \geq 2$, we consider two limiting regimes: (i) $\frac{B-\lambda}{\sqrt{\lambda}} \to \alpha$ as $B \to \infty$ and (ii) $\frac{B-\lambda}{\log_d \lambda} \to \beta$ as $B \to \infty$. We call the former the heavy-traffic regime and

the latter the critically-loaded regime. The heavy-traffic has been studied extensively in the context of $M/M/B/B$ and $G/G/B/B$ systems [79, 80, 81].

Theorem 5.3. *Let $\lambda < B$ and $\frac{\lambda}{B} \to 1$ as $B \to \infty$, then*

$$\pi_B \lesssim (e^{-\frac{c^2}{2}})^{\frac{(B-\lambda)^2}{\lambda}} d^{(1-c)(B-\lambda)-1}, \tag{5.5}$$

where c is an arbitrary constant satisfying $0 < c < 1$.

In particular,

1. *If $\frac{B-\lambda}{\sqrt{\lambda}} \to \alpha$ as $B \to \infty$, where $\alpha > 0$, then*

$$\log_d \log \frac{1}{P_b} \gtrsim ((1-c)\alpha + o(1))\sqrt{\lambda}.$$

   That is, the blocking probability decays doubly exponentially in $\sqrt{\lambda}$.

2. *If $\frac{B-\lambda}{\log_d \lambda} \to \beta$ as $B \to \infty$, where $\beta > 1$, then there exists a constant $\gamma = (1-c)\beta - 1 > 0$ such that*

$$\log \frac{1}{P_b} \gtrsim \lambda^{\gamma + o(1)}.$$

   That is, the blocking probability decays exponentially in λ^γ.

Remark. Theorem 5.3 shows that the fluid limit of the equilibrium blocking probability is dominated by an asymptotic upper bound, which exhibits very different behavior depending on the relationship between λ and B as B goes to infinity. In particular, if $\frac{B-\lambda}{\sqrt{\lambda}} \to \alpha$, the upper bound is doubly exponential in $\sqrt{\lambda}$ and if $\frac{B-\lambda}{\log \lambda} \to \beta$, $\beta > 1$, the upper bound is exponential in λ^η. This is in contrast with the result for random routing, where the blocking probability scales as $O(\frac{1}{\sqrt{\lambda}})$ even if $\frac{B-\lambda}{\sqrt{\lambda}} \to \alpha$.

Numerical Results: Figure 5.1 shows the blocking probability for the power-of-two-choices algorithm with $B - \lambda = \sqrt{\lambda}$ and $B - \lambda = 2\log\lambda$, both by solving Eq. (5.2) numerically and by simulating a finite system with $N = 1000$. Note that the y-axis is in log scale. We can see that even for small B, the blocking probability P_b exhibits qualitatively different behavior in these two regions: with $\log\lambda$ load gap, P_b decays exponentially; while for $\sqrt{\lambda}$ load gap, P_b decays much faster. For $B = 30$, P_b is of order 10^{-15} with $\sqrt{\lambda}$ load gap. It requires very long simulation time to observe a blocking

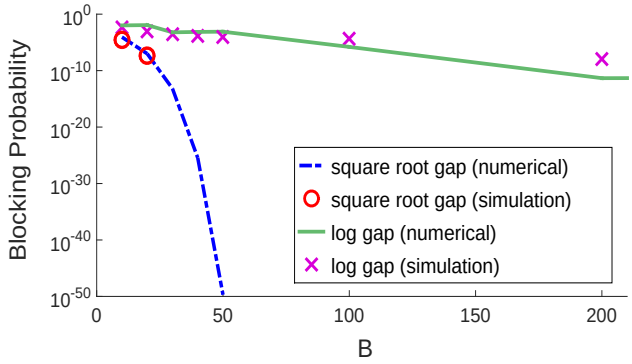


Figure 5.1: Blocking probability for the power-of-two-choices algorithm with different load gap. Line curves are obtained by solving Eq. (5.2) numerically. Markers are from simulations with $N = 1000$.

event. We simulated around 10^{10} arrivals and no job blocking was observed for $B \geq 30$.

To extend the results in this section to the heterogeneous job case, we present a well-known alternative viewpoint of the derivation of $\boldsymbol{\pi}$. Suppose we assume that, in steady-state, the servers become independent of each other and due to symmetry, the tail of the equilibrium queue-size distribution at each server is given by $\boldsymbol{\pi}$. In this case, let us focus on a particular server, say server 1, and write down the Markov chain corresponding to the number of jobs in the server. To describe the transition rate of this Markov chain, suppose that the server has k jobs currently in service. Then, the arrival rate of jobs to this server (call it q_k) is $N\lambda$ times the probability that an arriving job selects this server. It is easy that q_k is given by

$$\begin{aligned} q_k &= N\lambda \times \frac{d}{N} \times \left(\sum_{i=1}^{d} \frac{1}{i} \binom{d-1}{i-1} (\pi_k - \pi_{k+1})^{i-1} (\pi_{k+1})^{d-i} \right) \\ &= \lambda \left(\frac{\pi_k^d - \pi_{k+1}^d}{\pi_k - \pi_{k+1}} \right). \end{aligned}$$

Thus, the Markov chain can be represented by the transition diagram in Figure 5.2. It is now easy to see that the steady-state distribution of this Markov chain is given by Eq. (5.2). This independence ansantz will be used in the next section to derive blocking probability results for the heterogeneous job case.

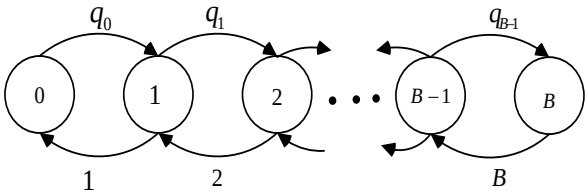


Figure 5.2: State-transition-rate diagram for server 1 with B units of resource and homogeneous job arrivals.

5.1.2 Heterogeneous Jobs

We use the independence ansatz in the previous subsection as follows. Consider a particular server, say server 1, and let $\mathbf{n} = (n_1, ..., n_J)$ be the number of jobs of different types in this server. Let $\{p_{\mathbf{n}}\}_{\mathbf{n}\in\mathcal{C}}$ denote the asymptotic equilibrium distribution for server 1. Then $p_{\mathbf{n}}$ is also the asymptotic fraction of servers in state $\mathbf{n}$.

Under the asymptotic independence assumption, the arrival process of type j jobs to server 1 is a state-dependent Poisson process with rate $\lambda_j(\mathbf{n})$, which is given by

$$\lambda_j(\mathbf{n}) = \lambda_j \left(\sum_{i=1}^{d} \binom{d}{i} E_{\mathbf{n}}^{i-1} G_{\mathbf{n}}^{d-i} \right), \tag{5.6}$$

where

$$E_{\mathbf{n}} = \sum_{\substack{\hat{\mathbf{n}}\in\mathcal{C} \\ \langle\hat{\mathbf{n}},\mathbf{b}\rangle=\langle\mathbf{n},\mathbf{b}\rangle}} p_{\hat{\mathbf{n}}} \quad , \; G_{\mathbf{n}} = \sum_{\substack{\mathbf{n}'\in\mathcal{C} \\ \langle\mathbf{n}',\mathbf{b}\rangle>\langle\mathbf{n},\mathbf{b}\rangle}} p_{\mathbf{n}'}.$$

Let $B_j = \lfloor \frac{B}{b_j} \rfloor$ denote the maximum number of type-j jobs that a server can serve simultaneously. In the case of two job types, the Markov chain is shown in Figure 5.3. However, it is difficult to analyze the equilibrium distribution of this Markov and obtain a simple expression for the blocking probability. Therefore, we study a one-dimensional recursion as in [82] and [83].

Theorem 5.4. *The tail distribution* $\mathbf{r}$ *of the number of occupied resource*

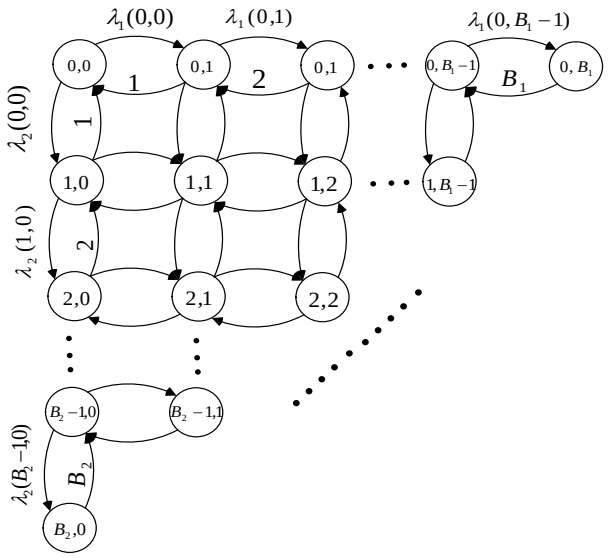


Figure 5.3: State-transition-rate diagram for server 1 with B units of resource and two types of jobs arrivals.

units satisfies the following equation for any $k = 0, 1, ..., B$*:*

$$\sum_{j=1}^{J} \lambda_j b_j (r^d_{k-b_j} - r^d_{k-b_j+1}) \quad = \quad k(r_k - r_{k+1}), \tag{5.7}$$

where $r_x = 1$ *for any* $x \leq 0$ *and* $r_{B+1} = 0$*.*

As for the blocking probability, we obtain analogous results as for homogeneous jobs. Let $b = \max_{j=1,...,J} b_j$, and denote the blocking probability for jobs of type j by P_{b_j}. We have the following theorem.

Theorem 5.5. *Let* $\lambda < B$ *and* $\frac{\lambda}{B} \to 1$ *as* $B \to \infty$*,*

$$r_{B-b+1} \lesssim (e^{-\frac{c^2}{2}})^{\frac{(B-\lambda)^2}{b\lambda} d^{(1-c)(\frac{B-\lambda}{b})-1}}, \tag{5.8}$$

where c *is an arbitrary constant satisfying* $0 < c < 1$*.*

In particular,

1. *If* $\frac{B-\lambda}{\sqrt{\lambda}} \to \alpha$ *as* $B \to \infty$*, where* $\alpha > 0$*, then*

$$\log_d \log \frac{1}{P_{b_j}} \gtrsim ((1-c)\frac{\alpha}{b} + o(1))\sqrt{\lambda},$$

$\forall j \in \{1, 2, ..., J\}$*. That is, for any type of jobs, the blocking probability*

decays doubly exponentially in $\sqrt{\lambda}$.

2. *If* $\frac{B-\lambda}{\log_d \lambda} \to \beta$ *as* $B \to \infty$, *where* $\beta > b$, *then there exists a constant* $\eta = (1-c)\frac{\beta}{b} - 1 > 0$ *such that*

$$\log \frac{1}{P_{b_j}} \gtrsim \lambda^{\eta + o(1)},$$

for all $j \in \{1, 2, ..., J\}$. *That is, for any type of jobs, the blocking probability decays exponentially in* λ^{η}.

The results in this section are derived under the independence ansatz. The existing technique to establish asymptotic independence depends on monotonicity, which does not hold for our problem. Although we do not have the tools to prove the ansatz without monotonicity, we believe that it is true in terms of the random nature of power-of-d-choices algorithm. Alternatively, one can use the fluid approximation: first show convergence of the stochastic system to a differential equation, then show that the differential equation has a unique stationary point to which it converges starting from any initial condition, and finally prove certain tightness results. We have done all of this for the homogeneous case in the next section. In the heterogeneous case, we only have partial results: we can prove convergence to a differential equation and also show that Eq. (5.7) is one of the stationary points of the differential equation. The rest of the steps need to be verified.

5.2 Convergence Results for Homogeneous Jobs

In this section, we focus on the convergence results that justify the approximation of the sample paths $\mathbf{S}^{(N)}(t)$ of sufficiently large systems using the solution $\mathbf{s}(t)$ to the fluid model. Before showing the convergence results rigorously, we introduce some notation for system state and provide some interpretation of the fluid model defined in Section 5.1.1.

5.2.1 Preliminaries

Fix the number of servers N. With homogeneous jobs, system state can be represented by $\mathbf{Q}^{(N)}(t) = (n_1^{(N)}(t), n_2^{(N)}(t), ..., n_N^{(N)}(t))$, where $n_m^{(N)}(t)$ is

the number of jobs in server m at time t. Under the Poisson arrivals and i.i.d. exponential service time assumption, the process $\{\mathbf{Q}^{(N)}(t), t \geq 0\}$ is Markov with state space $\mathcal{Q}^{(N)}$. Note that $0 \leq n_m^{(N)}(t) \leq B$ as each server can accommodate at most B jobs simultaneously. Define

$$S_k^{(N)}(t) = \frac{1}{N} \sum_{i=1}^{(N)} \mathbb{I}_{[k,B]}(n_i^{(N)}(t)), \ \forall k \in \{0, 1, 2, ..., B\},$$

where $S_k^{(N)}(t)$ represents the fraction of servers with at least k jobs in service. Note that $S_0^{(N)}(t) = 1$ for all t. Since the system is fully symmetric, the evolution of the system can be described by the process $\{\mathbf{S}^{(N)}(t), t \geq 0\}$, which is also Markov. Moreover, the system is stable for any $\lambda \geq 0$, as the amount of resource at each server is finite and there is no extra waiting room for arrivals. Hence the Markov process $\{\mathbf{S}^{(N)}(t), t \geq 0\}$ is positive recurrent, and then has a unique equilibrium distribution $\boldsymbol{\pi}^{(N)}$.

Explanation for the drift of $s_k(t)$ in Eq. (5.1): Consider a system with N servers. We will identify the expected change in the fraction of servers with at least k jobs in service over a small period of time of length dt.

(I). The first term corresponds to the change caused by the arrivals. When an arriving job is assigned to a server with $k-1$ jobs, $S_k^{(N)}$ increases by $\frac{1}{N}$. Observe that the number of servers with at least j jobs for $j \neq k$ does not change. Thus $S_k^{(N)}$ is increased by $\frac{1}{N}$ if only if an arriving job joins a server with $k-1$ jobs. Note that the probability that all d sampled servers have at least $k-1$ jobs is s_{k-1}^d. The difference $s_{k-1}^d - s_k^d$ is the probability that at least one of the sampled servers has $k-1$ jobs. With total arrival rate $N\lambda$, the increment for $S_k^{(N)}$ during this time period due to arrival is hence $dt \times N\lambda \times \frac{1}{N} \times (s_{k-1}^d - s_k^d) = \lambda(s_{k-1}^d - s_k^d)dt$.

(II). The second term corresponds to the decrease due to the completion of jobs. The argument is similar to that of the first term.

5.2.2 Convergence Results

We first provide an overview of the convergence results:

First we prove some properties of the fluid model. We will show that there exists a unique solution $\boldsymbol{\pi}$ to the differential equations (5.1) which is

stationary with respect to t, i.e., $\mathbf{F}(\boldsymbol{\pi}) = \mathbf{0}$ (Lemma 5.1). Moreover, given any finite initial condition, the solution to the fluid equation is unique and converges to the stationary solution as $t \to \infty$ (Lemma 5.2).

The second step is to show that as $N \to \infty$, the evolution of process $\mathbf{S}^{(N)}(t)$ converges uniformly, over any finite time interval, to the unique solution of the fluid model (Lemma 5.5). The result is derived by applying Kurtz's theorem ([84, 69]) for density dependent jump Markov processes.

The last step is to prove that the sequence of the stationary probability measure of $\mathbf{S}^{(N)}(t)$ (denoted by $\boldsymbol{\pi}^{(N)}$), concentrates at the unique stationary point $\boldsymbol{\pi}$ of the fluid model as $N \to \infty$ (Theorem 5.1).

Lemma 5.1. *There exists a unique solution $\boldsymbol{\pi} \in \mathcal{S}$ of the differential equation (5.1) that is invariant with respect to t, i.e., $\mathbf{F}(\boldsymbol{\pi}) = \mathbf{0}$.*

Proof outline of Lemma 5.1

Existence: The stationary solution $\boldsymbol{\pi}$ satisfies the equation $\mathbf{F}(\boldsymbol{\pi}) = \mathbf{0}$. We construct a continuous mapping $\mathbf{G} : \mathcal{S} \to \mathcal{S}$, such that a fixed point of $\mathbf{G}$ is a solution to $\mathbf{F}(\boldsymbol{\pi}) = \mathbf{0}$. By Brouwer Fixed Point Theorem, $\mathbf{G}$ has at least one fixed point, i.e., there exists $\boldsymbol{\pi} \in \mathcal{S}$ such that $\mathbf{F}(\boldsymbol{\pi}) = \mathbf{0}$.

Uniqueness: We prove the uniqueness of stationary solution by contradiction and induction. First we show that if there exist two stationary solutions $\boldsymbol{\pi}$ and $\hat{\boldsymbol{\pi}}$ satisfying $\pi_B = \hat{\pi}_B$, then $\pi_k = \hat{\pi}_k$ for any k. Therefore if there exist two different solutions $\boldsymbol{\pi}$ and $\hat{\boldsymbol{\pi}}$, $\pi_B \neq \hat{\pi}_B$. Assume $\pi_B < \hat{\pi}_B$, by induction, we can show that $\pi_k < \hat{\pi}_k$ for any $k = 0, 1, ..., B$, which contradicts with the fact that $\pi_0 = \hat{\pi}_0 = 1$.

Lemma 5.2. *Given any initial condition $\mathbf{s}^0 \in \mathcal{S}$,*

1. *the fluid model has a unique solution $\mathbf{s}(\mathbf{s}^0, t)$ in $\mathcal{S}$,*
2. *as $t \to \infty$, the solution $\mathbf{s}(\mathbf{s}^0, t)$ converges to the unique stationary solution $\boldsymbol{\pi}$.*

We need the following lemmas to prove Lemma 5.2. The proofs of Lemma 5.3-5.4 are provided in the Appendix C.

Lemma 5.3. *Let $\bar{\mathbf{s}}(t)$ and $\mathbf{s}(t)$ be the solutions to differential equations (5.1) with initial condition $\bar{\mathbf{s}}^0$ and $\mathbf{s}^0$ respectively. If $\bar{s}_k^0 \leq s_k^0$ for $k = 1, 2, ..., B$, then $\bar{s}_k(t) \leq s_k(t)$ for any $t \geq 0$.*

Lemma 5.4. *Let $\psi(t) = \sum_{k=0}^{B} |s_k(t) - \pi_k|$, where $\mathbf{s}(t)$ is the solution to differential equations (5.1) with initial condition $\mathbf{s}^0$ satisfying $s_k^0 \geq \pi_k$ for any k (or $s_k^0 \leq \pi_k$ for any k), then $\psi(t)$ converges to 0 as $t \to \infty$.*

Proof of Lemma 5.2. Item 1 follows by the arguments in Theorem 1.(a) of [71]. For any initial values $\mathbf{s}^0 \in \mathcal{S}$, define two initial conditions $\mathbf{s}^u$ and $\mathbf{s}^l$: $s_k^u = \max\{s_k^0, \pi_k\}$, $s_k^l = \min\{s_k^0, \pi_k\}$ for any k. Let $\mathbf{s}^u(t)$ and $\mathbf{s}^l(t)$ denote the solutions with initial conditions $\mathbf{s}^u$ and $\mathbf{s}^l$ respectively. From Lemma 5.3, we have $s_k^u(t) \geq \pi_k \geq s_k^l(t)$ for all t and any k. Thus it is sufficient to show that $\lim_{t\to\infty} |\mathbf{s}^u(t) - \pi| = \lim_{t\to\infty} |\mathbf{s}^l(t) - \pi| = 0$, where $|\cdot|$ is l_1 norm. The result follows directly from Lemma 5.4. ∎

Lemma 5.5. *Consider a sequence of systems with the number of servers N increasing to infinity. Fix any $T > 0$. If the sequence of initial system state $\{\mathbf{S}^{(N)}(0)\}_{N=1}^{\infty}$ concentrates on some $\mathbf{s}^0 \in \mathcal{S}$ as $N \to \infty$, then*

$$\lim_{N\to\infty} \sup_{t\in[0,T]} |\mathbf{S}^{(N)}(t) - \mathbf{s}(\mathbf{s}^0, t)| = 0 \quad a.s.. \tag{5.9}$$

where $\mathbf{s}(\mathbf{s}^0, t)$ is the solution to the differential equation (5.1) given initial condition $\mathbf{s}^0$.

The following lemma is used to prove Lemma 5.5.

Lemma 5.6. *The drift function $\mathbf{F}(\mathbf{s})$ is Lipschitz, i.e., there exists a constant $M > 0$ such that for any $\mathbf{x}, \mathbf{y} \in \mathcal{S}$,*

$$|\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y})| \leq M|\mathbf{x} - \mathbf{y}|,$$

where $|\cdot|$ is l_1 norm.

Proof of Lemma 5.5. We prove this lemma by Kurtz's theorem [84].

(a). It is easy to check that $\{\mathbf{S}^{(N)}(t), t \geq 0\}$ is a density dependent jump Markov process with state space $\mathcal{S}^{(N)}$.

(b). When the system is in state $\mathbf{s}$, the possible transitions is given by $\mathcal{L} = \{\pm\mathbf{e}_k : 1 \leq k \leq B\}$, where $\mathbf{e}_k$ are vectors with only the $k-$th element equal to $1/N$ and all other elements zero. The transition rates are given by $q_{\mathbf{s},\mathbf{s}+\mathbf{l}}^{(N)} = N\beta_{\mathbf{l}}(\mathbf{s})$, where $\beta_{\mathbf{e}_k}(\mathbf{s}) = \lambda(s_{k-1}^d - s_k^d)$ and $\beta_{-\mathbf{e}_k}(\mathbf{s}) = k(s_k - s_{k+1})$. Therefore the rate at which jumps occur is bounded above by $\lambda + B$ everywhere.

(c). Lemma 5.6 states that the differential equation for the limiting deterministic process satisfies the Lipschitz condition.

Then the result follows by Kurtz's Theorem. ∎

Proof of Theorem 5.1. We will use $\Rightarrow$ for weak convergence throughout the proof. Note that set $\mathcal{S}$ is compact. By a corollary of Prokhorov's theorem, for any subsequence of $\{N\}$, there exists a subsubsequence $\{N_k\}$ such that $\boldsymbol{\pi}^{(N_k)}$ converges weakly to some probability distribution $\bar{\boldsymbol{\pi}}$. By the Skorokhod's representation theorem, there exist a sequence of random vector $\{\mathbf{X}^{(N_k)}\}$ and a random vector $\bar{\mathbf{X}}$ such that

$$\mathbf{X}^{(N_k)} \stackrel{d}{=} \boldsymbol{\pi}^{(N_k)}, \quad \bar{\mathbf{X}} \stackrel{d}{=} \bar{\boldsymbol{\pi}},$$

and

$$\mathbf{X}^{(N_k)} \stackrel{a.s.}{\longrightarrow} \bar{\mathbf{X}} \text{ as } k \to \infty.$$

Let $\mathbf{S}^{(N_k)}(0) = \mathbf{X}^{(N_k)}$, i.e., start the system with N_k servers at an initial condition specified by its stationary distribution. We use $\bar{\mathbf{S}}(t)$ to denote the random state of the dynamic system with initial condition $\bar{\mathbf{X}}$.

We have the following claim:

Claim 1: For any $t \geq 0$,

$$\mathbf{S}^{(N_k)}(t) \Rightarrow \bar{\mathbf{S}}(t) \text{ as } k \to \infty.$$

Then the result follows from the arguments in Theorem 5.1 of [85]. We present it here for completeness. Since $\mathbf{S}^{(N_k)}(t)$ was started at the steady-state distribution $\boldsymbol{\pi}^{(N_k)}$,

$$\mathbf{S}^{(N_k)}(t) \stackrel{d}{=} \boldsymbol{\pi}^{(N_k)}, \quad \text{for all } t.$$

Thus, Claim 1 implies that the distribution of $\bar{\mathbf{S}}(t)$ is independent of time t, i.e., $\bar{\boldsymbol{\pi}}$ represents an invariant distribution of the dynamic system $\bar{\mathbf{S}}(t)$. On the other hand, the solution to the ODE for any initial condition converges to a unique fixed point $\boldsymbol{\pi}$. Therefore, the invariant measure $\bar{\boldsymbol{\pi}}$ concentrates at the fixed point $\boldsymbol{\pi}$. That is, $\bar{\boldsymbol{\pi}} = \delta_{\boldsymbol{\pi}}$. Hence

$$\boldsymbol{\pi}^{(N_k)} \Rightarrow \delta_{\boldsymbol{\pi}}.$$

So every convergent subsequence of $\{\boldsymbol{\pi}^{(N)}\}$ converges weakly to $\delta_{\boldsymbol{\pi}}$. Therefore $\boldsymbol{\pi}^{(N)} \Rightarrow \delta_{\boldsymbol{\pi}}$ by a corollary of Prohorov's theorem.

■

Remark. Lemma 5.5 and Theorem 5.1 state that the behavior of sufficiently large systems can be approximated by that of the deterministic infinite system, which is described by a system of differential equations defined in Eq. (5.1).

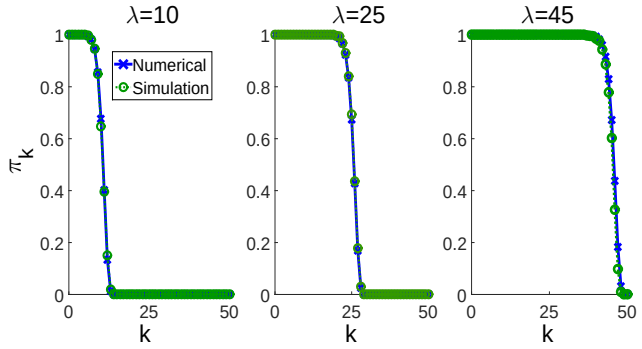


Figure 5.4: Equilibrium tail distribution for the power-of-two-choices algorithm with $B = 50$ at three different loads. The values for the stationary point are obtained numerically by solving Eq. (5.2). Simulation results are from a finite system with $N = 500$.

Numerical Result. Figure 5.4 shows the equilibrium tail distributions of the number of jobs at a server under the power-of-two-choices algorithm with $B = 50$ at three different loads, both by solving Eq. (5.2) numerically and by simulating a finite system with $N = 500$. We can see that the coincidence of the empirical distribution with the stationary point is almost exact. That is, values of the stationary point in the large system limit predict that of a finite system very well.

5.3 Asymptotic Blocking Probability for Homogeneous Jobs

In this section, we focus on the asymptotic blocking probability of power-of-d-choices routing algorithm with homogeneous jobs. We first develop an explicit upper bound for the blocking probability and then analyze the blocking probability in two limiting regimes.

5.3.1 An Upper Bound

Unlike the supermarket model operating under the power-of-d-choices policy [69, 71], there is no explicit expression for the stationary point $\boldsymbol{\pi}$ of the loss model. We establish an explicit upper bound for $\boldsymbol{\pi}$. Observe that the proposed upper bound $\bar{\boldsymbol{\pi}}$ (defined in Eq. (5.3)) can be expressed by a recursive formula as follows:

$$\bar{\pi}_k = \begin{cases} 1, & 0 \le k \le i_0 + 1 \\ \frac{\lambda}{k-1}\bar{\pi}_{k-1}^d, & i_0 + 1 < k \le B, \end{cases}$$

where $i_0 = \lfloor \lambda \rfloor$.

Proof of Theorem 5.2:. We complete the proof in two steps.

(i) First we show that $\pi_k \le \frac{\lambda}{k}\pi_{k-1}^d$ for $1 \le k \le B$ by backward induction. The inequality holds for $k = B$:

$$\pi_B - \frac{\lambda}{B}\pi_{B-1}^d = -\frac{\lambda}{B}\pi_B^d \le 0.$$

Assume that $\pi_{k+1} \le \frac{\lambda}{k+1}\pi_k^d$ hold for $k + 1 \le B$. Then

$$\pi_k - \frac{\lambda}{k}\pi_{k-1}^d = \pi_{k+1} - \frac{\lambda}{k}\pi_k^d \le \pi_{k+1} - \frac{\lambda}{k+1}\pi_k^d \le 0.$$

Hence $\pi_k \le \frac{\lambda}{k}\pi_{k-1}^d,\ \forall k = 1, 2, ..., B$.

(ii) Next we prove the theorem by induction.

For any $k \le i_0 + 1$, $\bar{\pi}_k = 1 \ge \pi_k$. Assume that $\bar{\pi}_k \ge \pi_k$ hold for some $k \ge i_0 + 1$. Then

$$\bar{\pi}_{k+1} = \frac{\lambda}{k}\bar{\pi}_k^d \ge \frac{\lambda}{k}\pi_k^d \ge \frac{\lambda}{k}\pi_k^d + \pi_k - \frac{\lambda}{k}\pi_{k-1}^d = \pi_{k+1},$$

where the first inequality comes from the assumption and the second one follows by the property of $\boldsymbol{\pi}$ we just proved.

■

Figure 5.5 compares the equilibrium tail distribution of the stationary point and the proposed distribution $\bar{\boldsymbol{\pi}}$ with $B = 50$ at three different loads. We can see that the upper bound always holds. Moreover, the proposed distribution characterizes the steep slope of the stationary point, i.e., π_k decreases drastically from 1 to 0 at some k.

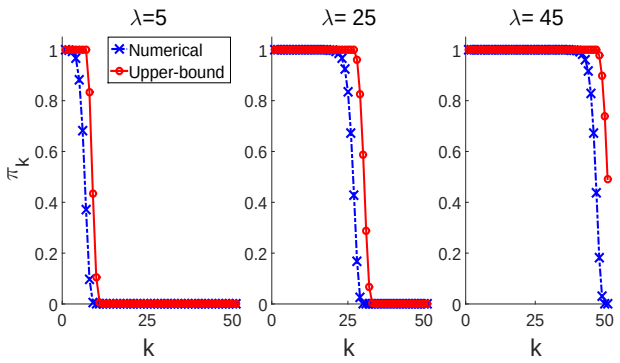


Figure 5.5: An upper bound for the stationary point.

Since the performance measure of primary interest is the blocking probability P_b, we are interested in the tightness of the upper-bound blocking probability.

Table 5.1: The blocking probability for the power-of-two-choices policy with $B = 50$ at different load.

$\rho = \lambda/B$	Fluid limit	Upper bound
0.6	0	0
0.8	0	4.508×10^{-27}
0.84	0.0000	7.003×10^{-7}
0.88	1.873×10^{-25}	0.0426
0.9	5.229×10^{-13}	0.2419
0.92	8.240×10^{-7}	0.5535
0.94	7.854×10^{-4}	0.8122

Table 5.1 compares the upper-bound blocking probability and values of the stationary fluid limit under the power-of-two-choices policy with $B = 50$ at different loads. The values given by the upper bound are quite close to that of the stationary fluid limit at low to medium load. With B fixed, as the load increases towards 1, the gap increases. We have seen that the proposed upper bound resembles a shift of the stationary fluid limit from Fig. 5.5. At high load, the upper bound shifts too much that the resulting bound for blocking probability becomes loose. However, if we fix the load $\rho = \frac{\lambda}{B}$ for the system, we can see that the upper bound blocking probability $\bar{\pi}_B^d$ decays to 0 as B increases. This implies that the upper bound becomes tight for sufficiently large B.

5.3.2 Proof of Theorem 5.3

We devote this section to the proof of Theorem 5.3. We begin by proving the following lemma.

Lemma 5.7. *Let $\lambda < B$ and $\frac{\lambda}{B} \to 1$ as $B \to \infty$. If $\frac{k}{B-\lambda} \to \theta$ as $B \to \infty$, where θ is a constant satisfying $0 \le \theta < 1$, then*

$$\frac{\lambda^{B-i_0-k} \cdot i_0!}{(B-k)!} \sim e^{-\frac{(1-\theta)^2(B-\lambda)^2}{2\lambda}}, \tag{5.10}$$

where $i_0 = \lfloor \lambda \rfloor$.

Proof. By Stirling's formula, we have

$$\begin{aligned}
\frac{\lambda^{B-i_0-k} \cdot i_0!}{(B-k)!} &\sim \lambda^{B-i_0-k} \frac{\sqrt{2\pi i_0} \cdot (\frac{i_0}{e})^{i_0}}{\sqrt{2\pi(B-k)} \cdot (\frac{B-k}{e})^{B-k}} \\
&\sim \sqrt{\frac{i_0}{B-k}} \cdot e^{B-k-\lambda} \left(\frac{\lambda}{B-k}\right)^{\lambda} \cdot \left(\frac{\lambda}{B-k}\right)^{B-k-\lambda} \\
&\sim e^{B-k-\lambda} \left(\frac{\lambda}{B-k}\right)^{\lambda} \cdot \left(\frac{\lambda}{B-k}\right)^{B-k-\lambda}.
\end{aligned} \tag{5.11}$$

Define $\Delta = B - \lambda$. Note that $\lambda/\Delta \to \infty$ as $B \to \infty$. And $(B-k-\lambda) \sim (1-\theta)\Delta$. Then we have

$$\begin{aligned}
\left(\frac{\lambda}{B-k}\right)^{B-k-\lambda} &\sim \left(\frac{\lambda}{\lambda + (1-\theta)\Delta}\right)^{(1-\theta)\Delta} \\
&\sim \left(1 + \frac{(1-\theta)\Delta}{\lambda}\right)^{-(1-\theta)\Delta} \\
&\sim \left[\left(1 + \frac{(1-\theta)}{\lambda/\Delta}\right)^{-\lambda/\Delta}\right]^{-(1-\theta)\frac{\Delta^2}{\lambda}} \\
&\sim e^{-(1-\theta)^2 \frac{\Delta^2}{\lambda}}.
\end{aligned} \tag{5.12}$$

Now consider the first two terms in Eq. (5.11).

$$\begin{aligned}
& \log\left(e^{B-k-\lambda}\left(\frac{\lambda}{B-k}\right)^{\lambda}\right) \\
\sim\ & (1-\theta)\Delta - \lambda\log\left(1+\frac{(1-\theta)\Delta}{\lambda}\right) \\
\sim\ & (1-\theta)\Delta - \lambda\left(\frac{(1-\theta)\Delta}{\lambda} - \frac{1}{2}\left(\frac{(1-\theta)\Delta}{\lambda}\right)^2 + o(\lambda)\right) \\
\sim\ & \frac{(1-\theta)^2\Delta^2}{2\lambda} + o(1).
\end{aligned}$$

That is,

$$e^{B-k-\lambda}\left(\frac{\lambda}{B-k}\right)^{\lambda} \sim e^{\frac{(1-\theta)^2\Delta^2}{2\lambda}}. \tag{5.13}$$

Equations (5.12)-(5.13) yield the asymptotic approximation in Eq. (5.10).

■

Proof of Theorem 5.3. From Theorem 5.2, it is sufficient to show that the upper bound $\bar{\pi}_B$ defined in (5.3) satisfies Eq. (5.5). We establish this result using Lemma 5.7. We can write $\bar{\pi}_B$ as

$$\begin{aligned}
\bar{\pi}_B \ =\ & \left(\frac{\lambda^{B-i_0-1}\cdot i_0!}{(B-1)!}\right)\cdot\left(\frac{\lambda^{B-i_0-2}\cdot i_0!}{(B-2)!}\right)^{(d-1)\cdot d^0} \\
& \cdot\left(\frac{\lambda^{B-i_0-3}\cdot i_0!}{(B-3)!}\right)^{(d-1)d}\cdots\left(\frac{\lambda\cdot i_0!}{(i_0+1)!}\right)^{(d-1)d^{B-i_0-3}}.
\end{aligned} \tag{5.14}$$

Note that each term within the bracket in Eq. (5.14) is no greater than 1. We can obtain an upper bound for $\bar{\pi}_B$ by discarding some terms in Eq. (5.14). In particular, consider keeping the first m terms, where $m = (1-c)(B-\lambda)$, c is an arbitrary constant satisfying $0 < c < 1$. From Lemma 5.7, each term we keep here can be approximated by using Eq. (5.10). Define $\Delta = B - \lambda$.

Then we have

$$\begin{aligned}
\bar{\pi}_B &\leq \left(\frac{\lambda^{B-i_0-1}\cdot i_0!}{(B-1)!}\right)\cdot\left(\frac{\lambda^{B-i_0-2}\cdot i_0!}{(B-2)!}\right)^{(d-1)d^0} \\
&\quad \cdots\left(\frac{\lambda^{B-i_0-m}\cdot i_0!}{(B-m)!}\right)^{(d-1)d^{m-2}} \\
&\sim e^{-\frac{\Delta^2}{2\lambda}\left[(1-\frac{1}{\Delta})^2+(1-\frac{2}{\Delta})^2\cdot(d-1)+\cdots+(1-\frac{m}{\Delta})^2\cdot(d-1)d^{m-2}\right]} \\
&\lesssim e^{-\frac{c^2\Delta^2}{2\lambda}\cdot d^{m-1}} \\
&= \left(e^{-\frac{c^2}{2}}\right)^{\frac{\Delta^2}{\lambda}\cdot d^{(1-c)\Delta-1}}.
\end{aligned}$$

We complete the proof for Eq. (5.5). As discussed in Section 5.1.1, we have $P_b = \pi_B^d$. Thus,

$$P_b \lesssim \left(e^{-\frac{c^2}{2}}\right)^{\frac{\Delta^2}{\lambda}\cdot d^{(1-c)\Delta}}.$$

Now we can study the blocking probability with various load gap by analyzing the exponent $\frac{c^2}{2}\frac{\Delta^2}{\lambda}\cdot d^{(1-c)\Delta}$.

1. $\frac{B-\lambda}{\sqrt{\lambda}} \to \alpha$: we have:

$$\begin{aligned}
\log_d \log\frac{1}{P_b} &\gtrsim 2\log_d\Delta - \log_d\lambda + \log_d\frac{c^2}{2} + (1-c)\Delta \\
&\sim ((1-c)\alpha + o(1))\sqrt{\lambda}.
\end{aligned}$$

2. $\frac{B-\lambda}{\log_d\lambda} \to \beta$: As $\beta > 1$ and $0 < c < 1$ is an arbitrary constant, we can select c to make $\gamma = (1-c)\beta - 1 > 0$. Then we have:

$$\begin{aligned}
\log_d \log\frac{1}{P_b} &\gtrsim ((1-c)\beta - 1)\log_d\lambda + 2\log_d\log\lambda + o(1) \\
&\sim (\gamma + o(1))\log_d\lambda.
\end{aligned}$$

Hence

$$\log\frac{1}{P_b} \gtrsim \lambda^{\gamma+o(1)}.$$

■

5.4 Heterogeneous Jobs

In this section, we focus on the heterogeneous job case. In particular, we will employ the ansatz in [72], which asserts that in equilibrium, any finite set of queues in a randomized load balancing system become asymptotically independent as the number of queues goes to infinity. This will allows us to derive the equilibrium distribution by studying a single server, which has state-dependent Poisson arrivals.

5.4.1 Independence Ansatz

The asymptotic independence for a supermarket model operating under the power-of-d policy with exponentially distributed service time was established by Graham [73] using the propagation of chaos approach. And the independence ansatz for general service time distributions was demonstrated in [72]. A key step of the existing approaches involves standard coupling to establish a monotonicity property for the supermarket model, which is essential to proving the independence ansatz. The monotonicity property states that there exists a coupling such that the evolution of a system with any non-zero initial condition stochastically dominates the evolution of the same system with the all-zeros initial condition. The monotonicity argument is used to demonstrate uniform convergence, i.e., the distance between the two evolutions of the system monotonically decreases with time. This ensures convergence of the system under the arbitrary initial condition to the limiting equilibrium distribution.

We found that it is difficult to establish the independence ansatz using such approach as the loss model with the power-of-d policy does not satisfy the monotonicity property. Consider two copies $\mathbf{X}_1(\cdot)$ and $\mathbf{X}_2(\cdot)$ of the loss model under the power-of-d policy. And assume element-wise dominance of $\mathbf{X}_1(\cdot)$ over $\mathbf{X}_2(\cdot)$. With exponential service times, departures of the two systems can always be coupled. The problem comes from blocking for arrivals. As an arrival is blocked when it is assigned to a server with insufficient resource, it is possible that jobs are blocked in the heavier-loaded system $\mathbf{X}_1(\cdot)$ while enter the lighter-loaded system $\mathbf{X}_2(\cdot)$. This might break the dominance. Therefore monotonicity does not hold for the loss model by standard coupling.

Justification of the independence ansatz for our model remains to be done.

However, we believe that it is true considering the randomized nature of power-of-d algorithms. In the following section, we derive some interesting results under the independence ansatz.

5.4.2 Equilibrium Distribution for A Single Queue

We assume asymptotic independence for the loss model with the power-of-d algorithm. Consider server 1 (by symmetry, any server) in the large N limit. Under the asymptotic independence assumption, the arrival process of type j jobs to server 1 is a state-dependent Poisson process with rate $\lambda_j(\mathbf{n})$, which is given in Eq. (5.6).

We can explain Eq. (5.6) as follows: Assume that server 1 is of state $\mathbf{n}$. When a type j job arrives at the system, it will join server 1 *only if* server 1 is chosen and the state $\mathbf{n}'$ of any other selected server satisfies the condition $\langle \mathbf{n}', \mathbf{b} \rangle \geq \langle \mathbf{n}, \mathbf{b} \rangle$, i.e., server 1 has the largest amount of available resource. Note that server 1 is selected as one of the d sampled servers with probability $\frac{\binom{N-1}{d-1}}{\binom{N}{d}} = \frac{d}{N}$. Consider the case where $i-1$ out of the other $d-1$ selected servers have the same amount of available resource, $i \in \{1, 2, ..., d\}$. Such an event happens with probability $\binom{d-1}{i-1} E_{\mathbf{n}}^{i-1} G_{\mathbf{n}}^{d-i}$, where $E_{\mathbf{n}}$ ($G_{\mathbf{n}}$) represents the fraction of servers with the same (larger) amount of resource occupied. As ties are broken randomly, server 1 is selected with probability $\frac{1}{i}$. Hence the probability that the arrival is routed to server 1 is given by

$$\sum_{i=1}^{d} \frac{d}{N} \cdot \frac{1}{i} \cdot \binom{d-1}{i-1} E_{\mathbf{n}}^{i-1} G_{\mathbf{n}}^{d-i} = \frac{1}{N} \sum_{i=1}^{d} \binom{d}{i} E_{\mathbf{n}}^{i-1} G_{\mathbf{n}}^{d-i}.$$

Multiplying this probability by the arrival rate of type j jobs and letting $N \to \infty$ yield Eq. (5.6).

Note that queue 1 is a birth-death process with state-dependent arrival and departure rates. The global balance equation is given by:

$$\begin{aligned} & \left[\sum_{j=1}^{J} n_j \delta_j^-(\mathbf{n}) + \sum_{j=1}^{J} \lambda_j(\mathbf{n}) \delta_j^+(\mathbf{n}) \right] p_{\mathbf{n}} \\ = & \sum_{j=1}^{J} \lambda_j(\mathbf{n}_j^-) \delta_j^-(\mathbf{n}) p_{\mathbf{n}_j^-} + \sum_{j=1}^{J} (n_j + 1) \delta_j^+(\mathbf{n}) p_{\mathbf{n}_j^+}, \end{aligned} \tag{5.15}$$

where

$$\mathbf{n}_j^+ = (n_1, n_2, ..., n_{j-1}, n_j + 1, n_{j+1}, ..., n_J),$$
$$\mathbf{n}_j^- = (n_1, n_2, ..., n_{j-1}, n_j - 1, n_{j+1}, ..., n_J),$$
$$\delta_j^+(\mathbf{n}) = \begin{cases} 1, & \text{if } \mathbf{n}_j^+ \in \mathcal{C} \\ 0, & \text{otherwise,} \end{cases}$$
$$\delta_j^-(\mathbf{n}) = \begin{cases} 1, & \text{if } \mathbf{n}_j^- \in \mathcal{C} \\ 0, & \text{otherwise.} \end{cases}$$

Moreover, the local balance equation is given by

$$\lambda_j(\mathbf{n}_j^-)\delta_j^-(\mathbf{n})p_{\mathbf{n}_j^-} = n_j\delta_j^-(\mathbf{n})p_{\mathbf{n}},\ \forall j \in \{1, 2, ..., J\}, \forall \mathbf{n} \in \mathcal{C}.$$

Remark. We notice that if the local balance equations are satisfied, the global balance equations are satisfied. However, we have not established that the global balance equations have a unique solution. This is normally true for queueing systems where the arrival rate is fixed; however, since the derivation here follows from the independence ansantz, the arrival rate depends on $\mathbf{p}$. Thus, establishing the uniqueness of the solution to Eq. (5.15) remains to be done.

5.4.3 One-dimensional Recursion

We are interested in the probability P_{b_j} that an arriving job of type j is blocked. Note that

$$P_{b_j} = \left(\sum_{\mathbf{n} \in \mathcal{T}_j^+} p_{\mathbf{n}} \right)^d, \tag{5.16}$$

where $\mathcal{T}_j^+ = \{\mathbf{n} \in \mathcal{C} : \mathbf{n}_j^+ \notin \mathcal{C}\}$.

The underlying high dimension of the state $\mathbf{n}$ makes it difficult to obtain the equilibrium distribution from Eq. (5.16). In order to quantify the blocking probability, we will use Kaufman-Roberts recursion [82, 83] to establish a one-dimensional recursion, regardless of the dimensionality of jobs types (Theorem 5.4). The key idea is to pay attention to the random variable $R(\mathbf{n}) = \sum_{j=1}^J n_j b_j$, which denote the amount of occupied resource. We use $\mathbf{r}$

to represent the tail distribution of $R(\mathbf{n})$, i.e.,

$$r_k = \Pr[R \geq k] = \sum_{\mathbf{n}\in\mathcal{C}:\langle \mathbf{n},\mathbf{b}\rangle \geq k} p_{\mathbf{n}}, \text{ for } k = 0, 1, ..., B.$$

Note that r_k is also the asymptotic *fraction* of servers having at least k units of resource occupied. For ease of exposition, throughout this section, we define $r_x = 1$ for any $x \leq 0$, and $r_{B+1} = 0$.

In order to prove Theorem 5.4, we need the following lemma.

Lemma 5.8. *For any $j \in \mathcal{J}$, and $k \in \{0, 1, ..., B\}$,*

$$\lambda_j(r^d_{k-b_j} - r^d_{k-b_j+1}) = \mathbb{E}\left[n_j | \langle \mathbf{n}, \mathbf{b}\rangle = k\right](r_k - r_{k+1}), \tag{5.17}$$

where $r_x = 1$ for any $x \leq 0$ and $r_{B+1} = 0$.

Proof. Equation (5.16) can be written as :

$$\lambda_j(\mathbf{n}^-_j)\gamma_j(\mathbf{n})p_{\mathbf{n}^-_j} = n_j p_{\mathbf{n}}, \tag{5.18}$$

where

$$\gamma_j(\mathbf{n}) \quad = \quad \begin{cases} 1 & \text{if } n_j \geq 1 \\ 0 & \text{if } n_j = 0. \end{cases}$$

For any $k \in \{0, 1, ..., B\}$, define $\mathcal{D}_k = \{\mathbf{n} \in \mathcal{C} : k = \sum_{j=1}^{J} n_j b_j\}$. Note that for any $\mathbf{n} \in \mathcal{D}_k$,

$$E_{\mathbf{n}} = r_k - r_{k+1}, \; G_{\mathbf{n}} = r_{k+1}.$$

Hence $\lambda_j(\mathbf{n})$ depends on $k = \langle \mathbf{n}, \mathbf{b}\rangle$ only. Summing Eq. (5.18) over the set $\mathcal{D}_k$, we have

$$\sum_{\mathbf{n}\in\mathcal{D}_k} \lambda_j(\mathbf{n}^-_j)\gamma_j(\mathbf{n})p_{\mathbf{n}^-_j} = \sum_{\mathbf{n}\in\mathcal{D}_k} n_j p_{\mathbf{n}}. \tag{5.19}$$

Consider the left-hand side (LHS) of (5.19).

$$\begin{aligned} LHS &= \sum_{\mathbf{n}\in\mathcal{D}_k} \lambda_j(\mathbf{n}_j^-)\gamma_j(\mathbf{n})p_{\mathbf{n}_j^-} \\ &= \lambda_j \sum_{\mathbf{n}\in\mathcal{D}_k} \left(\sum_{i=1}^{d} \binom{d}{i} E_{\mathbf{n}_j^-}^{i-1} G_{\mathbf{n}_j^-}^{d-i}\right) \gamma_j(\mathbf{n})p_{\mathbf{n}_j^-} \\ &= \lambda_j \sum_{\mathbf{n}\in\mathcal{D}_k\cap\{\mathbf{n}:n_j\geq 1\}} \left(\sum_{i=1}^{d} \binom{d}{i} E_{\mathbf{n}_j^-}^{i-1} G_{\mathbf{n}_j^-}^{d-i}\right) p_{\mathbf{n}_j^-}. \end{aligned}$$

Note that

$$\mathcal{D}_k \cap \{\mathbf{n} : n_j \geq 1\} = \left\{\mathbf{n}\in\mathcal{C} : \sum_{i\neq j} n_i b_i + (n_j-1)b_j = k - b_j, n_j \geq 1\right\}.$$

Let $\hat{\mathbf{n}} = \mathbf{n}_j^-$. Then

$$\begin{aligned} LHS &= \lambda_j \sum_{\hat{\mathbf{n}}\in\mathcal{D}_{k-b_j}} \left(\sum_{i=1}^{d} \binom{d}{i} E_{\hat{\mathbf{n}}}^{i-1} G_{\hat{\mathbf{n}}}^{d-i}\right) p_{\hat{\mathbf{n}}} \\ &= \lambda_j \left(\sum_{i=1}^{d} \binom{d}{i} (r_{k-b_j} - r_{k-b_j+1})^{i-1} r_{k-b_j+1}^{d-i}\right) \sum_{\hat{\mathbf{n}}\in\mathcal{D}_{k-b_j}} p_{\hat{\mathbf{n}}} \\ &= \lambda_j \left(\sum_{i=1}^{d} \binom{d}{i} (r_{k-b_j} - r_{k-b_j+1})^{i} r_{k-b_j+1}^{d-i}\right) \\ &= \lambda_j (r_{k-b_j}^d - r_{k-b_j+1}^d). \end{aligned} \tag{5.20}$$

The right-hand side (RHS) of (5.19) can be written as

$$\begin{aligned} RHS &= \sum_{\mathbf{n}\in\mathcal{D}_k} n_j \frac{p_{\mathbf{n}}}{\mathbb{P}\left[\{\mathbf{n} : \langle \mathbf{n}, \mathbf{b}\rangle = k\}\right]} \mathbb{P}\left[\{\mathbf{n} : \langle \mathbf{n}, \mathbf{b}\rangle = k\}\right] \\ &= \sum_{\mathbf{n}\in\mathcal{D}_k} n_j \mathbb{P}\left[\mathbf{n} | \langle \mathbf{n}, \mathbf{b}\rangle =\right] (r_k - r_{k+1}) \\ &= \mathbb{E}\left[n_j | \langle \mathbf{n}, \mathbf{b}\rangle = k\right] (r_k - r_{k+1}). \end{aligned} \tag{5.21}$$

Equation (5.17) follows from Eq. (5.20) and (5.21). ∎

Proof of Theorem 5.4. Multiplying Eq. (5.17) by b_j on both side and sum-

ming over j yields

$$\begin{aligned}\sum_{j=1}^{J}\lambda_j b_j(r^d_{k-b_j}-r^d_{k-b_j+1}) &= \sum_{j=1}^{J} b_j\mathbb{E}\left[n_j|k\right](r_k-r_{k+1})\\ &= \mathbb{E}\left[\sum_{j=1}^{J} b_j n_j|k\right](r_k-r_{k+1})\\ &= k(r_k-r_{k+1}).\end{aligned}$$

■

Remark. We can write the blocking probability for jobs of type j as

$$P_{b_j}=\left(\sum_{\mathbf{n}\in\mathcal{C}:\langle\mathbf{n},\mathbf{b}\rangle>B-b_j} p_{\mathbf{n}}\right)^d = r^d_{B-b_j+1}.$$

By solving Eq. (5.7), we can obtain P_{b_j} immediately. Compared with the formula (5.16), the one-dimensional recursion brings a significant reduction in computation.

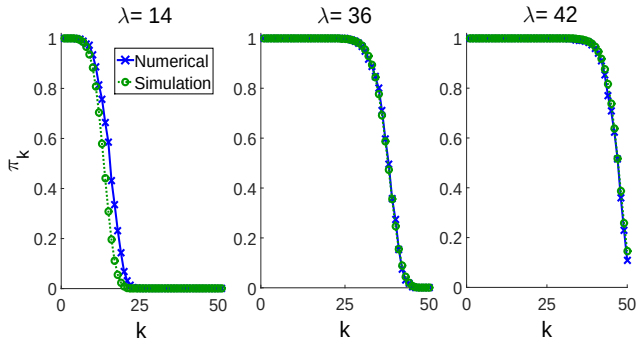


Figure 5.6: Equilibrium distribution of the number of occupied resource units for the power-of-two-choices algorithm with $B = 50$ and three types of jobs, where $\mathbf{b} = (1, 2, 4)$ and $\lambda_j = \lambda/7$. The values for the stationary point are obtained numerically by solving Eq. (5.7). Simulation results are from a finite system with $N = 1000$.

Numerical Results. Figure 5.6 compares the empirical distribution from simulation of a finite system with $N = 1000$ with the stationary point at three different loads. Simulation results coincide with the stationary point very well, which also verifies the validness of independence ansatz.

5.4.4 Upper Bound

We first establish an upper bound for the tail distribution $\mathbf{r}$ of the number of occupied resource units. Let $\lambda = \sum_{j=1}^{J} \lambda_j b_j$ be the total traffic intensity. We have the following theorem.

Theorem 5.6. *Define* $\{\bar{r}_k\}_{k=0}^{B}$ *as follows:*

$$\bar{r}_k = \begin{cases} 1, & 0 \le k \le k_0 + 1 \\ \frac{1}{k-1} \sum_{j=1}^{J} \lambda_j b_j \bar{r}_{k-b_j}^{d}, & k_0 + 1 < k \le B, \end{cases} \tag{5.22}$$

where $k_0 = \lfloor \lambda \rfloor$, $\bar{r}_x = 1$ *for any* $x \le 0$ *and* $\bar{r}_{B+1} = 0$.

Let $\{r_k\}_{k=0}^{B}$ *denote the solution to Eq (5.7). Then for any* $k = 0, 1, ..., B$,

$$\bar{r}_k \ge r_k.$$

Proof of Theorem 5.6 is essentially the same as that of Theorem 5.2.

Lemma 5.9. *Define* $\{\tilde{r}_k\}_{k=0}^{B}$ *as follows:*

$$\tilde{r}_k = \begin{cases} 1, & 0 \le k < b(k_0' + 2) \\ \frac{\lambda}{(m-1)b} \tilde{r}_{k-b}^{d}, & mb \le k < (m+1)b,\ k \le B, \\ & m \in \mathbb{N} \text{ and } k_0' + 1 < m \le \frac{B}{b}, \end{cases} \tag{5.23}$$

where $b = \max_{j=1,...,J} b_j$, *and* $k_0' = \lfloor \frac{\lambda}{b} \rfloor$

Then $\tilde{\mathbf{r}}$ *gives an upper bound for* $\bar{\mathbf{r}}$, *i.e., for any* $k = 0, 1, ..., B$,

$$\tilde{r}_k \ge \bar{r}_k.$$

The following corollary follows immediately by Theorem 5.6 and Lemma 5.9.

Corollary 1. $\tilde{\mathbf{r}}$ *is an upper bound for* $\mathbf{r}$, *i.e.,*

$$\tilde{r}_k \ge r_k, \forall k = 0, 1, ..., B.$$

Remark. Although the upper bound $\bar{\mathbf{r}}$ has no explicit expression, the recursion is straightforward and no further iterative calculation is needed here.

Lemma 5.9 provides a further upper bound on $\bar{\mathbf{r}}$ which is used in the analysis of the blocking probability in the heavy-traffic and critically-loaded traffic regimes (Theorem 5.5).

5.4.5 Proof Outline of Theorem 5.5

Proof outline of Theorem 5.5. Note that $b = \max_{j=1,\dots,J} b_j$. By the monotonicity of the tail distribution $\{r_k\}_{k=0}^{B}$, the blocking probability P_{b_j} for type j jobs ($\forall j \in \{1, 2, ..., J\}$) satisfies

$$P_{b_j} = r^d_{B-b_j+1} \leq r^d_{B-b+1} \leq \tilde{r}^d_{B-b+1}.$$

Hence it is sufficient to show that the upper bound $\tilde{r}_{B-b+1}$ satisfies (5.8).

From the definition of $\tilde{\mathbf{r}}$, we can see that $\{\tilde{r}_k\}_{k=0}^{B}$ consists of consecutive subsequences of size b, where elements in each subsequence have the same value. That is, $\forall k \in [mb, (m+1)b)$, $m \in \mathbb{N}$, $\tilde{r}_k = \tilde{r}_{mb}$. To analyze its asymptotic behavior, we consider the subsequence $\{\tilde{r}_{mb}\}_{m\in\mathbb{N}}$. Define the scaled arrival rate $\lambda' = \lambda/b$, and resource units $B' = \lfloor B/b \rfloor$. Then the recursion of $\{\tilde{r}_{mb}\}_{m=0}^{B'}$ is the same as $\bar{\pi}$ with arrival rate λ' and B' units of resource.

By following the proof for Theorem 5.3, we can establish the asymptotic behavior of $\tilde{r}_{B'b}$ in large B' limit, which gives Eq. (5.8). The analysis for the two limiting regimes is the same as that in Theorem 5.3.

Remark. Theorem 5.5 states that for the general case with multiple types of jobs, the blocking probability for jobs of any type under the power-of-d algorithm has exactly the same asymptotic behavior as that of homogeneous job case.

Numerical Results. We simulate a system of $N = 1000$ servers under the power-of-two-choices algorithm with different load gap. We consider three types of jobs with same arrival rate, i.e., $\lambda_1 = \lambda_2 = \lambda_3$, and $\mathbf{b} = (1, 2, 4)$. For each B, we simulate this system with different load gap $B - \lambda = \sqrt{\lambda}$ and $B - \lambda = 2\log\lambda$, where $\lambda = \sum_j \lambda_j b_j$ is the total traffic intensity. Figure 5.7 compares the blocking probability for jobs that require the maximum amount of resource, i.e., type 3 jobs, with different load gap, both by solving Eq. (5.7) numerically and by simulation. Note that the y-axis is in log

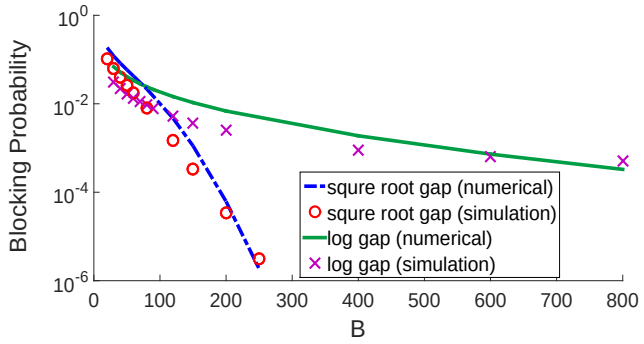


Figure 5.7: Blocking probability for the power-of-two-choices algorithm with different load gap. There are three types of jobs with $\mathbf{b} = (1, 2, 4)$. Line curves are obtained by solving Eq. (5.7) numerically. Markers are from simulations with $N = 1000$.

scale. Observe that the blocking probability for jobs of type 3 exhibits similar behavior as that of the homogeneous job case (Fig. 5.1). That is, P_{b_3} decays exponentially with $\log \lambda$ load gap, while it decays much faster with $\sqrt{\lambda}$ load gap. Similar behavior can be observed for the blocking probability of the other two types of jobs.

5.5 Conclusion

This chapter considered a loss model for the VM assignment problem in a cloud system. The overall goal is to study how to route arriving jobs to the servers in order to minimize the probability that an arriving job does not find the required number of resources in the system. Using the fluid model approach, we showed that when arrivals are routed to the least utilized of $d \geq 2$ randomly selected servers, the blocking probability decays exponentially or doubly exponentially. This is a substantial improvement over the random policy. In addition, we developed an explicit upper bound for the stationary fluid limit. The analysis of the upper bound revealed significant insight into the asymptotic behavior of large systems with the power-of-d-choices ($d \geq 2$) algorithm.

We have seen that for a fixed B, the gap between the proposed upper bound and the stationary fluid limit increases with the load. For future work, we are interested in characterizing the gap and establishing an approximation with higher accuracy. Some of current model assumptions could be relaxed

to make the model closer to the real system, including the assumption of exponential service times and the constraint on the one-dimensionality of requested resources.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this thesis, we have studied the scheduling and resource allocation problems in two typical cloud systems: data-intensive clouds and a IaaS cloud system.

For data-intensive clouds, we first investigated the scheduling problem from a stochastic perspective. We have proposed two novel priority algorithms, Pandas and balanced-Pandas. Using stochastic analysis, we have established optimality of both algorithms with respect to throughput and delay performance. In particular, we remark that the result of Pandas settles a version of an open problem in affinity scheduling, where we want to minimize delay without knowing job arrival rates. Moreover, we have implemented Pandas in Hadoop clusters, and demonstrated that Pandas achieves an order of magnitude improvement over existing schedulers. On the technical end, the prioritized service poses challenges to the state-space collapse analysis and makes the proof of heavy-traffic optimality go beyond applying existing approaches. We used a novel ideal load decomposition to separate the system into several subsystems, which require distinct treatments. The techniques developed could be useful to the general affinity scheduling problem.

For the VM allocation problem in a IaaS cloud, we considered a loss model, which characterizes the loss of unfulfilled VM requests in real systems. The overall goal is to study how to route arriving VM requests to servers in order to minimize the blocking probability. Using the fluid model approach, we showed that under the well-known power-of-d-choice routing, the blocking probability decays exponentially or doubly exponentially. This is a substantial improvement over the random policy.

There are several interesting and important questions which we did not address in this thesis. For the scheduling problem with multi-level locality, we have left out the question of what happens to the proposed Balanced-Pandas when overloaded servers exist in underloaded racks. Based on the

state-space collapse result, we conjecture that the delay under balanced-Pandas is within some constant factor of a universal lower bound. It will be interesting to investigate the factor here. A more fundamental question is what algorithm can achieve delay optimality for all traffic scenarios with unknown arrival rates. Addressing this issue for general affinity scheduling has been a longstanding open problem.

Another question of practical relevance is the design of a distributed scheduler. In this thesis, we have focused on centralized data-centric scheduling algorithms. The large scale of data centers and massive request rates makes the overhead of a centralized scheduler excessive, and calls for algorithms working with distributed schedulers, which is a harder problem. An alternative is random sample-based approaches, such as power-of-d-choice. Future work to design and analyze distributed schedulers could be of great interest.

On the modeling end, an interesting direction is to relax some of our current assumptions. For instance, the service times are assumed to be memoryless (geometric or exponential), while more general traffic distributions are observed in reality. Also, we assume a one-dimensional resource constraint for the VM assignment problem. It would be interesting to extend the model by incorporating more realistic constraints.

APPENDIX A

ADDITIONAL PROOFS FOR PANDAS

A.1 Proofs for Ideal Load Decomposition

We first prove Lemma 3.3 and then use the resulting decomposition of Lemma 3.3 to prove Lemma 3.4.

A.1.1 Proof of Lemma 3.3.

Given $\lambda \in \Lambda$, there exists a decomposition $\{\lambda_{\bar{L},n,m}\}$ satisfying Eq. (3.1). We apply an iterative approach to construct $\{\tilde{\lambda}_{\bar{L},n,m}\}$ from $\{\lambda_{\bar{L},n,m}\}$.

We denote by $\{\lambda^{(k)}_{\bar{L},n,m}\}$, $k \geq 0$, the decomposition after the k-th iteration. Let $\mathcal{M}_h^{(k)}$ and $\mathcal{M}_b^{(k)}$ denote the corresponding locally underloaded queues and locally overloaded queues, respectively. That is, $\mathcal{M}_h^{(k)} = \{n \in \mathcal{M} | \sum_{\bar{L}:n\in\bar{L}} \sum_m \lambda^{(k)}_{\bar{L},n,m} < \alpha\}$ and $\mathcal{M}_b^{(k)} = \{n \in \mathcal{M} | \sum_{\bar{L}:n\in\bar{L}} \sum_m \lambda^{(k)}_{\bar{L},n,m} \geq \alpha\}$. And $\mathcal{L}_b^{(k)}$ is used to denote the set of task types that are only local to $\mathcal{M}_b^{(k)}$, $\mathcal{L}_s^{(k)}$ the set of task types local both to $\mathcal{M}_h^{(k)}$ and $\mathcal{M}_b^{(k)}$. Initialize $\{\lambda^{(0)}_{\bar{L},n,m}\}$ as the given decomposition $\{\lambda_{\bar{L},n,m}\}$. If there exists $\bar{L} \in \mathcal{L}_s^{(k)}$ such that $\lambda^{(k)}_{\bar{L},n_1,m} > 0$ for some $n_1 \in \mathcal{M}_b^{(k)}$, $m \in \mathcal{M}$, $\{\lambda^{(k+1)}_{\bar{L},n,m}\}$ will be updated as follows. Otherwise, the iterative processing ends up with $\{\tilde{\lambda}_{\bar{L},n,m}\} = \{\lambda^{(k)}_{\bar{L},n,m}\}$.

The $k+1$-th iteration will redistribute $\lambda^{(k)}_{\bar{L},n_1,m}$ from temporal overloaded queue n_1 to temporal underloaded queue n_2 which is also local to $\bar{L}$. Consider the following four cases.

Case (i): $\lambda^{(k)}_{n_1} - \lambda^{(k)}_{\bar{L},n_1,m} \geq \alpha$, $\lambda^{(k)}_{n_2} + \lambda^{(k)}_{\bar{L},n_1,m} < \alpha$.

Set

$$\lambda^{(k+1)}_{\bar{L},n_1,m} = 0, \quad \lambda^{(k+1)}_{\bar{L},n_2,m} = \lambda^{(k)}_{\bar{L},n_2,m} + \lambda^{(k)}_{\bar{L},n_1,m}.$$

All other components $\lambda^{(k+1)}_{\bar{L},n,m'}$ remain the same as the previous iteration.

Hence after the $k+1$-th iteration, n_1 is still overloaded, while n_2 is still underloaded. Observe that for $\forall m' \in \mathcal{M}, m' \neq m$, Eq. (3.1) still holds under $\{\lambda^{(k+1)}_{\bar{L},n,m}\}$. The total amount of remote load for m remains the same as the k-th iteration, which ensures the correctness of Eq. (3.1) for m.

Case (ii): $\lambda^{(k)}_{n_1} - \lambda^{(k)}_{\bar{L},n_1,m} < \alpha$, $\lambda^{(k)}_{n_2} + \lambda^{(k)}_{\bar{L},n_1,m} < \alpha$.

Update $\{\lambda^{(k+1)}_{\bar{L},n,m} : \bar{L} \in \mathcal{L}, n \in \bar{L}, m \in \mathcal{M}\}$ as case (i). Thus the $k+1$-th iteration redistributes the shared load between n_1 and n_2, making n_1 underloaded as n_2. It is obvious that the load decomposition after the $k+1$-th iteration satisfies Eq. (3.1).

Case (iii): $\lambda^{(k)}_{n_1} - \lambda^{(k)}_{\bar{L},n_1,m} \geq \alpha$, $\lambda^{(k)}_{n_2} + \lambda^{(k)}_{\bar{L},n_1,m} \geq \alpha$.

Let

$$\delta = \min\{\lambda^{(k)}_{\bar{L},n_1,m}, \frac{\lambda^{(k)}_{n_1} - \lambda^{(k)}_{n_2}}{2}\}, \lambda^{(k+1)}_{\bar{L},n_1,m} = \lambda^{(k)}_{\bar{L},n_1,m} - \delta, \lambda^{(k+1)}_{\bar{L},n_2,m} = \lambda^{(k)}_{\bar{L},n_2,m} + \delta.$$

Keep all other components $\lambda^{(k+1)}_{\bar{L},n,m'}$ unchanged. Observe that such an exchange makes n_2 be overloaded as n_1 and minimizes the local load difference between n_1 and n_2. Again Eq. (3.1) holds for $\forall m \in \mathcal{M}$ after $k+1$-th iteration.

Case (iv):: $\lambda^{(k)}_{n_1} - \lambda^{(k)}_{\bar{L},n_1,m} < \alpha$, $\lambda^{(k)}_{n_2} + \lambda^{(k)}_{\bar{L},n_1,m} \geq \alpha$

Follow the same update as case (iii).

If $\lambda^{(k)}_{n_1} + \lambda^{(k)}_{n_2} \geq 2\alpha$, the update turns n_2 into an overloaded queue like case (iii).

If $\lambda^{(k)}_{n_1} + \lambda^{(k)}_{n_2} < 2\alpha$, $\delta = \frac{\lambda^{(k)}_{n_1} - \lambda^{(k)}_{n_2}}{2} < \lambda^{(k)}_{\bar{L},n_1,m}$. Thus $\lambda^{(k+1)}_{n_1} < \alpha$ and $\lambda^{(k+1)}_{n_2} < \alpha$, i.e., both n_1 and n_2 are underloaded after the $k+1$-th iteration.

Consider the system load ρ defined in (3.3). It is easy to verify that $\rho\left(\{\lambda^{(k+1)}_{\bar{L},n,m}\}\right) < \rho\left(\{\lambda^{(k)}_{\bar{L},n,m}\}\right)$. We note the important fact that $\rho\left(\{\lambda_{\bar{L},n,m}\}\right)$ is minimized by a decomposition satisfying condition (3.2). Observe that any arrival exchange among underloaded queues only or among overloaded queues only will not decrease the total system utilization. When all shared type tasks join underloaded queue, the corresponding total load is minimized as there is no possible arrival exchange that will reduce the total load. This ensures the convergence of the above iterative approach. Consequently, the decomposition after the algorithm stops gives the desired decomposition. This completes the proof of Lemma 3.3.

■

A.1.2 Proof of Lemma 3.4.

We construct the ideal decomposition iteratively from $\{\tilde{\lambda}_{\bar{L},n,m}\}$ given in Lemma 3.3 by exchanging remote load for local load in each buffer. First consider load exchange for $\mathcal{D}^c = \mathcal{M}/\mathcal{D}$ to construct $\mathcal{H}$. Define

$$\psi(\{\tilde{\lambda}_{\bar{L},n,m}\}) = \sum_{n\in\mathcal{D}^c} \sum_{m:m\neq n} \nu_{n,m}$$

as the total amount of remote service *received* by $\mathcal{D}^c$ with the decomposition $\{\tilde{\lambda}_{\bar{L},n,m}\}$. Whenever there exists some remote sub-queue of queue $n \in \mathcal{H}$ with non-zero load, for instance $\nu_{n,m} > 0(\ m \neq n)$, we move all the traffic from this remote sub-queue (n, m) to the local sub-queue (n, n). In order to maintain validity of Eq. (3.1) for server n, we reduce the amount of remote service provided by n. We can move $\min\{\nu_{n,m}, \sum_{k\neq n} \nu_{k,n}\}$ amount of load from remote sub-queues at the n-th column to the corresponding sub-queues at the m-th column (within the same row). Then Eq. (3.1) still holds for n. It is easy to see that such an exchange reduces ψ by $\nu_{n,m}$ at least. The iterative process ends when no remote load is left in the buffers of $\mathcal{D}^c$, i.e., $\psi = 0$, and $\mathcal{D}^c$ become $\mathcal{H}$ defined in (3.7).

Next we exchange load for $\mathcal{D}$ to construct $\mathcal{B}$. Define

$$\phi(\{\tilde{\lambda}_{\bar{L},n,m}\}) = \sum_{m_1\in\mathcal{D}} \sum_{\substack{m_2\in D \\ m_2\neq m_1}} \nu_{m_2,m_1}$$

as the total amount of remote service *offered* by $\mathcal{D}$ with the updated decomposition $\{\tilde{\lambda}_{\bar{L},n,m}\}$ satisfying Eq. (3.7). If some overloaded buffer $m_1 \in \mathcal{D}$ offers remote service, i.e., $\exists \nu_{m_2,m_1} > 0$ where $m_2 \in D$ and $m_2 \neq m_1$, we can exchange the remote service offered by m_1 for local service as follows: Pick any non-empty remote sub-queue (m_1, k) within Q_{m_1} $(k \in \mathcal{H})$, then move $\min\{\nu_{m_2,m_1}, \nu_{m_1,k}\}$ amount of load from sub-queue $\nu_{m_1,k}$ to the local sub-queue (m_1, m_1), and move the same amount of load from the sub-queue (m_2, m_1) to the sub-queue (m_2, k) within Q_{m_2}. Note that such movement does not increase remote service offered by other beneficiaries. Hence ψ is reduced by at least $\min\{\nu_{m_2,m_1}, \nu_{m_1,k}\}$. Again Eq. (3.1) holds for all $m \in \mathcal{M}$ after such an exchange. Similarly, the iterative process ends when $\psi = 0$, i.e., $\mathcal{D}$ become $\mathcal{B}$ defined in (3.8). ∎

A.2 Additional Proofs for Theorem 3.1

For ease of exposition, we temporally omit the superscript $^{(\mathcal{H})}$.

A.2.1 Proof of Lemma 3.7.

Under Pandas, every arriving task at the beginning of each time slot will join its shortest local queue. For $\forall \bar{L} \in \mathcal{L}^*_{\mathcal{H}}$, define $Q^*_{\bar{L}}(t) = \min_{m \in \bar{L} \cap \mathcal{H}} \{Q_m(t)\}$. For any task type that is only local to $\mathcal{H}$, i.e., $\bar{L} \in \mathcal{L}_{\mathcal{H}}$, it will be routed to queue $Q^*_{\bar{L}}(t)$ at the beginning of time slot t. Meanwhile, a task local both to $\mathcal{B}$ and $\mathcal{H}$ might join $Q^*_{\bar{L}}(t)$ or its shortest local queue in $\mathcal{B}$.

$$
\begin{aligned}
\mathbb{E}\left[\langle \mathbf{Q}^{(\mathcal{H})}(t), \mathbf{A}^{(\mathcal{H})}(t)\rangle \mid Z(t)\right] &= \mathbb{E}\left[\sum_{m\in\mathcal{H}} Q_m(t)A_m(t) \mid Z(t)\right] \\
&= \mathbb{E}\left[\sum_{\bar{L}\in\mathcal{L}^*_{\mathcal{H}}}\sum_{m\in\bar{L}\cap\mathcal{H}} Q_m(t)A_{\bar{L},m}(t) \mid Z(t)\right] \\
&\overset{(a)}{\leq} \sum_{\bar{L}\in\mathcal{L}^*_{\mathcal{H}}} Q^*_{\bar{L}}(t)\lambda_{\bar{L}} \\
&\overset{(b)}{=} \sum_{\bar{L}\in\mathcal{L}^*_{\mathcal{H}}} Q^*_{\bar{L}}(t) \sum_{m\in\bar{L}\cap\mathcal{H}}\sum_{n=1}^{M} \lambda^*_{\bar{L},m,n} \\
&\overset{(c)}{\leq} \sum_{\bar{L}\in\mathcal{L}^*_{\mathcal{H}}}\sum_{m\in\bar{L}\cap\mathcal{H}} \lambda^*_{\bar{L},m,m} Q_m(t) \\
&= \mathbb{E}\left[\langle \mathbf{Q}^{(\mathcal{H})}(t), \boldsymbol{\lambda}^{*(\mathcal{H})}\rangle \mid Z(t_0)\right],
\end{aligned}
$$

where step (a) follows from the fact that shared-type tasks might join $\mathcal{B}$ upon arrival; (b) uses the definition of ideal load decomposition; (c) is true since $\forall \bar{L}$, $\forall m \in \bar{L} \cap \mathcal{H}$, $Q_m(t) \geq Q^*_{\bar{L}}(t)$.

Therefore, we have

$$
\begin{aligned}
&\mathbb{E}\left[\langle \mathbf{Q}^{(\mathcal{H})}(t), \mathbf{A}^{(\mathcal{H})}(t)\rangle - \langle \mathbf{Q}^{(\mathcal{H})}(t), \boldsymbol{\lambda}^{*(\mathcal{H})}\rangle \mid Z(t_0)\right] \\
= \;& \mathbb{E}\left[\mathbb{E}\left[\langle \mathbf{Q}^{(\mathcal{H})}(t), \mathbf{A}^{(\mathcal{H})}(t)\rangle - \langle \mathbf{Q}^{(\mathcal{H})}(t), \boldsymbol{\lambda}^{*(\mathcal{H})}\rangle \mid Z(t)\right] \mid Z(t_0)\right] \leq 0.
\end{aligned}
$$

■

A.2.2 Proof of Lemma 3.8.

Consider the following random variables:

$$t_m^* = \min\{\tau : \tau \geq t_0, f_m(\tau) = -1\}, m \in \mathcal{M}, \tag{A.1}$$

$$t^* = \max_{1 \leq m \leq M} t_m^*. \tag{A.2}$$

So server m makes the first scheduling decision after t_0 at t_m^*. And t^* is the first time slot that every server has made at least one scheduling decision after t_0. Let $T = JK$, where $J > 0$ and $K > 0$. We then decompose the probability space into two parts by using t^*: $A_1 = \{t^* > t_0 + K \mid Z(t_0)\}$ and $A_2 = \{t^* \leq t_0 + K \mid Z(t_0)\}$. Let B_i denote the expectation term that is further conditioned on A_i, $i = 1, 2$, i.e.,

$$B_i = \mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} \left(\langle \mathbf{Q}^{(\mathcal{H})}(t), \boldsymbol{\lambda}^{*(\mathcal{H})}\rangle - \langle \mathbf{Q}^{(\mathcal{H})}(t), \mathbf{S}^{(\mathcal{H})}(t)\rangle\right) \mid Z(t_0), A_i\right].$$

Thus the expectation term in (3.10) is broken down into two parts: $B_1\mathbb{P}[A_1]$ and $B_2\mathbb{P}[A_2]$. That is,

$$\begin{aligned} & \mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} \left(\langle \mathbf{Q}^{(\mathcal{H})}(t), \boldsymbol{\lambda}^{*(\mathcal{H})}\rangle - \langle \mathbf{Q}^{(\mathcal{H})}(t), \mathbf{S}^{(\mathcal{H})}(t)\rangle\right) \mid Z(t_0)\right] \\ = \; & B_1\mathbb{P}[A_1] + B_2\mathbb{P}[A_2]. \end{aligned}$$

The following lemma gives an bound on $\mathbb{P}[A_1]$ and $\mathbb{P}[A_2]$, which will be used later.

Lemma A.1. *Consider the random variables t^* and t_m^* for $m \in \mathcal{M}$ defined in (A.1)-(A.2). Then*

$$\mathbb{P}[t^* < t_0 + K \mid Z(t_0)] \geq (1 - (1-\gamma)^K)^M,$$
$$\mathbb{P}[t^* \geq t_0 + K \mid Z(t_0)] \leq 1 - (1 - (1-\gamma)^K)^M.$$

Since both of arrivals and departures are bounded, for $\forall t_1, t \in [t_0, t_0 + T]$, where $t_1 < t$,

$$Q_m(t) \leq Q_m(t_1) + (t - t_1)C_A,$$
$$Q_m(t) \geq Q_m(t_1) - (t - t_1)M.$$

As $\lambda \in \Lambda$, there exists $\vartheta > 0$ such that for $\forall m \in \mathcal{M}$, the decomposition satisfies

$$\sum_{\bar{L}:m\in\bar{L}} \frac{\lambda^*_{\bar{L},m,m}}{\alpha} + \sum_{\bar{L}:m\notin\bar{L}} \sum_{n:n\in\bar{L}} \frac{\lambda^*_{\bar{L},n,m}}{\gamma} \leq \frac{1}{1+\vartheta}.$$

For any $m \in \mathcal{H}$, $\lambda^*_m = \sum_{\bar{L}:m\in\bar{L}} \lambda^*_{\bar{L},m,m} < \alpha$. Together with the bounded difference between $Q_m(t_0)$ and $Q_m(t)$, we can bound B_1 as

$$\begin{aligned}
B_1 &\leq \mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} \langle \mathbf{Q}^{(\mathcal{H})}(t), \boldsymbol{\lambda}^{*(\mathcal{H})}\rangle \mid Z(t_0), A_1\right] \\
&\leq \mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} \sum_{m\in\mathcal{H}} \alpha Q_m(t) \mid Z(t_0), A_1\right] \\
&\leq \mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} \sum_{m\in\mathcal{H}} \alpha(Q_m(t_0) + (t-t_0)C_A) \mid Z(t_0), A_1\right] \\
&\leq \alpha T \sum_{m\in\mathcal{H}} Q_m(t_0) + \alpha T^2 M C_A.
\end{aligned}$$

To bound the term B_2, we divide the summation into two parts: from $t = t_0$ to $t = t^*$ and from $t = t^* + 1$ to $t = t_0 + T - 1$. The first part can be bounded in a similar way as term B_1 :

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=t_0}^{t^*} \left(\langle \mathbf{Q}^{(\mathcal{H})}(t), \boldsymbol{\lambda}^{*(\mathcal{H})}\rangle - \langle \mathbf{Q}^{(\mathcal{H})}(t), \mathbf{S}^{(\mathcal{H})}(t)\rangle\right) \mid Z(t_0), t^* \leq t_0 + K\right] \\
\leq\ & \alpha(t^* - t_0) \sum_{m\in\mathcal{H}} Q_m(t_0) + \alpha(t^* - t_0) T M C_A. \qquad \text{(A.3)}
\end{aligned}$$

For the second part, we first let it condition on t^*, and then further conditioned on $Z(t)$. Note that $\forall t \in (t^*, t_0 + T)$ and $m \in \mathcal{H}$,

$$\begin{aligned}
&\mathbb{E}\left[\left(\langle \mathbf{Q}^{(\mathcal{H})}(t), \boldsymbol{\lambda}^{*(\mathcal{H})}\rangle - \langle \mathbf{Q}^{(\mathcal{H})}(t), \mathbf{S}^{(\mathcal{H})}(t)\rangle\right) \mid Z(t_0), t^* \leq t_0 + K\right] \\
=\ & \mathbb{E}\left[\sum_{m\in\mathcal{H}} (Q_m(t)\lambda^*_m - Q_m(t)S^l_m(t) - Q_m(t)S^r_m(t)) \mid Z(t_0), t^* \leq t_0 + K\right] \\
\leq\ & \mathbb{E}\left[\sum_{m\in\mathcal{H}} (Q_m(t)\lambda^*_m - Q_m(t)S^l_m(t)) \mid Z(t_0), t^* \leq t_0 + K\right] \\
=\ & \sum_{m\in\mathcal{H}} \mathbb{E}\left[Q_m(t)\left(\sum_{\bar{L}:m\in\bar{L}} \lambda^*_{\bar{L},m,m} - \alpha I_{\{\eta_m(t)=m\}}\right) \mid Z(t_0), t^* \leq t_0 + K\right].
\end{aligned}$$

As $t > t^*$, given $Z(t)$, $\eta(t)$ is independent of all the previous system state. Thus we have

$$\begin{aligned} & \mathbb{E}\left[Q_m(t)\left(\sum_{\bar{L}:m\in\bar{L}} \lambda^*_{\bar{L},m,m} - \alpha I_{\{\eta_m(t)=m\}}\right) \mid Z(t_0), Z(t), t^* \le t_0 + K\right] \\ = \; & Q_m(t) \sum_{\bar{L}:m\in\bar{L}} \lambda^*_{\bar{L},m,m} - Q_m(t)\alpha\mathbb{E}\left[I_{\{\eta_m(t)=m\}} \mid Z(t)\right]. \end{aligned} \tag{A.4}$$

Note that $\eta_m(t)$ is conditionally independent of $Q(t)$ given $Z(t)$.

Consider the following random variables

$$\tau_m^t := \max\{\tau : \tau \le t, f_m(\tau) = -1\}, m \in \mathcal{M}.$$

Hence τ_m^t is the last moment before t at which server m makes a scheduling decision. Therefore the status of server m remains the same from time τ_m^t to t, i.e., $\eta_m(t) = \eta_m(\tau_m^t)$. Observe that $\eta_m(\tau_m^t) = m$ if $Q_m(\tau_m^t) > 0$, as local tasks will be scheduled first. If $Q_m(\tau_m^t) = 0$, $\eta_m(\tau_m^t) \neq m$. Thus,

$$Q_m(\tau_m^t)\mathbb{E}\left[I_{\{\eta_m(t)=m\}} \mid Z(\tau_m^t)\right] = Q_m(\tau_m^t).$$

Using the bounded difference between $Q_m(t_0)$, $Q_m(\tau_m^t)$ and $Q_m(t)$, we have

$$\begin{aligned} & \mathbb{E}\left[Q_m(t) I_{\{\eta_m(t)=m\}} \mid Z(t)\right] \\ = \; & \mathbb{E}\left[Q_m(t)\mathbb{E}\left[I_{\{\eta_m(t)=m\}} \mid Z(\tau_m^t)\right] \mid Z(t)\right] \\ \ge \; & \mathbb{E}\left[(Q_m(\tau_m^t) - TM)\mathbb{E}\left[I_{\{\eta_m(t)=m\}} \mid Z(\tau_m^t)\right] \mid Z(t)\right] \\ \ge \; & \mathbb{E}\left[Q_m(\tau_m^t)\mathbb{E}\left[I_{\{\eta_m(t)=m\}} \mid Z(\tau_m^t)\right] \mid Z(t)\right] - TM \\ = \; & \mathbb{E}\left[Q_m(\tau_m^t) \mid Z(t)\right] - TM \\ \ge \; & \mathbb{E}\left[Q_m(t_0) \mid Z(t)\right] - 2TM. \end{aligned} \tag{A.5}$$

As $\sum_{\bar{L}:m\in\bar{L}} \lambda^*{}_{\bar{L},m,m} \le \frac{\alpha}{1+\vartheta}$, together with (A.5), we can upper bound (A.4) as

$$(A.4) \;\le\; -\frac{\alpha\vartheta}{1+\vartheta} Q_m(t_0) + 2\alpha TM.$$

Thus the summation from $t = t^* + 1$ to $t = t_0 + T - 1$ can be upper bounded

by

$$-(t_0+T-t^*)\left[\frac{\alpha\vartheta}{1+\vartheta}\sum_{m\in\mathcal{H}}Q_m(t_0)-2\alpha TMM_h\right]. \tag{A.6}$$

Now we can bound the term B_2 by combining the bounds for two summations in (A.3) and (A.6):

$$B_2 \le K\alpha\left(1-\frac{(J-1)\vartheta}{1+\vartheta}\right)\sum_{m\in\mathcal{H}}Q_m(t_0)+C,$$

where $C>0$ is a constant.

Let $\zeta=\frac{\vartheta}{1+\vartheta}$, and $J_1=1+\frac{1}{\zeta}$. Pick any $J>J_1$, then $K\alpha\left(1-\frac{(J-1)\vartheta}{1+\vartheta}\right)<0$.

From Lemma A.1, we have

$$\begin{aligned}
&\mathbb{P}[t^*<t_0+K\mid Z(t_0)]\ge(1-(1-\gamma)^K)^M,\\
&\mathbb{P}[t^*\ge t_0+K\mid Z(t_0)]\le 1-(1-(1-\gamma)^K)^M.
\end{aligned}$$

Applying the bound for B_1 and B_2, together with the above two inequalities, we can obtain

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1}\left(\langle\mathbf{Q}^{(\mathcal{H})}(t),\boldsymbol{\lambda}^{*(\mathcal{H})}\rangle-\langle\mathbf{Q}^{(\mathcal{H})}(t),\mathbf{S}^{(\mathcal{H})}(t)\rangle\right)\mid Z(t_0)\right]\\
\le\;&\alpha T\left(\sum_{m\in\mathcal{H}}Q_m(t_0)\right)(1-(1-(1-\gamma)^K)^M)\\
&+K\alpha\left(1-\frac{(J-1)\vartheta}{1+\vartheta}\right)\left(\sum_{m\in\mathcal{H}}Q_m(t_0)\right)(1-(1-\gamma)^K)^M+C\\
=\;&\alpha T\left(D_1(K)+\frac{1}{J}(1+\zeta)D_2(K)-\zeta D_2(K)\right)\sum_{m\in\mathcal{H}}Q_m(t_0)+C,
\end{aligned}$$

where $D_1(K)=1-(1-(1-\gamma)^K)^M$, $D_2(K)=(1-(1-\gamma)^K)^M$, and C is a constant independent of $Z(t_0)$.

Next we select K and J to make the coefficient of $\sum_{m\in\mathcal{H}}Q_m(t_0)$ negative. First pick any $\theta\in(0,\zeta)$. Note that $D_1(K)\to 0$ as $K\to\infty$, there exists K_1 such that $\forall K>K_1$, $D_1(K)\le\frac{\zeta-\theta}{3}$. Since $D_2(K)\to 1$ as $K\to\infty$, there exists K_2 such that $\forall K>K_2$, $D_2(K)\ge 1-\frac{\zeta-\theta}{3\zeta}$. Let $J_2=\frac{3(1+\zeta)}{\zeta-\theta}$, then $\forall J>J_2$, $\frac{1}{J}(1+\zeta)D_2(K)<\frac{\zeta-\theta}{3}D_2(K)<\frac{\zeta-\theta}{3}$. Thus, by picking $K>\max\{K_1,K_2\}$ and

$J > \max\{J_1, J_2\}$, we obtain

$$D_1(K) + \frac{1}{J}(1+\zeta)D_2(K) - \zeta D_2(K) \leq \frac{\zeta-\theta}{3} + \frac{\zeta-\theta}{3} - \zeta(1 - \frac{\zeta-\theta}{3\zeta}) = -\theta.$$

Therefore

$$\begin{aligned} & \mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1}\left(\langle \mathbf{Q}^{(\mathcal{H})}(t), \boldsymbol{\lambda}^{*(\mathcal{H})}\rangle - \langle \mathbf{Q}^{(\mathcal{H})}(t), \mathbf{S}^{(\mathcal{H})}(t)\rangle\right) \mid Z(t_0)\right] \\ \leq & -\theta\alpha T \sum_{m\in\mathcal{H}} Q_m(t_0) + C \\ = & -\theta_1 \left\|\mathbf{Q}^{(\mathcal{H})}(t_0)\right\|_1 + C, \end{aligned}$$

where $\theta_1 = \theta\alpha T$ and C are independent of $Z(t_0)$. ■

A.2.3 Proof of Lemma A.1

Given the server status vector $f(t_0)$, the service time distribution for all servers are determined. Hence t_m^*, $m \in \mathcal{M}$ are independent. We have

$$\begin{aligned} \mathbb{P}[t^* < t_0 + K \mid Z(t_0)] &= \mathbb{P}[t_1^* < t_0 + K, \cdots, t_M^* < t_0 + K \mid Z(t_0)] \\ &= \Pi_{m=1}^{M} \mathbb{P}[t_m^* < t_0 + K \mid Z(t_0)] \\ &= (1-(1-\alpha)^K)^{M_1}(1-(1-\gamma)^K)^{M_2}, \end{aligned}$$

where $M_1 = \sum_m I_{\{f_m(t)=m\}}$, $M_2 = \sum_m I_{\{f_m(t)\neq m, f_m(t)\neq -1\}}$. Note that $M_1 + M_2 \leq M$, and $0 < 1-(1-\gamma)^K < 1-(1-\alpha)^K < 1$. Thus

$$(1-(1-\alpha)^K)^{M_1}(1-(1-\gamma)^K)^{M_2} \geq (1-(1-\gamma)^K)^{M_1+M_2} \geq (1-(1-\gamma)^K)^M.$$

Therefore,

$$\begin{aligned} &\mathbb{P}[t^* < t_0 + K \mid Z(t_0)] \geq (1-(1-\gamma)^K)^M, \\ &\mathbb{P}[t^* \geq t_0 + K \mid Z(t_0)] \leq 1-(1-(1-\gamma)^K)^M. \end{aligned}$$

■

A.3 Heavy-traffic Optimality with Locally Overloaded Traffic

A.3.1 Proof of Lemma 3.11

We will prove this lemma by constructing a decomposition that meets the three conditions.

Consider a decomposition $\left\{\lambda_{\bar{L},n,m}\right\}$ that satisfies Lemma 3.4. We fix the decomposition of $\mathcal{L}_{\mathcal{H}}$ over $\mathcal{H}$. In the following argument, we will focus on the decomposition of $\mathcal{L}_{\mathcal{B}}$ to achieve the goal. We start with the coarse decomposition of $\lambda_{\bar{L}}$ into $\lambda_{\bar{L}} \equiv \sum_{m\in\bar{L}} \lambda_{\bar{L},m}$. For ease of exposition, we model the relationship between the task types $\mathcal{L}_{\mathcal{B}}$ and the beneficiaries $\mathcal{B}$ by a bipartite graph $\mathbb{G} = (\mathcal{X}, \mathcal{Y}, \mathcal{E})$. Each vertex $x \in \mathcal{X}$ corresponds to a task type $\bar{L} \in \mathcal{L}_{\mathcal{B}}$ and we assign x a budget $b(x) = \lambda_{\bar{L}}$. Each vertex $y \in \mathcal{Y}$ represents a server $m \in \mathcal{B}$. If server m is local to task type $\bar{L}$, we put an edge xy in $\mathcal{E}$. For any vertex v in the graph, we denote the set of its neighbor vertices by $\mathcal{N}(v)$. And let $\mathcal{N}(\mathcal{V}) = \cup_{v\in\mathcal{V}}\mathcal{N}(v)$ for any vertex set $\mathcal{V}$. Consider the weight function

$$\begin{aligned} w: \quad & \mathcal{E} \to [0, +\infty) \\ & xy \to w(xy) \end{aligned}$$

Let $w(x) = \sum_{y\in\mathcal{N}(x)} w(xy)$ and $w(y) = \sum_{x\in\mathcal{N}(y)} w(xy)$. If a weight function w satisfies that $\forall x \in \mathcal{X}$, $w(x) = b(x)$ and $w(y) \geq \alpha$, it is said to be a *proper weight function*. Let $\mathcal{W}$ be the set of proper weight functions. Then $\mathcal{W}$ is nonempty by Lemma 3.4.

For any proper weight function, we can further decompose $w(xy)$ into $w(xy) = \sum_{z\in\mathcal{M}} u(xy, z)$ to satisfy Eq. (3.1) and (3.8), where the function u

$$\begin{aligned} u: \quad & \mathcal{E} \times \mathcal{M} \to [0, +\infty) \\ & (xy, z) \to u(xy, z), \forall xy \in \mathcal{E}, \forall z \in \mathcal{M} \end{aligned}$$

For any such refined decomposition, let $w^l(y) = \sum_{x\in\mathcal{N}(y)} u(xy, y)$, which denotes the rate of arrivals that are served locally at server y. Then $w^r(y) = w(y) - w^l(y)$ is the rate of arrivals served remotely by other servers. In the rest of the proof we only consider proper weight functions. To prove the lemma, it suffices to find a weight function w and its refined decomposition

u such that

$$\forall x \in \mathcal{X}, w(x) = b(x), \tag{A.7}$$

$$\forall y \in \mathcal{Y}, w(y) \geq \alpha, w^l(y) = \alpha(1 - \epsilon_b), \tag{A.8}$$

$$\forall x \in \mathcal{X}, \forall y \in \mathcal{Y}, \forall z \in \mathcal{B}, z \neq y, u(xy, z) = 0 \tag{A.9}$$

$$\forall y \in \mathcal{Y}, \exists x \in \mathcal{X}, \sum_{z \in \mathcal{H}} u(xy, z) \geq \lambda_0. \tag{A.10}$$

Step 1. We first find a weight function w that satisfies (A.7) and

$$w(y) \geq \alpha + \kappa_0,$$

where $\kappa_0 > 0$ is a constant that does not depend on ϵ.

For any $\mathcal{G} \subseteq \mathcal{B}$, let $\mathcal{L}(\mathcal{G}) = \{\bar{L} \in \mathcal{L}_{\mathcal{B}} \mid \exists m \in \mathcal{G}, s.t., m \in \bar{L}\}$, i.e., the set of task types that are local to some servers in $\mathcal{G}$. Define

$$\kappa_1 = \min_{\mathcal{G} \subseteq \mathcal{B}} \left\{ \sum_{\bar{L} \in \mathcal{L}(\mathcal{G})} \lambda_{\bar{L}} - |\mathcal{G}|\alpha \right\}.$$

From the heavy locally overloaded traffic assumption, $\kappa_1 > 0$, and for any $\mathcal{G} \subseteq \mathcal{B}$,

$$\sum_{\bar{L} \in \mathcal{L}(\mathcal{G})} \lambda_{\bar{L}} \geq |\mathcal{G}|\alpha + \kappa_1. \tag{A.11}$$

First we obtain a proper weight function w such that for any $y \in \mathcal{Y}$, $w(y) \geq \alpha + \frac{\kappa_2}{M_b}$, where

$$\kappa_2 = \min \left\{ \kappa_1, \min_{\bar{L} \in \mathcal{L}_{\mathcal{B}}} \frac{\lambda_{\bar{L}}}{|\bar{L}| + 1} \right\}.$$

We have the following claim.

Claim 1. For any proper weight function $w \in \mathcal{W}$, if there exists $y_0 \in \mathcal{Y}$ with $w(y_0) < \alpha + \frac{\kappa_2}{M_b}$, then there exists a path $\mathcal{P} = y_0 x_0 y_1 x_1 \cdots y_k$ such that for $i = 0, 1, ..., k-1$, $w(x_i y_{i+1}) > 0$, and for $i = 1, ..., k-1$, $w(y_i) \leq \alpha + \frac{\kappa_2}{M_b}$, and $w(y_k) > \alpha + \frac{\kappa_2}{M_b}$.

Proof. If there exists $x_0 \in \mathcal{N}(y_0)$ and $y_1 \in \mathcal{N}(x_0)$ such that $w(x_0 y_1) > 0$ and $w(y_1) > \alpha + \frac{\kappa_2}{M_b}$, then let $\mathcal{P} = y_0 x_0 y_1$ and it is done. Otherwise $\forall x \in$

$\mathcal{N}(y_0)$ and $y \in \mathcal{N}(x)$, either $w(xy) = 0$ or $w(y) \leq \alpha + \frac{\kappa_2}{M_b}$. Consider the sets $\mathcal{X}_0 = \mathcal{N}(y_0)$ and $\mathcal{Y}_1 = \mathcal{N}(\mathcal{X}_0)$. Let $\mathcal{V} = \{y \in \mathcal{Y} \mid \exists x \in \mathcal{N}(y), s.t.\ x \notin \mathcal{X}_0\}$, which is the set of vertices that have neighbors outside $\mathcal{X}_0$. Note that for any $y \in \mathcal{Y}_1 \backslash \mathcal{V}$, $\exists x \in \mathcal{X}_0$ such that $w(xy) > 0$, otherwise $w(y) = 0$, which contradicts with $w(y) \geq \alpha$. Moreover, $\mathcal{V} \neq \emptyset$ and $\exists x_0 \in \mathcal{X}_0$, $y_1 \in \mathcal{V}$ such that $w(x_0 y_1) > 0$, since if this is not true, we have

$$\begin{aligned} \sum_{x \in \mathcal{X}_0} b(x) &= \sum_{x \in \mathcal{X}_0} w(x) = \sum_{y \in \mathcal{Y}_1 \backslash \mathcal{V}} w(y) = \sum_{y \in \mathcal{Y}_1 \backslash \mathcal{V}} w(y) \\ &< |\mathcal{Y}_1 \backslash \mathcal{V}| \left(\alpha + \frac{\kappa_2}{M_b} \right) \leq |\mathcal{Y}_1 \backslash \mathcal{V}| \alpha + \kappa_2, \end{aligned}$$

which contradicts with the heavy traffic assumption in (A.11). Thus there exists $\exists x_0 \in \mathcal{X}_0$, $y_1 \in \mathcal{V}$ such that $w(x_0 y_1) > 0$. If $\exists x_1 \in \mathcal{N}(y_1)$, and $\exists y_2 \in \mathcal{N}(x_1)$ such that $w(x_1 y_2) > 0$ and $w(y_2) > \alpha + \frac{\kappa_2}{M_b}$, then let $\mathcal{P} = y_0 x_0 y_1 x_1 y_2$ and we are done. Otherwise, let $\mathcal{X}_1 = \mathcal{N}(\{y_0, y_1\})$, and $\mathcal{Y}_2 = \mathcal{N}(\mathcal{X}_1)$. Arguing similarly, we can find $x_1 \in \mathcal{X}_1$ and $y_2 \in \mathcal{Y}_2$ such that y_2 has neighbors outside of $\mathcal{X}_1$ and $w(x_1 y_2) > 0$. Then if $\exists x_2 \in \mathcal{N}(y_2)$, and $\exists y_3 \in \mathcal{N}(x_2)$ such that $w(x_2 y_3) > 0$ and $w(y_3) > \alpha + \frac{\kappa_2}{M_b}$, then let $\mathcal{P} = y_0 x_0 y_1 x_1 y_2 x_2 y_3$ and we are done. Otherwise we can continue to consider $\mathcal{X}_2 = \mathcal{N}(\{y_0, y_1, y_2\})$, and $\mathcal{Y}_2 = \mathcal{N}(\mathcal{X}_2)$. The procedure will end in finite steps for the following reason. In the connected component $(\mathcal{X}', \mathcal{Y}', \mathcal{E}')$ that contains y_0, there exists at least $y \in \mathcal{Y}'$ such that $w(y) > \alpha + \frac{\kappa_2}{M_b}$. Otherwise

$$\sum_{x \in \mathcal{X}'} b(x) = \sum_{x \in \mathcal{X}'} w(x) = \sum_{\substack{y \in \mathcal{Y}', \exists x \in \mathcal{X}' \\ s.t.\ w(xy) > 0}} w(y) < |\mathcal{Y}'| \left(\alpha + \frac{\kappa_2}{M_b} \right) \leq |\mathcal{Y}'| \alpha + \kappa_2,$$

which contradicts with the heavy traffic assumption.

Following the above procedure, we obtain a sequence $\mathcal{Y}_0 \subsetneq \mathcal{Y}_1 \subsetneq \cdots$ in $\mathcal{Y}'$. The procedure ends when the sequence hits some $y \in \mathcal{Y}'$ with $w(y) > \alpha + \frac{\kappa_2}{M_b}$. So it takes at most $|\mathcal{Y}'|$ steps. This completes the proof for the claim. ∎

Consider a proper weight function w such that $\min_{y \in \mathcal{Y}} w(y)$ is maximized. Then for any $y \in \mathcal{Y}$, $w(y) \geq \alpha + \frac{\kappa_2}{M_b}$. If w does not satisfy this condition, let $y_0 \in \arg\min_{y \in \mathcal{Y}} w(y)$. Then $\alpha \leq w(y_0) < \alpha + \frac{\kappa_2}{M_b}$. From Claim 1, there exists a path $\mathcal{P} = y_0 x_0 y_1 x_1 \cdots y_k$ such that $w(x_i y_{i+1}) > 0$ for $i = 0, 1, ..., k-1$, and $w(y_i) \leq \alpha + \frac{\kappa_2}{M_b}$ for $i = 1, ..., k-1$ and $w(y_k) > \alpha + \frac{\kappa_2}{M_b}$. Let $\delta =$

$\min\{w(x_0y_1), w(x_1y_2), ..., w(x_{k-1}y_k), w(y_k) - (\alpha + \frac{\kappa_2}{M_b})\}$. Then $\delta > 0$. We modify w to get another weight function $\tilde{w}$ as follows:

$$\tilde{w}(xy) = \begin{cases} w(xy) + \delta & \text{if } x = x_i, y = y_i, \text{ where } i = 0, 1, ..., k \\ w(xy) - \delta & \text{if } x = x_i, y = y_{i+1}, \text{ where } i = 0, 1, ..., k-1 \\ w(xy) & \text{otherwise.} \end{cases}$$

By the definition of δ, $\tilde{w}(xy) \geq 0$ for any $xy \in \mathcal{E}$. And for any $x \in \mathcal{X}$, $b(x) = \tilde{w}(x)$. For any $y \in \mathcal{Y}, y \neq y_0, y \neq y_k$, $\tilde{w}(y) = w(y) \geq \alpha$. And $\tilde{w}(y_k) \geq \alpha + \frac{\kappa_2}{M_b}$, $\tilde{w}(y_0) = w(y_0) + \delta > w(y_0) \geq \alpha$. We then modify other vertices in $\arg\min_{y\in\mathcal{Y}} w(y)$ using similar method, which results a proper weight function $\hat{w}$. Then $\min_{y\in\mathcal{Y}} \hat{w}(y) > \min_{y\in\mathcal{Y}} w(y)$, which contradicts with the assumption that w maximize $\min_{y\in\mathcal{Y}} w(y)$. Let $\kappa_0 = \frac{\kappa_2}{M_b}$.

Step 2. Next we further decompose $w(xy)$ into $w(xy) \equiv \sum_{z\in\mathcal{M}} u(xy, z)$ that satisfies conditions (A.8)-(A.10).

Define $\epsilon_b = \frac{\epsilon}{2\alpha M_b}$, $\epsilon_h = \frac{\epsilon}{2\gamma M_h}$. For any $y \in \mathcal{Y}$, since $w(y) = \sum_{x\in\mathcal{N}(y)} w(xy) \geq \alpha + \kappa_0$, we can pick a subset of $\mathcal{X}' \subset \mathcal{N}(y)$, and assign appropriate values for $u(xy, z)$ such that

$$0 < u(xy, y) \leq w(xy), \ \forall x \in \mathcal{X}',$$

$$\sum_{x\in\mathcal{X}'} u(xy, y) = \alpha(1 - \epsilon_b),$$

$$u(xy, y) = 0, \ \forall x \in \mathcal{N}(y)\backslash\mathcal{X}'.$$

Further, $\forall xy \in \mathcal{E}$, $\forall z \in \mathcal{B}$, $z \neq y$, let $u(xy, z) = 0$. Therefore, conditions (A.7)-(A.9) are satisfied. Next we focus on distributing the remaining weight of y, $w^r(y) = \sum_{x\in\mathcal{N}(y)}(w(xy) - u(xy, y))$, over helpers $\mathcal{H}$ to ensure that Eq. (3.1) holds for any helper, and condition (A.10) is satisfied.

Observe that

$$\begin{aligned}\sum_{y\in\mathcal{Y}} w^r(y) &= \sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{N}(y)} (w(xy)-u(xy,y)) \\ &= \sum_{x\in\mathcal{X}} w(x) - \sum_{y\in\mathcal{Y}} w^l(y) \\ &= \sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}}} \lambda_{\bar{L}} - M_b\alpha(1-\epsilon_b) \\ &= M_h\gamma - \frac{\gamma}{\alpha}\sum_{\bar{L}\in\mathcal{L}^*_{\mathcal{H}}} \lambda_{\bar{L}} - M_h\gamma\epsilon_h.\end{aligned}$$

For any edge xy with $w(xy)-u(xy,y)=0$, let $u(xy,z)=0$ for any $z\in\mathcal{H}$. If $w(xy)-u(xy,y)>0$, we can assign an appropriate non-negative value for $u(xy,z)$, $z\in\mathcal{H}$, such that $\sum_{z\in\mathcal{H}} u(xy,z) = w(xy)-u(xy,y)$, and

$$\sum_{xy\in\mathcal{E}} \frac{u(xy,z)}{\gamma} + \frac{\lambda^l_z}{\alpha} = 1-\epsilon_h, \quad \forall z\in\mathcal{H},$$

where $\lambda^l_z = \sum_{\bar{L}:z\in\bar{L}} \lambda_{\bar{L},z,z}$ is the amount of local arrivals for helper z from task types $\mathcal{L}^*_{\mathcal{H}}$.

In this way, the refined decomposition $u(xy,z)$ maintains the validity of Eq.(3.1). Last, we will show that condition (A.10) is satisfied. For any $y\in\mathcal{Y}$,

$$\sum_{x\in\mathcal{N}(y)} (w(xy)-u(xy,y)) = w^r(y) = w(y)-w^l(y) \geq \alpha+\kappa_0-\alpha(1-\epsilon_b) \geq \kappa_0.$$

By the pigeon hole principle, there exists $x'\in\mathcal{N}(y)$ such that

$$w(x'y)-u(x'y,y) \geq \frac{\kappa_0}{|\mathcal{N}(y)|} \geq \frac{\kappa_0}{|\mathcal{L}_{\mathcal{B}}|}.$$

That is,

$$\sum_{z\in\mathcal{H}} u(x'y,z) \geq \lambda_0,$$

where $\lambda_0 = \frac{\kappa_0}{|\mathcal{L}_{\mathcal{B}}|}$, a constant not depending on ϵ. Therefore, condition (A.10) holds for each beneficiary. This completes the proof. ∎

A.3.2 Proof of Lemma 3.12

By the queue dynamics,

$$\begin{aligned}
&\left\|\mathbf{Q}_{||}^{(\mathcal{B})}(t+1)\right\|^2 - \left\|\mathbf{Q}_{||}^{(\mathcal{B})}(t)\right\|^2 \\
= \;& \langle \mathbf{c_b}, \mathbf{A}^{(\mathcal{B})}(t) - \mathbf{S}^{(\mathcal{B})}(t) + \mathbf{U}^{(\mathcal{B})}(t)\rangle^2 + 2\langle \mathbf{c_b}, \mathbf{Q}^{(\mathcal{B})}(t)\rangle\langle \mathbf{c_b}, \mathbf{A}^{(\mathcal{B})}(t) - \mathbf{S}^{(\mathcal{B})}(t)\rangle \\
& +2\langle \mathbf{c_b}, \mathbf{Q}^{(\mathcal{B})}(t)\rangle\langle \mathbf{c_b}, \mathbf{U}^{(\mathcal{B})}(t)\rangle \\
\geq \;& 2\langle \mathbf{c_b}, \mathbf{Q}^{(\mathcal{B})}(t)\rangle\langle \mathbf{c_b}, \mathbf{A}^{(\mathcal{B})}(t) - \mathbf{S}^{(\mathcal{B})}(t)\rangle.
\end{aligned}$$

■

A.3.3 Proof of Lemma 3.13

Note that $(\mathbf{Q}^{(\mathcal{B})}(t) - \mathbf{Q}^{(\mathcal{B})}(t_0))_\perp = \mathbf{Q}_\perp^{(\mathcal{B})}(t) - \mathbf{Q}_\perp^{(\mathcal{B})}(t_0)$. By the boundedness of arrivals and service, we have

$$\begin{aligned}
&\left|\left\|\mathbf{Q}_\perp^{(\mathcal{B})}(t)\right\| - \left\|\mathbf{Q}_\perp^{(\mathcal{B})}(t_0)\right\|\right| \leq \left\|\mathbf{Q}_\perp^{(\mathcal{B})}(t) - \mathbf{Q}_\perp^{(\mathcal{B})}(t_0)\right\| \\
\leq \;& \left\|\mathbf{Q}^{(\mathcal{B})}(t) - \mathbf{Q}^{(\mathcal{B})}(t_0)\right\| \leq T\sqrt{M_b}\max\{M, C_A\}.
\end{aligned}$$

■

A.3.4 Proof of Lemma 3.14

$$\begin{aligned}
G(t) &= \langle \mathbf{Q}^{(\mathcal{B})}(t), \mathbf{A}^{(\mathcal{B})}(t) - \mathbf{S}^{(\mathcal{B})}(t)\rangle - \langle \mathbf{c_b}, \mathbf{Q}^{(\mathcal{B})}(t)\rangle\langle \mathbf{c_b}, \mathbf{A}^{(\mathcal{B})}(t) - \mathbf{S}^{(\mathcal{B})}(t)\rangle \\
&= \langle \mathbf{Q}_\perp^{(\mathcal{B})}(t), \mathbf{A}_\perp^{(\mathcal{B})}(t) - \mathbf{S}_\perp^{(\mathcal{B})}(t)\rangle \\
&\overset{(a)}{\leq} \left\|\mathbf{Q}_\perp^{(\mathcal{B})}(t)\right\| \cdot \left\|\mathbf{A}_\perp^{(\mathcal{B})}(t) - \mathbf{S}_\perp^{(\mathcal{B})}(t)\right\| \\
&\leq \left\|\mathbf{Q}_\perp^{(\mathcal{B})}(t)\right\| \cdot \left\|\mathbf{A}^{(\mathcal{B})}(t) - \mathbf{S}^{(\mathcal{B})}(t)\right\|, \\
&\overset{(b)}{\leq} \left(\left\|\mathbf{Q}_\perp^{(\mathcal{B})}(t_0)\right\| + T\sqrt{M_b}\max\{M, C_A\}\right) \cdot \sqrt{M_b}\max\{M, C_A\},
\end{aligned}$$

where (a) follows from Cauchy-Schwartz inequality, (b) is true due to the boundedness of arrivals and service. Thus the proof is complete if we set $h = \sqrt{M_b}\max\{M, C_A\}$ and $F_0 = M_b T(\max\{M, C_A\})^2$.

A.3.5 Proof of Lemma 3.15

For $\forall \bar{L} \in \mathcal{L}_{\mathcal{B}}$, define $Q^*_{\bar{L}}(t) = \min_{m \in \bar{L}}\{Q_m(t)\}$. For any task type that is only local to $\mathcal{B}$, i.e., $\bar{L} \in \mathcal{L}_{\mathcal{B}}$, it will be routed to queue $Q^*_{\bar{L}}(t)$ at the beginning of time slot t. By the definition of ideal arrival process $\hat{A}$,

$$\sum_{m \in \mathcal{B}} \hat{A}_m(t) = \sum_{\bar{L} \in \mathcal{L}_{\mathcal{B}}} A_{\bar{L}}(t).$$

From Lemma 3.11, for any $m \in \mathcal{B}$, $\exists \bar{L}_m \in \mathcal{L}_{\mathcal{B}}$ such that $\sum_{n:n\neq m} \lambda^*_{\bar{L}_m,m,n} \geq \lambda_0$.

$$\begin{aligned}
&\mathbb{E}\left[\langle \mathbf{Q}^{(\mathcal{B})}(t), \hat{\mathbf{A}}^{(\mathcal{B})}(t)\rangle \mid Z^{(\mathcal{B})}(t)\right] = \sum_{\bar{L} \in \mathcal{L}_{\mathcal{B}}} Q^*_{\bar{L}} \lambda_{\bar{L}} \\
\overset{(a)}{=}\ & \sum_{m \in \mathcal{B}} \sum_{\substack{\bar{L} \in \mathcal{L}_{\mathcal{B}} \\ m \in \bar{L}}} Q^*_{\bar{L}} \lambda^*_{\bar{L},m,m} + \sum_{m \in \mathcal{B}} \sum_{\substack{\bar{L} \in \mathcal{L}_{\mathcal{B}} \\ \bar{L} \neq \bar{L}_m \\ m \in \bar{L}}} \sum_{n:n\neq m} Q^*_{\bar{L}} \lambda^*_{\bar{L},m,n} + \sum_{m \in \mathcal{B}} \sum_{n:n\neq m} Q^*_{\bar{L}_m} \lambda^*_{\bar{L},m,n} \\
\overset{(b)}{\leq}\ & \sum_{m \in \mathcal{B}} \sum_{\substack{\bar{L} \in \mathcal{L}_{\mathcal{B}} \\ m \in \bar{L}}} Q_m \lambda^*_{\bar{L},m,m} + \sum_{m \in \mathcal{B}} \sum_{\substack{\bar{L} \in \mathcal{L}_{\mathcal{B}} \\ \bar{L} \neq \bar{L}_m \\ m \in \bar{L}}} \sum_{n:n\neq m} Q^{max} \lambda^*_{\bar{L},m,n} \\
& + \sum_{m \in \mathcal{B}} Q^{max} \left(\sum_{n:n\neq m} \lambda^*_{\bar{L}_m,m,n} - \lambda_0 \right) + \sum_{m \in \mathcal{B}} Q_m \lambda_0 \\
=\ & \mathbb{E}\left[\langle \mathbf{Q}^{(\mathcal{B})}(t), \boldsymbol{\lambda}^{*l(\mathcal{B})}\rangle + \langle Q^{max}(t)\mathbf{e}, \boldsymbol{\lambda}^{*r(\mathcal{B})}\rangle + \langle \mathbf{Q}^{(\mathcal{B})}(t) - Q^{max}(t)\mathbf{e}, \lambda_0 \mathbf{e}\rangle \mid Z^{(\mathcal{B})}(t_0)\right],
\end{aligned}$$

where (a) follows from the definition of ideal load decomposition; (b) follows from the fact that $Q^*_{\bar{L}} \leq Q_m \leq Q^{max}$ for any $m \in \bar{L}$. ∎

A.3.6 Proof of Lemma 3.16

The proof is similar to the derivation of (A.6) in the proof of Lemma 3.8.

A.3.7 Proof of Lemma 3.17

$$
\begin{aligned}
& \mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1} \langle Q^{max}(t)\mathbf{e}, \boldsymbol{\lambda}^{*r(\mathcal{B})}\rangle \mid t^* < t_0+K, Z^{(\mathcal{B})}(t_0)\right] \\
= \;& \mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1}\sum_{m\in\mathcal{B}} Q^{max}(t)\lambda_m^{*r} \mid t^* < t_0+K, Z^{(\mathcal{B})}(t_0)\right] \\
\leq \;& \mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1}\sum_{m\in\mathcal{B}} (Q^{max}(t_0)+TC_A)\lambda_m^{*r} \mid t^* < t_0+K, Z^{(\mathcal{B})}(t_0)\right] \\
\leq \;& (t_0+T-t^*)Q^{max}(t_0)\sum_{m\in\mathcal{B}}\lambda_m^{*r} + C,
\end{aligned}
$$

where C is a constant.

$$
\begin{aligned}
& \mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1} \langle \mathbf{Q}^{(\mathcal{B})}(t), \mathbf{S}^{r(\mathcal{B})}(t)\rangle \mid t^* < t_0+K, Z^{(\mathcal{B})}(t_0)\right] \\
= \;& \mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1}\sum_{m\in\mathcal{B}} \left(Q_m(t)\sum_{n:n\neq m} R_n(t)I_{\{\eta_n(t)=m\}}\right) \mid t^* < t_0+K, Z^{(\mathcal{B})}(t_0)\right] \\
= \;& \mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1}\sum_{n\in\mathcal{M}} R_n(t)\left(\sum_{m\in\mathcal{B}:m\neq n} Q_m(t)I_{\{\eta_n(t)=m\}}\right) \mid t^* < t_0+K, Z^{(\mathcal{B})}(t_0)\right] \\
\geq \;& \mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1}\sum_{n\in\mathcal{H}} \gamma\left(\sum_{m\in\mathcal{B}} Q_m(t)I_{\{\eta_n(t)=m\}}\right) \mid t^* < t_0+K, Z^{(\mathcal{B})}(t_0)\right].
\end{aligned}
$$

Similar to the proof of Lemma 3.8, consider the following random variables

$$\tau_n^t := \max\{\tau : \tau \leq t, f_n(\tau) = -1\}, n \in \mathcal{M}.$$

Hence τ_n^t is the last moment before t at which server n makes a scheduling decision. Therefore the status of server m remains the same from time τ_n^t to t, i.e., $\eta_n(t) = \eta_n(\tau_n^t)$. Note that if a remote task is scheduled for server n, it must come from the longest queue. Hence,

$$
\begin{aligned}
\sum_{m\in\mathcal{B}} \mathbb{E}\left[Q_m(\tau_n^t)I_{\{\eta_n(t)=m\}} \mid Z(\tau_n^t)\right] &= \sum_{m\in\mathcal{B}} \mathbb{E}\left[Q_m(\tau_n^t)I_{\{\eta_n(\tau_n^t)=m\}} \mid Z(\tau_n^t)\right] \\
&= Q^{max}(\tau_n^t)I_{\{\eta_n(\tau_n^t)\in\mathcal{B}\}} = Q^{max}(\tau_n^t)I_{\{\eta_n(t)\in\mathcal{B}\}}.
\end{aligned}
$$

Applying the bounded difference between $Q(t_0)$, $Q(\tau_n^t)$ and $Q(t)$ yields

$$\begin{aligned}
& \mathbb{E}\left[\sum_{m\in\mathcal{B}} Q_m(t) I_{\{\eta_n(t)=m\}} \mid Z^{(\mathcal{B})}(t_0)\right] \\
= \;& \mathbb{E}\left[\sum_{m\in\mathcal{B}} Q_m(t)\mathbb{E}\left[\mathbb{E}\left[I_{\{\eta_n(t)=m\}} \mid Z(\tau_n^t)\right] \mid Z(t)\right] \mid Z^{(\mathcal{B})}(t_0)\right] \\
\geq \;& \mathbb{E}\left[\sum_{m\in\mathcal{B}} (Q_m(\tau_n^t) - TM)\mathbb{E}\left[\mathbb{E}\left[I_{\{\eta_n(t)=m\}} \mid Z(\tau_n^t)\right] \mid Z(t)\right] \mid Z^{(\mathcal{B})}(t_0)\right] \\
\geq \;& \mathbb{E}\left[\mathbb{E}\left[Q^{max}(\tau_n^t) I_{\{\eta_n(t)\in\mathcal{B}\}} \mid Z(t)\right] \mid Z^{(\mathcal{B})}(t_0)\right] - TM \\
\geq \;& \mathbb{E}\left[Q^{max}(t_0)\mathbb{E}\left[I_{\{\eta_n(t)\in\mathcal{B}\}} \mid Z(t)\right] \mid Z^{(\mathcal{B})}(t_0)\right] - 2TM \\
= \;& Q^{max}(t_0)\mathbb{E}\left[I_{\{\eta_n(t)\in\mathcal{B}\}} \mid Z^{(\mathcal{B})}(t_0)\right] - 2TM.
\end{aligned}$$

Thus

$$\begin{aligned}
& \mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1} \left(\langle Q^{max}(t)\mathbf{e}, \boldsymbol{\lambda}^{*r(\mathcal{B})}\rangle - \langle \mathbf{Q}^{(\mathcal{B})}(t), \mathbf{S}^{r(\mathcal{B})}(t)\rangle\right) \mid t^* < t_0 + K, Z^{(\mathcal{B})}(t_0)\right] \\
\leq \;& (t_0 + T - t^*) Q^{max}(t_0) \sum_{m\in\mathcal{B}} \lambda_m^{*r} \\
& -\mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1} \sum_{n\in\mathcal{H}} I_{\{\eta_n(t)\in\mathcal{B}\}} \mid t^* < t_0 + K, Z^{(\mathcal{B})}(t_0)\right] + C.
\end{aligned}$$

By the boundedness of arrivals and service, we have

$$\langle \mathbf{c_b}, \mathbf{Q}(t_0)\rangle - \frac{TM}{\sqrt{M_b}} \leq \langle \mathbf{c_b}, \mathbf{Q}(t)\rangle \leq \langle \mathbf{c_b}, \mathbf{Q}(t_0)\rangle + \frac{TC_A}{\sqrt{M_b}}.$$

So

$$\begin{aligned}
& \mathbb{E}\left[\langle \mathbf{c_b}, \mathbf{Q}(t)\rangle\langle \mathbf{c_b}, \hat{\mathbf{A}}(t) - \mathbf{S}(t)\rangle \mid t^* < t_0 + K, Z^{(\mathcal{B})}(t_0)\right] \\
\geq \;& \langle \mathbf{c_b}, \mathbf{Q}(t_0)\rangle \frac{1}{\sqrt{M_b}} \mathbb{E}\left[\sum_{m\in\mathcal{B}} (\hat{A}_m(t) - S_m(t)) \mid t^* < t_0 + K, Z^{(\mathcal{B})}(t_0)\right] - C.
\end{aligned}$$

Observe that

$$\mathbb{E}\left[\sum_{m\in\mathcal{B}}\hat{A}_m(t) \mid t^* < t_0+K, Z^{(\mathcal{B})}(t_0)\right] = \sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}}}\lambda_{\bar{L}}.$$

$$\begin{aligned}
&\mathbb{E}\left[\sum_{m\in\mathcal{B}}S_m(t) \mid t^* < t_0+K, Z^{(\mathcal{B})}(t_0)\right] \\
= \;& \mathbb{E}\left[\sum_{m\in\mathcal{B}}(\alpha I_{\{\eta_m(t)=m\}}+\gamma I_{\{\eta_m(t)\in\mathcal{B}\}})+\sum_{n\in\mathcal{H}}\gamma I_{\{\eta_n(t)\in\mathcal{B}\}} \mid t^* < t_0+K, Z^{(\mathcal{B})}(t_0)\right] \\
\leq \;& \alpha M_b + \mathbb{E}\left[\sum_{n\in\mathcal{H}}\gamma I_{\{\eta_n(t)\in\mathcal{B}\}} \mid t^* < t_0+K, Z^{(\mathcal{B})}(t_0)\right].
\end{aligned}$$

Hence the conditional expectation of $\langle \mathbf{c_b}, \mathbf{Q}(t)\rangle\langle \mathbf{c_b}, \hat{\mathbf{A}}(t)-\mathbf{S}(t)\rangle$ can be lower bounded by

$$\frac{\langle \mathbf{c_b}, \mathbf{Q}(t_0)\rangle}{\sqrt{M_b}}\left(\sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}}}\lambda_{\bar{L}} - \alpha M_b - \mathbb{E}\left[\sum_{n\in\mathcal{H}}\gamma I_{\{\eta_n(t)\in\mathcal{B}\}} \mid t^* < t_0+K, Z^{(\mathcal{B})}(t_0)\right]\right) - C.$$

Then we can obtain an upper bound on the summation from $t = t^*+1$ to t_0+T-1 :

$$\begin{aligned}
&(t_0+T-t^*)\alpha\epsilon_0\sum_{m\in\mathcal{B}}Q_m(t_0) + C + \frac{1}{M_b}\left(\sum_{m\in\mathcal{B}}Q_m(t_0) - M_bQ^{max}(t_0)\right) \\
&\cdot\left(\mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1}\sum_{n\in\mathcal{H}}\gamma I_{\{\eta_n(t)\in\mathcal{B}\}} \mid t^* < t_0+K, Z^{(\mathcal{B})}(t_0)\right] - (t_0+T-t^*)\sum_{m\in\mathcal{B}}\lambda_m^{*r}\right).
\end{aligned} \tag{A.12}$$

We will show that $\forall\epsilon < \frac{M_b\lambda_0}{4}$, there exist a constant $L_r > 0$ not depending on ϵ such that $\forall Z^{(\mathcal{B})}(t_0)$ with $\left\|\mathbf{Q}_{\perp}^{(\mathcal{B})}(t_0)\right\| \geq L_r$,

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1}\sum_{n\in\mathcal{H}}\gamma I_{\{\eta_n(t)\in\mathcal{B}\}} \mid t^* < t_0+K, Z^{(\mathcal{B})}(t_0)\right] \\
\geq \;& (t_0+T-t^*)\left(\sum_{m\in\mathcal{B}}\lambda_m^{*r} - \frac{M_b\lambda_0}{4}\right).
\end{aligned} \tag{A.13}$$

Then for any $Z^{(\mathcal{B})}$ with sufficiently large $\left\|\mathbf{Q}_{\perp}^{(\mathcal{B})}\right\|$, (A.12) can be bounded as:

$$(A.12) \leq (t_0+T-t^*)\left[\alpha\epsilon_0 \sum_{m\in\mathcal{B}} Q_m(t_0) + \frac{\lambda_0}{4}\left(\sum_{m\in\mathcal{B}} Q_m(t_0) - M_b Q^{max}(t_0)\right)\right] + C.$$

This finishes the proof of Lemma 3.17.

We now prove inequality (A.13) by contradiction. Assume that $\exists \epsilon < \frac{M_b\lambda_0}{4}$, $\forall L_1 > 0$ there exists $\left\|\mathbf{Q}_{\perp}^{(\mathcal{B})}(t_0)\right\| > L_1$ such that (A.13) does not hold. Then we can bound the total amount of service received by beneficiaries when $Z^{(\mathcal{B})}$ hits the state $Z^{(\mathcal{B})}(t_0)$ as

$$\begin{aligned}
& \mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1}\sum_{m\in\mathcal{B}} S_m(t) \mid t^* < t_0 + KZ^{(\mathcal{B})}(t_0)\right] \\
= \;& \mathbb{E}\left[\sum_{m\in\mathcal{B}}(\alpha I_{\{\eta_m(t)=m\}} + \gamma I_{\{\eta_m(t)\in\mathcal{B}\}}) + \sum_{n\in\mathcal{H}} \gamma I_{\{\eta_n(t)\in\mathcal{B}\}} \mid t^* < t_0 + K, Z^{(\mathcal{B})}(t_0)\right] \\
< \;& (t_0 + T - t^*)\left(M_b\alpha + \sum_{m\in\mathcal{B}} \lambda_m^{*r} - \frac{M_b\lambda_0}{4}\right).
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1}\sum_{m\in\mathcal{B}} A_m(t)\right] &\geq \sum_{t=t^*+1}^{t_0+T-1}\sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}}} \lambda_{\bar{L}} \\
&> \mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1}\sum_{m\in\mathcal{B}} S_m(t) \mid t^* < t_0 + K, Z^{(\mathcal{B})}(t_0)\right].
\end{aligned}$$

That is, when $Z^{(\mathcal{B})}$ hits the state $Z^{(\mathcal{B})}(t_0)$, the amount of service beneficiaries receive is insufficient for the arrival. Arguing similarly to the proof for stability of the beneficiary system, all beneficiary queues will grow together. Then shared arrivals will join helper queues, i.e., the helper subsystem receives arrivals with maximum rate $\sum_{\bar{L}\in\mathcal{L}_{\mathcal{H}}^*} \lambda_{\bar{L}}$. From Section 3.3, the helper subsystem will be stable with such arrivals and any moment of $\left\|\mathbf{Q}^{\mathcal{H}}\right\|$ is bounded. Consider $\hat{Z}^{(\mathcal{B})}$ with $\left\|\hat{\mathbf{Q}}_{\perp}^{(\mathcal{B})}\right\| > \left\|\mathbf{Q}_{\perp}^{(\mathcal{B})}(t_0)\right\| > L_1$ and $\hat{Q}_m > Q_m(t_0)$ for any $m \in \mathcal{B}$, then $\hat{Q}_B^{max} \geq \frac{1}{M_b}\left\|\hat{\mathbf{Q}}_{\perp}^{(\mathcal{B})}\right\| \geq \frac{L_1}{M_b}$. Note that we can make $\mathbb{P}[Q_H^{max} > \hat{Q}_B^{max}]$ arbitrarily small by selecting sufficiently large L_1. An upper bound on the amount of remote service provided by helpers and devoted

to helpers, denoted by δ_{HH}, is given by $\delta_{HH} \leq R_{\mathcal{H}}\mathbb{P}[Q_H^{max} > \hat{Q}_B^{max}]$, which can be arbitrarily small. Hence $\exists L_1 > 0$ such that for any $\hat{Z}^{(\mathcal{B})}$ with $\left\|\mathbf{Q}_{\perp}^{(\mathcal{B})}\right\| > L_1$, $\delta_{HH} < \frac{M_b\lambda_0}{4}$. Thus we can obtain an lower bound on the amount of remote service provided by helpers and devoted to beneficiaries

$$\mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1}\sum_{n\in\mathcal{H}}\gamma I_{\{\eta_n(t)\in\mathcal{B}\}} \mid \hat{Z}^{(\mathcal{B})}\right] \geq (t_0+T-t^*)\left(R_{\mathcal{H}} - \frac{M_b\lambda_0}{4}\right).$$

This contradicts with the assumption. Note that L_1 does not depend on ϵ, as λ_0 is independent of ϵ. ∎

A.3.8 Proof of Lemma 3.18

For each $m \in \mathcal{B}$, define

$$A_m^e(t) = A_m(t) - \hat{A}_m(t) = \sum_{\bar{L}:\bar{L}\notin\mathcal{L}_{\mathcal{B}}, m\in\bar{L}} A_{\bar{L},m}(t),$$

which gives the extra arrivals of shared types for m. For any $L > 0$, we have

$$\begin{aligned}
&\mathbb{E}\left[\langle\mathbf{Q}(t), \mathbf{A}(t) - \hat{\mathbf{A}}(t)\rangle \mid Z^{(\mathcal{B})}(t_0)\right] = \mathbb{E}\left[\sum_{m\in\mathcal{B}} Q_m(t)A_m^e(t) \mid Z^{(\mathcal{B})}(t_0)\right] \\
= \;& \mathbb{E}\left[\sum_{m\in\mathcal{B}}\left(Q_m(t)I_{\{Q_m(t)<L\}}A_m^e(t) + Q_m(t)I_{\{Q_m(t)\geq L\}}A_m^e(t)\right) \mid Z^{(\mathcal{B})}(t_0)\right] \\
\leq \;& \mathbb{E}\left[L\sum_{m\in\mathcal{B}} A_m^e(t) + \sum_{m\in\mathcal{B}} Q_m(t)I_{\{Q_m(t)\geq L\}}A_m^e(t) \mid Z^{(\mathcal{B})}(t_0)\right] \\
\leq \;& C + Q^{max}(t_0)\mathbb{E}\left[\sum_{m\in\mathcal{B}} I_{\{Q_m(t)\geq L\}}A_m^e(t) \mid Z^{(\mathcal{B})}(t_0)\right].
\end{aligned}$$

For brevity, we use Y to denote the event $\{t^* < t_0 + K, Z^{(\mathcal{B})}(t_0)\}$. Since

$$\begin{aligned} & \mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1} \langle \mathbf{c_b}, \mathbf{Q}(t)\rangle \langle \mathbf{c_b}, \mathbf{A}(t) - \hat{\mathbf{A}}(t)\rangle \mid Y\right] \\ = & \mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1} \frac{\sum_{m\in\mathcal{B}} Q_m(t)}{M_b} \sum_{m\in\mathcal{B}} A_m^e(t) \mid Y\right], \\ \geq & \frac{\sum\limits_{m\in\mathcal{B}} Q_m(t_0)}{M_b} \mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1} \sum_{m\in\mathcal{B}} I_{\{Q_m(t)\geq L\}} A_m^e(t) \mid Y\right] - C, \end{aligned}$$

we can upper bound the conditional expectation of the summation over $[t^* + 1, t_0 + T - 1]$ by

$$\frac{M_b Q^{max}(t_0) - \sum\limits_{m\in\mathcal{B}} Q_m(t_0)}{M_b} \mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1} \sum_{m\in\mathcal{B}} I_{\{Q_m(t)\geq L\}} A_m^e(t) \mid Y\right] + C. \quad \text{(A.14)}$$

Next we will show that $\forall \epsilon < \frac{M_b\lambda_0(\alpha-\gamma)}{4\alpha}$, there exists $L_a > 0$ not depending on ϵ, $\forall L > L_a > 0$,

$$\mathbb{E}\left[\sum_{t=t^*+1}^{t_0+T-1} \sum_{m\in\mathcal{B}} I_{\{Q_m(t)\geq L\}} A_m^e(t) \mid Y\right] \leq (t_0 + T - t^*)\frac{M_b\lambda_0}{4}. \quad \text{(A.15)}$$

Then we can bound term (A.14) as

$$(A.14) \leq (t_0 + T - t^*)\frac{\lambda_0}{4}\left(M_b Q^{max}(t_0) - \sum_{m\in\mathcal{B}} Q_m(t_0)\right) + C.$$

Similar to the proof of inequality (A.13), we prove (A.15) by contradiction. Assume that $\exists \epsilon < \frac{M_b\lambda_0(\alpha-\gamma)}{4\alpha}$, such that $\forall L > 0$, $\exists Z^{(\mathcal{B})}$ such that

$$\mathbb{E}\left[\sum_{m\in\mathcal{B}} I_{\{Q_m(t)\geq L\}} A_m^e(t) \mid Z^{(\mathcal{B})}(t_0)\right] > \frac{M_b\lambda_0}{4}.$$

Then total arrival for $\mathcal{B}$ is bounded as

$$\begin{aligned}
& \mathbb{E}\left[\sum_{m\in\mathcal{B}} A_m(t) \mid Z^{(\mathcal{B})}(t_0)\right] \\
\geq\ & \mathbb{E}\left[\sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}}} A_{\bar{L}}(t) \mid Z^{(\mathcal{B})}(t_0)\right] + \mathbb{E}\left[\sum_{m\in\mathcal{B}} I_{\{Q_m(t)\geq L\}} A^e_m(t) \mid Z^{(\mathcal{B})}(t_0)\right] \\
>\ & \sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}}} \lambda_{\bar{L}} + \frac{M_b\lambda_0}{4} \sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}}} \lambda_{\bar{L}} + \frac{\alpha}{\alpha-\gamma}\epsilon \\
\geq\ & \mathbb{E}\left[\sum_{m\in\mathcal{B}} S_m(t) \mid Z^{(\mathcal{B})}(t_0)\right].
\end{aligned}$$

Thus all beneficiaries grow together when sub-system hits the state $Z^{(\mathcal{B})}(t_0)$. Again shared arrivals will join helper queues. Consider stable helper subsystem with maximum arrival rate $\sum_{\bar{L}\in\mathcal{L}^*_{\mathcal{H}}} \lambda_{\bar{L}}$. Note that there exists uniform bound for stable helper subsystem. The bounded moments of $\left\|\mathbf{Q}^{(\mathcal{H})}\right\|$ ensure that $\mathbb{P}[Q_H^{max} > L_a]$ can be arbitrarily small with sufficiently large L_a. Hence $\exists L_a > 0$ such that the amount of shared arrivals that join $\mathcal{B}$, denoted by a^e, is upper bounded by

$$a^e \leq C_A \mathbb{P}[Q_H^{max} > Q_B^{min} \mid Q^{min} > L_a] < \frac{M_b\lambda_0}{4}.$$

We note the fact that L_a does not depend on ϵ. This contradicts with the assumption.

A.3.9 Proof of Lemma 3.20.

In the following argument, we will focus on the steady state of the system and omit the time (t). We will show that

$$\mathbb{E}\left[\left\|\hat{\mathbf{S}} - \mathbf{S}\right\|^2\right] \leq C_1\epsilon, \quad \mathbb{E}\left[\left\|\mathbf{A} - \hat{\mathbf{A}}\right\|^2\right] \leq C_2\epsilon, \quad \mathbb{E}\left[\|\mathbf{U}\|^2\right] \leq C_3\epsilon,$$

where C_1, C_2, C_3 are constants independent of ϵ. Then

$$\begin{aligned}\mathbb{E}\left[\left\|\hat{\mathbf{U}}\right\|^2\right] &= \mathbb{E}\left[\left\|\hat{\mathbf{S}}-\mathbf{S}+\mathbf{A}-\hat{\mathbf{A}}+\mathbf{U}\right\|^2\right] \\ &\leq 2e\left\|\hat{\mathbf{S}}-\mathbf{S}\right\|^2+2\mathbb{E}\left[\left\|\mathbf{A}-\hat{\mathbf{A}}\right\|^2\right]+2\mathbb{E}\left[\|\mathbf{U}\|^2\right] \\ &\leq 2(C_1+C_2+C_3)\epsilon.\end{aligned}$$

We first show that $\mathbb{E}\left[\left\|\hat{\mathbf{S}}-\mathbf{S}\right\|^2\right] \leq C_1\epsilon$. Since we consider steady state,

$$\mathbb{E}\left[\sum_{m\in\mathcal{B}} S_m\right] = \mathbb{E}\left[\sum_{m\in\mathcal{B}} A_m\right]+\mathbb{E}\left[\sum_{m\in\mathcal{B}} U_m\right] \geq \mathbb{E}\left[\sum_{m\in\mathcal{B}} A_m\right] \geq \sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}}}\lambda_{\bar{L}}$$

$$\mathbb{E}\left[\sum_{m=1}^{M} S_m\right] = \mathbb{E}\left[\sum_{m=1}^{M} A_m\right]+\mathbb{E}\left[\sum_{m=1}^{M} U_m\right] \geq \mathbb{E}\left[\sum_{m=1}^{M} A_m\right] = \sum_{\bar{L}\in\mathcal{L}}\lambda_{\bar{L}}.$$

Let $N_{bb}(t)$ and $N_{bh}(t)$ denote the number of servers in $\mathcal{B}$ that are scheduled to serve beneficiary and remote helper queues at time slot t, respectively. Similarly define $N_{hb}(t)$ and $N_{hb}(t)$ as the number of servers in $\mathcal{H}$ that are scheduled to serve remote beneficiary and helper queues at time slot t, respectively. Then

$$\mathbb{E}\left[\sum_{m\in\mathcal{B}} S_m\right] = \alpha\left(M_b-\mathbb{E}\left[N_{bb}+N_{bh}\right]\right)+\gamma\mathbb{E}\left[N_{hb}+N_{bb}\right] \geq \sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}}}\lambda_{\bar{L}}, \quad \text{(A.16)}$$

$$\begin{aligned}\mathbb{E}\left[\sum_{m=1}^{M} S_m\right] &= \alpha\left(M_b-\mathbb{E}\left[N_{bb}+N_{bh}\right]+M_h-\mathbb{E}\left[N_{hb}+N_{hh}\right]\right) \\ &\quad +\gamma\mathbb{E}\left[N_{hb}+N_{bb}+N_{bh}+N_{hh}\right] \geq \sum_{\bar{L}\in\mathcal{L}}\lambda_{\bar{L}}. \quad \text{(A.17)}\end{aligned}$$

Eliminating $\mathbb{E}\left[N_{hb}\right]$ by adding $\frac{\gamma}{\alpha-\gamma}*$(A.17) to (A.16) yields:

$$(\alpha+\gamma)\mathbb{E}\left[N_{bh}\right]+\alpha\mathbb{E}\left[N_{bb}\right]+\gamma\mathbb{E}\left[N_{hh}\right] \leq \frac{\alpha}{\alpha-\gamma}\epsilon.$$

Therefore

$$\mathbb{E}\left[N_{bh}\right] \leq \frac{\alpha\epsilon}{(\alpha+\gamma)(\alpha-\gamma)},\ \mathbb{E}\left[N_{bb}\right] \leq \frac{\epsilon}{\alpha-\gamma},\ \mathbb{E}\left[N_{hh}\right] \leq \frac{\alpha}{\gamma(\alpha-\gamma)}\epsilon. \quad \text{(A.18)}$$

For $\forall m \in \mathcal{B}$,

$$\begin{aligned}\hat{S}_m - S_m &= X_m^l + \sum_{n\in\mathcal{H}} X_n^r \cdot I_{\{\hat{\eta}_n = m\}} - S_m^l - \sum_{n:n\neq m} R_n I_{\{\eta_n = m\}} \\ &= X_m^l(1 - I_{\{\eta_m = m\}}) + \sum_{n\in\mathcal{H}} X_n^r (I_{\{\hat{\eta}_n = m\}} - I_{\{\eta_n = m\}}) - \sum_{\substack{n\in\mathcal{B}\\ n\neq m}} R_n I_{\{\eta_n = m\}}.\end{aligned}$$

We have

$$\mathbb{E}\left[\sum_{m\in\mathcal{B}} (\hat{S}_m - S_m) \mid Z\right] = \alpha(N_{bb} + N_{bh}) + \gamma N_{hh} - \gamma N_{bb}.$$

It is easy to see that $-(M_b - 1) \leq \hat{S}_m - S_m \leq 1 + M_h$. So $|\hat{S}_m - S_m| \leq M$. Define $\mathcal{M}_b^- = \{m \in \mathcal{B} : \hat{S}_m - S_m \leq 0\}$ and $\mathcal{M}_b^+ = \{m \in \mathcal{B} : \hat{S}_m - S_m > 0\}$.

$$\begin{aligned}&\mathbb{E}\left[\sum_{m\in\mathcal{M}_b^-} (\hat{S}_m - S_m) \mid Z\right] \geq \mathbb{E}\left[\sum_{m\in\mathcal{M}_b^-} \sum_{\substack{n\in\mathcal{B}\\ n\neq m}} -R_n I_{\{\eta_n = m\}} \mid Z\right] \\ \geq\ & -\mathbb{E}\left[\sum_{n\in\mathcal{B}} R_n I_{\{\eta_n \neq n \text{ and } \eta_n \in \mathcal{B}\}} \mid Z\right] = -\gamma N_{bb}.\end{aligned}$$

Thus

$$\begin{aligned}&\mathbb{E}\left[\sum_{m\in\mathcal{B}} (\hat{S}_m - S_m)^2 \mid Z\right] \leq \mathbb{E}\left[\sum_{m\in\mathcal{B}} M|\hat{S}_m - S_m| \mid Z\right] \\ =\ & M\mathbb{E}\left[\sum_{m\in\mathcal{B}} (\hat{S}_m - S_m) - 2\sum_{m\in\mathcal{M}_b^-} (\hat{S}_m - S_m) \mid Z\right] \\ \leq\ & M\left[(\alpha - \gamma)N_{bb} + \alpha N_{bh} + \gamma N_{hh}\right] + 2M\gamma N_{bb} \\ =\ & M\left[(\alpha + \gamma)N_{bb} + \alpha N_{bh} + \gamma N_{hh}\right].\end{aligned}$$

For $\forall m \in \mathcal{H}$,

$$\hat{S}_m - S_m = -\sum_{n:n\neq m} R_n I_{\{\eta_n = m\}}.$$

It is obvious that $-M \leq \hat{S}_m - S_m \leq 0$. Hence

$$\mathbb{E}\left[\sum_{m\in\mathcal{H}}(\hat{S}_m - S_m)^2 \mid Z\right] \leq \mathbb{E}\left[\sum_{m\in\mathcal{H}} -M(\hat{S}_m - S_m) \mid Z\right] = M\gamma(N_{bh} + N_{hh}).$$

Therefore

$$\begin{aligned}\mathbb{E}\left[\sum_{m=1}^{M}(\hat{S}_m - S_m)^2\right] &= \mathbb{E}\left[\mathbb{E}\left[\sum_{m=1}^{M}(\hat{S}_m - S_m)^2 \mid Z\right]\right] \\ \leq\ & M(\alpha+\gamma)(\mathbb{E}[N_{bb}] + \mathbb{E}[N_{bh}]) + 2M\gamma\mathbb{E}[N_{hh}] \leq C_1\epsilon,\end{aligned}$$

where $C_1 = \frac{M(4\alpha+\gamma)}{\alpha-\gamma}$ is a constant not depending on ϵ.

Next we will show that $\mathbb{E}\left[\left\|\mathbf{A} - \hat{\mathbf{A}}\right\|^2\right] \leq C_2\epsilon$. Note that $\forall m \in \mathcal{M}$, $|A_m - \hat{A}_m| \leq C_A$. In particular, for $\forall m \in \mathcal{B}$, $A_m - \hat{A}_m \geq 0$ and for $\forall m \in \mathcal{H}$, $A_m - \hat{A}_m \leq 0$. Let A_s^b denote the total amount of shared tasks that are routed to beneficiary queues. Then we have

$$\begin{aligned}\mathbb{E}\left[\left\|\mathbf{A} - \hat{\mathbf{A}}\right\|^2\right] &\leq \mathbb{E}\left[\sum_{m=1}^{M} C_A|A_m - \hat{A}_m|\right] \\ =\ & C_A\mathbb{E}\left[\sum_{m\in\mathcal{B}}(A_m - \hat{A}_m) - \sum_{m\in\mathcal{H}}(A_m - \hat{A}_m)\right] = 2C_A\mathbb{E}\left[A_s^b\right].\end{aligned}$$

In steady state,

$$\mathbb{E}\left[\sum_{m\in\mathcal{H}} S_m\right] = \mathbb{E}\left[\sum_{m\in\mathcal{H}} A_m\right] + \mathbb{E}\left[\sum_{m\in\mathcal{H}} U_m\right] \geq \mathbb{E}\left[\sum_{m\in\mathcal{H}} A_m\right] = \sum_{\bar{L}\in\mathcal{L}_{\mathcal{H}}^*}\lambda_{\bar{L}} - \mathbb{E}\left[A_s^b\right]$$

$$\mathbb{E}\left[\sum_{m\in\mathcal{B}} S_m\right] = \mathbb{E}\left[\sum_{m\in\mathcal{B}} A_m\right] + \mathbb{E}\left[\sum_{m\in\mathcal{B}} U_m\right] \geq \mathbb{E}\left[\sum_{m\in\mathcal{B}} A_m\right] = \sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}}}\lambda_{\bar{L}} + \mathbb{E}\left[A_s^b\right].$$

On the other hand,

$$\begin{aligned}\mathbb{E}\left[\sum_{m\in\mathcal{H}} S_m\right] &= \alpha(M_h - \mathbb{E}[N_{hh} + N_{hb}]) + \gamma\mathbb{E}[N_{hh} + N_{bh}] \\ &\leq \alpha M_h - \alpha\mathbb{E}[N_{hb}] + \gamma\mathbb{E}[N_{bh}] \qquad \text{(A.19)} \\ \mathbb{E}\left[\sum_{m\in\mathcal{B}} S_m\right] &= \alpha(M_b - \mathbb{E}[N_{bh} + N_{bb}]) + \gamma\mathbb{E}[N_{hb} + N_{bb}] \\ &\leq \alpha M_b - \alpha\mathbb{E}[N_{bh}] + \gamma\mathbb{E}[N_{hb}]. \qquad \text{(A.20)}\end{aligned}$$

Eliminating N_{hb} on the right hand sides of (A.19) and (A.20) yields:

$$\begin{aligned}\frac{\gamma}{\alpha}\mathbb{E}\left[\sum_{m\in\mathcal{H}} S_m\right] + \mathbb{E}\left[\sum_{m\in\mathcal{B}} S_m\right] &\leq \alpha M_b + \gamma M_h - \frac{1}{\alpha}(\alpha^2-\gamma^2)\mathbb{E}\left[N_{bh}\right] \\ &\leq \alpha M_b + \gamma M_h.\end{aligned}$$

We have

$$\begin{aligned}\alpha M_b + \gamma M_h &\geq \frac{\gamma}{\alpha}\sum_{\bar{L}\in\mathcal{L}^*_{\mathcal{H}}} \lambda_{\bar{L}} - \frac{\gamma}{\alpha}\mathbb{E}\left[A^b_s\right] + \sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}}} \lambda_{\bar{L}} + \mathbb{E}\left[A^b_s\right] \\ &= \alpha M_b + \gamma M_h - \epsilon + \frac{\alpha-\gamma}{\alpha}\mathbb{E}\left[A^b_s\right].\end{aligned}$$

Hence $\mathbb{E}\left[A^b_s\right] \leq \frac{\alpha-\gamma}{\alpha}\epsilon$. Therefore,

$$\mathbb{E}\left[\left\|\mathbf{A}-\hat{\mathbf{A}}\right\|^2\right] \leq 2C_A \cdot \frac{\alpha-\gamma}{\alpha}\epsilon = C_2\epsilon,$$

where $C_2 = 2C_A\frac{\alpha-\gamma}{\alpha}$ is a constant.

Now consider the term $\mathbb{E}\left[\|\mathbf{U}\|^2\right]$. Since $0 \leq U_m \leq M$, $\mathbb{E}\left[\|\mathbf{U}\|^2\right] \leq M\mathbb{E}\left[\sum_{m=1}^{M} U_m\right]$. In steady state,

$$\mathbb{E}\left[\sum_{m=1}^{M} U_m\right] = \mathbb{E}\left[\sum_{m=1}^{M} S_m\right] - \mathbb{E}\left[\sum_{m=1}^{M} A_m\right].$$

From (A.20), we have

$$\begin{aligned}\mathbb{E}\left[N_{hb}\right] &\geq \frac{1}{\gamma}\left(\mathbb{E}\left[\sum_{m\in\mathcal{B}} S_m\right] - \alpha M_b + \alpha\mathbb{E}\left[N_{bh}\right]\right) \\ &\geq \frac{1}{\gamma}\left(\sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}}} \lambda_{\bar{L}} - \alpha M_b + \alpha\mathbb{E}\left[N_{bh}\right]\right).\end{aligned}$$

It follows from (A.19) and (A.20) that

$$\begin{aligned}\mathbb{E}\left[\sum_{m=1}^{M} S_m\right] &\leq \alpha M_b + \gamma M_h - (\alpha-\gamma)\mathbb{E}\left[N_{bh}\right] - (\alpha-\gamma)\mathbb{E}\left[N_{hb}\right] \\ &\leq \sum_{\bar{L}\in\mathcal{L}} \lambda_{\bar{L}} + \frac{\alpha}{\gamma}\epsilon.\end{aligned}$$

Therefore

$$\mathbb{E}\left[\|\mathbf{U}(t)\|^2\right] \leq \frac{\alpha M}{\gamma}\epsilon.$$

■

A.3.10 Proof of Lemma 3.21.

$$\begin{aligned}
&\langle \mathbf{Q}(t), \hat{\mathbf{S}}(t) - \mathbf{S}(t)\rangle \\
= &\sum_{m\in\mathcal{H}} Q_m(t)\left(-\sum_{n:n\neq m} R_n(t) I_{\{\eta_n(t)=m\}}\right) + \sum_{m\in\mathcal{B}} Q_m(t)\left(X_m^l(t)\right. \\
&\left. + \sum_{n\in\mathcal{H}} X_n^r(t)\cdot I_{\{\hat{\eta}_n(t)=m\}} - S_m^l(t) - \sum_{n:n\neq m} R_n(t) I_{\{\eta_n(t)=m\}}\right) \\
= &\sum_{m\in\mathcal{H}}\left(X_m^r(t)\sum_{n\in\mathcal{B}} Q_n(t) I_{\{\hat{\eta}_m(t)=n\}} - R_m(t)\sum_{n:n\neq m} Q_n(t) I_{\{\eta_m(t)=n\}}\right) \qquad \text{(A.21)} \\
&+ \sum_{m\in\mathcal{B}}\left(Q_m(t)(X_m^l(t) - S_m^l(t)) - R_m(t)\sum_{n:n\neq m} Q_n(t) I_{\{\eta_m(t)=n\}}\right). \qquad \text{(A.22)}
\end{aligned}$$

By the coupling of $\{X_m^r(t), t \geq 0\}$ with $\{R_m(t), t \geq 0\}$, the expectation of each term in (A.21) can be written as

$$\gamma \sum_{m\in\mathcal{H}} \mathbb{E}\left[\sum_{n\in\mathcal{B}} Q_n(t)(I_{\{\hat{\eta}_m(t)=n\}} - I_{\{\eta_m(t)=n\}}) - \sum_{\substack{n:n\neq m \\ n\in\mathcal{H}}} Q_n(t) I_{\{\eta_m(t)=n\}}\right].$$

Consider the random variable τ_m^t, which is the last time slot before t at

which server m makes a scheduling decision. Then

$$\gamma\mathbb{E}\left[\sum_{n\in\mathcal{B}} Q_n(t)(I_{\{\hat{\eta}_m(t)=n\}} - I_{\{\eta_m(t)=n\}}) - \sum_{\substack{n:n\neq m\\ n\in\mathcal{H}}} Q_n(t) I_{\{\eta_m(t)=n\}}\right] \tag{A.23}$$

$$= \sum_{i=1}^{t} \gamma\mathbb{E}\left[\sum_{n\in\mathcal{B}} Q_n(t)(I_{\{\hat{\eta}_m(t)=n\}} - I_{\{\eta_m(t)=n\}}) \right.$$

$$\left. - \sum_{\substack{n:n\neq m\\ n\in\mathcal{H}}} Q_n(t) I_{\{\eta_m(t)=n\}} \mid \tau_m^t = t-i\right] \cdot \mathbb{P}\left[\tau_m^t = t-i\right].$$

For a particular $\tau_m^t = t-i$, we decompose the probability space based on $Q_m(\tau_m^t)$.

Case (i): $Q_m(\tau_m^t) > 0$

Under the proposed algorithm, $\eta_m(\tau_m^t) = m$ when $Q_m(\tau_m^t) > 0$. Hence the corresponding term equals zero.

Case (ii): $Q_m(\tau_m^t) = 0$

Let $Q_b^{max}(t) = \max_{m:m\in\mathcal{B}}\{Q_m(t)\}$ and $Q_h^{max}(t) = \max_{m:m\in\mathcal{H}}\{Q_m(t)\}$. Under the proposed algorithm, $\eta_m(\tau_m^t) = \arg\max_{n:n\neq m}\{Q_n(\tau_m^t)\}$ if $Q_m(\tau_m^t) = 0$. We further decompose the probability space based on the values of $Q_b^{max}(\tau_m^t)$ and $Q_h^{max}(\tau_m^t)$.

Observe that if $Q_b^{max}(\tau_m^t) > Q_h^{max}(\tau_m^t)$, $\hat{\eta}_m(\tau_m^t) = \eta_m(\tau_m^t) = Q_b^{max}(\tau_m^t)$. Hence the expectation in Eq.(A.23) is equal to zero under this case.

If $Q_b^{max}(\tau_m^t) \leq Q_h^{max}(\tau_m^t)$, $\eta_m(\tau_m^t) = Q_h^{max}(\tau_m^t)$ and $\hat{\eta}_m(\tau_m^t) = Q_b^{max}(\tau_m^t)$. By the boundedness of arrivals and departures, we can upper bound the conditional expectation by

$$\mathbb{E}\big[\quad Q_b^{max}(\tau_m^t) + iC_A - Q_h^{max}(\tau_m^t) + inM$$
$$\mid Q_b^{max}(\tau_m^t) \leq Q_h^{max}(\tau_m^t), Q_m(\tau_m^t) = 0, \tau_m^t\big] \leq i(C_A + M).$$

Then we can obtain an upper bound the term (A.23)

$$\sum_{i=1}^{t} \gamma i(C_A+M) \cdot \mathbb{P}\left[Q_b^{max}(\tau_m^t) \leq Q_h^{max}(\tau_m^t), Q_m(\tau_m^t) = 0, \tau_m^t\right]. \tag{A.24}$$

The event $\{Q_b^{max}(\tau_m^t) \leq Q_h^{max}(\tau_m^t), Q_m(\tau_m^t) = 0, \tau_m^t = t-i\}$ is equivalent

to the event that at time slot $t-i$, server m is idle and is scheduled to the maximum helper queue. Let $k = \arg\max_{n\in\mathcal{H}}\{Q_n(\tau_m^t)\}$. For any time slot between $t-i$ and t, the working status of server m is equal to k. Hence

$$\begin{aligned} & \{Q_b^{max}(\tau_m^t) \leq Q_h^{max}(\tau_m^t), Q_m(\tau_m^t) = 0, \tau_m^t = t-i\} \\ = \; & \{f_m((t-i)^-) = 0, \eta_m(t-i) = k, f_m(t-i+1) = k, ..., f_m(t) = k\}. \end{aligned}$$

We have

$$\begin{aligned} & \mathbb{P}\left[Q_b^{max}(\tau_m^t) \leq Q_h^{max}(\tau_m^t), Q_m(\tau_m^t) = 0, \tau_m^t = t-i\right] \\ = \; & \mathbb{P}\left[f_m((t-i)^-) = 0, \eta_m(t-i) = k, f_m(t-i+1) = k, ..., f_m(t) = k\right] \\ = \; & \mathbb{P}\left[f_m(t-i+1) = k, ..., f_m(t) = k \mid f_m((t-i)^-) = 0, \eta_m(t-i) = k\right] \\ & \cdot\mathbb{P}\left[f_m((t-i)^-) = 0, \eta_m(t-i) = k\right]. \end{aligned}$$

Let Y_m denote the event that when server m becomes idle, it is scheduled to the maximum helper queue. As we consider the steady state,

$$\mathbb{P}\left[f_m((t-i)^-) = 0, \eta_m(t-i) \in \mathcal{H}\right] = \mathbb{P}\left[Y_m\right].$$

By using the chain rule of probability, we have

$$\begin{aligned} & \mathbb{P}\left[f_m(t-i+1) = k, ..., f_m(t) = k \mid f_m((t-i)^-) = 0, \eta_m(t-i) = k\right] \\ = \; & \mathbb{P}\left[f_m(t-i+1) = k, \mid f_m((t-i)^-) = 0, \eta_m(t-i) = k\right] \\ & \cdot \sum_{j=0}^{n-2} \mathbb{P}\left[f_m(t-j) = k \mid f_m(t-j-1) = k, \cdots,\right. \\ & \quad \left. f(t-i+1) = k, f_m((t-i)^-) = 0, \eta_m(t-i) = k\right]. \end{aligned}$$

Given that server m is scheduled to serve a remote task from another helper queue at time slot $t-i$, the working status $f_m(t-i+1)$ is determined by the random variable $R_m(t-i+1) \sim Bern(\gamma)$. Hence

$$\mathbb{P}\left[f_m(t-i+1) = k, \mid f_m((t-i)^-) = 0, \eta_m(t-i) = k\right] = 1-\gamma.$$

Similarly, for any $j = 0, 1, ..., n-2$, given $f_m(t-j-1) = k$, $f_m(t-j)$ is

determined by $R_m(t-j-1) \sim Bern(\gamma)$, thus

$$\mathbb{P}\left[f_m(t-j) = k \mid f_m(t-j-1) = k, \cdots, \right.$$
$$\left. f(t-i+1) = k, f_m((t-i)^-) = 0, \eta_m(t-i) = k \right] = 1-\gamma.$$

Now we can bound (A.23) as

$$\begin{aligned}
& \mathbb{E}\left[\sum_{n\in\mathcal{B}} Q_n(t)(I_{\{\hat{\eta}_m(t)=n\}} - I_{\{\eta_m(t)=n\}}) - \sum_{\substack{n:n\neq m \\ n\in\mathcal{H}}} Q_n(t) I_{\{\eta_m(t)=n\}}\right] \\
\leq & \sum_{i=1}^{t} i(C_A+M)(1-\gamma)^i \cdot \mathbb{P}\left[Y_m\right] \\
\leq & (C_A+M)(1-\gamma)\cdot\mathbb{P}\left[Y_m\right]\sum_{i=1}^{\infty} i(1-\gamma)^{i-1} \\
= & \frac{(C_A+M)(1-\gamma)}{\gamma^2}\cdot\mathbb{P}\left[Y_m\right].
\end{aligned}$$

Next we will bound the expectation of term (A.22) in a similar way. Again consider the random variable τ_m^t and first decompose the probability space based on τ_m^t. For a particular $\tau_m^t = t-i$, consider $Q_m(\tau_m^t)$.

Case (i): $Q_m(\tau_m^t) > 0$

Under the proposed algorithm, $\eta_m(t) = \eta_m(\tau_m^t) = m$ with $Q_m(\tau_m^t) > 0$. And $X_m^l(t) = S_m^l(t)$. Hence the term (A.22) is equal to zero.

Case (ii): $Q_m(\tau_m^t) = 0$

When $Q_m(\tau_m^t) = 0$, $\eta_m(\tau_m^t) = \arg\max_{n\neq m}\{Q_n(\tau_m^t)\}$. By the bounded difference between $Q_m(t)$ and $Q_m(\tau_m^t)$, similarly we can upper bound the conditional expectation by $i(\alpha C_A + \gamma M)$. Thus we can upper bound the expectation of each term in (A.22) by

$$\sum_{i=1}^{t} i(\alpha C_A+\gamma M)\cdot\mathbb{P}\left[Q_m(\tau_m^t)=0 \mid \tau_m^t = t-i\right]\cdot\mathbb{P}\left[\tau_m^t = t-i\right]. \quad \text{(A.25)}$$

For $\forall m \in \mathcal{B}$, let V_m denote the event that when server m is idle, its local queue is empty so it is scheduled to serve the maximum queue in the system.

In steady state,

$$\mathbb{P}\left[Q_m(\tau_m^t)=0, \tau_m^t = t-n\right] = \mathbb{P}\left[V_m\right].$$

Similar to the analysis for (A.24), we can bound the term (A.25) by

$$\frac{(\alpha C_A + \gamma M))(1-\gamma)}{\gamma^2} \cdot \mathbb{P}\left[V_m\right].$$

Therefore,

$$\begin{aligned} & \mathbb{E}\left[\langle \mathbf{Q}(t), \hat{\mathbf{S}}(t) - \mathbf{S}(t)\rangle\right] \\ \leq & \frac{(C_A+M)(1-\gamma)}{\gamma} \sum_{m\in\mathcal{H}} \mathbb{P}\left[Y_m\right] + \frac{(\alpha C_A + \gamma M)(1-\gamma)}{\gamma^2} \sum_{m\in\mathcal{B}} \mathbb{P}\left[V_m\right]. \end{aligned}$$

On the other hand,

$$\begin{aligned} & \sqrt{M}\langle \mathbf{e}, \hat{\mathbf{S}}(t) - \mathbf{S}(t)\rangle \\ = & \sum_{m\in\mathcal{H}} \left(- \sum_{n:n\neq m} R_n(t) I_{\{\eta_n(t)=m\}} \right) \\ & + \sum_{m\in\mathcal{B}} \left(X_m^l(t) + \sum_{n\in\mathcal{H}} X_n^r(t) \cdot I_{\{\hat{\eta}_n(t)=m\}} - S_m^l(t) - \sum_{n:n\neq m} R_n(t) I_{\{\eta_n(t)=m\}} \right) \\ = & \sum_{m\in\mathcal{H}} \left(X_m^r(t) I_{\{\hat{\eta}_m(t)\neq m\}} - R_m(t) I_{\{\eta_m(t)\neq m\}} \right) \\ & + \sum_{m\in\mathcal{B}} \left(X_m^l(t) - S_m^l(t) - R_m(t) I_{\{\eta_m(t)\neq m\}} \right) \\ \overset{(a)}{=} & \sum_{m\in\mathcal{B}} \left(X_m^l(t)(1 - I_{\{\eta_m(t)=m\}}) - R_m(t) I_{\{\eta_m(t)\neq m\}} \right), \end{aligned}$$

where (a) comes from the coupling of X_m^r and R_m for $m \in \mathcal{H}$.

For $\forall m \in \mathcal{B}$,

$$\begin{aligned} & \mathbb{E}\left[X_m^l(t)(1 - I_{\{\eta_m(t)=m\}}) - R_m(t) I_{\{\eta_m(t)\neq m\}}\right] \\ = & (\alpha-\gamma)\mathbb{P}\left[I_{\{\eta_m(t)\neq m\}}\right] = (\alpha-\gamma)\mathbb{P}\left[V_m\right]. \end{aligned}$$

Thus

$$\mathbb{E}\left[\langle \mathbf{e}, \hat{\mathbf{S}}(t) - \mathbf{S}(t)\rangle\right] \geq (\alpha-\gamma) \sum_{m\in\mathcal{B}} \mathbb{P}\left[V_m\right].$$

It follows from (A.18) in the proof of Lemma 3.20 that

$$\sum_{m\in\mathcal{H}} \mathbb{P}\left[Y_m\right] = \mathbb{E}\left[N_{hh}(t)\right] \leq \frac{\alpha}{\gamma(\alpha-\gamma)}\epsilon.$$

Therefore

$$\begin{aligned}
& \mathbb{E}\left[\langle \mathbf{Q}(t), \hat{\mathbf{S}}(t) - \mathbf{S}(t)\rangle\right] \\
\leq\ & \frac{(C_A+M)(1-\gamma)}{\gamma}\sum_{m\in\mathcal{H}}\mathbb{P}\left[Y_m\right] + \frac{(\alpha C_A+\gamma M)(1-\gamma)}{\gamma^2(\alpha-\gamma)}\mathbb{E}\left[\langle \mathbf{e}, \hat{\mathbf{S}}(t) - \mathbf{S}(t)\rangle\right] \\
\leq\ & R_0\epsilon + R_1\sqrt{M}e\langle \mathbf{e}, \hat{\mathbf{S}}(t) - \mathbf{S}(t)\rangle,
\end{aligned}$$

where $R_0 = \frac{\alpha(C_A+M)(1-\gamma)}{\gamma^2(\alpha-\gamma)}$, $R_1 = \frac{(\alpha C_A+\gamma M)(1-\gamma)}{\gamma^2(\alpha-\gamma)\sqrt{M}}$ are constants independent of ϵ. ∎

A.3.11 Proof of Lemma 3.22 .

We denote by $\mathcal{L}_s$ the set of task types that are local both to helpers and beneficiaries. By the definition of $\hat{\mathbf{A}}$:

$$\begin{aligned}
& \langle \mathbf{Q}, \mathbf{A} - \hat{\mathbf{A}}\rangle \\
=\ & \sum_{m\in\mathcal{B}}\left(Q_m(t)\sum_{\bar{L}\in\mathcal{L}_s: m\in\bar{L}} A_{\bar{L},m}\right) - \sum_{m\in\mathcal{H}}\left(Q_m(t)\sum_{\bar{L}\in\mathcal{L}_s: m\in\bar{L}}\frac{\sum_{n\in\bar{L}\cap\mathcal{B}} A_{\bar{L},n}}{|\{k: k\in\bar{L}\cap\mathcal{H}\}|}\right) \\
=\ & \sum_{\bar{L}\in\mathcal{L}_s}\left[\sum_{n\in\bar{L}\cap\mathcal{B}} Q_n(t)A_{\bar{L},n} - \sum_{m\in\bar{L}\cap\mathcal{H}} Q_m(t)\frac{\sum_{n\in\bar{L}\cap\mathcal{B}} A_{\bar{L},n}}{|\{k: k\in\bar{L}\cap\mathcal{H}\}|}\right] \\
=\ & \sum_{\substack{\bar{L}=(m_1,m_2,m_3)\\ m_1,m_2\in\mathcal{H},m_3\in\mathcal{B}}}\left[Q_{m_3}(t)A_{\bar{L},m_3} - (Q_{m_1}(t)+Q_{m_2}(t))\frac{A_{\bar{L},m_3}}{2}\right] \\
+\ & \sum_{\substack{\bar{L}=(m_1,m_2,m_3)\\ m_1\in\mathcal{H},m_2,m_3\in\mathcal{B}}}\left[Q_{m_2}(t)A_{\bar{L},m_2} + Q_{m_3}(t)A_{\bar{L},m_3} - Q_{m_1}(t)(A_{\bar{L},m_2}+A_{\bar{L},m_3})\right].
\end{aligned}$$

Case (i): $\bar{L} = (m_1, m_2, m_3)$ with $m_1, m_2 \in \mathcal{H}, m_3 \in \mathcal{B}$.

As arriving tasks are routed to the shortest local queue, $A_{\bar{L},m_3} > 0$ only if $Q_{m_3} \leq Q_{m_1}(t)$ and $Q_{m_3} \leq Q_{m_2}(t)$. Hence $A_{\bar{L},m_3}\left(Q_{m_3} - (Q_{m_1}(t)+Q_{m_2}(t))/2\right) \leq 0$.

Case (ii): $\bar{L} = (m_1, m_2, m_3)$ with $m_1 \in \mathcal{H}, m_2, m_3 \in \mathcal{B}$.

Similarly, $A_{\bar{L},m_2} > 0$ only if $Q_{m_2} \leq Q_{m_1}(t)$. Hence $A_{\bar{L},m_2}(Q_{m_2} - Q_{m_1}(t)) \leq 0$. Similarly $A_{\bar{L},m_3}(Q_{m_3} - Q_{m_1}(t)) \leq 0$.

Therefore,

$$\langle \mathbf{Q}, \mathbf{A} - \hat{\mathbf{A}} \rangle \leq 0.$$

■

A.3.12 Proof of Lemma 3.23.

Proof. By the definition of $U_m(t)$, $0 \leq U_m(t) \leq M$. If $U_m(t) = 0$, $Q_m(t)U_m(t) = 0$. We note the fact that $U_m(t) > 0$ only if the number of tasks in Q_m is less than the number of available servers scheduled to Q_m at time t. Since $S_m(t) \leq M$, we have $Q_m(t) \leq Q_m(t) + A_m(t) < M$. Hence $Q_m(t)U_m(t) < MU_m(t)$. Therefore, $\langle \mathbf{Q}(t), \mathbf{U}(t) \rangle < \sum_{m \in \mathcal{M}} MU_m(t) = M\sqrt{M}\langle \mathbf{e}, \mathbf{U}(t) \rangle$, where $e = \frac{1}{\sqrt{M}} \underbrace{(1, 1, \cdots, 1)}_{M}$. ■

A.4 Heavy-traffic Optimality with Evenly Loaded Traffic

A.4.1 Proof of Lemma 3.24

Fix an $0 < \epsilon < \frac{M\alpha}{2}$, i.e., $0 < \epsilon_0 < \frac{1}{2}$. There exists a constant $\theta > 0$ such that $0 < \epsilon_0 \leq \frac{1}{1+\theta}$. Since $\bar{\boldsymbol{\lambda}} \in \mathcal{F}$ satisfies the resource pooling condition, there exists a decomposition $\{\bar{\lambda}^*_{\bar{L},n,m}\}$ of $\bar{\boldsymbol{\lambda}}$ such that Eq. (3.36) and all servers are connected. As $\boldsymbol{\lambda}^{(\epsilon)} = (1 - \epsilon_0)\bar{\boldsymbol{\lambda}}$, it is easy to see that $\{\lambda^*_{\bar{L},n,m}\}$ with $\lambda^*_{\bar{L},n,m} = (1 - \epsilon_0)\bar{\lambda}^*_{\bar{L},n,m}$ for any $\bar{L} \in \mathcal{L}$, any $n \in \mathcal{L}$, and any $m \in \mathcal{M}$ gives a decomposition of $\boldsymbol{\lambda}$. By the property of $\{\bar{\lambda}^*_{\bar{L},n,m}\}$, condition 1 holds naturally under the decomposition $\{\lambda^*_{\bar{L},n,m}\}$, and $\mathcal{H} = \mathcal{M}$, i.e., Lemma 3.4 is satisfied. Define

$$\kappa = \min_{\substack{\forall \bar{L} \in \mathcal{L}, \forall m \in \bar{L} \\ \bar{\lambda}^*_{\bar{L},m,m} > 0}} \left\{ \bar{\lambda}^*_{\bar{L},m,m} \right\}.$$

It follows that for any $\lambda^*_{\bar{L},m,m} > 0$,

$$\lambda^*_{\bar{L},m,m} = (1 - \epsilon_0)\bar{\lambda}^*_{\bar{L},m,m} \geq \frac{\kappa}{2}.$$

Let $\lambda_{min} = \frac{\kappa}{2}$, which is independent of ϵ. Since all servers are connected under $\{\lambda^*_{\bar{L},n,m}\}$, condition 2 is satisfied with λ_{min}. ∎

A.4.2 Proof of Lemma 3.28

The proof is similar to that of Lemma 3.15. For $\forall \bar{L} \in \mathcal{L}$, define $Q^*_{\bar{L}}(t) = \min_{m\in\bar{L}}\{Q_m(t)\}$. Thus tasks of type $\bar{L}$ will be routed to queue $Q^*_{\bar{L}}(t)$ at the beginning of time slot t. We have

$$\mathbb{E}\left[\langle \mathbf{Q}(t), \mathbf{A}(t)\rangle \mid Z(t)\right] = \sum_{\bar{L}} \sum_{m:m\in\bar{L}} \lambda^*_{\bar{L},m,m} Q^*_{\bar{L}}(t).$$

Hence

$$\begin{aligned} & \mathbb{E}\left[\langle \mathbf{Q}(t), \mathbf{A}(t)\rangle - \langle \mathbf{Q}(t), \boldsymbol{\lambda}^*\rangle \mid Z(t)\right] \\ = & -\sum_{\bar{L}} \sum_{m:m\in\bar{L}} \lambda^*_{\bar{L},m,m}(Q_m(t) - Q^*_{\bar{L}}(t)). \end{aligned} \tag{A.26}$$

Assume that $m_1 = \arg\max_{m\in\mathcal{M}}\{Q_m(t)\}$, and $m' = \arg\min_{m\in\mathcal{M}}\{Q_m(t)\}$. Denote the maximum queue length at time slot t by $Q^{max}(t)$. That is, $Q^{max}(t) = Q_{m_1}(t)$. Note that for any $\bar{L} \in \mathcal{L}$ such that $m' \in \bar{L}$, $Q^*_{\bar{L}}(t) = Q_{m'}(t)$ as $Q_{m'}$ is the minimum queue at time t.

From Lemma 3.24, there exists a sequence of servers $(m_1, m_2, ..., m_{k-1}, m_k)$ such that $m_k = m'$, and m_i is connected directly with m_{i+1} under the ideal decomposition, for all $i = 1, 2, \cdots, k-1$. That is, there exists a task type $\bar{L}_{i,i+1}$ local to both server m_i and m_{i+1}, satisfying $\lambda_{\bar{L}_{i,i+1},m_i,m_i} \geq \lambda_{min}$, and $\lambda_{\bar{L}_{i,i+1},m_{i+1},m_{i+1}} \geq \lambda_{min}$. For the summation in (A.26), we keep terms of types $\bar{L}_{1,2}, \bar{L}_{2,3}, ..., \bar{L}_{k-1,k}$, and for each task type $\bar{L}_{i,i+1}$, we only keep $m = m_i$ term.

All other terms are discarded. It follows that

$$\begin{aligned}
& \mathbb{E}\left[\langle \mathbf{Q}(t), \mathbf{A}(t)\rangle - \langle \mathbf{Q}(t), \boldsymbol{\lambda}^*\rangle \mid Z(t)\right] \\
\leq\ & -\lambda_{min}\left(Q_{m_1}(t) - Q^*_{\bar{L}_{1,2}} + Q_{m_2}(t) - Q^*_{\bar{L}_{2,3}} + \cdots + Q_{m_{k-1}}(t) - Q^*_{\bar{L}_{k-1,k}}\right) \\
\leq\ & -\lambda_{min}\left(Q_{m_1}(t) - Q^*_{\bar{L}_{k-1,k}}\right) \\
=\ & -\lambda_{min}\left(Q^{max}(t) - Q^{min}(t)\right) \\
\leq\ & -\frac{\lambda_{min}}{\sqrt{M}}\sqrt{\sum_m \left(Q_m(t) - \frac{\sum_i Q_i(t)}{M}\right)^2} \\
=\ & -\frac{\lambda_{min}}{\sqrt{M}}\|\mathbf{Q}_\perp(t)\| \\
\overset{(a)}{\leq}\ & -\frac{\lambda_{min}}{\sqrt{M}}\|\mathbf{Q}_\perp(t_0)\| + F_1',
\end{aligned}$$

where the inequality (a) comes from Lemma 3.26. This completes the proof. ■

A.4.3 Proof of Lemma 3.29

The proof is the same as that of Lemma 3.8. Observe that for any $m \in \mathcal{M}$, $\sum_{\bar{L}:m\in\bar{L}} \lambda_{\bar{L},m,m} = \alpha(1 - \frac{\epsilon}{M})$ by Lemma 3.24. Replacing $\frac{\alpha}{1+\vartheta}$ with $\alpha(1 - \frac{\epsilon}{M})$ gives the part on the right hand side in Lemma 3.29.

A.4.4 Proof of Lemma 3.30

By the boundedness of arrivals and service, we have

$$\langle \mathbf{c_e}, \mathbf{Q}(t_0)\rangle - \frac{TM}{\sqrt{M}} \leq \langle \mathbf{c_e}, \mathbf{Q}(t)\rangle \leq \langle \mathbf{c_e}, \mathbf{Q}(t_0)\rangle + \frac{TC_A}{\sqrt{M}}.$$

Hence

$$\begin{aligned}
& \mathbb{E}\left[\langle \mathbf{c_e}, \mathbf{Q}(t)\rangle\langle \mathbf{c_e}, \mathbf{A}(t) - \mathbf{S}(t)\rangle \mid t^*, Z(t_0)\right] \\
\geq\ & \langle \mathbf{c_e}, \mathbf{Q}(t_0)\rangle\frac{1}{\sqrt{M}}\mathbb{E}\left[\sum_{m\in\mathcal{M}}(A_m(t) - S_m(t)) \mid t^*, Z(t_0)\right] - F_3',
\end{aligned}$$

where F_3' is a constant.

Note that

$$\mathbb{E}\left[\sum_{m\in\mathcal{M}} A_m(t)\right] = \mathbb{E}\left[\sum_{\bar{L}\in\mathcal{L}} A_{\bar{L}}(t)\right] = \sum_{\bar{L}\in\mathcal{L}} \lambda_{\bar{L}} = M\alpha - \epsilon,$$

$$\mathbb{E}\left[\sum_{m\in\mathcal{M}} S_m(t) \mid t^*, Z(t_0)\right] = \mathbb{E}\left[\sum_{m\in\mathcal{M}} (\alpha I_{\{\eta_m(t)=m\}} + \gamma I_{\{\eta_m(t)\neq m\}}) \mid t^*, Z(t_0)\right]$$
$$\leq \quad M\alpha.$$

Combining the above inequalities yields

$$\begin{aligned} & \mathbb{E}\left[\langle \mathbf{c_e}, \mathbf{Q}(t)\rangle \langle \mathbf{c_e}, \mathbf{A}(t) - \mathbf{S}(t)\rangle \mid t^*, Z(t_0)\right] \\ \geq \quad & \langle \mathbf{c_e}, Q(t_0)\rangle \frac{1}{\sqrt{M}}(-\epsilon) - F_3' \\ = \quad & -\frac{\epsilon}{M}\sum_m Q_m(t_0) - F_3'. \end{aligned}$$

■

APPENDIX B

ADDITIONAL PROOFS FOR BALANCED-PANDAS

B.1 Proof of Theorem 4.1

To prove Theorem 4.1, we first establish the equivalence of the capacity region $\bar{\Lambda}$ and then use the refined decomposition as an intermediary.

Lemma B.1. *The set $\bar{\Lambda}$ is equivalent to Λ.*

Proof. The proof is straightforward. For Λ, The rate $\lambda_{\bar{L}}$ is decomposed into $\lambda_{\bar{L},m}$, which is the rate of type-$\bar{L}$ arrival allocated to server m. We further refine the decomposition by simply writing $\lambda_{\bar{L},m} \equiv \sum_n \lambda_{\bar{L},n,m}$, where n is the index of the queue to which a task is local. It is easy to show that $\bar{\Lambda} \subset \Lambda$ by defining $\lambda_{\bar{L},m} \equiv \sum_n \lambda_{\bar{L},n,m}$.

To show the reverse direction, $\forall \boldsymbol{\lambda} \in \Lambda$, $\forall \bar{L}$, $\forall n \in \mathcal{L},\ m \in \mathcal{M}$, define

$$\lambda'_{\bar{L},n,m} = \frac{\lambda_{\bar{L},m}}{|\bar{L}|}.$$

It is obvious that the constructed decomposition $\{\lambda'_{\bar{L},n,m}\}$ satisfies condition (4.3). Thus $\boldsymbol{\lambda} \in \bar{\Lambda}$, i.e., $\Lambda \subset \bar{\Lambda}$. ∎

For any arrival rate vector $\boldsymbol{\lambda} \in \bar{\Lambda}$, there exists $\delta > 0$ such that $\boldsymbol{\lambda'} = (1+\delta)\boldsymbol{\lambda} \in \bar{\Lambda}$. Thus there exists a decomposition $\{\lambda'_{\bar{L},n,m}\}$ of $\boldsymbol{\lambda'}$, which satisfies condition (4.3). Let

$$\lambda_{\bar{L},n,m} = \frac{\lambda'_{\bar{L},n,m}}{1+\delta}, \forall \bar{L},\ \forall n \in \mathcal{L},\ m \in \mathcal{M}.$$

Hence $\{\lambda_{\bar{L},n,m}\}$ gives a decomposition of $\boldsymbol{\lambda}$. For any m,

$$\sum_{\bar{L}:m\in\bar{L}} \sum_{n:n\in\bar{L}} \frac{\lambda_{\bar{L},n,m}}{\alpha} + \sum_{\bar{L}:m\in\bar{L}_k} \sum_{n:n\in\bar{L}} \frac{\lambda_{\bar{L},n,m}}{\beta} + \sum_{\bar{L}:m\notin\bar{L}} \sum_{n:n\in\bar{L}} \frac{\lambda_{\bar{L},n,m}}{\gamma} \leq \frac{1}{1+\delta}.$$

Define $\boldsymbol{\psi} = (\psi_1, \psi_2, \cdots, \psi_M)$ as

$$\psi_n = \sum_{\bar{L}: n \in \bar{L}} \sum_{m=1}^{M} \lambda_{\bar{L},n,m}, \ \forall n, \tag{B.1}$$

which can be regarded as the arrival rate for server n under the JSQ-MaxWeight algorithm. The following two lemmas use $\boldsymbol{\psi}$ as an intermediary to show some properties of arrival distribution and service for the system under the JSQ-MaxWeight algorithm, which are the key steps for the proof of Theorem 4.1. The proofs of these two lemmas are identical to that of Lemma 2 and 3 in [37]. We omit the details here.

Lemma B.2. *Consider any arrival rate vector $\boldsymbol{\lambda} \in \bar{\Lambda}$ and the corresponding $\boldsymbol{\psi}$ defined in (B.1). Under the JSQ routing algorithm, for any t_0 and any $t \geq t_0$,*

$$\mathbb{E}\left[\langle \mathbf{Q}(t), \mathbf{A}(t) \rangle - \langle \mathbf{Q}(t), \boldsymbol{\psi} \rangle | Z(t_0)\right] \leq 0.$$

Lemma B.3. *Consider any arrival rate vector $\boldsymbol{\lambda} \in \bar{\Lambda}$ and the corresponding $\boldsymbol{\psi}$ defined in (B.1). Under the MaxWeight scheduling algorithm, there exists $T_1 > 0$ such that for any $T > T_1$ and any t_0,*

$$\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1} \left(\langle \mathbf{Q}(t), \boldsymbol{\psi} \rangle - \langle \mathbf{Q}(t), \mathbf{S}(t) \rangle\right) | Z(t_0)\right] \leq -\theta_1 \|\mathbf{Q}(t_0)\|_1 + C_1,$$

where $\theta_1 > 0$ and C_1 are constants independent of $Z(t_0)$.

We also need the following lemma for the proof of Theorem 4.1.

Lemma B.4. *For any t,*

$$\langle \mathbf{Q}(t), \mathbf{U}(t) \rangle \leq M^2.$$

Proof. By the definition of $U_m(t)$, if $U_m(t) > 0$, the number of tasks in queue m must be less than the number of available servers scheduled to this queue at time slot t. Since $S_m(t) \leq M$, $Q_m(t) < M$. Note that $U_m(t) \leq M$ If $U_m(t) = 0$, $Q_m(t)U_m(t) = 0$. Hence $Q_m(t)U_m(t) < MU_m(t)$. Therefore, $\langle \mathbf{Q}(t), \mathbf{U}(t) \rangle < \sum_{m \in \mathcal{M}} MU_m(t) = M^2$. ∎

Proof of Theorem 4.1. Consider the following Lyapunov function:

$$F(Z(t)) = \|\mathbf{Q}(t))\|^2.$$

The corresponding T-period drift is given by:

$$\begin{aligned}\Delta F(Z(t_0)) &= \mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1}\left(F(t+1)-F(t)\right)|Z(t_0)\right] \\ &= \mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1}\left(2\langle\mathbf{Q}(t),\mathbf{A}(t)-\mathbf{S}(t)\rangle+2\langle\mathbf{Q}(t),\mathbf{U}(t)\rangle\right.\right. \\ &\quad \left.\left.+\|\mathbf{A}(t)-\mathbf{S}(t)+\mathbf{U}(t)\|^2\right)|Z(t_0)\right].\end{aligned}$$

By Lemma B.4, the term $\langle\mathbf{Q}(t),\mathbf{U}(t)\rangle \leq M^2$. Since both the arrival vector $\mathbf{A}(t)$ and the service vector $\mathbf{S}(t)$ are bounded, so as the unused vector $\mathbf{U}(t)$, the term $\|\mathbf{A}(t)-\mathbf{S}(t)+\mathbf{U}(t)\|^2$ can be bounded by a constant. Thus the T-time slot drift can be bounded as

$$\Delta F(Z(t_0)) = 2\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1}\langle\mathbf{Q}(t),\mathbf{A}(t)-\mathbf{S}(t)\rangle|Z(t_0)\right]+C.$$

For any arrival rate vector $\boldsymbol{\lambda} \in \bar{\Lambda}$ and the corresponding $\boldsymbol{\psi}$, we split the expectation term into two terms using $\boldsymbol{\psi}$:

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1}\langle\mathbf{Q}(t),\mathbf{A}(t)-\mathbf{S}(t)\rangle|Z(t_0)\right] \\ =\ &\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1}\left(\langle\mathbf{Q}(t),\mathbf{A}(t)\rangle-\langle\mathbf{Q}(t),\boldsymbol{\psi}\rangle\right)|Z(t_0)\right] \\ +\ &\mathbb{E}\left[\sum_{t=t_0}^{t_0+T-1}\left(\langle\mathbf{Q}(t),\boldsymbol{\psi}\rangle-\langle\mathbf{Q}(t),\mathbf{S}(t)\rangle\right)|Z(t_0)\right].\end{aligned}$$

By Lemma B.2-B.3, we have

$$\Delta F(Z(t_0)) \leq -2\theta_1\|\mathbf{Q}(t_0)\|_1 + C_2,$$

where $C > 0$ is a constant.

Pick $T \geq T_0$ and any $\epsilon > 0$. Let $\mathcal{P} = \left\{Z = (\mathbf{Q},\mathbf{f})|\ \|\mathbf{Q}\|_1 \leq \frac{C+\epsilon}{2\theta_1}\right\}$. Then $\mathcal{P}$ is a finite subset of state space. For any $Z \in \mathcal{P}^c$, $\Delta F(Z) \leq -\epsilon$. Therefore

the Markov process $\{Z(t), t \geq 0\}$ is positive recurrent. As a result, $\bar{\Lambda}$ and Λ are the capacity region of the system, and the JSQ-MaxWeight algorithm is throughput optimal.

B.2 Proof of Theorem 4.2

We can show that the JSQ-MaxWeight algorithm achieves first-order heavy-traffic optimality in this special scenario. The proof follows the Lyapunov drift-based approach developed in [59], which consists of three steps: first obtain a lower bound; then show state space collapse; and finally use the state space collapse result to obtain an upper bound. In particular, the proof of state-space collapse and upper bound are identical to that in [37]. We omit the proof details here and just state the results.

For any arrival rate in the capacity region, we know that a steady state distribution exists under the JSQ-MaxWeight algorithm. Let $\bar{\mathbf{Q}}$ denote the steady state random vector. Let $\sigma_1^{(\epsilon)}$ be the standard deviation of the arrival rate vector $\{A_{\bar{L}}^{(\epsilon)}(t), t \geq 0\}_{\bar{L} \in \mathcal{L}}$, which converges to a constant σ_1.

Lower Bound

Consider a single server queueing system with arrival process $\sum_{\bar{L}} A_{\bar{L}}^{(\epsilon)}(t)$, and service process $b_1(t)$:

$$b_1(t) = \sum_{i \in \mathcal{B}_o} X_i(t) + \sum_{j \in \mathcal{H}_o} Y_j(t) + \sum_{n \in \mathcal{H}_u} V_n(t),$$

where all $\{X_i(t)\}_{i \in \mathcal{B}_o}$, $\{Y_j(t)\}_{j \in \mathcal{H}_o}$ and $\{V_n(t)\}_{n \in \mathcal{H}_u}$ are independent and each process is i.i.d. In particular, $X_i(t) \sim \text{Bern}(\alpha)$, $Y_j(t) \sim \text{Bern}(\beta)$ and $V_n(t) \sim \text{Bern}(\gamma)$. We denote $\text{Var}(b_1(t))$ by ν_1^2. As the corresponding queue length process of this single server system is stochastically smaller than the sum of queue length in the original system, we can obtain the following lower bound:

$$\mathbb{E}\left[\sum_{m=1}^{M} \bar{Q}_m^{(\epsilon)}\right] \geq \frac{(\sigma_1^{(\epsilon)})^2 + \nu_1^2 + \epsilon^2}{2\epsilon} - \frac{M}{2}.$$

Therefore, in the heavy traffic limit, we have

$$\liminf_{\epsilon \to 0^+} \epsilon \mathbb{E}\left[\sum_{m=1}^{M} \bar{Q}_m^{(\epsilon)}\right] \geq \frac{\sigma_1^2 + \nu_1^2}{2}.$$

State Space Collapse

We expect that state space under the JSQ-MaxWeight algorithm collapses along the direction where all $\mathcal{B}_o$ queues are equal. Let $\mathbf{c}_1 \in \mathbb{R}^M$ be the unit vector where

$$c_{1m} = \begin{cases} \frac{1}{\sqrt{M_{B_o}}}, & \forall m \in \mathcal{B}_o \\ 0, & \text{else.} \end{cases}$$

The parallel and perpendicular components of $\mathbf{Q}$ with respect to the direction $\mathbf{c}_1$ are defined as:

$$\mathbf{Q}_{||} = \langle \mathbf{c}_1, \mathbf{Q} \rangle \mathbf{c}_1, \quad \mathbf{Q}_{\perp} = \mathbf{Q} - \mathbf{Q}_{||}.$$

Consider the Lyapunov function $F_{\perp}(\mathbf{Z}) = \|\mathbf{Q}_{\perp}\|$. We can show that the drift of $F_{\perp}(\mathbf{Z})$ is always finite and becomes negative for sufficiently large $F_{\perp}$. We then obtain state space collapse by Lemma 3.6. That is, there exists a sequence of finite numbers $\{C_r : r \in \mathbb{N}\}$ such that for each positive integer r,

$$\mathbb{E}\left[\left\|\bar{\mathbf{Q}}_{\perp}^{(\epsilon)}\right\|^r\right] \leq C_r,$$

where $\bar{\mathbf{Q}}_{\perp}$ is the component of $\bar{\mathbf{Q}}$ perpendicular to $\mathbf{c}_1$.

Upper Bound

By utilizing the result of state space collapse, we can obtain the following upper bound on the expected queue length in steady state:

$$\mathbb{E}\left[\sum_{m=1}^{M} \bar{Q}_m^{(\epsilon)}\right] \leq \frac{(\sigma_1^{(\epsilon)})^2 + \nu_1^2}{2\epsilon} + B^{(\epsilon)},$$

where $B^{(\epsilon)} = o(\frac{1}{\epsilon})$, i.e., $\lim_{\epsilon \to 0^+} \epsilon B^{(\epsilon)} = 0$. Therefore, in the heavy-traffic limit, we have

$$\limsup_{\epsilon \to 0^+} \epsilon \mathbb{E}\left[\sum_{m=1}^{M} \bar{Q}_m^{(\epsilon)}\right] \leq \frac{\sigma^2 + \nu_1^2}{2}.$$

The heavy-traffic optimality of the proposed algorithm follows by the coincidence of lower and upper bounds.

B.3 Proof of Theorem 4.3

By Theorem 4.1, it is equivalent to prove that balanced-Pandas stabilizes any arrival rate vector within Λ. For any arrival rate vector $\boldsymbol{\lambda} \in \Lambda$, since Λ is an open set, similarly there exists a $\delta > 0$ such that there exists a decomposition of $\boldsymbol{\lambda}$, $\{\lambda_{\bar{L},m}\}$ satisfying the following condition:

$$\sum_{\bar{L}:m\in\bar{L}} \frac{\lambda_{\bar{L},m}}{\alpha} + \sum_{\bar{L}:m\in\bar{L}_k} \frac{\lambda_{\bar{L},m}}{\beta} + \sum_{\bar{L}:m\in\bar{L}_r} \frac{\lambda_{\bar{L},m}}{\gamma} \leq \frac{1}{1+\delta},\ \forall m. \tag{B.2}$$

Define $\boldsymbol{\omega} = (\omega_1, \omega_2, \cdots, \omega_M)$ as

$$\omega_m = \sum_{\bar{L}:m\in\bar{L}} \frac{\lambda_{\bar{L},m}}{\alpha} + \sum_{\bar{L}:m\in\bar{L}_k} \frac{\lambda_{\bar{L},m}}{\beta} + \sum_{\bar{L}:m\in\bar{L}_r} \frac{\lambda_{\bar{L},m}}{\gamma},\ \forall m, \tag{B.3}$$

which can be regarded as the workload for server m under balanced-Pandas. Note that the dynamics of the expected workload can be described as

$$W_m(t+1) = W_m(t) + A_m(t) + S_m(t) + \tilde{U}_m(t),$$

where

$$A_m(t) = \frac{A_m^l(t)}{\alpha} + \frac{A_m^k(t)}{\beta} + \frac{A_m^r(t)}{\gamma},$$

$$S_m(t) = \frac{S_m^l(t)}{\alpha} + \frac{S_m^k(t)}{\beta} + \frac{S_m^r(t)}{\gamma},$$

$$\tilde{U}_m(t) = \frac{U_m(t)}{\gamma}.$$

With a slight abuse of notation, we use $\mathbf{A} = (A_1, A_2, \cdots, A_M)$, $\mathbf{S} = (S_1, S_2, \cdots, S_M)$ and $\tilde{\mathbf{U}} = (\tilde{U}_1, \tilde{U}_2, \cdots, \tilde{U}_M)$ throughout the proofs. Then the dynamics of $\mathbf{W}$ can be expressed as

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \mathbf{A}(t) - \mathbf{S}(t) + \tilde{\mathbf{U}}(t).$$

Proof of Theorem 4.3. Consider the Lyapunov function $V(Z(t)) = \|\mathbf{W}(t))\|^2$.

The corresponding drift is given by:

$$\begin{aligned}\Delta V(Z(t)) &= \mathbb{E}\left[V(t+1)-V(t)|Z(t)\right] \\ &= \mathbb{E}\Big[\Big(2\langle \mathbf{W}(t), \mathbf{A}(t)-\mathbf{S}(t)\rangle + 2\langle \mathbf{W}(t), \tilde{\mathbf{U}}(t)\rangle \\ &\quad + \Big\|\mathbf{A}(t)-\mathbf{S}(t)+\tilde{\mathbf{U}}(t)\Big\|^2\Big)\,|Z(t)\Big].\end{aligned}$$

The remaining steps are the same as the proof of Theorem 4.1. We need the following lemmas analogous to Lemmas B.2-B.4.

Lemma B.5. *Consider any arrival rate vector $\boldsymbol{\lambda} \in \Lambda$ and the corresponding $\boldsymbol{\omega}$ defined in (B.3). Under balanced-Pandas, for any $t \geq 0$,*

$$\mathbb{E}\left[\langle \mathbf{W}(t), \mathbf{A}(t)\rangle - \langle \mathbf{W}(t), \boldsymbol{\omega}\rangle | Z(t)\right] \leq 0. \tag{B.4}$$

Lemma B.6. *Consider any arrival rate vector $\boldsymbol{\lambda} \in \Lambda$ and the corresponding $\boldsymbol{\omega}$ defined in (B.3). Under balanced-Pandas, for any $t \geq 0$,*

$$\mathbb{E}\left[\langle \mathbf{W}(t), \boldsymbol{\omega}\rangle - \langle \mathbf{W}(t), \mathbf{S}(t)\rangle | Z(t)\right] \leq -\theta_2 \|\mathbf{Q}(t)\|_1, \tag{B.5}$$

where $\theta_2 > 0$ is constant independent of $Z(t)$.

Lemma B.7. *For any t, $\langle \mathbf{W}(t), \tilde{\mathbf{U}}(t)\rangle = 0$.*

Therefore, we can bound the drift of $V(Z(t))$ as

$$\Delta V(Z(t)) \leq -2\theta_2 \|\mathbf{Q}(t)\|_1 + C,$$

where θ_2 and C are positive constants independent of $Z(t)$.

Pick any $\epsilon > 0$. Let $\mathcal{P} = \left\{Z = (\mathbf{Q}, \mathbf{f}) |\ \|\mathbf{Q}\|_1 \leq \frac{\epsilon}{2\theta_2}\right\}$. Then $\mathcal{P}$ is a finite subset of state space. For any $Z \in \mathcal{P}^c$, $\Delta F(Z) \leq -\epsilon$. Therefore the Markov process $\{Z(t), t \geq 0\}$ is positive recurrent. Therefore balanced-Pandas stabilizes the system for any $\boldsymbol{\lambda} \in \Lambda$, i.e., balanced-Pandas is throughput optimal.

■

B.3.1 Proof of Lemma B.5

$$\begin{aligned}\langle \mathbf{W}(t), \mathbf{A}(t)\rangle &= \sum_{\bar{L}\in\mathcal{L}}\Bigg(\sum_{m:m\in\bar{L}}\frac{W_m(t)}{\alpha}A_{\bar{L},m}(t)+\\ &\qquad \sum_{m:m\in\bar{L}_k}\frac{W_m(t)}{\beta}A_{\bar{L},m}(t)+\sum_{m:m\in\bar{L}_r}\frac{W_m(t)}{\gamma}A_{\bar{L},m}(t)\Bigg).\end{aligned}$$

For any task type $\bar{L}\in\mathcal{L}$, define

$$W^*_{\bar{L}}(t)=\min_{m\in\mathcal{M}}\left\{\frac{W_m(t)}{\alpha}I_{\{m\in\bar{L}\}},\frac{W_m(t)}{\beta}I_{\{m\in\bar{L}_k\}},\frac{W_m(t)}{\gamma}I_{\{m\in\bar{L}_r\}}\right\}.$$

Note that for any task of type $\bar{L}\in\mathcal{L}_{\mathcal{H}}$, it will be routed to queue m^* with expected workload $W^*_{\bar{L}}(t)$ at the beginning of time slot t. That is, type-$\bar{L}$ tasks will not join any server m with $W_m(t)>W^*_{\bar{L}}(t)$. Thus

$$\mathbb{E}\left[\langle \mathbf{W}(t),\mathbf{A}(t)\rangle|Z(t)\right]=\mathbb{E}\left[\sum_{\bar{L}\in\mathcal{L}}W^*_{\bar{L}}(t)A_{\bar{L}}(t)|Z(t)\right]=\sum_{\bar{L}\in\mathcal{L}}W^*_{\bar{L}}(t)\lambda_{\bar{L}}.$$

On the other hand,

$$\begin{aligned}&\mathbb{E}\left[\langle \mathbf{W}(t),\boldsymbol{\omega}\rangle|Z(t)\right]\\ =\ &\sum_{\bar{L}\in\mathcal{L}}\left(\sum_{m:m\in\bar{L}}\frac{W_m(t)}{\alpha}\lambda_{\bar{L},m}+\sum_{m:m\in\bar{L}_k}\frac{W_m(t)}{\beta}\lambda_{\bar{L},m}+\sum_{m:m\in\bar{L}_r}\frac{W_m(t)}{\gamma}\lambda_{\bar{L},m}\right).\end{aligned}$$

Note that for any task of type $\bar{L}\in\mathcal{L}$, $\frac{W_m(t)}{\alpha}\geq W^*_{\bar{L}}(t)$ for any $m\in\bar{L}$, $\frac{W_m(t)}{\beta}\geq W^*_{\bar{L}}(t)$ for any $m\in\bar{L}_k$, and $\frac{W_m(t)}{\gamma}\geq W^*_{\bar{L}}(t)$ for any $m\in\bar{L}_r$. Therefore

$$\mathbb{E}\left[\langle \mathbf{W}(t),\boldsymbol{\omega}\rangle|Z(t)\right]\geq\sum_{\bar{L}\in\mathcal{L}}\left(W^*_{\bar{L}}(t)\sum_m\lambda_{\bar{L},m}\right)=\sum_{\bar{L}\in\mathcal{L}}W^*_{\bar{L}}(t)\lambda_{\bar{L}}.$$

Consequently, we have

$$\mathbb{E}\left[\langle \mathbf{W}(t),\boldsymbol{\omega}\rangle|Z(t)\right]\geq\mathbb{E}\left[\langle \mathbf{W}(t),\mathbf{A}\rangle|Z(t)\right].$$

■

B.3.2 Proof of Lemma B.6

From Eq. (B.2)-(B.3), we have $\omega_m \leq \frac{1}{1+\delta},\ \forall m$. Thus

$$\mathbb{E}\left[\langle \mathbf{W}(t), \boldsymbol{\omega}\rangle | Z(t)\right] = \sum_m W_m(t)\omega_m \leq \frac{1}{1+\delta}\sum_m W_m(t).$$

On the other hand,

$$\begin{aligned}
& \mathbb{E}\left[\langle \mathbf{W}(t), \mathbf{S}(t)\rangle | Z(t)\right] \\
= & \sum_m W_m(t) \sum_{i=0}^{2} \mathbb{E}\left[\mathbb{E}\left[\frac{S_m^l(t)}{\alpha} + \frac{S_m^k(t)}{\beta} + \frac{S_m^r(t)}{\gamma} | Z(t), \eta_m(t) = i\right] | Z(t)\right] \\
= & \sum_m W_m(t).
\end{aligned}$$

We now are ready to prove (B.5):

$$\mathbb{E}\left[\langle \mathbf{W}(t), \boldsymbol{\omega}\rangle - \langle \mathbf{W}(t), \mathbf{S}(t)\rangle | Z(t)\right] \leq -\frac{\delta}{1+\delta}\sum_m W_m(t) \leq -\theta_2 \left\|\bar{\mathbf{Q}}(t)\right\|_1,$$

where $\theta_2 = \frac{\delta}{(1+\delta)\alpha}$. ∎

B.3.3 Proof of Lemma B.7

$$\langle \mathbf{W}(t), \tilde{\mathbf{U}}(t)\rangle = \sum_m \left(\frac{Q_m^l(t)}{\alpha} + \frac{Q_m^k(t)}{\beta} + \frac{Q_m^r(t)}{\gamma}\right)\tilde{U}_m(t).$$

For any m, by the definition of $\tilde{U}_m(t)$, $\tilde{U}_m(t) > 0$ implies that server m is idle, i.e., all of its three sub-queues are empty. That is, $W_m(t) = 0$. Therefore $W_m(t)\tilde{U}U_m(t) = 0$ for all m. ∎

B.4 Heavy-traffic Optimality without Overloaded Racks

For the case without overloaded racks, we consider two traffic scenarios: the set of overloaded servers in underloaded racks $\mathcal{B}_u = \emptyset$ and $\mathcal{B}_u \neq \emptyset$, which correspond to the *evenly loaded* and *locally overloaded* scenario for the system

with two levels of locality respectively.

We first establish the heavy-traffic optimality for the *evenly loaded* case where $\mathcal{B}_u = \emptyset$ in subsection B.4.1 and then show the proof for the *locally overloaded* case where $\mathcal{B}_u \neq \emptyset$ in subsection B.4.2.

B.4.1 Evenly loaded

The proof follows exactly the same three steps for the evenly loaded traffic scenario in a system with two levels of locality in Chapter 3. We will skip the proof details and just present the main steps and results here.

We consider the heavy-traffic regime where the limiting arrival rate vector satisfies the *resource pooling* condition introduced in Chapter 3. We use the same notation $\mathcal{F}$ to denote the set of arrival rate vector on the boundary of the capacity region Λ, such that all servers are fully utilized to handle its local load. That is, all servers in the system are helpers in underloaded racks.

Assumption 4 (Assumption for the heavy evenly loaded traffic). Consider the arrival processes $\{A_{\bar{L}}^{(\epsilon)}(t), t \geq 0\}_{\bar{L}\in\mathcal{L}}$, parameterized by $\epsilon > 0$, with mean arrival rate vector $\boldsymbol{\lambda}^{(\epsilon)} = (1-\epsilon_0)\bar{\boldsymbol{\lambda}}$, where $\epsilon_0 = \frac{\epsilon}{M\alpha}$, and $\bar{\boldsymbol{\lambda}} \in \mathcal{F}$ satisfies the resource pooling condition. The variance of the number of arrivals, $\mathrm{Var}(\sum_{\bar{L}\in\mathcal{L}} A_{\bar{L}}^{(\epsilon)}(t))$, is denoted as $(\sigma_e^{(\epsilon)})^2$, which converges to σ_e^2 as $\epsilon \downarrow 0$.

We have shown that the corresponding Markov chain $\{Z^{(\epsilon)}(t) = (\mathbf{Q}^{(\epsilon)}(t), \mathbf{f}^{(\epsilon)}(t))\}$ under balanced-Pandas is positive recurrent. The queue-length vector process $\mathbf{Q}^{(\epsilon)}(t)$ hence converges in distribution to a random vector $\bar{\mathbf{Q}}^{(\epsilon)}$ for any $0 < \epsilon < \bar{\epsilon}$, where $\bar{\epsilon}$ is a positive constant. All theorems in this section concern the *steady-state* queueing process $\bar{\mathbf{Q}}^{(\epsilon)}$.

Lower Bound

Consider a single server system with arrival process $\{a^{(\epsilon)}(t), t \geq 0\}$ and service process $\{b_e^{(\epsilon)}(t), t \geq 0\}$, where

$$a^{(\epsilon)}(t) = \sum_{\bar{L}\in\mathcal{L}} A_{\bar{L}}^{(\epsilon)}(t),\ b_e^{(\epsilon)}(t) = \sum_{m=1}^{M} X_m(t).$$

Here $\{X_m(t), t \geq 0\}_{m\in\mathcal{M}}$ are independent, and each process is temporally i.i.d. with $X_m(t) \sim Bern(\alpha)$. Note that the mean of $a^{(\epsilon)}(t)$ is $M\alpha - \epsilon$ and the variance is given by $(\sigma_e^{(\epsilon)})^2$. Then the corresponding queue-length process is stochastically smaller than $\sum_{m=1}^{M}\left(Q_m^{l(\epsilon)}(t) + Q_m^{k(\epsilon)}(t) + Q_m^{r(\epsilon)}(t)\right)$. Hence

$$\mathbb{E}\left[\sum_{m=1}^{M}\left(\bar{Q}_m^{l(\epsilon)} + \bar{Q}_m^{k(\epsilon)} + \bar{Q}_m^{r(\epsilon)}\right)\right] \geq \frac{(\sigma_e^{(\epsilon)})^2 + \nu_e^2 + \epsilon^2}{2\epsilon} - \frac{M}{2},$$

where ν_e^2 is the variance for $\{b_e^{(\epsilon)}(t)\}$. Therefore, in the heavy traffic limit, we have

$$\liminf_{\epsilon\to 0^+} \epsilon\mathbb{E}\left[\sum_{m=1}^{M}\left(\bar{Q}_m^{l(\epsilon)} + \bar{Q}_m^{k(\epsilon)}) + \bar{Q}_m^{r(\epsilon)}\right)\right] \geq \frac{\sigma_e^2 + \nu_e^2}{2}. \tag{B.6}$$

State Space Collapse

We will show that the expected workload $\mathbf{W}$ collapses to the direction $\mathbf{c_e}$, where

$$\mathbf{c_e} = \frac{1}{\sqrt{M}}(\underbrace{1, 1, \cdots, 1}_{M}).$$

Let $\mathbf{W}_{||}$ and $\mathbf{W}_{\perp}$ be the components of $\mathbf{W}$ parallel and perpendicular to the direction $\mathbf{c_e}$. We will establish state space collapse by showing that $\mathbf{W}_{\perp}$ is bounded and independent of the heavy-traffic parameter ϵ. That is, there exists a sequence of finite numbers $\{C_r : r \in \mathcal{N}\}$ such that for each positive integer r,

$$\mathbb{E}\left[\|\mathbf{W}_{\perp}\|^r\right] \leq C_r,$$

where $\mathbf{W}_{\perp}$ is the component of $\mathbf{W}$ perpendicular to $\mathbf{c_e}$.

To establish state space collapse, we consider the Lyapunov functions $F(Z) = \|\mathbf{W}_{\perp}\|$. By Lemma 3.6, it is sufficient to show that the drift of $F(Z)$, denoted by $\Delta F(Z)$, satisfies two conditions: (i) $\Delta F(Z)$ is finite with probability 1; (ii) $\Delta F(Z)$ is negative for sufficiently large $F(Z)$. The analysis is identical to that for the two-level locality system, with queue-length vector $\mathbf{Q}$ replaced by the workload vector $\mathbf{W}$. We put the details here for completeness.

We first need the following lemma for the ideal load decomposition, which is analogous to Lemma 3.24 for the evenly loaded traffic scenario in Chapter

3.

Lemma B.8. *Consider any arrival rate vector* $\boldsymbol{\lambda} = (1-\epsilon_0)\bar{\boldsymbol{\lambda}}$, *where* $\epsilon_0 = \frac{\epsilon}{M\alpha}$, *and* $\bar{\boldsymbol{\lambda}} \in \mathcal{F}$ *satisfies the resource pooling condition. Consider any* $0 < \epsilon < \bar{\epsilon}$, *where* $\bar{\epsilon}$ *is a positive constant. Then there exists an ideal decomposition* $\left\{\lambda^*_{\bar{L},n,m}\right\}$ *of* λ *satisfying the following conditions:*

1. $\forall m \in \mathcal{M}$,
$$\sum_{\bar{L}:m\in\bar{L}} \lambda^*_{\bar{L},m,m} = \alpha(1-\epsilon_0).$$
2. *There exists a positive constant* λ_{min} *not depending on* ϵ, *such that for any two servers* m *and* m' *that are connected directly, there exists a task type* $\bar{L} \in \mathcal{L}$, *such that* $\lambda^*_{\bar{L},m,m} \geq \lambda_{min}$, $\lambda^*_{\bar{L},m',m'} \geq \lambda_{min}$.

We need the following additional lemmas analogues to Lemmas 3.25-3.26, with queue-length vector $\mathbf{Q}$ replaced by the workload vector $\mathbf{W}$.

Lemma B.9. *Let* $\mathbf{c}$ *be a vector with unit norm in* $\mathbb{R}^M$. *Then for any* $t \geq 0$,

$$\left\|\mathbf{W}_{||}(t+1)\right\|^2 - \left\|\mathbf{W}_{||}(t)\right\|^2 \geq 2\langle \mathbf{c}, \mathbf{W}(t)\rangle\langle \mathbf{c}, \mathbf{A}(t) - \mathbf{S}(t)\rangle,$$

where $\mathbf{W}_{||}$ *is the parallel component of the workload* $\mathbf{W}$ *with respect to the direction* $\mathbf{c}$.

Lemma B.10. *Let* $\mathbf{c}$ *be a vector with unit norm in* $\mathbb{R}^M$. *Then for any* $t \geq 0$,

$$\|\mathbf{W}_{\perp}(t+1)\| - \|\mathbf{W}_{\perp}(t)\| \leq \frac{\sqrt{M}}{\gamma}\max\{1, C_A\}, \tag{B.7}$$

where $\mathbf{W}_{\perp}$ *is the perpendicular component of the workload* $\mathbf{W}$ *with respect to the direction* $\mathbf{c}$.

From lemma B.10, we can see that $\Delta F(Z)$ satisfies finite condition, since $Pr(\Delta F(Z) \leq C) = 1$ with $C = \sqrt{M}\max\{M, C_A\}$.

Next we focus on the negative drift condition. Consider the following Lyapunov functions:

$$V(Z) = \|\mathbf{W}\|^2, V_{||}(Z) = \left\|\mathbf{W}_{||}\right\|^2.$$

The rest of the proof follows the same line of reasoning as in the proof of Theorem 3.5 to bound $\Delta F(Z)$:

$$\begin{aligned} & \mathbb{E}\left[\Delta F(Z(t))|Z(t)\right] \\ \leq \; & \frac{1}{2\|\mathbf{W}_\perp\|}\left(2\mathbb{E}\left[\langle \mathbf{W}(t), \mathbf{A}(t)-\mathbf{S}(t)\rangle - \langle \mathbf{c_e}, \mathbf{W}(t)\rangle\langle \mathbf{c_e}, \mathbf{A}(t)-\mathbf{S}(t)\rangle | Z(t)\right] + C_1\right). \end{aligned}$$

To obtain the a bound on the expectation term on the right-hand side of the above inequality, we need the following lemmas analogous to Lemmas 3.28-3.30.

Lemma B.11. *Under balanced-Pandas, for any $t \geq 0$,*

$$\mathbb{E}\left[\langle \mathbf{W}(t), \mathbf{A}(t)\rangle - \langle \mathbf{W}(t), \boldsymbol{\omega}\rangle | Z(t)\right] \leq -\lambda_{min}\|\mathbf{W}_\perp(t)\|.$$

Lemma B.12. *Under balanced-Pandas, for any $t \geq 0$,*

$$\mathbb{E}\left[\langle \mathbf{W}(t), \boldsymbol{\omega}\rangle - \langle \mathbf{W}(t), \mathbf{S}(t)\rangle | Z(t)\right] = -\frac{\epsilon}{M\alpha}\sum_m W_m(t).$$

Lemma B.13. *Under balanced-Pandas, for any $t \geq 0$,*

$$\mathbb{E}\left[\langle \mathbf{c_e}, \mathbf{W}(t)\rangle\langle \mathbf{c_e}, \mathbf{A}(t)-\mathbf{S}(t)\rangle | Z(t)\right] \geq -\frac{\epsilon}{M\alpha}\sum_m W_m(t).$$

Utilizing the above three inequalities yields:

$$\mathbb{E}\left[\Delta F(Z(t))|Z(t)\right] \leq -\lambda_0 + \frac{C_1}{\|\mathbf{W}_\perp(t)\|},$$

where λ_0 and C_1 are positive constants independent of ϵ. This inequality verifies the negative drift condition, and hence establishes the existence of finite constants $\{C_r\}_{r\in\mathbb{N}}$ for which $\mathbb{E}\left[\left\|\mathbf{W}_\perp^{(\epsilon)}(t)\right\|^r\right] \leq C_r$, for all $\epsilon \in (0, M\alpha)$. ∎

Proof of Lemma B.11. is similar to that of Lemma B.5

Proof of Lemma B.12. is similar to that of Lemma B.6. We omit details here.

Proof of Lemma B.13. Note that

$$\langle \mathbf{c_e}, \mathbf{A}(t)\rangle \;\; \geq \;\; \frac{1}{\sqrt{M}\alpha}\sum_m \left(A_m^l(t) + A_m^k(t) + A_m^r(t)\right) = \frac{1}{\sqrt{M}\alpha}\sum_{\bar{L}\in\mathcal{L}} A_{\bar{L}}(t).$$

From the proof of Lemma B.6, we have

$$\mathbb{E}\left[\langle \mathbf{c_e}, \mathbf{S}(t)\rangle | Z(t)\right] = \frac{1}{\sqrt{M}} \sum_m 1 = \sqrt{M}.$$

Consequently,

$$\begin{aligned} & \mathbb{E}\left[\langle \mathbf{c_e}, \mathbf{W}(t)\rangle \langle \mathbf{c_e}, \mathbf{A}(t) - \mathbf{S}(t)\rangle | Z(t)\right] \\ = \; & \langle \mathbf{c_e}, \mathbf{W}(t)\rangle \mathbb{E}\left[\langle \mathbf{c_e}, \mathbf{A}(t) - \mathbf{S}(t)\rangle | Z(t)\right] \\ \geq \; & -\frac{\epsilon}{M\alpha} \sum_m W_m(t). \end{aligned}$$

■

Upper Bound

Again we construct a series of ideal arrival and service processes, which allows us to rewrite the dynamics of $\mathbf{W}$, and bound the terms using Lemma 8 in [59].

Ideal scheduling decision process $\hat{\boldsymbol{\eta}}(t)$: $\forall m \in \mathcal{M}$, $\hat{\eta}_m(t) = 0$. That is, each server is scheduled to serve its local sub-queue only under the *ideal scheduling*.

Ideal service process $\hat{\mathbf{S}}(t) : \forall m \in \mathcal{M}$

$$\hat{S}_m^l(t) = X_m^l(t),\ \hat{S}_m^k(t) = 0,\ \hat{S}_m^r(t) = 0,$$

where each process $X_m^l(t)$ is coupled with $\mathbf{S}_m(t)$ in the following way: If $\eta_m(t) = 0$, $X_m^l(t) = S_m^l(t)$; if $\eta_m(t) = 1$, $X_m^l(t) = 1$ when $S_m^k(t) = 1$, and $X_m^l(t) \sim \text{Bern}(\frac{\alpha-\beta}{1-\beta})$ when $S_m^k(t) = 0$; if $\eta_m(t) = 2$, $X_m^l(t) = 1$ when $S_m^r(t) = 1$, and $X_m^l(t) \sim \text{Bern}(\frac{\alpha-\gamma}{1-\gamma})$ when $S_m^k(t) = 0$. Hence each process $X_m^l(t)$ is i.i.d. with $X_m^l(t) \sim \text{Bern}(\alpha)$.

Ideal arrival process $\hat{\mathbf{A}}(t)$: Ideally, $\forall \bar{L} \in \mathcal{L}$, type-$\bar{L}$ tasks would join their local sub-queues. So we re-distribute unwanted arrivals $\sum_{m:m\notin\bar{L}} A_{\bar{L},m}$ among its local servers evenly.

Then we can rewrite the dynamics of $\mathbf{W}$ as

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \hat{\mathbf{A}}(t) - \hat{\mathbf{S}}(t) + \hat{\mathbf{U}}(t),$$

where $\hat{\mathbf{U}}(t) = \hat{\mathbf{S}}(t) - \hat{\mathbf{A}}(t) + \mathbf{A}(t) - \mathbf{S}(t) + \tilde{\mathbf{U}}(t)$. We consider the Lyapunov function $G_{||}(\mathbf{Z}) = \left\|\mathbf{W}_{||}\right\|^2$, where $\mathbf{W}_{||}$ is the parallel component of the vector $\mathbf{W}$ with respect to the direction $\mathbf{c_e}$. As shown in Lemma 8 [59] the drift of $G_{||}(\mathbf{Z})$ is zero in steady state, which yields

$$\begin{aligned} & 2\mathbb{E}\left[\langle \mathbf{c_e}, \mathbf{W}(t)\rangle \langle \mathbf{c_e}, \hat{\mathbf{S}}(t) - \hat{\mathbf{A}}(t)\rangle\right] \\ = \; & \mathbb{E}\left[\langle \mathbf{c_e}, \hat{\mathbf{A}}(t) - \hat{\mathbf{S}}(t)\rangle^2\right] + \mathbb{E}\left[\langle \mathbf{c_e}, \hat{\mathbf{U}}(t)\rangle^2\right] \qquad \text{(B.8)} \\ & +2\mathbb{E}\left[\langle \mathbf{c_e}, \mathbf{W}(t) + \hat{\mathbf{A}}(t) - \hat{\mathbf{S}}(t)\rangle \langle \mathbf{c_e}, \hat{\mathbf{U}}(t)\rangle\right]. \qquad \text{(B.9)} \end{aligned}$$

An upper bound on $\mathbb{E}\left[\langle \mathbf{c_e}, \mathbf{W}(t)\rangle\right]$ can be obtained by bounding each of the above terms, which gives an upper bound on $\mathbb{E}\left[\sum_{m=1}^{M}\left(Q_m^{l(\epsilon)}(t) + Q_m^{k(\epsilon)}(t) + Q_m^{r(\epsilon)}(t)\right)\right]$.

$$\begin{aligned} & \mathbb{E}\left[\langle \mathbf{c_e}, \mathbf{W}(t)\rangle \langle \mathbf{c_e}, \hat{\mathbf{S}}(t) - \hat{\mathbf{A}}(t)\rangle\right] \\ = \; & \frac{1}{M}\mathbb{E}\left[\left(\sum_m W_m(t)\right)\left(\sum_m \frac{\hat{S}_m^l(t)}{\alpha} - \sum_m \frac{\hat{A}_m^l(t)}{\alpha}\right)\right] \\ = \; & \frac{\epsilon}{M\alpha}\mathbb{E}\left[\left(\sum_m W_m(t)\right)\right]. \end{aligned}$$

By the definition of ideal service and arrival processes, we have

$$\mathbb{E}\left[\langle \mathbf{c_e}, \hat{\mathbf{A}}(t) - \hat{\mathbf{S}}(t)\rangle^2\right] = \frac{(\sigma_e^{(\epsilon)})^2 + \nu^2 + \epsilon}{M\alpha^2}.$$

For the term $\mathbb{E}\left[\langle \mathbf{c_e}, \hat{\mathbf{U}}(t)\rangle^2\right]$, we have the following lemma.

Lemma B.14.

$$\mathbb{E}\left[\langle \mathbf{c_e}, \hat{\mathbf{U}}(t)\rangle^2\right] \leq C'\epsilon,$$

where C' is a constant not depending on ϵ.

Next we will bound the term (B.9). We consider the system in steady state, which yields $\mathbb{E}\left[\langle \mathbf{c_e}, \hat{\mathbf{A}}(t) - \hat{\mathbf{S}}(t) + \hat{\mathbf{U}}(t)\rangle\right] = \mathbb{E}\left[\langle \mathbf{c_e}, \mathbf{W}(t+1) - \mathbf{W}(t)\rangle\right] = 0$. Hence

$$\mathbb{E}\left[\langle \mathbf{c_e}, \hat{\mathbf{U}}(t)\rangle\right] = \mathbb{E}\left[\langle \mathbf{c_e}, \hat{\mathbf{A}}(t) - \hat{\mathbf{S}}(t)\rangle\right] = \frac{\epsilon}{M\alpha}.$$

Note that

$$\begin{aligned} & \mathbb{E}\left[\langle \mathbf{c_e}, \hat{\mathbf{A}}(t) - \hat{\mathbf{S}}(t)\rangle \langle \mathbf{c_e}, \hat{\mathbf{U}}(t)\rangle\right] \leq \mathbb{E}\left[\langle \mathbf{c_e}, \hat{\mathbf{A}}(t)\rangle \langle \mathbf{c_e}, \hat{\mathbf{U}}(t)\rangle\right] \\ \leq & \frac{C_A}{\sqrt{M}\alpha}\mathbb{E}\left[\langle \mathbf{c_e}, \hat{\mathbf{U}}(t)\rangle\right] = \frac{C_A}{M\sqrt{M}\alpha^2}\epsilon. \end{aligned}$$

Then we have

$$\begin{aligned} & \mathbb{E}\left[\langle \mathbf{c_e}, \mathbf{W}(t) + \hat{\mathbf{A}}(t) - \hat{\mathbf{S}}(t)\rangle \langle \mathbf{c_e}, \hat{\mathbf{U}}(t)\rangle\right] \\ = & \mathbb{E}\left[\langle \mathbf{c_e}, \mathbf{W}(t)\rangle \langle \mathbf{c_e}, \hat{\mathbf{U}}(t)\rangle\right] + \mathbb{E}\left[\langle \mathbf{c_e}, \hat{\mathbf{A}}(t) - \hat{\mathbf{S}}(t)\rangle \langle \mathbf{c_e}, \hat{\mathbf{U}}(t)\rangle\right] \\ \leq & \mathbb{E}\left[\langle \mathbf{c_e}, \mathbf{W}(t)\rangle \langle \mathbf{c_e}, \hat{\mathbf{U}}(t)\rangle\right] + \frac{C_A}{M\sqrt{M}\alpha^2}\epsilon. \end{aligned}$$

We can rewrite the term $\langle \mathbf{c_e}, \mathbf{W}(t)\rangle \langle \mathbf{c_e}, \hat{\mathbf{U}}(t)\rangle$ as

$$\begin{aligned} & \langle \mathbf{c_e}, \mathbf{W}(t)\rangle \langle \mathbf{c_e}, \hat{\mathbf{U}}(t)\rangle \\ = & \langle \mathbf{W}(t), \hat{\mathbf{U}}(t)\rangle - \langle \mathbf{W}_\perp(t), \hat{\mathbf{U}}_\perp(t)\rangle \\ = & \langle \mathbf{W}(t), \hat{\mathbf{S}}(t) - \mathbf{S}(t)\rangle + \langle \mathbf{W}(t), \mathbf{A}(t) - \hat{\mathbf{A}}(t)\rangle \qquad \text{(B.10)} \\ & + \langle \mathbf{W}(t), \tilde{\mathbf{U}}(t)\rangle - \langle \mathbf{W}_\perp(t), \hat{\mathbf{U}}_\perp(t)\rangle. \qquad \text{(B.11)} \end{aligned}$$

The following two lemmas bound the two terms in (B.10).

Lemma B.15.

$$\mathbb{E}\left[\langle \mathbf{W}(t), \hat{\mathbf{S}}(t) - \mathbf{S}(t)\rangle\right] = 0.$$

Lemma B.16.

$$\mathbb{E}\left[\langle \mathbf{W}(t), \mathbf{A}(t) - \hat{\mathbf{A}}(t)\rangle\right] \leq 0.$$

The first term in (B.11) is equal to zero by Lemma B.7. To bound the second term in (B.11), we first show that

$$\mathbb{E}\left[\left\|\hat{\mathbf{U}}(t)\right\|^2\right] \leq R\epsilon,$$

where R is a constant independent of ϵ. Next we will use state space collapse

result to bound $-\langle \mathbf{W}_\perp(t), \hat{\mathbf{U}}_\perp(t)\rangle$. By Cauchy-Schwarz inequality, we have

$$\begin{aligned}\mathbb{E}\left[-\langle \mathbf{W}_\perp(t), \hat{\mathbf{U}}_\perp(t)\rangle\right] &\leq \sqrt{\mathbb{E}\left[\left\|\hat{\mathbf{W}}_\perp(t)\right\|^2\right]\mathbb{E}\left[\left\|\hat{\mathbf{U}}_\perp(t)\right\|^2\right]} \\ &\leq \sqrt{C_2 R\epsilon}.\end{aligned}$$

Combining these inequalities gives the bound on the term (B.9)

$$\mathbb{E}\left[\langle \mathbf{c_e}, \mathbf{W}(t) + \hat{\mathbf{A}}(t) - \hat{\mathbf{S}}(t)\rangle\langle \mathbf{c_e}, \hat{\mathbf{U}}(t)\rangle\right] \leq \frac{C_A}{M\sqrt{M}\alpha^2}\epsilon + \sqrt{C_2 R\epsilon}.$$

Now we are ready to revive the superscript $^{(\epsilon)}$. From the above analysis, we have

$$2\frac{\epsilon}{M\alpha}\mathbb{E}\left[\sum_m W_m^{(\epsilon)}(t)\right] \leq \frac{(\sigma_e^{(\epsilon)})^2 + \nu^2 + \epsilon}{M\alpha^2} + C'\epsilon + 2\frac{C_A}{M\sqrt{M}\alpha^2}\epsilon + 2\sqrt{C_2 R\epsilon},$$

i.e.,

$$\alpha\mathbb{E}\left[\sum_m W_m^{(\epsilon)}(t)\right] \leq \frac{(\sigma_e^{(\epsilon)})^2 + \nu^2}{2\epsilon} + D_e^{(\epsilon)},$$

where $D_e^{(\epsilon)} = \frac{\epsilon}{2} + \frac{C'M\alpha^2}{2} + \frac{C_A}{\sqrt{M}} + M\alpha^2\sqrt{\frac{C_2 R}{\epsilon}}$. On the other hand,

$$\mathbb{E}\left[\sum_{m=1}^M \left(Q_m^{l(\epsilon)}(t) + Q_m^{k(\epsilon)}(t) + Q_m^{r(\epsilon)}(t)\right)\right] \leq \alpha\mathbb{E}\left[\sum_m W_m^{(\epsilon)}(t)\right].$$

Thus

$$\mathbb{E}\left[\sum_{m=1}^M \left(Q_m^{l(\epsilon)}(t) + Q_m^{k(\epsilon)}(t) + Q_m^{r(\epsilon)}(t)\right)\right] \leq \frac{(\sigma_e^{(\epsilon)})^2 + \nu^2}{2\epsilon} + D_e^{(\epsilon)}.$$

Observe that $D_e^{(\epsilon)} = o(\frac{1}{\epsilon})$, i.e., $\lim_{\epsilon\downarrow 0}\epsilon D_e^{(\epsilon)} = 0$. Therefore, in the heavy-traffic limit, we obtain the following upper bound:

$$\limsup_{\epsilon\downarrow 0}\epsilon\mathbb{E}\left[\sum_{m=1}^M \left(Q_m^{l(\epsilon)}(t) + Q_m^{k(\epsilon)}(t) + Q_m^{r(\epsilon)}(t)\right)\right] \leq \frac{(\sigma_e)^2 + \nu^2}{2\epsilon},$$

which coincides with the lower bound in (B.6).

B.4.2 Locally overloaded

Consider the heavy traffic regime where $\mathcal{O} = \emptyset$ and $\mathcal{B}_u \neq \emptyset$. The system can be separated into two subsystems: racks with only $\mathcal{H}_u$ servers, denoted by $\mathcal{P}$, and racks mixed with servers of $\mathcal{H}_u$ and $\mathcal{B}_u$, denoted by $\mathcal{P}^c$. In the heavy-traffic regime, the behavior of subsystem $\mathcal{P}$ is exactly the same as a system with evenly loaded traffic. Here we focus on the subsystem $\mathcal{P}^c$. For simplicity, assume that $\mathcal{P} = \emptyset$. Let the local traffic for $\mathcal{H}_u$ be

$$\sum_{\bar{L} \in \mathcal{L}^*_{\mathcal{H}}} \lambda_{\bar{L}} \equiv \Phi\alpha. \tag{B.12}$$

We define the heavy-traffic regime to be

$$\sum_{\bar{L} \in \mathcal{L}_{\mathcal{B}_u}} \lambda_{\bar{L}} = |\mathcal{B}_u|\alpha + \beta(|\mathcal{H}_u| - \Phi) - \epsilon, \tag{B.13}$$

where $\epsilon > 0$ characterizes the distance of the arrival rate vector from the capacity boundary. We will make a further assumption that the $\{\lambda_{\bar{L}} : \bar{L} \in \mathcal{L}^*_{\mathcal{H}_u}\}$ are independent of ϵ.

Assumption 5 Consider the arrival processes $\{A^{(\epsilon)}_{\bar{L}}(t)\}_{\bar{L} \in \mathcal{L}}$ with arrival rate vector $\boldsymbol{\lambda}^{(\epsilon)}$ satisfying the above conditions. Note that the variance of $\{A^{(\epsilon)}_{\bar{L}}(t)\}_{\bar{L} \in \mathcal{L}^*_{\mathcal{H}_u}}$ is independent of ϵ. We denote by $(\sigma_l^{(\epsilon)})^2$ the variance of the number of arrivals that are only local to beneficiaries in overloaded racks, i.e., $\mathrm{Var}\left(\sum_{\bar{L} \in \mathcal{L}_{\mathcal{B}_u}} A^{(\epsilon)}_{\bar{L}}(t)\right) = (\sigma_l^{(\epsilon)})^2$, which converges to σ_l^2 as $\epsilon \downarrow 0$.

Again we follow the three-step framework to establish heavy-traffic optimality for the case $\mathcal{B}_u \neq \emptyset$. All results in this subsection concern the steady-state queue-length vector $\bar{\mathbf{Q}}$.

Helper queues

Similar to the case of $\mathcal{O} \neq \emptyset$, we first need to show that the helper subsystem is uniformly bounded and independent of ϵ. That is, there exist two sequences of finite numbers $\{N_r : r \in \mathbb{N}\}$ such that for each positive integer r,

$$\mathbb{E}\left[\sum_{m \in \mathcal{H}_u} \bar{Q}_m^{l(\epsilon)}\right] \leq N_r.$$

Thus,

$$\lim_{\epsilon \downarrow 0} \epsilon \mathbb{E}\left[\sum_{m \in \mathcal{H}_u} \bar{Q}_m^{l(\epsilon)}\right] = 0.$$

Therefore, we only need to consider

$$\Phi_l(t) = \sum_{m \in \mathcal{B}_u} \left(Q_m^{l(\epsilon)}(t) + Q_m^{k(\epsilon)}(t) + Q_m^{r(\epsilon)}(t)\right) + \sum_{m \in \mathcal{H}_u} \left(Q_m^{k(\epsilon)}(t) + Q_m^{r(\epsilon)}(t)\right).$$

Lower Bound

Consider a single server system with an arrival process $\{a_l^{(\epsilon)}(t), t \geq 0\}$ and service process $\{b_l^{(\epsilon)}(t), t \geq 0\}$, where

$$a_l^{(\epsilon)}(t) = \sum_{\bar{L} \in \mathcal{L}_{\mathcal{B}_u}} A_{\bar{L}}^{(\epsilon)}(t),\ b_l^{(\epsilon)}(t) = \sum_{i \in \mathcal{B}_u} X_i(t) + \sum_{j \in \mathcal{H}_u} Y_j(t).$$

Here $\{X_i(t)\}_{i \in \mathcal{B}_u}$, $\{Y_j(t)\}_{j \in \mathcal{H}_u}$ are independent and each process is i.i.d. For all $i \in \mathcal{B}_u$, $X_i(t) \sim \text{Bern}(\alpha)$. For all $j \in \mathcal{H}_u$, $Y_j(t) \sim \text{Bern}(\beta(1-\rho_j^l))$, where ρ_j^l is the proportion of time helper j spends on local tasks in steady state. The definition of X_i and Y_j is such that $\mathbb{E}\left[\sum_{i \in \mathcal{B}_u} X_i(t)\right]$ and $\mathbb{E}\left[\sum_{j \in \mathcal{H}_u} Y_j(t)\right]$ are the maximum amount of local and rack-local services that can be provided for $\sum_{\bar{L} \in \mathcal{L}_{\mathcal{B}_u}} A_{\bar{L}}^{(\epsilon)}(t)$. We denote $\text{Var}(b_l^{(\epsilon)}(t))$ by $(\nu_l^{(\epsilon)})^2$, which converges to a constant ν_l^2 as $\epsilon \downarrow 0$.

Then in steady state the corresponding queue-length process is stochastically smaller than $\Phi_l^{(\epsilon)}(t)$. Hence

$$\mathbb{E}\left[\Phi_l^{(\epsilon)}\right] \geq \frac{(\sigma_l^{(\epsilon)})^2 + \nu_l^2 + \epsilon^2}{2\epsilon} - \frac{M}{2},$$

where ν_l^2 is the variance for $\{b_l^{(\epsilon)}(t)\}$.

Therefore, in the heavy traffic limit, we have

$$\liminf_{\epsilon \to 0^+} \epsilon \mathbb{E}\left[\Phi_l^{(\epsilon)}\right] \geq \frac{\sigma_l^2 + \nu_l^2}{2}. \tag{B.14}$$

State Space Collapse

The weighted-workload routing distributes the tasks local only to $\mathcal{B}_u$ in the ratio of $\alpha : \beta$ in terms of server workload across $\mathcal{B}_u$ and $\mathcal{H}_u$. We will show that the workload vector $\mathbf{W}$ collapses to the direction $\mathbf{c}_l = \frac{\tilde{\mathbf{c}}_l}{||\tilde{\mathbf{c}}_l||} \in \mathbb{R}_+^M$ as a vector with unit l_2 norm, where

$$\tilde{c}_{lm} = \begin{cases} \beta, & \forall m \in \mathcal{H}_u \\ \alpha, & \forall m \in \mathcal{B}_u \end{cases}.$$

The parallel and perpendicular components of the steady-state weighted queue-length vector $\mathbf{W}$ with respect to $\mathbf{c}_l$ are

$$\mathbf{W}_{||} = \langle \mathbf{c}_l, \mathbf{W} \rangle \mathbf{c}_l, \quad \mathbf{W}_{\perp} = \mathbf{W} - \mathbf{W}_{||}.$$

We establish state space collapse by showing that $\mathbf{W}_{\perp}$ is bounded and independent of the heavy-traffic parameter ϵ. That is, there exists a sequence of finite numbers $\{C_r : r \in \mathbb{N}\}$ such that for each positive integer r,

$$\mathbb{E}\left[\|\mathbf{W}_{\perp}\|^r\right] \leq C_r.$$

Upper Bound

Utilizing the property of state-space collapse in the heavy-traffic limit, we can obtain an upper bound on $\mathbb{E}\left[\Phi_l^{(\epsilon)}\right]$:

$$\mathbb{E}\left[\Phi_l^{(\epsilon)}\right] \leq \frac{(\sigma_l^{(\epsilon)})^2 + (\nu_l^{(\epsilon)})^2}{2\epsilon} + B_l^{(\epsilon)},$$

where $B_l^{(\epsilon)} = o(\frac{1}{\epsilon})$, i.e., $\lim_{\epsilon \downarrow 0} \epsilon B_l^{(\epsilon)} = 0$. Therefore, in the heavy-traffic limit, we have

$$\limsup_{\epsilon \downarrow 0} \epsilon \mathbb{E}\left[\Phi_l^{(\epsilon)}\right] \leq \frac{\sigma_l^2 + \nu_l^2}{2},$$

which coincides with the lower bound (B.14).

B.5 Heavy-traffic Optimality with Overloaded Racks

B.5.1 Proof of Theorem 4.4

We need the following lemma to prove Theorem 4.4.

Lemma B.17. *There exists a constant ρ_h, $0 \leq \rho_h < 1$, not depending on ϵ, such that for any $m \in \mathcal{H}_o \cup \mathcal{H}_u$,*

$$\mathbb{E}\left[\frac{A_m^l}{\alpha}\right] \leq \rho_h.$$

Proof of Lemma B.17. We prove this lemma by contradiction. Assume that there exists a server $n \in \mathcal{H}_u$, s.t. $\mathbb{E}\left[\frac{A_n^l}{\alpha}\right] \xrightarrow{\epsilon \to 0} 1$.

Let $\mathcal{S}_n$ denote the set of servers in $\mathcal{H}_u$ that have shared traffic with n, i.e., $\mathcal{S}_n = \{i : \exists \bar{L} \in \mathcal{L} \; s.t. \; n \in \bar{L}, \; i \in \bar{L}\}$. For each $i \in \mathcal{S}_n$, define $P_i = \mathbb{P}[W_n < W_i] + \frac{1}{2}\mathbb{P}[W_n = W_i]$. Among tasks that are local to both n and i (might also local to other servers), we denote by $L_i^{(n)}$ the amount that are routed to server n, and L_i the amount routed to i. Then

$$L_i^{(n)} : L_i = P_i : (1 - P_i).$$

Consider remote tasks from $\mathcal{L}_{\mathcal{B}_o}$. For each i, let R_i denote the amount of remote tasks from $\mathcal{L}_{\mathcal{B}_o}$. Thus

$$R_n : R_i = P_i : (1 - P_i) = L_i^{(n)} : L_i.$$

Note that local load on n is contributed by $\{L_i^{(n)}\}_{i \in \mathcal{S}_n}$. Since $\mathbb{E}\left[\frac{A_n^l}{\alpha}\right] \xrightarrow{\epsilon \to 0} 1$, there must exist a subset $\mathcal{S}_n^* \subset \mathcal{S}_n$, such that for any $i \in \mathcal{S}_n^*$, $\lim_{\epsilon \to 0} L_i^{(n)} > 0$. In addition, as the system is stable, $\mathbb{E}\left[\frac{A_n^r}{\alpha}\right] \xrightarrow{\epsilon \to 0} 0$, i.e., $R_n \xrightarrow{\epsilon \to 0} 0$. Hence $\forall n \in \mathcal{S}_n^*$,

$$R_i = R_n \frac{L_i}{L_i^{(n)}} \xrightarrow{\epsilon \to 0} 0.$$

Thus the amount of tasks from $\mathcal{L}_{\mathcal{B}_o}$ that are served remotely by $n \cup \mathcal{S}_n^*$ vanishes as $\epsilon \to 0$.

From Lemma B.7, we know that the amount of rack-local tasks and that of

remote tasks not from $\mathcal{L}_{\mathcal{B}_o}$ have order $o(\epsilon)$. As a result, for any $i \in \mathcal{S}_n^* \cup \{n\}$,

$$\mathbb{E}\left[\frac{A_i^l}{\alpha}\right] \xrightarrow{\epsilon \to 0} 1.$$

Since $\forall j \in \mathcal{S}_n \setminus \mathcal{S}_n^*$, $L_j^{(n)} \to 0$, local load on n comes from tasks types that are only local to $n \cup \mathcal{S}_n^*$. If there exists $i \in \mathcal{S}_n^*$, such that its local load includes tasks that are also local to servers not in the set $n \cup \mathcal{S}_n^*$, since $\mathbb{E}\left[\frac{A_i^l}{\alpha}\right] \xrightarrow{\epsilon \to 0} 1$, similarly we can define $\mathcal{S}_i$ and find a set $\mathcal{S}_i^*$, such that shared traffic between i and $\mathcal{S}_i \setminus \mathcal{S}_i^*$ rarely contributes to the local load on i. Repeat the above procedure for any server in $\mathcal{S}_n^* \cup \{n\}$, we will end up with a set $\mathcal{S}^* \subset \mathcal{H}_u$, such that the local load on each server in $\mathcal{S}^*$ converges to 1 as $\epsilon \to 0$, i.e.,

$$\mathbb{E}\left[A_i^l\right] \xrightarrow{\epsilon \to 0} \alpha. \tag{B.15}$$

And the amount of shared traffic between $\mathcal{S}^*$ and $\mathcal{H}_u \setminus \mathcal{S}^*$ routed to $\mathcal{S}^*$ vanishes. That is,

$$\sum_{i \in \mathcal{S}^*} \mathbb{E}\left[A_i^l\right] \xrightarrow{\epsilon \to 0} \sum_{\bar{L}: \forall m \in \bar{L}, m \in \mathcal{S}^*} \lambda_{\bar{L}}. \tag{B.16}$$

Equations (B.15) and (B.16) imply that $\sum_{\bar{L}: \forall m \in \bar{L}, m \in \mathcal{S}^*} \lambda_{\bar{L}} = |\mathcal{S}^*|\alpha$. However, by the definition of $\mathcal{H}_u$,

$$\sum_{\bar{L}: \forall m \in \bar{L}, m \in \mathcal{S}^*} \lambda_{\bar{L}} \leq \sum_{m \in \mathcal{S}^*} \phi_m < |\mathcal{S}^*|\alpha.$$

Contradiction. Thus the assumption is not valid. Therefore, for any $n \in \mathcal{H}_u$, its local load in steady state is strictly less than 1 as $\epsilon \to 1$. That is, there exists a constant $\rho_h^* < 1$, such that $\mathbb{E}\left[A_n^l\right] \leq \rho_h^*$. Similarly, we can show that this holds for $\mathcal{H}_o$ as well.

Now we are ready to prove Theorem 4.4. Consider the system in steady state. For any $m \in \mathcal{H}_u$, define

$$\hat{Q}_m(t) = Q_m^l(t) + Q_m^k(t),\ \hat{A}_m(t) = A_m^l(t) + A_m^k(t),\ \hat{S}_m(t) = S_m^l(t) + S_m^k(t).$$

The dynamics of $\hat{\mathbf{Q}}$ can be written as

$$\hat{\mathbf{Q}}(t+1) = \hat{\mathbf{Q}}(t) + \hat{\mathbf{A}}(t) - \hat{\mathbf{S}}(t).$$

Consider the ideal arrival process $\mathbf{F}(t)$ defined in the proof of Theorem 4.7. Let

$$\hat{F}_m(t) = F_m^l(t) + F_m^k(t).$$

Then we can rewrite the dynamics of $\hat{\mathbf{Q}}$ as

$$\hat{\mathbf{Q}}(t+1) = \hat{\mathbf{Q}}(t) + \hat{\mathbf{F}}(t) - \hat{\mathbf{S}}(t) + \hat{\mathbf{A}}(t) - \hat{\mathbf{F}}(t).$$

Let $\mathbf{c}_h \in \mathbb{R}_+^{M_{\mathcal{H}_o}}$ be a unit vector with all elements equal, i.e.,

$$\mathbf{c_h} = \frac{1}{\sqrt{M_{\mathcal{H}_u}}}(\underbrace{1, 1, \cdots, 1}_{M_{\mathcal{H}_u}}).$$

Since we consider the system in steady state, the drift of function $\left\|\hat{\mathbf{Q}}_{||}\right\|^2 = \left\|\langle \mathbf{c_h}, \hat{\mathbf{Q}}_{||}\rangle\right\|^2$ should be zero, which yields

$$\begin{aligned} & 2\mathbb{E}\left[\langle \mathbf{c_h}, \hat{\mathbf{Q}}(t)\rangle\langle \mathbf{c_h}, \hat{\mathbf{S}}(t) - \hat{\mathbf{F}}(t)\rangle\right] \\ = \; & \mathbb{E}\left[\langle \mathbf{c_h}, \hat{\mathbf{F}}(t) - \hat{\mathbf{S}}(t)\rangle^2\right] + \mathbb{E}\left[\langle \mathbf{c_h}, \hat{\mathbf{A}}(t) - \hat{\mathbf{F}}(t)\rangle^2\right] \\ & +2\mathbb{E}\left[\langle \mathbf{c_h}, \hat{\mathbf{Q}}(t) + \hat{\mathbf{F}}(t) - \hat{\mathbf{S}}(t)\rangle\langle \mathbf{c_h}, \hat{\mathbf{A}}(t) - \hat{\mathbf{F}}(t)\rangle\right]. \end{aligned} \tag{B.17}$$

According to the definition of ideal arrival process,

$$\langle \mathbf{c_h}, \hat{\mathbf{F}}(t)\rangle = \frac{1}{\sqrt{M_{\mathcal{H}_u}}} \sum_{m \in \mathcal{H}_u} F_m^l(t) = \frac{1}{\sqrt{M_{\mathcal{H}_u}}} \sum_{\bar{L} \in \mathcal{L}^*_{\mathcal{H}_u}} A_{\bar{L}}(t).$$

Thus the sum of ideal arrivals on $\mathcal{H}_u$ and the queue lengths are independent. We have

$$\begin{aligned} & \mathbb{E}\left[\langle \mathbf{c_h}, \hat{\mathbf{Q}}(t)\rangle\langle \mathbf{c_h}, \hat{\mathbf{S}}(t) - \hat{\mathbf{F}}(t)\rangle\right] \\ = \; & \frac{1}{M_{\mathcal{H}_u}}\mathbb{E}\left[\left(\sum_{m \in \mathcal{H}_u} \hat{Q}_m(t)\right)\left(\sum_{m \in \mathcal{H}_u} \hat{S}_m(t)\right)\right] - \frac{1}{M_{\mathcal{H}_u}}\mathbb{E}\left[\sum_{m \in \mathcal{H}_u} \hat{Q}_m(t)\right]\left(\sum_{\bar{L} \in \mathcal{L}^*_{\mathcal{H}_u}} \lambda_{\bar{L}}\right). \end{aligned}$$

Note that $\hat{S}_m(t) = S_m^l(t) + S_m^k(t)$ only depends on the state of $m-$th queue.

Hence

$$\begin{aligned}
&\mathbb{E}\left[\left(\sum_{m\in\mathcal{H}_u}\hat{Q}_m(t)\right)\left(\sum_{m\in\mathcal{H}_u}\hat{S}_m(t)\right)\right]\\
=&\sum_{m\in\mathcal{H}_u}\mathbb{E}\left[\hat{S}_m(t)\hat{Q}_m(t)\right]+\sum_{m\in\mathcal{H}_u}\mathbb{E}\left[\hat{S}_m(t)\right]\mathbb{E}\left[\sum_{n\in\mathcal{H}_u:n\neq m}\hat{Q}_n(t)\right]\\
=&\sum_{m\in\mathcal{H}_u}\mathbb{E}\left[\hat{S}_m(t)\hat{Q}_m(t)\right]+\mathbb{E}\left[\sum_{m\in\mathcal{H}_u}\hat{S}_m(t)\right]\mathbb{E}\left[\sum_{n\in\mathcal{H}_u}\hat{Q}_n(t)\right]\\
&-\sum_{m\in\mathcal{H}_u}\left(\mathbb{E}\left[\hat{S}_m(t)\right]\mathbb{E}\left[\hat{Q}_m(t)\right]\right).
\end{aligned}$$

We have the following bound on the term $\sum_{m\in\mathcal{H}_u}\mathbb{E}\left[\hat{S}_m(t)\hat{Q}_m(t)\right]$.

Lemma B.18.

$$\sum_{m\in\mathcal{H}_u}\mathbb{E}\left[\hat{S}_m(t)\hat{Q}_m(t)\right]\geq\sum_{m\in\mathcal{H}_u}\alpha\mathbb{E}\left[\hat{Q}_m(t)\right]-C_1,$$

where $C_1>0$ is a constant.

Since $\hat{\mathbf{Q}}$ is in steady state, we have $\mathbb{E}\left[\hat{S}_m(t)\right]=\mathbb{E}\left[\hat{A}_m(t)\right]$. So

$$\begin{aligned}
&\mathbb{E}\left[\sum_{m\in\mathcal{H}_u}\hat{S}_m(t)\right]\mathbb{E}\left[\sum_{n\in\mathcal{H}_u}\hat{Q}_n(t)\right]-\sum_{m\in\mathcal{H}_u}\left(\mathbb{E}\left[\hat{S}_m(t)\right]\mathbb{E}\left[\hat{Q}_m(t)\right]\right)\\
=&\mathbb{E}\left[\sum_{m\in\mathcal{H}_u}\hat{A}_m(t)\right]\mathbb{E}\left[\sum_{n\in\mathcal{H}_u}\hat{Q}_n(t)\right]-\sum_{m\in\mathcal{H}_u}\left(\mathbb{E}\left[\hat{A}_m(t)\right]\mathbb{E}\left[\hat{Q}_m(t)\right]\right)\\
=&\mathbb{E}\left[\sum_{m\in\mathcal{H}_u}\left(A_m^l(t)+A_m^k(t)\right)\right]\mathbb{E}\left[\sum_{n\in\mathcal{H}_u}\hat{Q}_n(t)\right]\\
&-\sum_{m\in\mathcal{H}_u}\left(\mathbb{E}\left[A_m^l(t)+A_m^k(t)\right]\mathbb{E}\left[\hat{Q}_m(t)\right]\right)\\
\overset{(a)}{\geq}&\mathbb{E}\left[\sum_{m\in\mathcal{H}_u}\left(A_m^l(t)+A_m^k(t)\right)\right]\mathbb{E}\left[\sum_{n\in\mathcal{H}_u}\hat{Q}_n(t)\right]-\sum_{m\in\mathcal{H}_u}\left(\alpha\rho_h^*\mathbb{E}\left[\hat{Q}_m(t)\right]\right),
\end{aligned}$$

where inequality (a) follows from Lemma B.17.

Together we have

$$\begin{aligned}
&\mathbb{E}\left[\langle \mathbf{c_h}, \hat{\mathbf{Q}}(t)\rangle \langle \mathbf{c_h}, \hat{\mathbf{S}}(t) - \hat{\mathbf{F}}(t)\rangle\right] \\
\geq\ & \frac{1}{M_{\mathcal{H}_u}} \left\{ \sum_{m\in\mathcal{H}_u} \alpha \mathbb{E}\left[\hat{Q}_m(t)\right] - C_1 + \mathbb{E}\left[\sum_{m\in\mathcal{H}_u} \left(A_m^l(t) + A_m^k(t)\right)\right] \mathbb{E}\left[\sum_{n\in\mathcal{H}_u} \hat{Q}_n(t)\right] \right. \\
& \left. - \sum_{m\in\mathcal{H}_u} \left(\alpha\rho_h^* \mathbb{E}\left[\hat{Q}_m(t)\right]\right) - \mathbb{E}\left[\sum_{m\in\mathcal{H}_u} \hat{Q}_m(t)\right] \left(\sum_{\bar{L}\in\mathcal{L}^*_{\mathcal{H}_u}} \lambda_{\bar{L}}\right) \right\} \\
=\ & \frac{1}{M_{\mathcal{H}_u}} \left\{ \alpha(1-\rho_h^*) + \mathbb{E}\left[\sum_{m\in\mathcal{H}_u} \left(A_m^l(t) + A_m^k(t)\right)\right] - \sum_{\bar{L}\in\mathcal{L}^*_{\mathcal{H}_u}} \lambda_{\bar{L}} \right\} \mathbb{E}\left[\sum_{m\in\mathcal{H}_u} \hat{Q}_m(t)\right] \\
& - \frac{C_1}{M_{\mathcal{H}_u}}.
\end{aligned}$$

Following the same line of reasoning as in the proof of Lemma 3.20, we can show that

$$\sum_{\bar{L}\in\mathcal{L}^*_{\mathcal{H}_u}} \lambda_{\bar{L}} - \mathbb{E}\left[\sum_{m\in\mathcal{H}_u} \left(A_m^l(t) + A_m^k(t)\right)\right] \leq C\epsilon,$$

where C is a constant only depending on α, β and γ. Also, By the definition of ideal arrival process,

$$\sum_{m\in\mathcal{H}_u} \hat{F}_m(t) = \sum_{\bar{L}\in\mathcal{L}^*_{\mathcal{H}_u}} \lambda_{\bar{L}} \geq \sum_{m\in\mathcal{H}_u} \hat{A}_m(t).$$

Therefore we have

$$\mathbb{E}\left[\langle \mathbf{c_h}, \hat{\mathbf{Q}}(t)\rangle \langle \mathbf{c_h}, \hat{\mathbf{S}}(t) - \hat{\mathbf{F}}(t)\rangle\right] \tag{B.18}$$

$$\geq \frac{1}{M_{\mathcal{H}_u}} \left[\alpha(1-\rho_h^*) - C\epsilon\right] \mathbb{E}\left[\sum_{m\in\mathcal{H}_u} \hat{Q}_m(t)\right] - \frac{C_1}{M_{\mathcal{H}_u}}. \tag{B.19}$$

By the boundedness of arrivals and service, there exist constants $C_2 > 0$ and $C_3 > 0$ not depending on ϵ such that

$$\mathbb{E}\left[\langle \mathbf{c_h}, \hat{\mathbf{F}}(t) - \hat{\mathbf{S}}(t)\rangle^2\right] \leq C_2, \tag{B.20}$$

$$\mathbb{E}\left[\langle \mathbf{c_h}, \hat{\mathbf{A}}(t) - \hat{\mathbf{F}}(t)\rangle^2\right] \leq C_3. \tag{B.21}$$

For the last on the RHS of (B.17),

$$\begin{aligned}
& \mathbb{E}\left[\langle \mathbf{c_h}, \hat{\mathbf{Q}}(t) + \hat{\mathbf{F}}(t) - \hat{\mathbf{S}}(t)\rangle \langle \mathbf{c_h}, \hat{\mathbf{A}}(t) - \hat{\mathbf{F}}(t)\rangle\right] \\
= \; & \mathbb{E}\left[\langle \mathbf{c_h}, \hat{\mathbf{Q}}(t)\rangle \langle \mathbf{c_h}, \hat{\mathbf{A}}(t) - \hat{\mathbf{F}}(t)\rangle\right] + \mathbb{E}\left[\langle \mathbf{c_h}, \hat{\mathbf{F}}(t) - \hat{\mathbf{S}}(t)\rangle \langle \mathbf{c_h}, \hat{\mathbf{A}}(t) - \hat{\mathbf{F}}(t)\rangle\right] \\
\overset{(a)}{\leq} \; & \mathbb{E}\left[\langle \mathbf{c_h}, \hat{\mathbf{F}}(t) - \hat{\mathbf{S}}(t)\rangle \langle \mathbf{c_h}, \hat{\mathbf{A}}(t) - \hat{\mathbf{F}}(t)\rangle\right] \\
\overset{(b)}{\leq} \; & C_4, \qquad \text{(B.22)}
\end{aligned}$$

where (a) follows by the fact that $\langle \mathbf{c_h}, \hat{\mathbf{A}}(t) - \hat{\mathbf{F}}(t)\rangle \leq 0$, and (b) follows from the boundedness of arrivals, and $C_4 > 0$ is a constant.

From Eq. (B.17) and inequalities (B.19)-(B.22), we have

$$\frac{2}{M_{\mathcal{H}_u}} \left[\alpha(1-\rho_h^*) - C\epsilon\right] \mathbb{E}\left[\sum_{m \in \mathcal{H}_u} \hat{Q}_m(t)\right] \leq \frac{2C_1}{M_{\mathcal{H}_u}} + C_2 + C_3 + 2C_4. \quad \text{(B.23)}$$

Thus for any $0 < \epsilon < \frac{\alpha(1-\rho_h^*)}{C}$,

$$\mathbb{E}\left[\sum_{m \in \mathcal{H}_u} \hat{Q}_m(t)\right] \leq \frac{C_5}{\alpha(1-\rho_h^*) - C\epsilon},$$

where $C_5 = C_1 + (C_2 + C_3 + 2C_4)\frac{M_{\mathcal{H}_u}}{2}$. Therefore

$$\lim_{\epsilon \downarrow 0} \mathbb{E}\left[\sum_{m \in \mathcal{H}_u} \left(Q_m^{l(\epsilon)}(t) + Q_m^{k(\epsilon)}(t)\right)\right] \leq \frac{C_5}{\alpha(1-\rho_h^*)}. \qquad \text{(B.24)}$$

That is,

$$\lim_{\epsilon \downarrow 0} \epsilon \mathbb{E}\left[\sum_{m \in \mathcal{H}_u} \left(Q_m^{l(\epsilon)}(t) + Q_m^{k(\epsilon)}(t)\right)\right] = 0.$$

Similarly we can show that

$$\lim_{\epsilon \downarrow 0} \epsilon \mathbb{E}\left[\sum_{m \in \mathcal{H}_o} Q_m^{l(\epsilon)}(t)\right] = 0.$$

■

B.5.2 Proof of Theorem 4.6

Analogue to Lemma B.8, we have the following lemma for the ideal load decomposition for the case $\mathcal{O} \neq \emptyset$.

Lemma B.19. *Consider an arrival rate vector $\boldsymbol{\lambda}$ that satisfies the heavy traffic rack overloaded assumption, with $0 < \epsilon < \bar{\epsilon}$, where $\bar{\epsilon}$ is a positive constant. Then there exists a decomposition $\left\{\lambda^*_{\bar{L},n,m}\right\}$ of $\boldsymbol{\lambda}$ satisfying the following conditions:*

1. *$\forall m \in \mathcal{M}$, define*

$$\omega_m = \sum_{\bar{L}:m\in\bar{L}} \frac{\lambda^*_{\bar{L},m}}{\alpha} + \sum_{\bar{L}:m\in\bar{L}_k} \frac{\lambda^*_{\bar{L},m}}{\beta} + \sum_{\bar{L}:m\in\bar{L}_r} \frac{\lambda^*_{\bar{L},m}}{\gamma}.$$

   Then

$$\omega_m = \begin{cases} 1-\gamma\epsilon_0, & \forall m \in \mathcal{H}_u \\ 1-\beta\epsilon_0, & \forall m \in \mathcal{H}_o \\ 1-\alpha\epsilon_0, & \forall m \in \mathcal{B}_o \end{cases}$$

   where $\epsilon_0 = \frac{\epsilon}{\|\tilde{\mathbf{c}}\|^2}$.

2. *Let $\mathcal{L}_{\mathcal{B}_o}$ denote the set of task types that are only local to $\mathcal{B}_o$. $\forall \bar{L} \in \mathcal{L}_{\mathcal{B}_o}$, $\forall m \in \{i \in \mathcal{M} | i \in \bar{L}$, or $i \in \mathcal{H}_u$, or $i \in \bar{L}_k \cap \mathcal{H}_o\}$, $\lambda^*_{\bar{L},m} = \sum_{n\in\bar{L}} \lambda^*_{\bar{L},n,m} \geq \kappa$, where κ is a positive constant independent of ϵ.*

We need the following additional lemmas analogue to Lemmas B.11-B.13 for the evenly loaded case.

Lemma B.20. *Under balanced-Pandas, for any $t \geq 0$,*

$$\mathbb{E}\left[\langle \mathbf{W}(t), \mathbf{A}(t)\rangle - \langle \mathbf{W}(t), \boldsymbol{\omega}\rangle | Z(t)\right] \leq -\lambda_{min}\|\mathbf{W}_\perp(t)\|, \tag{B.25}$$

where λ_{min} is a constant independent of ϵ.

Lemma B.21. *Under balanced-Pandas, for any $t \geq 0$,*

$$\mathbb{E}\left[\langle \mathbf{W}(t), \boldsymbol{\omega}\rangle - \langle \mathbf{W}(t), \mathbf{S}(t)\rangle | Z(t)\right] = -\frac{\epsilon}{\|\tilde{\mathbf{c}}\|}\langle \mathbf{c}, \mathbf{W}\rangle.$$

Lemma B.22. *Under balanced-Pandas, for any $t \geq 0$,*

$$\mathbb{E}\left[\langle \mathbf{c}, \mathbf{W}(t)\rangle \langle \mathbf{c}, \mathbf{A}(t) - \mathbf{S}(t)\rangle | Z(t)\right] \geq -\frac{\epsilon}{\|\tilde{\mathbf{c}}\|}\langle \mathbf{c}, \mathbf{W}\rangle.$$

Proof of Theorem 4.6.: We consider Lyapunov function

$$F(Z) = \|\mathbf{W}_{\perp}\|,$$

whose drift can be bounded as

$$\Delta F(Z) \leq \frac{1}{2\|\mathbf{W}_{\perp}\|}(\Delta V(Z) - \Delta V_{||}(Z)), \tag{B.26}$$

where $\Delta V(Z)$ and $\Delta V_{||}(Z)$ are the drifts for $V(Z) = \|\mathbf{W}\|^2$ and $V_{||}Z = \left\|\mathbf{W}_{||}\right\|^2$ respectively. We then have

$$\begin{aligned} & \mathbb{E}\left[\Delta V(Z(t)) - \Delta V_{||}(Z(t))|Z(t)\right] \qquad (B.27) \\ \leq \; & 2\mathbb{E}\left[\langle \mathbf{W}(t), \mathbf{A}(t) - \mathbf{S}(t)\rangle - \langle \mathbf{c}, \mathbf{W}(t)\rangle \langle \mathbf{c}, \mathbf{A}(t) - \mathbf{S}(t)\rangle | Z(t)\right] + C_1. \end{aligned}$$

Lemma B.20-B.22 gives an bound on $\Delta F(Z(t))$:

$$\mathbb{E}\left[\Delta F(Z(t))|Z(t)\right] \leq -\lambda_0 + \frac{C}{\|\mathbf{W}_{\perp}(t)\|},$$

where λ_0 and C are positive constants independent of ϵ. This inequality verifies the negative drift condition, and hence establishes the existence of finite constants $\{C'_r\}_{r\in\mathbb{N}}$ for which $\mathbb{E}\left[\left\|\mathbf{W}_{\perp}^{(\epsilon)}(t)\right\|^r\right] \leq C'_r$. ∎

Proof of Lemmas B.20-B.22

Proof of Lemma B.20. From the proof of Lemma 3.7, we have

$$\mathbb{E}\left[\langle \mathbf{W}(t), \mathbf{A}(t)\rangle | Z(t)\right] \;=\; \sum_{\bar{L}\in\mathcal{L}} \left(W^*_{\bar{L}}(t) \sum_m \lambda^*_{\bar{L},m} \right),$$

where

$$W^*_{\bar{L}}(t) = \min_{m\in\mathcal{M}} \left\{ \frac{W_m(t)}{\alpha} I_{\{m\in\bar{L}\}}, \frac{W_m(t)}{\beta} I_{\{m\in\bar{L}_k\}}, \frac{W_m(t)}{\gamma} I_{\{m\in\bar{L}_r\}} \right\}.$$

Note that

$$\begin{aligned}
&\mathbb{E}\left[\langle \mathbf{W}(t), \boldsymbol{\omega}\rangle | Z(t)\right] \\
= &\sum_{\bar{L}\in\mathcal{L}} \left(\sum_{m:m\in\bar{L}} \frac{W_m(t)}{\alpha} \lambda^*_{\bar{L},m} + \sum_{m:m\in\bar{L}_k} \frac{W_m(t)}{\beta} \lambda^*_{\bar{L},m} + \sum_{m:m\in\bar{L}_r} \frac{W_m(t)}{\gamma} \lambda^*_{\bar{L},m} \right).
\end{aligned}$$

So

$$\begin{aligned}
&\mathbb{E}\left[\langle \mathbf{W}(t), \mathbf{A}(t)\rangle | Z(t)\right] - \mathbb{E}\left[\langle \mathbf{W}(t), \boldsymbol{\omega}\rangle | Z(t)\right] \\
= &\sum_{m} \Bigg[\sum_{\substack{\bar{L}\in\mathcal{L}_{\mathcal{B}_o} \\ s.t.\ m\in\bar{L}}} \left(W^*_{\bar{L}}(t) - \frac{W_m(t)}{\alpha} \right) \lambda^*_{\bar{L},m} + \sum_{\substack{\bar{L}\in\mathcal{L}_{\mathcal{B}_o} \\ s.t.\ m\in\bar{L}_k}} \left(W^*_{\bar{L}}(t) - \frac{W_m(t)}{\beta} \right) \lambda^*_{\bar{L},m} \\
&+ \sum_{\substack{\bar{L}\in\mathcal{L}_{\mathcal{B}_o} \\ s.t.\ m\in\bar{L}_r}} \left(W^*_{\bar{L}}(t) - \frac{W_m(t)}{\gamma} \right) \lambda^*_{\bar{L},m} \Bigg].
\end{aligned}$$

Let $\lambda_0 = \frac{\kappa}{M}$, where κ is defined in Lemma B.19. Then $\lambda_0 > 0$ and does not depend on ϵ. By Lemma B.19, for any $\bar{L} \in \mathcal{L}_{\mathcal{B}_o}$, $\forall m \in \{i \in \mathcal{M} | i \in \bar{L},$ or $i \in \mathcal{H}_u,$ or $i \in \bar{L}_k \cap \mathcal{H}_o\}$, $\lambda^*_{\bar{L},m} \geq \lambda_0$, and $\frac{\lambda^*_{\bar{L},m}}{M} \geq \lambda_0$.

Define $\bar{L}^* \in \arg\min_{\bar{L}\in\mathcal{L}_{\mathcal{B}_o}} W^*_{\bar{L}}(t)$, and $W_{min} = W^*_{\bar{L}^*}(t)$. Consider a particular server $m_0 \in \mathcal{H}_u$, and discard the terms with indices not equal to $\bar{L}^*$ in its summation. For each $m \in \mathcal{B}_o$, keep one term with index $\bar{L}^{(m)} \in \mathcal{L}_{\mathcal{B}_o}$ $s.t.$ $m \in \bar{L}$; for each $m \in \mathcal{H}_o$, keep one term with index $\bar{L}^{(m)} \in \mathcal{L}_{\mathcal{B}_o}$ $s.t.$ $m \in \bar{L}_k$; for each $m \in \mathcal{H}_u$, except m_0, keep one term with index $\bar{L}^{(m)} \in \mathcal{L}_{\mathcal{B}_o}$ $s.t.$ $m \in \bar{L}_r$. Discarding all other terms yields:

$$
\begin{aligned}
& \mathbb{E}\left[\langle \mathbf{W}(t), \mathbf{A}(t)\rangle | Z(t)\right] - \mathbb{E}\left[\langle \mathbf{W}(t), \boldsymbol{\omega}\rangle | Z(t)\right] \\
\leq\ & -\sum_{m\in\mathcal{B}_o} \lambda_0 \left(\frac{W_m(t)}{\alpha} - W^*_{\bar{L}^{(m)}}(t)\right) - \sum_{m\in\mathcal{H}_o} \lambda_0 \left(\frac{W_m(t)}{\beta} - W^*_{\bar{L}^{(m)}}(t)\right) \\
& - \sum_{m\in\mathcal{H}_u, m\neq m_0} \lambda_0 \left(\frac{W_m(t)}{\gamma} - W^*_{\bar{L}^{(m)}}(t)\right) - \lambda_0 M \left(\frac{W_{m_0}(t)}{\gamma} - W_{min}(t)\right) \\
=\ & -\sum_{m\in\mathcal{B}_o} \lambda_0 \left(\frac{W_m(t)}{\alpha} - W_{min}(t)\right) - \sum_{m\in\mathcal{H}_o} \lambda_0 \left(\frac{W_m(t)}{\beta} - W_{min}(t)\right) \\
& - \sum_{m\in\mathcal{H}_u} \lambda_0 \left(\frac{W_m(t)}{\gamma} - W_{min}(t)\right) - \lambda_0 \sum_{m\neq m_0} \left(\frac{W_{m_0}(t)}{\gamma} - W^*_{\bar{L}^{(m)}}(t)\right) \\
\overset{(a)}{\leq}\ & -\lambda_0 \left[\sum_{m\in\mathcal{B}_o} \left(\frac{W_m(t)}{\alpha} - W_{min}(t)\right) + \sum_{m\in\mathcal{H}_o} \left(\frac{W_m(t)}{\beta} - W_{min}(t)\right)\right. \\
& \left. + \sum_{m\in\mathcal{H}_u} \left(\frac{W_m(t)}{\gamma} - W_{min}(t)\right)\right] \\
\leq\ & -\frac{\lambda_0}{\alpha} \sum_m \left(W_m(t) - \tilde{c}_m W_{min}(t)\right),
\end{aligned}
$$

where (a) follows from the fact that $\frac{W_{m_0}(t)}{\gamma} \geq W^*_{\bar{L}^{(m)}}(t)$ for all $\bar{L}^{(m)}$.

By the definition of $W_{min}(t)$, for any m, $W_m(t) \geq \tilde{c}_m W_{min}(t)$. Hence

$$
\begin{aligned}
\sum_m \left(W_m(t) - \tilde{c}_m W_{min}(t)\right) &= \|\mathbf{W}(t) - W_{min}(t)\tilde{\mathbf{c}}\|_1 \\
&\geq \|\mathbf{W}(t) - W_{min}(t)\|\tilde{\mathbf{c}}\| \cdot \mathbf{c}\|_2,
\end{aligned}
$$

where $\|\cdot\|_1$ and $\|\cdot\|_2$ are the l_1 and l_2 norm, respectively. The inequality follows by the fact that the l_1 norm of a vector is no smaller than its l_2 norm. As the convex function $\|\mathbf{W}(t) - x\mathbf{c}\|_2$ is minimized at $x = \langle \mathbf{c}, \mathbf{W}\rangle$,

$$
\|\mathbf{W}(t) - W_{min}(t)\|\tilde{\mathbf{c}}\| \cdot \mathbf{c}\|_2 \geq \|\mathbf{W}(t) - \langle \mathbf{c}, \mathbf{W}\rangle \mathbf{c}\|_2 = \|\mathbf{W}_\perp(t)\|.
$$

Therefore, we have

$$
\mathbb{E}\left[\langle \mathbf{W}(t), \mathbf{A}(t)\rangle | Z(t)\right] - \mathbb{E}\left[\langle \mathbf{W}(t), \boldsymbol{\omega}\rangle | Z(t)\right] \leq -\lambda_0 \|\mathbf{W}_\perp(t)\|,
$$

where $\lambda_{min} = \frac{\lambda_0}{\alpha}$, independent of ϵ. ∎

Proof of Lemma B.21. It is similar to that of Lemma B.12. We skip the proof details here.

Proof of Lemma B.22. It is similar to that of Lemma B.13. We skip the proof details here.

B.5.3 Proof of Theorem 4.7

First we define a series of *ideal processes*.

Ideal scheduling decision process $\boldsymbol{\eta'}(t)$: $\forall m \in \mathcal{B}_o$, $\eta'_m(t) = 0$; $\forall m \in \mathcal{H}_o$, $\eta'_m(t) = \eta_m(t)$ if $\eta_m(t) = 0$, and $\eta'_m(t) = 1$ when $f_m(t^-) = -1$, $Q^l_m(t) = 0$; $\forall m \in \mathcal{H}_u$, $\eta'_m(t) = \eta_m(t)$. That is, each beneficiary in the overloaded racks is scheduled to serve its local sub-queue only under the *ideal scheduling*, and an idle helper with an empty local sub-queue in an overloaded rack is scheduled to serve its rack-local sub-queue only.

Ideal service process $\mathbf{D}(t)$: For each $m \in \mathcal{B}_o$,

$$D^k_m(t) = 0,\ D^r_m(t) = 0,$$

and each process $D^l_m(t)$ is i.i.d. with $D^l_m(t) \sim \text{Bern}(\alpha)$ and is coupled with $\mathbf{S}_m(t)$ in the following way: If $\eta_m(t) = 0$, $D^l_m(t) = S^l_m(t)$; if $\eta_m(t) = 1$, $D^l_m(t) = 1$ when $S^k_m(t) = 1$, and $D^l_m(t) \sim \text{Bern}(\frac{\alpha-\beta}{1-\beta})$ when $S^k_m(t) = 0$; if $\eta_m(t) = 2$, $D^l_m(t) = 1$ when $S^r_m(t) = 1$, and $D^l_m(t) \sim \text{Bern}(\frac{\alpha-\gamma}{1-\gamma})$ when $S^k_m(t) = 0$.

For each $m \in \mathcal{H}_o$,

$$D^l_m(t) = S^l_m(t),\ D^r_m(t) = 0,$$

and each process $D^k_m(t)$ is i.i.d. with $D^k_m(t) \sim \text{Bern}(\beta I_{\{\eta_m(t)\neq 0\}})$ and is coupled with $\mathbf{S}_m(t)$ in the following way: If $\eta_m(t) \neq 2$, $D^k_m(t) = S^k_m(t)$; if $\eta_m(t) = 2$, $D^k_m(t) = 1$ when $S^r_m(t) = 1$, and $D^l_m(t) \sim \text{Bern}(\frac{\beta-\gamma}{1-\gamma})$ when $S^r_m(t) = 0$.

For each $m \in \mathcal{H}_u$, $\mathbf{D}_m(t) = \mathbf{S}_m(t)$.

$$D^l_m(t) = S^l_m(t), D^k_m(t) = 0,$$

and each process $D^r_m(t)$ is i.i.d. with $D^r_m(t) \sim \text{Bern}(\gamma I_{\{\eta_m(t)\neq 0\}})$ and is cou-

pled with $\mathbf{S}_m(t)$ in the following way: If $\eta_m(t) \neq 1$, $D_m^r(t) = S_m^r(t)$; if $\eta_m(t) = 1$, $D_m^r(t) = 0$ when $S_m^k(t) = 0$, and $D_m^r(t) \sim \text{Bern}(\frac{\beta-\gamma}{1-\beta})$ when $S_m^k(t) = 1$.

Ideal arrival process $\mathbf{F}(t)$: Ideally, $\forall \bar{L} \in \mathcal{L}^*_{\mathcal{H}_u}$, type-$\bar{L}$ tasks would join their local sub-queues at $\mathcal{H}_u$. So we re-distribute unwanted arrivals $\sum_{m:m\notin\bar{L},m\notin\mathcal{H}_u} A_{\bar{L},m}$ among its local servers at $\mathcal{H}_u$ evenly. Similarly, all tasks of types $\mathcal{L}_{\mathcal{H}_o}$ would be routed to their local sub-queues at $\mathcal{H}_o$. That is, unwanted arrivals $\sum_{m:m\notin\bar{L},m\notin\mathcal{H}_o} A_{\bar{L},m}$ would be re-distributed evenly among its local servers at $\mathcal{H}_o$. For any $\bar{L} \in \mathcal{L}_{\mathcal{B}_o}$, type-$\bar{L}$ tasks would only join their local sub-queues at $\mathcal{B}_o$, or rack-local sub-queues at $\mathcal{H}_o$, or remote sub-queues at $\mathcal{H}_u$. Hence we re-distribute unwanted arrivals that are routed to other sub-queues evenly among its local servers at $\mathcal{B}_o$. Then we can rewrite the dynamics of $\tilde{\mathbf{Q}}$ as

$$\tilde{\mathbf{Q}}(t+1) = \tilde{\mathbf{Q}}(t) + \tilde{\mathbf{F}}(t) - \tilde{\mathbf{D}}(t) + \tilde{\mathbf{V}}(t),$$

where $\tilde{\mathbf{V}}(t) = \tilde{\mathbf{A}}(t)(t) - \tilde{\mathbf{F}}(t) + \tilde{\mathbf{D}}(t) - \tilde{\mathbf{S}}(t) + \tilde{\mathbf{U}}(t)$.

In steady state, we have

$$\begin{aligned} & 2\mathbb{E}\left[\langle \mathbf{c}, \tilde{\mathbf{Q}}(t)\rangle \langle \mathbf{c}, \tilde{\mathbf{D}}(t) - \tilde{\mathbf{F}}(t)\rangle\right] && \text{(B.28)} \\ = \; & \mathbb{E}\left[\langle \mathbf{c}, \tilde{\mathbf{F}}(t) - \tilde{\mathbf{D}}(t)\rangle^2\right] + \mathbb{E}\left[\langle \mathbf{c}, \tilde{\mathbf{V}}(t)\rangle^2\right] && \text{(B.29)} \\ & +2\mathbb{E}\left[\langle \mathbf{c}, \tilde{\mathbf{Q}}(t) + \tilde{\mathbf{F}}(t) - \tilde{\mathbf{D}}(t)\rangle \langle \mathbf{c}, \tilde{\mathbf{V}}(t)\rangle\right]. && \text{(B.30)} \end{aligned}$$

Note that

$$\begin{aligned} \Psi^{(\epsilon)} \; &\leq \; \sum_{m\in\mathcal{H}_u} \gamma \frac{Q_m^r}{\gamma} + \sum_{m\in\mathcal{H}_o} \beta\left(\frac{Q_m^k}{\beta} + \frac{Q_m^r}{\gamma}\right) + \sum_{m\in\mathcal{B}_o} \alpha\left(\frac{Q_m^l}{\alpha} + \frac{Q_m^k}{\beta} + \frac{Q_m^r}{\gamma}\right) \\ &= \; ||\tilde{\mathbf{c}}||\langle \mathbf{c}, \tilde{\mathbf{Q}}\rangle. \end{aligned}$$

An upper bound on $\mathbb{E}\left[\langle \mathbf{c}, \tilde{\mathbf{Q}}(t)\rangle\right]$ can be obtained by bounding each of the above terms, which gives an upper bound on $\mathbb{E}\left[\Psi^{(\epsilon)}(t)\right]$.

For convenience, we temporarily omit the superscript $^{(\epsilon)}$. We study each term in (B.28)-(B.30).

According to the definition of ideal arrival processes,

$$\langle \tilde{\mathbf{c}}, \tilde{\mathbf{F}}(t)\rangle \; = \; \sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}_o}} A_{\bar{L}}(t).$$

Hence

$$\mathbb{E}\left[\langle\tilde{\mathbf{c}},\tilde{\mathbf{F}}(t)\rangle\right]=\sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}_o}}\lambda_{\bar{L}},\quad \mathrm{Var}\left[\langle\tilde{\mathbf{c}},\tilde{\mathbf{F}}(t)\rangle\right]=(\sigma^{(\epsilon)})^2.$$

By the definition of ideal service processes,

$$\langle\tilde{\mathbf{c}},\tilde{\mathbf{D}}(t)\rangle \;=\; \sum_{m\in\mathcal{B}_o}\alpha\cdot\frac{D_m^l(t)}{\alpha}+\sum_{m\in\mathcal{H}_o}\beta\cdot\frac{D_m^k(t)}{\beta}+\sum_{m\in\mathcal{H}_u}\gamma\cdot\frac{D_m^r(t)}{\gamma}.$$

For each server m, we denote by ρ_m^l the proportion of time it spends on serving local sub-queue in steady state. Then

$$\begin{aligned}
\mathbb{E}\left[\langle\tilde{\mathbf{c}},\tilde{\mathbf{D}}(t)\rangle\right] &= \alpha M_{\mathcal{B}_o}+\sum_{m\in\mathcal{H}_o}\beta(1-\rho_m^l)+\sum_{m\in\mathcal{H}_u}\gamma(1-\rho_m^l),\\
\mathrm{Var}\left[\langle\tilde{\mathbf{c}},\tilde{\mathbf{D}}(t)\rangle\right] &= \alpha(1-\alpha)M_{\mathcal{B}_o}+\sum_{m\in\mathcal{H}_o}\beta(1-\rho_m^l)[1-\beta(1-\rho_m^l)]\\
&\quad+\sum_{m\in\mathcal{H}_u}\gamma(1-\rho_m^l)[1-\gamma(1-\rho_m^l)]\\
&= (\nu^{(\epsilon)})^2.
\end{aligned}$$

It is easy to verify that

$$\mathbb{E}\left[\langle\tilde{\mathbf{c}},\tilde{\mathbf{D}}(t)\rangle\right]-\mathbb{E}\left[\langle\tilde{\mathbf{c}},\tilde{\mathbf{F}}(t)\rangle\right]=\epsilon+\beta(\Phi_{H_o}-\sum_{m\in\mathcal{H}_o}\rho_m^l)+\gamma(\Phi_{H_u}-\sum_{m\in\mathcal{H}_u}\rho_m^l)=\epsilon+\delta,$$

where $\delta=\beta(\Phi_{H_o}-\sum_{m\in\mathcal{H}_o}\rho_m^l)+\gamma(\Phi_{H_u}-\sum_{m\in\mathcal{H}_u}\rho_m^l)\geq 0$, and $\delta\to 0$ as $\epsilon\downarrow 0$. Therefore,

$$\begin{aligned}
\mathbb{E}\left[\langle\mathbf{c},\tilde{\mathbf{Q}}(t)\rangle\langle\mathbf{c},\tilde{\mathbf{D}}(t)-\tilde{\mathbf{F}}(t)\rangle\right] &= \frac{1}{||\tilde{\mathbf{c}}||}\mathbb{E}\left[\langle\mathbf{c},\tilde{\mathbf{Q}}(t)\rangle\left(\langle\tilde{\mathbf{c}},\tilde{\mathbf{D}}(t)\rangle-\langle\tilde{\mathbf{c}},\tilde{\mathbf{F}}(t)\rangle\right)\right]\\
&= \frac{\epsilon+\delta}{||\tilde{\mathbf{c}}||}\mathbb{E}\left[\langle\mathbf{c},\tilde{\mathbf{Q}}(t)\rangle\right]. \qquad \text{(B.31)}
\end{aligned}$$

For the first term in (B.29), we have

$$\begin{aligned}
&\mathbb{E}\left[\langle\mathbf{c},\tilde{\mathbf{F}}(t)-\tilde{\mathbf{D}}(t)\rangle^2\right]\\
=\;& \frac{1}{||\tilde{\mathbf{c}}||^2}\left\{\mathrm{Var}\left[\langle\tilde{\mathbf{c}},\tilde{\mathbf{D}}(t)\rangle\right]+\mathrm{Var}\left[\langle\tilde{\mathbf{c}},\tilde{\mathbf{D}}(t)\rangle\right]+\left(\mathbb{E}\left[\langle\mathbf{c},\tilde{\mathbf{F}}(t)-\tilde{\mathbf{D}}(t)\rangle\right]\right)^2\right\}\\
=\;& \frac{1}{||\tilde{\mathbf{c}}||^2}\left[(\sigma^{(\epsilon)})^2+(\nu^{(\epsilon)})^2+(\epsilon+\delta)^2\right]. \qquad \text{(B.32)}
\end{aligned}$$

The following lemma provides an upper bound on the second term in (B.29).

Lemma B.23.

$$\mathbb{E}\left[\langle \mathbf{c}, \tilde{\mathbf{V}}(t)\rangle^2\right] \leq C\epsilon, \tag{B.33}$$

where C is a constant not depending on ϵ.

Next we will bound the term (B.30). We consider the system in steady state, which yields

$$\mathbb{E}\left[\langle \mathbf{c}, \tilde{\mathbf{F}}(t) - \tilde{\mathbf{D}}(t) + \tilde{\mathbf{V}}(t)\rangle\right] = \mathbb{E}\left[\langle \mathbf{c}, \tilde{\mathbf{Q}}(t+1) - \tilde{\mathbf{Q}}(t)\rangle\right] = 0.$$

Hence

$$\mathbb{E}\left[\langle \mathbf{c}, \tilde{\mathbf{V}}(t)\rangle\right] = \mathbb{E}\left[\langle \mathbf{c}, \tilde{\mathbf{F}}(t) - \tilde{\mathbf{D}}(t)\rangle\right] = \frac{\epsilon + \delta}{\|\tilde{\mathbf{c}}\|}.$$

Thus

$$\mathbb{E}\left[\langle \mathbf{c}, \tilde{\mathbf{F}}(t) - \tilde{\mathbf{D}}(t)\rangle\langle \mathbf{c}, \tilde{\mathbf{V}}(t)\rangle\right] \;\leq\; \mathbb{E}\left[\langle \mathbf{c}, \tilde{\mathbf{F}}(t)\rangle\langle \mathbf{c}, \tilde{\mathbf{V}}(t)\rangle\right] \leq \frac{C_A(\epsilon+\delta)}{\|\tilde{\mathbf{c}}\|^2}.$$

Together, we have

$$\mathbb{E}\left[\langle \mathbf{c}, \tilde{\mathbf{Q}}(t) + \tilde{\mathbf{F}}(t) - \tilde{\mathbf{D}}(t)\rangle\langle \mathbf{c}, \tilde{\mathbf{V}}(t)\rangle\right] \leq \mathbb{E}\left[\langle \mathbf{c}, \tilde{\mathbf{Q}}(t)\rangle\langle \mathbf{c}, \tilde{\mathbf{V}}(t)\rangle\right] + \frac{C_A(\epsilon+\delta)}{\|\tilde{\mathbf{c}}\|^2}.$$

We can rewrite the term $\langle \mathbf{c}, \tilde{\mathbf{Q}}(t)\rangle\langle \mathbf{c}, \tilde{\mathbf{V}}(t)\rangle$ as

$$\begin{aligned}
& \langle \mathbf{c}, \tilde{\mathbf{Q}}(t)\rangle\langle \mathbf{c}, \tilde{\mathbf{V}}(t)\rangle \\
= \;& \langle \tilde{\mathbf{Q}}(t), \tilde{\mathbf{V}}(t)\rangle - \langle \tilde{\mathbf{Q}}_\perp(t), \tilde{\mathbf{V}}_\perp(t)\rangle \\
= \;& \langle \tilde{\mathbf{Q}}(t), \tilde{\mathbf{D}}(t) - \mathbf{S}(t)\rangle + \langle \tilde{\mathbf{Q}}(t), \mathbf{A}(t) - \tilde{\mathbf{F}}(t)\rangle && \text{(B.34)} \\
& + \langle \tilde{\mathbf{Q}}(t), \tilde{\mathbf{V}}(t)\rangle - \langle \tilde{\mathbf{Q}}_\perp(t), \tilde{\mathbf{V}}_\perp(t)\rangle. && \text{(B.35)}
\end{aligned}$$

The following two lemmas bound the first two terms in (B.34).

Lemma B.24.

$$\mathbb{E}\left[\langle \tilde{\mathbf{Q}}(t), \tilde{\mathbf{D}}(t) - \tilde{\mathbf{S}}(t)\rangle\right] = 0.$$

Lemma B.25.

$$\mathbb{E}\left[\langle \tilde{\mathbf{Q}}(t), \tilde{\mathbf{A}}(t) - \tilde{\mathbf{F}}(t)\rangle\right] = o(\epsilon).$$

The first term in (B.35) is equal to zero by Lemma B.7. To bound the second term in (B.35), we first provide a bound on $\mathbb{E}\left[\left\|\tilde{\mathbf{V}}(t)\right\|^2\right]$. By Lemma B.23, we can show that

$$\mathbb{E}\left[\left\|\tilde{\mathbf{V}}(t)\right\|^2\right] \leq R\epsilon,$$

where R is a constant independent of ϵ. Again we will use state space collapse result to bound $-\langle\tilde{\mathbf{Q}}_\perp(t), \tilde{\mathbf{V}}_\perp(t)\rangle$. By Cauchy-Schwarz inequality, we have

$$\begin{aligned}\mathbb{E}\left[-\langle\tilde{\mathbf{Q}}_\perp(t), \tilde{\mathbf{V}}_\perp(t)\rangle\right] &\leq \sqrt{\mathbb{E}\left[\left\|\tilde{\mathbf{Q}}_\perp(t)\right\|^2\right]\mathbb{E}\left[\left\|\tilde{\mathbf{V}}_\perp(t)\right\|^2\right]} \\ &\leq \sqrt{C_2' R\epsilon}.\end{aligned}$$

Utilizing the above inequalities yields the following bound on the term (B.30):

$$\mathbb{E}\left[\langle\mathbf{c}, \tilde{\mathbf{Q}}(t) + \tilde{\mathbf{F}}(t) - \tilde{\mathbf{D}}(t)\rangle\langle\mathbf{c}, \tilde{\mathbf{V}}(t)\rangle\right] \leq \frac{C_A(\epsilon+\delta)}{\left\|\tilde{\mathbf{c}}\right\|^2}. + \sqrt{C_2' R\epsilon} + o(\epsilon). \quad \text{(B.36)}$$

We now reintroduce the superscript $^{(\epsilon)}$. Substituting (B.31)-(B.33) and (B.36) in (B.28)-(B.30) yields

$$\begin{aligned}&2\frac{\epsilon+\delta}{||\tilde{\mathbf{c}}||}\mathbb{E}\left[\langle\mathbf{c}, \tilde{\mathbf{Q}}(t)\rangle\right] \\ \leq\ & \frac{1}{||\tilde{\mathbf{c}}||^2}\left[(\sigma^{(\epsilon)})^2 + (\nu^{(\epsilon)})^2 + (\epsilon+\delta)^2\right] + C\epsilon + \frac{2C_A(\epsilon+\delta)}{\left\|\tilde{\mathbf{c}}\right\|^2}. + 2\sqrt{C_2' R\epsilon} + 2o(\epsilon).\end{aligned}$$

Since $\delta \geq 0$, we have

$$\|\tilde{\mathbf{c}}\|\mathbb{E}\left[\langle\mathbf{c}, \tilde{\mathbf{Q}}(t)\rangle\right] \leq \frac{(\sigma^{(\epsilon)})^2 + (\nu^{(\epsilon)})^2}{2(\epsilon+\delta)} + B^{(\epsilon)} \leq \frac{(\sigma^{(\epsilon)})^2 + (\nu^{(\epsilon)})^2}{2\epsilon} + B^{(\epsilon)},$$

where $B^{(\epsilon)} = \frac{C}{2}\frac{\epsilon}{\epsilon+\delta}\|\tilde{\mathbf{c}}\| + C_A + \left\|\tilde{\mathbf{c}}\right\|^2\sqrt{C_2' R}\frac{\sqrt{\epsilon}}{\epsilon+\delta} + o(1)$, which is $o(1/\epsilon)$. Since

$$\begin{aligned}\Psi^{(\epsilon)} &= \sum_{m\in\mathcal{B}_o}(Q_m^{l(\epsilon)} + Q_m^{k(\epsilon)} + Q_m^{r(\epsilon)}) + \sum_{m\in\mathcal{H}_o}(Q_m^{k(\epsilon)} + Q_m^{r(\epsilon)}) + \sum_{m\in\mathcal{H}_u} Q_m^{r(\epsilon)} \\ &\leq ||\tilde{\mathbf{c}}||\langle\mathbf{c}, \tilde{\mathbf{Q}}\rangle,\end{aligned}$$

we have

$$\mathbb{E}\left[\Psi^{(\epsilon)}\right] \leq \frac{(\sigma^{(\epsilon)})^2 + (\nu^{(\epsilon)})^2 + (\epsilon+\delta)^2}{2\epsilon} + B^{(\epsilon)}.$$

Taking the limit as $\epsilon \to 0^+$ gives the result (4.7). Observe that

$$\begin{aligned} & \mathbb{E}\left[\sum_m (Q_m^{l(\epsilon)} + Q_m^{k(\epsilon)} + Q_m^{r(\epsilon)})\right] \\ = \; & \mathbb{E}\left[\sum_{m\in\mathcal{H}_u} (Q_m^{l(\epsilon)} + Q_m^{k(\epsilon)}) + \sum_{m\in\mathcal{H}_o} Q_m^{l(\epsilon)}\right] + \mathbb{E}\left[\Psi^{(\epsilon)}\right]. \end{aligned}$$

We have established the coincidence of lower and upper bounds for $\epsilon\mathbb{E}\left[\Psi^{(\epsilon)}\right]$. Then the heavy-traffic optimality of the proposed algorithm follows by Theorem 4.4.

B.5.4 Proof of Lemmas B.23-B.25

Proof of Lemma B.23. We will show that

$$\mathbb{E}\left[\left\|\tilde{\mathbf{A}} - \tilde{\mathbf{F}}\right\|^2\right] \le C_1\epsilon,\ \mathbb{E}\left[\left\|\tilde{\mathbf{D}} - \tilde{\mathbf{S}}\right\|^2\right] \le C_2\epsilon,\ \mathbb{E}\left[\left\|\tilde{\mathbf{U}}\right\|^2\right] \le C_3\epsilon,$$

where C_1, C_2, C_3 are constants independent of ϵ. Since $\tilde{\mathbf{V}}(t) = \tilde{\mathbf{A}}(t) - \tilde{\mathbf{F}}(t) + \tilde{\mathbf{D}}(t) - \tilde{\mathbf{S}}(t) + \tilde{\mathbf{U}}(t)$, we have

$$\begin{aligned} \mathbb{E}\left[\left\|\tilde{\mathbf{V}}(t)\right\|^2\right] &= \mathbb{E}\left[\left\|\tilde{\mathbf{A}}(t) - \tilde{\mathbf{F}}(t) + \tilde{\mathbf{D}}(t) - \tilde{\mathbf{S}}(t) + \tilde{\mathbf{U}}(t)\right\|^2\right] \\ &\le 2\mathbb{E}\left[\left\|\tilde{\mathbf{A}}(t) - \tilde{\mathbf{F}}(t)\right\|^2 + \left\|\tilde{\mathbf{D}}(t) - \tilde{\mathbf{S}}(t)\right\|^2 + \left\|\tilde{\mathbf{U}}(t)\right\|^2\right] \\ &\le 2(C_1 + C_2 + C_3)\epsilon. \end{aligned}$$

In order to achieve maximum throughput, $\forall \bar{L} \in \mathcal{L}^*_{\mathcal{H}_u}$, type-$\bar{L}$ tasks would join their local sub-queues at $\mathcal{H}_u$ ideally. Let $A^l_{\mathcal{H}_u\mathcal{C}}$, $A^k_{\mathcal{H}_u\mathcal{C}}$ and $A^r_{\mathcal{H}_u\mathcal{C}}$ denote the amount of tasks from $\mathcal{L}^*_{\mathcal{H}_u}$ that are routed to local, rack-local and remote sub-queues at $\mathcal{C}$ respectively, where $\mathcal{C} \in \{\mathcal{H}_u, \mathcal{H}_o, \mathcal{B}_o\}$. Thus ideally $A^l_{\mathcal{H}_u\mathcal{H}_u} \ge 0$, and all others are zero. We call arrival types that are supposed to be zero as *unwanted arrivals*. Similarly, all tasks of types $\mathcal{L}_{\mathcal{H}_o}$ would be routed to their local sub-queues at $\mathcal{H}_o$ ideally. We denote by $A^l_{\mathcal{H}_o\mathcal{C}}$, $A^k_{\mathcal{H}_o\mathcal{C}}$ and $A^r_{\mathcal{H}_o\mathcal{C}}$ the amount of tasks from $\mathcal{L}_{\mathcal{H}_o}$ that are routed to local, rack-local and remote sub-queues at $\mathcal{C}$ respectively, where $\mathcal{C} \in \{\mathcal{H}_u, \mathcal{H}_o, \mathcal{B}_o\}$. All these arrivals except $A^l_{\mathcal{H}_o\mathcal{H}_o}$ are *unwanted*. For any $\bar{L} \in \mathcal{L}_{\mathcal{B}_o}$, type-$\bar{L}$ tasks would only join their local sub-queues at $\mathcal{B}_o$, or rack-local sub-queues at $\mathcal{H}_o$, or remote sub-queues

at $\mathcal{H}_u$. As task types $\mathcal{L}_{\mathcal{B}_o}$ are only local to servers in $\mathcal{B}_o$, *unwanted arrivals* include $A^k_{\mathcal{B}_o\mathcal{B}_o}$, $A^r_{\mathcal{B}_o\mathcal{H}_o}$ and $A^r_{\mathcal{B}_o\mathcal{B}_o}$.

First we show that the amount of *unwanted arrivals* is upper bounded by $C\epsilon$, where C is a constant not depending on ϵ. The expected load on $\mathcal{B}_o$ is given by

$$\begin{aligned}\Upsilon_{\mathcal{B}_o} &= \mathbb{E}\left[\frac{1}{\alpha}\left(A^l_{\mathcal{B}_o\mathcal{B}_o}+A^l_{\mathcal{H}_o\mathcal{B}_o}+A^l_{\mathcal{H}_u\mathcal{B}_o}\right)+\frac{1}{\beta}\left(A^k_{\mathcal{B}_o\mathcal{B}_o}+A^k_{\mathcal{H}_o\mathcal{B}_o}+A^k_{\mathcal{H}_u\mathcal{B}_o}\right)\right.\\ &\quad\left.+\frac{1}{\gamma}\left(A^r_{\mathcal{B}_o\mathcal{B}_o}+A^r_{\mathcal{H}_o\mathcal{B}_o}+A^r_{\mathcal{H}_u\mathcal{B}_o}\right)\right].\end{aligned}$$

Since the system is stable, $\Upsilon_{\mathcal{B}_o} < M_{\mathcal{B}_o}$. Similarly, the expected load on $\mathcal{H}_o$ satisfies

$$\begin{aligned}\Upsilon_{\mathcal{H}_o} &= \mathbb{E}\left[\frac{1}{\alpha}\left(A^l_{\mathcal{H}_o\mathcal{H}_o}+A^l_{\mathcal{H}_u\mathcal{H}_o}\right)+\frac{1}{\beta}\left(A^k_{\mathcal{B}_o\mathcal{H}_o}+A^k_{\mathcal{H}_o\mathcal{H}_o}+A^k_{\mathcal{H}_u\mathcal{H}_o}\right)\right.\\ &\quad\left.+\frac{1}{\gamma}\left(A^r_{\mathcal{B}_o\mathcal{H}_o}+A^r_{\mathcal{H}_o\mathcal{H}_o}+A^r_{\mathcal{H}_u\mathcal{H}_o}\right)\right]\\ &< M_{\mathcal{H}_o}.\end{aligned}$$

The expected load on $\mathcal{H}_u$ satisfies

$$\Upsilon_{\mathcal{H}_u} = \mathbb{E}\left[\frac{A^l_{\mathcal{H}_u\mathcal{H}_u}}{\alpha}+\frac{A^k_{\mathcal{H}_u\mathcal{H}_u}}{\beta}+\frac{1}{\gamma}\left(A^r_{\mathcal{B}_o\mathcal{H}_u}+A^r_{\mathcal{H}_o\mathcal{H}_u}+A^r_{\mathcal{H}_u\mathcal{H}_u}\right)\right] < M_{\mathcal{H}_u}.$$

Therefore,

$$\alpha\Upsilon_{\mathcal{B}_o}+\beta\Upsilon_{\mathcal{H}_o}+\gamma\Upsilon_{\mathcal{H}_u} < \alpha M_{\mathcal{B}_o}+\beta M_{\mathcal{H}_o}+\gamma M_{\mathcal{H}_u}. \tag{B.37}$$

LHS of (B.37) can be written as

$$
\begin{aligned}
&\mathbb{E}\left[A^l_{\mathcal{B}_o\mathcal{B}_o}+A^k_{\mathcal{B}_o\mathcal{H}_o}+A^r_{\mathcal{B}_o\mathcal{H}_u}+\frac{\alpha}{\beta}A^k_{\mathcal{B}_o\mathcal{B}_o}+\frac{\alpha}{\gamma}A^r_{\mathcal{B}_o\mathcal{B}_o}+\frac{\beta}{\gamma}A^r_{\mathcal{B}_o\mathcal{H}_o}\right]\\
&+\mathbb{E}\left[\frac{\beta}{\alpha}A^l_{\mathcal{H}_o\mathcal{H}_o}+A^l_{\mathcal{H}_o\mathcal{B}_o}+\frac{\alpha}{\beta}A^k_{\mathcal{H}_o\mathcal{B}_o}+A^k_{\mathcal{H}_o\mathcal{H}_o}\right]\\
&+\mathbb{E}\left[\frac{\beta}{\gamma}A^r_{\mathcal{H}_o\mathcal{H}_o}+\frac{\alpha}{\gamma}A^r_{\mathcal{H}_o\mathcal{B}_o}+A^r_{\mathcal{H}_o\mathcal{H}_u}\right]\\
&+\mathbb{E}\left[\frac{\gamma}{\alpha}A^l_{\mathcal{H}_u\mathcal{H}_u}+\frac{\beta}{\alpha}A^l_{\mathcal{H}_u\mathcal{H}_o}+A^l_{\mathcal{H}_u\mathcal{B}_o}+\frac{\alpha}{\beta}A^k_{\mathcal{H}_u\mathcal{B}_o}+A^k_{\mathcal{H}_u\mathcal{H}_o}\right]\\
&+\mathbb{E}\left[\frac{\gamma}{\beta}A^k_{\mathcal{H}_u\mathcal{H}_u}+\frac{\alpha}{\gamma}A^r_{\mathcal{H}_u\mathcal{B}_o}+\frac{\beta}{\gamma}A^r_{\mathcal{H}_u\mathcal{H}_o}+A^r_{\mathcal{H}_u\mathcal{H}_u}\right]\\
=\;&\sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}_o}}\lambda_{\bar{L}}+\mathbb{E}\left[\frac{\alpha-\beta}{\beta}A^k_{\mathcal{B}_o\mathcal{B}_o}+\frac{\alpha-\gamma}{\gamma}A^r_{\mathcal{B}_o\mathcal{B}_o}+\frac{\beta-\gamma}{\gamma}A^r_{\mathcal{B}_o\mathcal{H}_o}\right]\\
&+\frac{\beta}{\alpha}\sum_{\bar{L}\in\mathcal{L}_{\mathcal{H}_o}}\lambda_{\bar{L}}+\mathbb{E}\left[\frac{\alpha-\beta}{\alpha}A^l_{\mathcal{H}_o\mathcal{B}_o}+\frac{\alpha^2-\beta^2}{\alpha\beta}A^k_{\mathcal{H}_o\mathcal{B}_o}+\frac{\alpha-\beta}{\alpha}A^k_{\mathcal{H}_o\mathcal{H}_o}\right]\\
&+\mathbb{E}\left[\frac{(\alpha-\gamma)\beta}{\alpha\gamma}A^r_{\mathcal{H}_o\mathcal{H}_o}+\frac{\alpha^2-\beta\gamma}{\alpha\gamma}A^r_{\mathcal{H}_o\mathcal{B}_o}+\frac{\alpha-\beta}{\alpha}A^r_{\mathcal{H}_o\mathcal{H}_u}\right]\\
&+\frac{\gamma}{\alpha}\sum_{\bar{L}\in\mathcal{L}^*_{\mathcal{H}_u}}\lambda_{\bar{L}}+\mathbb{E}\left[\frac{\beta-\gamma}{\alpha}A^l_{\mathcal{H}_u\mathcal{H}_o}+\frac{\alpha-\gamma}{\alpha}A^l_{\mathcal{H}_u\mathcal{B}_o}+\frac{\alpha^2-\beta\gamma}{\alpha\beta}A^k_{\mathcal{H}_u\mathcal{H}_o}+\frac{\alpha-\gamma}{\alpha}A^k_{\mathcal{H}_u\mathcal{H}_o}\right]\\
&+\mathbb{E}\left[\frac{(\alpha-\beta)\gamma}{\alpha\beta}A^k_{\mathcal{H}_u\mathcal{H}_u}+\frac{\alpha^2-\gamma^2}{\alpha\gamma}A^r_{\mathcal{H}_u\mathcal{B}_o}+\frac{\alpha\beta-\gamma^2}{\alpha\gamma}A^r_{\mathcal{H}_u\mathcal{H}_o}+\frac{\alpha-\gamma}{\alpha}A^r_{\mathcal{H}_u\mathcal{H}_u}\right]. \quad \text{(B.38)}
\end{aligned}
$$

Since

$$\sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}_o}}\lambda_{\bar{L}}+\frac{\beta}{\alpha}\sum_{\bar{L}\in\mathcal{L}_{\mathcal{H}_o}}\lambda_{\bar{L}}+\frac{\gamma}{\alpha}\sum_{\bar{L}\in\mathcal{L}^*_{\mathcal{H}_u}}\lambda_{\bar{L}}=\alpha M_{\mathcal{B}_o}+\beta M_{\mathcal{H}_o}+\gamma M_{\mathcal{H}_u}-\epsilon,$$

and the coefficient for each unwanted arrival term in (B.38) is positive, each unwanted arrival term can be upper bounded by $C\epsilon$, where C is a constant only depending on α,β and γ. By the definition of ideal arrival processes, $\mathbb{E}\left[\left\|\tilde{\mathbf{A}}-\tilde{\mathbf{F}}\right\|_1\right]$ is a linear function of the *unwanted arrival* terms in (B.38). Thus

$$\mathbb{E}\left[\left\|\tilde{\mathbf{A}}-\tilde{\mathbf{F}}\right\|^2\right]\le C_A\mathbb{E}\left[\left\|\tilde{\mathbf{A}}-\tilde{\mathbf{F}}\right\|_1\right]\le C_1\epsilon,$$

where $C_1>0$ is a constant independent of ϵ.

As we consider the system in steady state, for any $m \in \mathcal{M}$,

$$\mathbb{E}\left[A_m(t) - S_m(t) + \tilde{U}_m(t)\right] = \mathbb{E}\left[W_m(t+1) - W_m(t)\right] = 0,$$

i.e., $\mathbb{E}\left[S_m(t) - \tilde{U}_m(t)\right] = \mathbb{E}\left[A_m(t)\right]$. Then we have

$$\begin{aligned}
& \mathbb{E}\left[\sum_{m\in\mathcal{B}_o} \alpha(S_m - \tilde{U}_m) + \sum_{m\in\mathcal{H}_o} \beta(S_m - \tilde{U}_m) + \sum_{m\in\mathcal{H}_u} \gamma(S_m - \tilde{U}_m)\right] \\
= \; & \mathbb{E}\left[\sum_{m\in\mathcal{B}_o} \alpha A_m + \sum_{m\in\mathcal{H}_o} \beta A_m + \sum_{m\in\mathcal{H}_u} \gamma A_m\right] \\
\geq \; & \sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}_o}} \lambda_{\bar{L}} + \frac{\beta}{\alpha} \sum_{\bar{L}\in\mathcal{L}_{\mathcal{H}_o}} \lambda_{\bar{L}} + \frac{\gamma}{\alpha} \sum_{\bar{L}\in\mathcal{L}^*_{\mathcal{H}_u}} \lambda_{\bar{L}}.
\end{aligned}$$

Consequently,

$$\begin{aligned}
& \mathbb{E}\left[\sum_{m\in\mathcal{B}_o} \alpha\tilde{U}_m + \sum_{m\in\mathcal{H}_o} \beta\tilde{U}_m + \sum_{m\in\mathcal{H}_u} \tilde{U}_m\right] \\
\leq \; & \alpha M_{\mathcal{B}_o} + \beta M_{\mathcal{H}_o} + \gamma M_{\mathcal{H}_u} - \left(\sum_{\bar{L}\in\mathcal{L}_{\mathcal{B}_o}} \lambda_{\bar{L}} + \frac{\beta}{\alpha} \sum_{\bar{L}\in\mathcal{L}_{\mathcal{H}_o}} \lambda_{\bar{L}} + \frac{\gamma}{\alpha} \sum_{\bar{L}\in\mathcal{L}^*_{\mathcal{H}_u}} \lambda_{\bar{L}}\right) \\
= \; & \epsilon.
\end{aligned}$$

As $0 \leq \tilde{U}_m(t) \leq \frac{1}{\gamma}$,

$$\begin{aligned}
\mathbb{E}\left[\left\|\tilde{\mathbf{U}}(t)\right\|^2\right] &\leq \frac{1}{\gamma}\mathbb{E}\left[\sum_m \tilde{U}_m(t)\right] \\
&\leq \frac{1}{\gamma^2}\mathbb{E}\left[\sum_{m\in\mathcal{B}_o} \alpha\tilde{U}_m(t) + \sum_{m\in\mathcal{H}_o} \beta\tilde{U}_m(t) + \sum_{m\in\mathcal{H}_u} \tilde{U}_m(t)\right] \\
&\leq \frac{\epsilon}{\gamma^2}.
\end{aligned}$$

Next we will focus on $\mathbb{E}\left[\left\|\tilde{\mathbf{D}}(t)-\tilde{\mathbf{S}}(t)\right\|^2\right]$. By the definition of $\tilde{\mathbf{D}}(t)$,

$$\begin{aligned}
&\mathbb{E}\left[\left\|\tilde{\mathbf{D}}(t)-\tilde{\mathbf{S}}(t)\right\|^2\right]\\
=\;&\mathbb{E}\left[\sum_{m\in\mathcal{B}_o}\left\|\frac{D_m^l}{\alpha}-\frac{S_m^l}{\alpha}-\frac{S_m^k}{\beta}-\frac{S_m^r}{\gamma}\right\|^2\right]\\
&+\mathbb{E}\left[\sum_{m\in\mathcal{H}_o}\left\|\frac{D_m^k}{\beta}+\frac{D_m^r}{\gamma}-\frac{S_m^k}{\beta}-\frac{S_m^r}{\gamma}\right\|^2+\sum_{m\in\mathcal{H}_u}\left\|\frac{D_m^r}{\gamma}-\frac{S_m^r}{\gamma}\right\|^2\right]\\
=\;&\mathbb{E}\left[\sum_{m\in\mathcal{B}_o}\left(\left\|\frac{D_m^l}{\alpha}-\frac{S_m^k}{\beta}\right\|^2 I_{\{\eta_m(t)=1\}}+\left\|\frac{D_m^l}{\alpha}-\frac{S_m^r}{\gamma}\right\|^2 I_{\{\eta_m(t)=2\}}\right)\right]\\
&+\mathbb{E}\left[\sum_{m\in\mathcal{H}_o}\left\|\frac{D_m^k}{\beta}-\frac{S_m^r}{\gamma}\right\|^2 I_{\{\eta_m(t)=2\}}\right]+\mathbb{E}\left[\sum_{m\in\mathcal{H}_u}\left\|\frac{D_m^r}{\gamma}\right\|^2 I_{\{\eta_m(t)=1\}}\right]\\
\leq\;&\frac{1}{\gamma^2}\mathbb{E}\left[\sum_{m\in\mathcal{B}_o}\left(I_{\{\eta_m(t)=1\}}+I_{\{\eta_m(t)=2\}}\right)\right]+\frac{1}{\gamma^2}\mathbb{E}\left[\sum_{m\in\mathcal{H}_o}I_{\{\eta_m(t)=2\}}\right]\\
&+\frac{1}{\gamma^2}\mathbb{E}\left[\sum_{m\in\mathcal{H}_u}I_{\{\eta_m(t)=1\}}\right].
\end{aligned}$$

Again as we consider the system in steady state, for any m,

$$\begin{aligned}
&\mathbb{E}\left[A_m^l(t)-S_m^l(t)\right]=\mathbb{E}\left[Q_m^l(t+1)-Q_m^l(t)\right]=0,\\
&\mathbb{E}\left[A_m^k(t)-S_m^k(t)\right]=\mathbb{E}\left[Q_m^k(t+1)-Q_m^k(t)\right]=0,\\
&\mathbb{E}\left[A_m^r(t)-S_m^r(t)+U_m(t)\right]=\mathbb{E}\left[Q_m^r(t+1)-Q_m^r(t)\right]=0.
\end{aligned}$$

That is,

$$\begin{aligned}
&\mathbb{E}\left[S_m^l(t)\right]=\alpha\mathbb{E}\left[I_{\{\eta_m(t)=0\}}\right]=\mathbb{E}\left[A_m^l(t)\right],\\
&\mathbb{E}\left[S_m^k(t)\right]=\beta\mathbb{E}\left[I_{\{\eta_m(t)=1\}}\right]=\mathbb{E}\left[A_m^k(t)\right],\\
&\mathbb{E}\left[S_m^r(t)\right]=\gamma\mathbb{E}\left[I_{\{\eta_m(t)=2\}}\right]=\mathbb{E}\left[A_m^r(t)+U_m(t)\right].
\end{aligned}$$

Thus

$$\begin{aligned}
&\mathbb{E}\left[\sum_{m\in\mathcal{B}_o}\left(I_{\{\eta_m(t)=1\}}+I_{\{\eta_m(t)=2\}}\right)\right]\\
=\;&\frac{1}{\beta}\mathbb{E}\left[\sum_{m\in\mathcal{B}_o}A_m^k(t)\right]+\frac{1}{\gamma}\mathbb{E}\left[\sum_{m\in\mathcal{B}_o}A_m^r(t)\right]+\frac{1}{\gamma}\mathbb{E}\left[\sum_{m\in\mathcal{B}_o}U_m^r(t)\right]\\
=\;&\mathbb{E}\left[\sum_{m\in\mathcal{B}_o}\frac{1}{\beta}\left(A^k_{\mathcal{B}_o\mathcal{B}_o}+A^k_{\mathcal{H}_o\mathcal{B}_o}+A^k_{\mathcal{H}_u\mathcal{B}_o}\right)+\frac{1}{\gamma}\left(A^r_{\mathcal{B}_o\mathcal{B}_o}+A^r_{\mathcal{H}_o\mathcal{B}_o}+A^r_{\mathcal{H}_u\mathcal{B}_o}\right)\right]\\
&+\frac{1}{\gamma}\mathbb{E}\left[\sum_{m\in\mathcal{B}_o}U_m^r(t)\right]\\
\leq\;&C\epsilon,
\end{aligned}$$

where C is a constant not depending on ϵ. Similarly,

$$\mathbb{E}\left[\sum_{m\in\mathcal{H}_o}I_{\{\eta_m(t)=2\}}\right]\leq C\epsilon,\quad \mathbb{E}\left[\sum_{m\in\mathcal{H}_u}I_{\{\eta_m(t)=1\}}\right]\leq C\epsilon.$$

Combining these inequalities yields:

$$\mathbb{E}\left[\left\|\tilde{\mathbf{D}}(t)-\tilde{\mathbf{S}}(t)\right\|^2\right]\leq C_2\epsilon.$$

■

Proof of Lemma B.24. The proof is similar to that of Lemma B.15.

Proof of Lemma B.25. By the definition of ideal arrival process $\mathbf{F}(t)$, for any $m\in\mathcal{B}_o$,

$$\begin{aligned}
F_m^l&=A_m^l+\sum_{\substack{\bar{L}:\bar{L}\in\mathcal{L}_{\mathcal{B}_o}\\ m\in\bar{L}}}\frac{1}{|\bar{L}|}\left(\sum_{\substack{n:n\in\mathcal{B}_o\\ n\in\bar{L}_k}}A_{\bar{L},n}+\sum_{\substack{n:n\in\mathcal{B}_o\\ n\in\bar{L}_r}}A_{\bar{L},n}+\sum_{\substack{n:n\in\mathcal{H}_o\\ n\in\bar{L}_r}}A_{\bar{L},n}\right)-\sum_{\substack{\bar{L}:\bar{L}\notin\mathcal{L}_{\mathcal{B}_o}\\ m\in\bar{L}}}A_{\bar{L},m},\\
F_m^k&=F_m^r=0.
\end{aligned}$$

For any $m \in \mathcal{H}_o$,

$$
\begin{aligned}
F_m^l &= A_m^l - \sum_{\substack{\bar{L}:\bar{L}\notin\mathcal{L}_{\mathcal{H}_o} \\ m\in\bar{L}}} A_{\bar{L},m} + \sum_{\substack{\bar{L}:\bar{L}\in\mathcal{L}_{\mathcal{H}_o} \\ m\in\bar{L}}} \frac{1}{|\bar{L}|} \sum_{n:n\notin\bar{L}\cap\mathcal{H}_o} A_{\bar{L},n}, \\
F_m^k &= A_m^k - \sum_{\substack{\bar{L}:\bar{L}\notin\mathcal{L}_{\mathcal{B}_o} \\ m\in\bar{L}_k}} A_{\bar{L},m}, \\
F_m^r &= 0.
\end{aligned}
$$

For any $m \in \mathcal{H}_u$,

$$
\begin{aligned}
F_m^l &= A_m^l + \sum_{\substack{\bar{L}:\bar{L}\in\mathcal{L}^*_{\mathcal{H}_o} \\ m\in\bar{L}}} \frac{1}{|\bar{L}|} \sum_{n:n\notin\bar{L}\cap\mathcal{H}_u} A_{\bar{L},n}, \\
F_m^k &= 0, \\
F_m^r &= A_m^r - \sum_{\substack{\bar{L}:\bar{L}\notin\mathcal{L}_{\mathcal{B}_o} \\ m\in\bar{L}_r}} A_{\bar{L},m}.
\end{aligned}
$$

We can write $\langle \tilde{\mathbf{Q}}(t), \tilde{\mathbf{A}}(t) - \tilde{\mathbf{F}}(t) \rangle$ as

$$\begin{aligned}
&\sum_{m\in\mathcal{B}_o} \tilde{Q}_m \left(\frac{A_m^l}{\alpha} + \frac{A_m^k}{\beta} + \frac{A_m^r}{\gamma} - \frac{F_m^l}{\alpha} - \frac{F_m^k}{\beta} - \frac{F_m^r}{\gamma} \right) \\
&+ \sum_{m\in\mathcal{H}_o} \tilde{Q}_m \left(\frac{A_m^k}{\beta} + \frac{A_m^r}{\gamma} - \frac{F_m^k}{\beta} - \frac{F_m^r}{\gamma} \right) + \sum_{m\in\mathcal{H}_u} \tilde{Q}_m \left(\frac{A_m^r}{\gamma} - \frac{F_m^r}{\gamma} \right) \\
= &\sum_{\bar{L}:\bar{L}\in\mathcal{L}_{\mathcal{B}_o}} \Bigg[\sum_{\substack{n:n\in\mathcal{B}_o \\ n\in\bar{L}_k}} \left(\frac{\tilde{Q}_n}{\beta} - \frac{1}{|\bar{L}|} \sum_{m\in\bar{L}} \frac{\tilde{Q}_m}{\alpha} \right) A_{\bar{L},n} \\
&+ \sum_{\substack{n:n\in\mathcal{B}_o \\ n\in\bar{L}_r}} \left(\frac{\tilde{Q}_n}{\gamma} - \frac{1}{|\bar{L}|} \sum_{m\in\bar{L}} \frac{\tilde{Q}_m}{\alpha} \right) A_{\bar{L},n} \\
&+ \sum_{\substack{n:n\in\mathcal{H}_o \\ n\in\bar{L}_r}} \left(\frac{\tilde{Q}_n}{\gamma} - \frac{1}{|\bar{L}|} \sum_{m\in\bar{L}} \frac{\tilde{Q}_m}{\alpha} \right) A_{\bar{L},n} \Bigg] \qquad \text{(B.39)} \\
&+ \sum_{\bar{L}:\bar{L}\in\mathcal{L}_{\mathcal{H}_o}} \left(\sum_{n\in\bar{L}_k} \frac{\tilde{Q}_n}{\beta} A_{\bar{L},n} + \sum_{n\in\bar{L}_r} \frac{\tilde{Q}_n}{\gamma} A_{\bar{L},n} + \sum_{\substack{n\in\bar{L} \\ n\notin\mathcal{H}_o}} \frac{\tilde{Q}_n}{\alpha} A_{\bar{L},n} \right) \qquad \text{(B.40)} \\
&+ \sum_{\bar{L}:\bar{L}\in\mathcal{L}^*_{\mathcal{H}_u}} \left(\sum_{n\in\bar{L}_k} \frac{\tilde{Q}_n}{\beta} A_{\bar{L},n} + \sum_{n\in\bar{L}_r} \frac{\tilde{Q}_n}{\gamma} A_{\bar{L},n} + \sum_{\substack{n\in\bar{L} \\ n\notin\mathcal{H}_u}} \frac{\tilde{Q}_n}{\alpha} A_{\bar{L},n} \right). \qquad \text{(B.41)}
\end{aligned}$$

For each term in (B.39), according to weighted workload routing, $A_{\bar{L},n} > 0$ only if n is in the set

$$\arg\min_{m\in\mathcal{M}} \left\{ \frac{W_m(t)}{\alpha} I_{\{m\in\bar{L}\}}, \frac{W_m(t)}{\beta} I_{\{m\in\bar{L}_k\}}, \frac{W_m(t)}{\gamma} I_{\{m\in\bar{L}_r\}} \right\}.$$

Note that $\forall n \in \mathcal{B}_o$, $\tilde{Q}_n = W_n$. Hence if $n \in \mathcal{B}_o$ and $n \in \bar{L}_k$,

$$\frac{\tilde{Q}_n}{\beta} \le \frac{1}{|\bar{L}|} \sum_{m\in\bar{L}} \frac{W_m}{\alpha} = \frac{1}{|\bar{L}|} \sum_{m\in\bar{L}} \frac{\tilde{Q}_m}{\alpha}.$$

So the first term in (B.39) is non-positive. Similarly we can show the second term in (B.39) is non-positive. As for the third term, since $\forall n \in \mathcal{H}_o$, $\tilde{Q}_n = \frac{Q_m^k}{\beta} + \frac{Q_m^r}{\gamma} \le W_n$, $\frac{\tilde{Q}_n}{\gamma} \le gamma \frac{W_n}{} \le \frac{1}{|\bar{L}|} \sum_{m\in\bar{L}} \frac{W_m}{\alpha} = \frac{1}{|\bar{L}|} \sum_{m\in\bar{L}} \frac{\tilde{Q}_m}{\alpha}$. Thus we

have $(B.39) \leq 0$. From the definition of $\tilde{\mathbf{Q}}$, (B.40) can be upper bounded by

$$\sum_{\bar{L}:\bar{L}\in\mathcal{L}_{\mathcal{H}_o}} \left(\sum_{n\in\bar{L}_k} \frac{W_n}{\beta} A_{\bar{L},n} + \sum_{n\in\bar{L}_r} \frac{W_n}{\gamma} A_{\bar{L},n} + \sum_{n\in\bar{L}\&n\notin\mathcal{H}_o} \frac{W_n}{\alpha} A_{\bar{L},n} \right). \quad \text{(B.42)}$$

Similarly, (B.41) can be upper bounded by

$$\sum_{\bar{L}:\bar{L}\in\mathcal{L}^*_{\mathcal{H}_u}} \left(\sum_{n\in\bar{L}_k} \frac{W_n}{\beta} A_{\bar{L},n} + \sum_{n\in\bar{L}_r} \frac{W_n}{\gamma} A_{\bar{L},n} + \sum_{n\in\bar{L}\&n\notin\mathcal{H}_u} \frac{W_n}{\alpha} A_{\bar{L},n} \right). \quad \text{(B.43)}$$

Next we will show that the expectation of each term in (B.42) and (B.43) will go to zero as $\epsilon \to 0$. Consider any $\bar{L} \in \mathcal{L}_{\mathcal{H}_o}$, and $n \in \bar{L}_k \cap \mathcal{B}_o$ such that $A_{\bar{L},n} > 0$. Pick any $m \in \bar{L}$ such that $m \in \mathcal{H}_o$. Define

$$W^*_{\bar{L}}(t) = \min_{m\in\mathcal{M}} \left\{ \frac{W_m(t)}{\alpha} I_{\{m\in\bar{L}\}}, \frac{W_m(t)}{\beta} I_{\{m\in\bar{L}_k\}}, \frac{W_m(t)}{\gamma} I_{\{m\in\bar{L}_r\}} \right\}.$$

So

$$\begin{aligned} & \frac{W_n}{\beta} A_{\bar{L},n} \leq C_A \frac{W_n}{\beta} I_{\{\frac{W_n}{\beta} = W^*_{\bar{L}}\}} \leq C_A \frac{W_n}{\beta} I_{\{\frac{W_n}{\beta} \leq \frac{W_m}{\alpha}\}} \\ = \; & C_A \frac{W_n}{\beta} I_{\{(\frac{\alpha^2}{\gamma^2}-1)\frac{W_n}{\alpha} \leq \frac{W_m}{\beta} - \frac{W_n}{\alpha}\}} \leq \frac{C_A \alpha}{\beta} \cdot \frac{W_n}{\alpha} I_{\{a\frac{W_n}{\alpha} \leq |\frac{W_m}{\beta} - \frac{W_n}{\alpha}|\}}, \end{aligned}$$

where $a = \frac{\alpha^2}{\gamma^2} - 1 > 0$ is a constant.

Next we will show that

$$\mathbb{E}\left[\frac{W_n}{\alpha} I_{\{a\frac{W_n}{\alpha} \leq |\frac{W_m}{\beta} - \frac{W_n}{\alpha}|\}} \right] = o(\epsilon).$$

We need the following lemma, which follows by the result of state space collapse.

Lemma B.26. *There exist a sequence of constants $\{C_r\}_{r\in\mathbb{N}}$ independent of ϵ such that for any $n, m \in \mathcal{M}$,*

$$\mathbb{E}\left[\left\| \frac{W_n}{c_n} - \frac{W_m}{c_m} \right\|^r \right] \leq C_r,$$

where $\mathbf{c}$ is the direction to which the state space $\mathbf{W}$ collapse.

Note that

$$\begin{aligned}\frac{W_n}{\alpha}I_{\{a\frac{W_n}{\alpha}\leq|\frac{W_m}{\beta}-\frac{W_n}{\alpha}|\}} &= \frac{W_n}{\alpha}I_{\{a\frac{W_n}{\alpha}\leq|\frac{W_m}{\beta}-\frac{W_n}{\alpha}|\}}I_{\{W_n>0\}} \\ &= \frac{W_n}{\alpha}I_{\{\frac{a^2W_n^2}{\alpha^2}\leq|\frac{W_m}{\beta}-\frac{W_n}{\alpha}|^2\}}I_{\{W_n>0\}} \\ &\leq \frac{|\frac{W_m}{\beta}-\frac{W_n}{\alpha}|^2}{\frac{a^2W_n}{\alpha^2}}I_{\{W_n>0\}}.\end{aligned}$$

Thus

$$\begin{aligned}\mathbb{E}\left[\frac{W_n}{\alpha}I_{\{a\frac{W_n}{\alpha}\leq|\frac{W_m}{\beta}-\frac{W_n}{\alpha}|\}}\right] &\leq \mathbb{E}\left[\frac{|\frac{W_m}{\beta}-\frac{W_n}{\alpha}|^2}{\frac{a^2W_n}{\alpha^2}}I_{\{W_n>0\}}\right] \\ &\overset{(a)}{\leq} \frac{\alpha^2}{a^2}\sqrt{\mathbb{E}\left[|\frac{W_m}{\beta}-\frac{W_n}{\alpha}|^4\right]\mathbb{E}\left[\frac{I_{\{W_n>0\}}}{W_n^2}\right]} \\ &\overset{(b)}{\leq} \frac{\alpha^2}{a^2}\sqrt{C_4\mathbb{E}\left[\frac{I_{\{W_n>0\}}}{W_n^2}\right]},\end{aligned}$$

where (a) comes from Cauchy-Schwarz inequality, and (b) follows by Lemma B.26. Since $W_n = \frac{Q_n^l(t)}{\alpha} + \frac{Q_n^k(t)}{\beta} + \frac{Q_n^r(t)}{\gamma}$, we have $\mathbb{E}\left[\frac{I_{\{W_n\geq 0\}}}{W_n^2}\right] \to 0$ as $\epsilon \to 0$. Therefore we have

$$\mathbb{E}\left[\frac{W_n}{\alpha}I_{\{a\frac{W_n}{\alpha}\leq|\frac{W_m}{\beta}-\frac{W_n}{\alpha}|\}}\right] \overset{\epsilon\to 0}{\longrightarrow} 0.$$

We can show that any other term in (B.42) and (B.43) is upperbounded by $o(\epsilon)$ in a similar way.

■

APPENDIX C

ADDITIONAL PROOFS FOR LOSS MODEL

C.1 Proof of Lemma 5.1

The case $\lambda = 0$ is trivial with a unique stationary solution $\boldsymbol{\pi} = (1, \underbrace{0, 0, ..., 0}_{B})$.

We focus on the case $\lambda > 0$.

Existence: For ease of exposition, throughout the proof we define $x_{B+1} = 0$ for any $\mathbf{x} \in \mathcal{S}$.

Step 1. Define $\mathbf{G}(\mathbf{x}),\ \forall \mathbf{x} \in \mathcal{S}$.

For any $\mathbf{x} \in \mathcal{S}$, let $\mathbf{G}(\mathbf{x}) = (G_0(\mathbf{x}), G_1(\mathbf{x}), ..., G_B(\mathbf{x}))$, where $G_0(\mathbf{x}) = 1$, and $\forall k = 1, 2, ..., B$, $G_k(\mathbf{x}) \geq 0$ satisfies

$$\lambda G_k^d(\mathbf{x}) + kG_k(\mathbf{x}) - \lambda x_{k-1}^d - kx_{k+1} = 0. \tag{C.1}$$

We will show that $\mathbf{G}$ is uniquely determined by $\mathbf{x}$. Consider a sequence of functions $\{H_k(y_k)\}_{k=1}^B$, where

$$H_k(y_k) = \lambda y_k^d + ky_k - \lambda x_{k-1}^d - kx_{k+1}.$$

Since $\mathbf{x} \in \mathcal{S}$,

$$\begin{aligned} H_k(x_{k-1}) &= kx_{k-1} - kx_{k+1} \geq 0, \\ H_k(x_{k+1}) &= \lambda x_{k+1}^d - \lambda x_{k-1}^d \leq 0. \end{aligned}$$

Note that $H_k(y_k)$ is strictly increasing in $y_k \in [0, \infty)$. Hence there exists a unique $y_k^* > 0$ such that $H_k(y_k^*) = 0$. By the definition of G_k in (C.1), $H_k(G_k) = 0$. Hence $G_k = y_k^*$ is determined by $\mathbf{x}$ uniquely, and

$$x_{k+1} \leq G_k(\mathbf{x}) \leq x_{k-1}. \tag{C.2}$$

Step 2. Show that $\mathbf{G}(\cdot)$ is mapping $\mathcal{S}$ into $\mathcal{S}$.

We will verify that $\forall \mathbf{x} \in \mathcal{S}$, $\mathbf{G}(\mathbf{x}) \in \mathcal{S}$, i.e., $1 = G_0(\mathbf{x}) \geq G_1(\mathbf{x}) \geq ... \geq G_B(\mathbf{x}) \geq 0$. For any $\mathbf{x} \in \mathcal{S}$, inequality in (C.2) ensures that $G_k \in [0,1]$ for all k. To prove that $G_k \geq G_{k+1}$, consider a function

$$\varphi_k(z) = \lambda z^d + kz,$$

which is strictly increasing in $[0,1]$. Hence it is sufficient to show that $\varphi_k(G_k) \geq \varphi_k(G_{k+1})$.

$$\begin{aligned}
\varphi_k(G_k) - \varphi_k(G_{k+1}) &= \lambda G_k^d + kG_k - \lambda G_{k+1}^d - (k+1)G_{k+1} + G_{k+1} \\
&\overset{(a)}{=} \lambda x_{k-1}^d + kx_{k+1} - \lambda x_k^d - (k+1)x_{k+2} + G_{k+1} \\
&= \lambda(x_{k-1}^d - x_k^d) + k(x_{k+1} - x_{k+2}) + G_{k+1} - x_{k+2} \\
&\overset{(b)}{\geq} G_{k+1} - \pi_{k+2} \\
&\overset{(c)}{\geq} 0,
\end{aligned}$$

where the equality (a) comes from the definition of G_k, G_{k+1} in (C.1), and the inequality (b) follows by the fact that $\mathbf{x} \in \mathcal{S}$, and the inequality (c) results from the property of G_k in (C.2).

Therefore $\mathbf{G}(\mathbf{x}) \in \mathcal{S}$.

Step 3. Show that $\mathbf{G}(\cdot)$ is continuous.

Consider any point $\mathbf{x} \in \mathcal{S}$. For every $\epsilon > 0$, set $\delta = \frac{\epsilon}{\lambda d + 1}$. Let $\mathbf{y}$ be any point in $\mathcal{S}$ such that $|\mathbf{x} - \mathbf{y}| < \delta$. By the definition of $\mathbf{G}(\cdot)$, $\forall k = 1, 2, ..., B$,

$$\begin{aligned}
& \lambda(G_k^d(\mathbf{x}) - G_k^d(\mathbf{y})) + k(G_k(\mathbf{x}) - G_k(\mathbf{y})) \\
= \; & (G_k(\mathbf{x}) - G_k(\mathbf{y})) \left(\lambda \sum_{i=0}^{d-1} G_k^{d-1-i}(\mathbf{x}) G_k^i(\mathbf{y}) + k \right) \\
= \; & \lambda(x_{k-1}^d - y_{k-1}^d) + k(x_{k+1} - y_{k+1}) \\
= \; & \lambda(x_{k-1} - y_{k-1}) \left(\sum_{i=0}^{d-1} x_{k-1}^{d-1-i} y_{k-1}^i \right) + k(x_{k+1} - y_{k+1}).
\end{aligned}$$

Then we have

$$\begin{aligned}
|G_k(\mathbf{x}) - G_k(\mathbf{y})| &= \frac{|\lambda(x_{k-1} - y_{k-1})\left(\sum_{i=0}^{d-1} x_{k-1}^{d-1-i} y_{k-1}^{i}\right) + k(x_{k+1} - y_{k+1})|}{\lambda \sum_{i=0}^{d-1} G_k^{d-1-i}(\mathbf{x}) G_k^{i}(\mathbf{y}) + k} \\
&\leq \frac{\lambda d|x_{k-1} - y_{k-1}| + k|(x_{k+1} - y_{k+1})|}{k} \\
&\leq \lambda d|x_{k-1} - y_{k-1}| + |(x_{k+1} - y_{k+1})|,
\end{aligned}$$

which implies that

$$\begin{aligned}
|\mathbf{G}(\mathbf{x}) - \mathbf{G}(\mathbf{y})| &= \sum_{k=0}^{B} |G_k(\mathbf{x}) - G_k(\mathbf{y})| \\
&\leq \sum_{k=1}^{B} (\lambda d|x_{k-1} - y_{k-1}| + |(x_{k+1} - y_{k+1})|) \\
&\leq (\lambda d + 1) \sum_{k=0}^{B} |x_k - y_k| \\
&< (\lambda d + 1)\delta \\
&= \epsilon.
\end{aligned}$$

Therefore $\mathbf{G}$ is continuous at any point $\mathbf{x} \in \mathcal{S}$.

Step 4. Show that a fixed point of $\mathbf{G}$ in $\mathcal{S}$ is a stationary point.

Note that set $\mathbf{S}$ is compact and convex. Step 1-3 ensures that there exists a fixed point of $\mathbf{G}$ in $\mathcal{S}$, denoted by $\hat{\mathbf{x}}$. That is, $\hat{\mathbf{x}} = \mathbf{G}(\hat{\mathbf{x}})$. From the definition of $\mathbf{G}$ in (C.1), we have

$$F_k(\hat{\mathbf{x}}) = \lambda \hat{x}_k^d + k\hat{x}_k - \lambda \hat{x}_{k-1}^d - k\hat{x}_{k+1} = 0.$$

That is, $\hat{\mathbf{x}}$ is a stationary point.

Uniqueness: We prove the uniqueness of stationary solution by contradiction. Assume that there exists two different solutions $\boldsymbol{\pi}$ and $\hat{\boldsymbol{\pi}}$. We claim that $\pi_B \neq \hat{\pi}_B$. Otherwise, we have

$$\pi_{B-1} = \sqrt[d]{\pi_B^d + \frac{B}{\lambda}\pi_B} = \hat{\pi}_{B-1}.$$

Note that

$$\pi_k = \sqrt[d]{\pi_{k+1}^d + \frac{k+1}{\lambda}(\pi_{k+1} - \pi_{k+2})}.$$

Hence by induction, we can show that $\pi_k = \hat{\pi}_k$ for any $k = 0, 1, ..., B$.

Consider the case that $\pi_B < \hat{\pi}_B$. Similarly, we can establish that $\pi_k < \hat{\pi}_k$ for any $k = 0, 1, ..., B$ by induction. Therefore, $\pi_0 < \hat{\pi}_0$, which contradicts with the fact that $\pi_0 = \hat{\pi}_0 = 1$. This completes the proof for the uniqueness. ■

C.2 Proof of Lemma 5.3

Due to continuous dependence of a solution on the initial values, it is sufficient to show that if $\bar{s}_k^0 < s_k^0$ for any $k \geq 1$, $\bar{s}_k(t) \leq s_k(t)$ for all $t \geq 0$ and any k. Assume that strict inequalities hold for $t < t_1$ and are broken at $t = t_1$. Consider two cases:

(i) $\bar{s}_k(t_1) = s_k(t_1)$ for any k.

The uniqueness of solution ensures that $\bar{s}_k(t) = s_k(t)$ for all $t \geq t_1$ and any k. Hence the claim holds.

(ii) $\exists k^* \geq 1$ such that $\bar{s}_{k^*}(t_1) < s_{k^*}(t_1)$.

Then there exists $k \geq 1$ such that $\bar{s}_k(t_1) = s_k(t_1)$, and at least of one following conditions hold: $\bar{s}_{k-1}(t_1) < s_{k-1}(t_1)$, $\bar{s}_{k+1}(t_1) < s_{k+1}(t_1)$. If $k < B$, we have

$$\begin{aligned} & \frac{d\bar{s}_k}{dt}(t_1) - \frac{ds_k}{dt}(t_1) \\ = \; & \lambda(\bar{s}_{k-1}^d - s_{k-1}^d) + k(\bar{s}_{k+1} - s_{k+1}) - \lambda(\bar{s}_k^d - s_k^d) - k(\bar{s}_k - s_k) < 0, \end{aligned}$$

where the inequality comes from the definition of k. Similarly, we can verify that $\frac{d\bar{s}_k}{dt}(t_1) - \frac{ds_k}{dt}(t_1) < 0$ if $k = B$.

Since $\bar{\mathbf{s}}(t)$ and $\mathbf{s}(t)$ are continuous functions of t, there exists $t_0 < t_1$ such that $\bar{s}_k(t_0) < s_k(t_0)$ and

$$\frac{d\bar{s}_k}{dt}(t) - \frac{ds_k}{dt}(t) < 0$$

for any $t \in (t_0, t_1)$. Thus

$$\bar{s}_k(t_1) - s_k(t_1) = \bar{s}_k(t_0) - s_k(t_0) + \int_{t_0}^{t_1} \left(\frac{d\bar{s}_k}{dt}(t) - \frac{ds_k}{dt}(t) \right) dt < 0,$$

which contradicts with the assumption that $\bar{s}_k(t_1) = s_k(t_1)$. ■

C.3 Proof of Lemma 5.4

We will show that $d\psi(t)/dt \leq -\psi$. Then $\psi(t) \leq \psi(0)e^{-t}$, which implies that $\psi(t)$ converges to 0 exponentially fast.

Consider the case where $s_k^0 \geq \pi_k$ for any k. From Lemma 4, $s_k(t) \geq \pi_k$ for any $t \geq 0, \forall k \in \{0, 1, ..., B\}$. We can rewrite $\psi(t)$ as $\psi(t) = \sum_{k=0}^{B}(s_k(t) - \pi_k)$. Since $\mathbf{F}(\boldsymbol{\pi}) = \mathbf{0}$, $\dot{\mathbf{s}} = \mathbf{F}(\mathbf{s})$, we have

$$\begin{aligned}
\frac{d\psi(t)}{dt} &= \sum_{k=0}^{B} \frac{ds_k(t)}{dt} = \sum_{k=1}^{B} F_k(\mathbf{s}(t)) - \sum_{k=1}^{B} F_k(\boldsymbol{\pi}) \\
&= \left(\lambda(s_0^d(t) - s_B^d(t)) - \sum_{k=1}^{B} s_k(t)\right) - \left(\lambda(\pi_0^d - \pi_B^d) - \sum_{k=1}^{B} \pi_k\right) \\
&= -\lambda(s_B^d(t) - \pi_B^d) - \psi(t) \leq -\psi(t),
\end{aligned}$$

where the last inequality follows by the fact that $s_B^d(t) \geq \pi_B^d$.

The other case where $s_k^0 \leq \pi_k$ for any k can be proved similarly. ■

C.4 Proof of Lemma 5.6

Since $\mathbf{x}, \mathbf{y} \in \mathcal{S}$, for any $0 \leq k \leq B$

$$0 \leq x_k \leq 1,\ 0 \leq y_k \leq 1.$$

Then we have:

$$\begin{aligned}
& |\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y})| \\
= & \sum_{k=1}^{B-1} |\lambda(x_{k-1}^d - x_k^d) - k(x_k - x_{k+1}) - \lambda(y_{k-1}^d - y_k^d) + k(y_k - y_{k+1})| \\
& + |\lambda(x_{B-1}^d - x_B^d) - Bx_B - \lambda(y_{B-1}^d - y_B^d) + By_B| \\
\leq & 2\sum_{k=0}^{B} k|x_k - y_k| + 2\sum_{k=0}^{B} \lambda|x_k^d - y_k^d| \\
\leq & 2B\sum_{k=0}^{B} |x_k - y_k| + 2\lambda \sum_{k=0}^{B} \left(|x_k - y_k| \sum_{i=0}^{d-1} x_k^{d-1-i} y_k^i \right) \\
\leq & 2(B + d\lambda) \sum_{k=0}^{B} |x_k - y_k| \\
= & M|\mathbf{x} - \mathbf{y}|,
\end{aligned}$$

where $M = 2(B + d\lambda)$. ∎

C.5 Proof of Claim 1

From Lemma 5.5, we have

$$\mathbf{S}^{(N_k)}(t) \Rightarrow \bar{\mathbf{S}}(t) \text{ as } k \to \infty.$$

By the definition of weak convergence, for a bounded continuous function f, if $\mathbf{S}^{(N_k)}(0) \to \bar{\mathbf{S}}(0)$ as $k \to \infty$,

$$\mathbb{E}\left[f(\mathbf{S}^{(N_k)}(t))|\mathbf{S}^{(N_k)}(0)\right] \overset{n\to\infty}{\longrightarrow} \mathbb{E}\left[f(\bar{\mathbf{S}}(t))|\bar{\mathbf{S}}(0)\right].$$

As $\mathbf{S}^{(N_k)}(0) = \mathbf{X}^{(N_k)}$ and $\bar{\mathbf{S}}(0) = \bar{\mathbf{X}}$, by Skorokhod's representation theorem,

$$\mathbf{S}^{(N_k)}(0) \to \bar{\mathbf{S}}(0).$$

Define

$$\mathbf{Y}_k = \mathbb{E}\left[f(\mathbf{S}^{(N_k)}(t))|\mathbf{X}^{(N_k)}\right], \quad \mathbf{Y} = \mathbb{E}\left[f(\bar{\mathbf{S}}(t))|\bar{\mathbf{X}}\right].$$

Since f is bounded, $\mathbf{Y}_k$ and $\mathbf{Y}$ are bounded. By the bounded convergence

theorem, we have

$$\mathbb{E}\left[\mathbf{Y}_k\right] \to \mathbb{E}\left[\mathbf{Y}\right].$$

This holds for all bounded, continuous f. Thus again by the definition of weak convergence,

$$\mathbf{S}^{(N_k)}(t) \Rightarrow \bar{\mathbf{S}}(t) \text{ as } k \to \infty.$$

■

REFERENCES

[1] "Amazon EC2," http://aws.amazon.com/ec2.

[2] "Google App Engine," https://cloud.google.com/appengine/docs?csw=1.

[3] "Rackspace," https://www.rackspace.com/.

[4] "Azure," http://azure.microsoft.com/en-us/.

[5] L. A. Barroso, J. Clidaras, and U. Hölzle, "The datacenter as a computer: An introduction to the design of warehouse-scale machines, second edition," *Synthesis Lectures on Computer Architecture*, vol. 8, no. 3, pp. 1–154, 2013.

[6] C. Delimitrou and C. Kozyrakis, "Quasar: Resource-efficient and qos-aware cluster management," in *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '14. ACM, 2014, pp. 127–144.

[7] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in *Proceedings of the Symposium on Operating Systems Design and Implementation (OSDI)*. USENIX, 2004.

[8] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly, "Dryad: Distributed data-parallel programs from sequential building blocks," in *Proceedings of the European Conference on Computer Systems (EuroSys)*, 2007.

[9] H. C. Yang, A. Dasdan, R. L. Hsiao, and D. S. Parker, "Map-reduce-merge: Simplified relational data processing on large clusters," in *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*. ACM, 2007, pp. 1029–1040.

[10] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google file system," in *Proceedings of the Symposium on Operating Systems Principles (SOSP)*. ACM, 2003.

[11] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The Hadoop distributed file system," in *Proceedings of the Symposium on Mass Storage Systems and Technology (MSST)*. IEEE, 2010.

[12] M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, "Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling," in *Proceedings of the European Conference on Computer Systems (EuroSys)*, 2010.

[13] G. Ananthanarayanan, S. Agarwal, S. Kandula, A. Greenberg, I. Stoica, D. Harlan, and E. Harris, "Scarlett: Coping with skewed popularity content in MapReduce clusters," in *Proceedings of the European Conference on Computer Systems (EuroSys)*, 2011.

[14] G. Ananthanarayanan, A. Ghodsi, A. Wang, D. Borthakur, S. Kandula, S. Shenker, and I. Stoica, "Pacman: Coordinated memory caching for parallel jobs," in *Proceedings of Symposium on Networked Systems Design and Implementation (NSDI)*. USENIX, 2012.

[15] M. Squillante, C. Xia, D. Yao, and L. Zhang, "Threshold-based priority policies for parallel-server systems with affinity scheduling," in *Proc. IEEE American Control Conf.*, 2001.

[16] J. M. Harrison, "Heavy traffic analysis of a system with parallel servers: Asymptotic optimality of discrete review policies," *Annals of Applied Probability*, vol. 8, no. 3, pp. 822–848, 1998.

[17] J. M. Harrison and M. J. López, "Heavy traffic resource pooling in parallel-server systems," *Queueing Syst. Theory Appl.*, vol. 33, no. 4, Apr. 1999.

[18] S. Bell and R. Williams, "Dynamic scheduling of a parallel server system in heavy traffic with complete resource pooling: Asymptotic optimality of a threshold policy," *Electron. J. Probab.*, vol. 10, pp. no. 33, 1044–1115, 2005.

[19] M. Isard, V. Prabhakaran, J. Currey, U. Wieder, K. Talwar, and A. Goldberg, "Quincy: Fair scheduling for distributed computing clusters," in *Proceedings of the Symposium on Operating Systems Principles (SOSP)*. ACM, 2009.

[20] J. Jin, J. Luo, A. Song, F. Dong, and R. Xiong, "Bar: An efficient data locality driven task scheduling algorithm for cloud computing," in *Proceedings of the International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*. IEEE, 2011.

[21] S. Ibrahim, H. Jin, L. Lu, B. He, G. Antoniu, and S. Wu, "Maestro: Replica-aware map scheduling for mapreduce," in *Proceedings of the International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*. IEEE, 2012.

[22] C. He, Y. Lu, and D. Swanson, "Matchmaking: A new mapreduce scheduling technique," in *Proceedings of the International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE, 2011.

[23] "Aws case study: Netflix," http://aws.amazon.com/solutions/case-studies/netflix/.

[24] M. Wang, X. Meng, and L. Zhang, "Consolidating virtual machines with dynamic bandwidth demand in data centers," in *Proceedings of IEEE INFOCOM*, April 2011, pp. 71–75.

[25] U. Sharma, P. Shenoy, S. Sahu, and A. Shaikh, "Kingfisher: Cost-aware elasticity in the cloud," in *Proceedings of IEEE INFOCOM*, April 2011, pp. 206–210.

[26] S. Maguluri, R. Srikant, and L. Ying, "Stochastic models of load balancing and scheduling in cloud computing clusters," in *Proceedings of IEEE INFOCOM*, Mar 2012, pp. 702–710.

[27] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," in *Proceedings of IEEE INFOCOM*, April 2011, pp. 1098–1106.

[28] A. L. Stolyar and Y. Zhong, "A large-scale service system with packing constraints: Minimizing the number of occupied servers," *SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 1, pp. 41–52, June 2013.

[29] K. Tsakalozos, H. Kllapi, E. Sitaridi, M. Roussopoulos, D. Paparas, and A. Delis, "Flexible use of cloud resources through profit maximization and price discrimination," in *2011 IEEE 27th International Conference on Data Engineering*, April 2011, pp. 75–86.

[30] Y. O. Yazir, C. Matthews, R. Farahbod, S. Neville, A. Guitouni, S. Ganti, and Y. Coady, "Dynamic resource allocation in computing clouds using distributed multiple criteria decision analysis," in *2010 IEEE 3rd International Conference on Cloud Computing*, July 2010, pp. 91–98.

[31] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in *Proceedings of IEEE INFOCOM*, 2010, pp. 1154–1162.

[32] S. T. Maguluri, R. Srikant, and L. Ying, "Heavy traffic optimal resource allocation algorithms for cloud computing clusters," in *Proc. of the 24th International Teletraffic Congress*, 2012, pp. 25:1–25:8.

[33] A. L. Stolyar, "An infinite server system with general packing constraints," 2012, ArXiv preprint arXiv:1205.4271.

[34] S. L. Bell and R. J. Williams, "Dynamic scheduling of a system with two parallel servers in heavy traffic with resource pooling: asymptotic optimality of a threshold policy," *Annals of Applied Probability*, vol. 11, 2001.

[35] A. Mandelbaum and A. Stolyar, "Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized $c\mu$-rule," *Operations Research*, vol. 52, 2004.

[36] A. L. Stolyar, "Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic," *The Annals of Applied Probability*, vol. 14, no. 1, pp. pp. 1–53, 2004.

[37] W. Wang, K. Zhu, L. Ying, J. Tan, and L. Zhang, "Map task scheduling in mapreduce with data locality: Throughput and heavy-traffic optimality," in *Proceedings of INFOCOM*. IEEE, 2013.

[38] W. Wang, M. Barnard, and L. Ying, "Decentralized scheduling with data locality for data-parallel computation on peer-to-peer networks," in *Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2015.

[39] X. Bu, J. Rao, and C. Z. Xu, "Interference and locality-aware task scheduling for mapreduce applications in virtual clusters," in *Proceedings of the International Symposium on High-performance Parallel and Distributed Computing*. ACM, 2013.

[40] B. Palanisamy, A. Singh, L. Liu, and B. Jain, "Purlieus: Locality-aware resource allocation for MapReduce in a cloud," in *Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, 2011.

[41] C. Abad, Y. Lu, and R. H. Campbell, "DARE: Adaptive data replication for efficient cluster scheduling," in *Proceedings of the International Conference on Cluster Computing (CLUSTER)*. IEEE, 2011.

[42] Y. Yao, J. Tai, B. Sheng, and N. Mi, "LsPS: A job size-based scheduler for efficient assignments in Hadoop," in *Transactions on Cloud Computing*. IEEE, 2014.

[43] P. Nguyen, T. Simon, M. Halem, D. Chapman, and Q. Le, "A hybrid scheduling algorithm for data intensive workloads in a mapreduce environment," in *Proceedings of the International Conference on Utility and Cloud Computing*. IEEE, 2012.

[44] C. Wang, Y. Qin, Z. Huang, Y. Peng, D. Li, and H. Li, "OPTAS: Optimal data placement in MapReduce," in *Proceedings of International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2013.

[45] J. Tan, S. Meng, X. Meng, and L. Zhang, "Improving reduce task data locality for sequential MapReduce jobs," in *Proceedings of INFOCOM*. IEEE, 2013.

[46] M. Hammoud, M. S. Rehman, and M. F. Sakr, "Center-of-gravity reduce task scheduling to lower MapReduce network traffic," in *Proceedings of the International Conference on Cloud Computing (CLOUD)*. IEEE, 2012.

[47] M. Lin, L. Zhang, A. Wierman, and J. Tan, "Joint optimization of overlapping phases in MapReduce," *Performance Evaluation*, 2013.

[48] J. Tan, X. Meng, and L. Zhang, "Coupling task progress for MapReduce resource-aware scheduling," in *Proceedings of INFOCOM*. IEEE, 2013.

[49] X. Ren, G. Ananthanarayanan, A. Wierman, and M. Yu, "Speculation-aware cluster scheduling," *SIGMETRICS Perform. Eval. Rev.*, vol. 43, no. 2, pp. 42–44, Sep. 2015. [Online]. Available: http://doi.acm.org/10.1145/2825236.2825254

[50] G. Ananthanarayanan, S. Kandula, A. Greenberg, I. Stoica, Y. Lu, B. Saha, and E. Harris, "Reining in the outliers in MapReduce clusters using Mantri," in *Proceedings of the Conference on Operating Systems Design and Implementation (OSDI)*. USENIX, 2010.

[51] G. Ananthanarayanan, A. Ghodsi, S. Shenker, and I. Stoica, "Effective straggler mitigation: Attack of the clones," in *Proceedings of the Symposium on Networked Systems Design and Implementation (NSDI)*. USENIX, 2013.

[52] J. Wolf, D. Rajan, K. Hildrum, R. Khandekar, V. Kumar, S. Parekh, K.-L. Wu, and A. Balmin, "Flex: A slot allocation scheduling optimizer for MapReduce workloads," in *Proceedings of the Middleware Conference*, 2010.

[53] F. Chen, M. Kodialam, and T. Lakshman, "Joint scheduling of processing and shuffle phases in MapReduce systems," in *Proceedings of INFOCOM*. IEEE, 2012.

[54] A. Verma, L. Cherkasova, and R. H. Campbell, "Two sides of a coin: optimizing the schedule of MapReduce jobs to minimize their makespan and improve cluster performance," in *Proceedings of International Symposium of Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*. IEEE, 2012.

[55] J. Polo, C. Castillo, D. Carrera, Y. Becerra, I. Whalley, M. Steinder, J. Torres, and E. Ayguad, "Resource-aware adaptive scheduling for MapReduce clusters," in *Proceedings of the Middleware Conference*, 2011.

[56] Y. Zhu, Y. Jiang, W. Wu, L. Ding, T. Ankur, D. Li, and W. Lee, "Minimizing makespan and total completion time in MapReduce-like systems," in *Proceedings of INFOCOM*, 2014.

[57] S. Venkataraman, A. Panda, G. Ananthanarayanan, M. J. Franklin, and I. Stoica, "The power of choice in data-aware cluster scheduling," in *Proceesings of the Conference on Operating Systems Design and Implementation (OSDI)*. ACM, 2014.

[58] Y. Chen, A. Ganapathi, R. Griffith, and R. Katz, "The case for evaluating MapReduce performance using workload suites," in *Proceedings of the International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*. IEEE, 2011.

[59] A. Eryilmaz and R. Srikant, "Asymptotically tight steady-state queue length bounds implied by drift conditions," *Queueing Syst. Theory Appl.*, vol. 72, no. 3-4, pp. 311–359, 2012.

[60] B. Hajek, "Hitting-time and occupation-time bounds implied by drift analysis with applications," *Advances in Applied Probability*, vol. 14, no. 3, pp. pp. 502–525, 1982.

[61] Y. Chen, S. Alspaugh, and R. Katz, "Interactive analytical processing in big data systems: A cross-industry study of MapReduce workloads," in *Proceedings of the VLDB Endowment*. VLDB Endowment, 2012.

[62] "OpenStack," http://www.openstack.org/.

[63] N. Bansal, A. Caprara, and M. Sviridenko, "A new approximation method for set covering problems, with applications to multidimensional bin packing," *SIAM Journal on Computing*, vol. 39, no. 4, pp. 1256–1278, 2010.

[64] J. Csirik, D. S. Johnson, C. Kenyon, J. B. Orlin, P. W. Shor, and R. R. Weber, "On the sum-of-squares algorithm for bin packing," *J. ACM*, vol. 53, no. 1, pp. 1–65, Jan. 2006.

[65] V. Gupta and A. Radovanovic, "Online stochastic bin packing," 2012, ArXiv preprint arXiv:1211.2687. [Online]. Available: http://arxiv.org/abs/1211.2687

[66] L. Wang, F. Zhang, A. V. Vasilakos, C. Hou, and Z. Liu, "Joint virtual machine assignment and traffic engineering for green data center networks," *SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 3, pp. 107–112, Jan. 2014.

[67] A. L. Stolyar and Y. Zhong, "An infinite server system with general packing constraints: Asymptotic optimality of a greedy randomized algorithm," in *Proc. 53th Annu. Allerton Conf. Commun., Control Comput.*, Oct 2013, pp. 575–582.

[68] Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal, "Balanced allocations," *SIAM J. Comput.*, vol. 29, no. 1, pp. 180–200, Sep. 1999.

[69] M. Mitzenmacher, "The power of two choices in randomized load balancing," Ph.D. dissertation, UC Berkeley, 1996.

[70] M. Mitzenmacher, "Studying balanced allocations with differential equations," *Combinatorics, Probability and Computing*, vol. 8, no. 5, pp. 473–482, Sep. 1999.

[71] N. D. Vvedenskaya, R. L. Dobrushin, and F. I. Karpelevich, "Queueing system with selection of the shortest of two queues: An asymptotic approach," *Probl. Peredachi Inf.*, vol. 32, no. 1, pp. 20–34, 1996.

[72] M. Bramson, Y. Lu, and B. Prabhakar, "Asymptotic independence of queues under randomized load balancing," *Queueing Systems: Theory and Applications (QUESTA)*, vol. 71, pp. 247–292, 2012.

[73] C. Graham, "Chaoticity on path space for a queueing network with selection of the shortest queue among several," *Journal of Appl. Prob.*, vol. 37, pp. 198–211, 2000.

[74] M. Luczak and C. McDiarmid, "On the maximum queue length in the supermarket model," *The Annals of Probability*, vol. 34, no. 2, pp. 493–527, 2006.

[75] A. Mukhopadhyay and R. R. Mazumdar, "Analysis of load balancing in large heterogeneous processor sharing systems," 2013, ArXiv preprint arXiv:1311.5806. [Online]. Available: http://arxiv.org/abs/1311.5806

[76] L. Ying, R. Srikant, and X. Kang, "The power of slightly more than one sample in randomized load balancing," in *Proceedings of IEEE INFOCOM*, 2015.

[77] J. N. Tsitsiklis and K. Xu, "On the power of (even a little) resource pooling," *Stochastic Systems*, vol. 2, no. 1, pp. 1–66, 2012.

[78] A. Mukhopadhyay, A. Karthik, R. R. Mazumdar, and F. Guillemin, "Mean field and propagation of chaos in multi-class heterogeneous loss models," *Performance Evaluation*, vol. 91, p. 117131, 2015.

[79] A. A. Borovkov, *Stochastic Processes in Queueing Theory*. Springer, 1976.

[80] W. Whitt, "Heavy-traffic approximations for service systems with blocking," *AT&T Bell Laboratories Technical Journal*, vol. 63, no. 5, pp. 689–708, 1984.

[81] R. Srikant and W. Whitt, "Simulation run lengths to estimate blocking probabilities," *ACM Trans. Model. Comput. Simul.*, vol. 6, no. 1, pp. 7–52, Jan. 1996.

[82] J. Kaufman, "Blocking in a shared resource environment," *IEEE Transactions on Communications*, vol. 29, no. 10, pp. 1474–1481, Oct 1981.

[83] J. W. Roberts, "A service system with heterogeneous user requirement," in *Performance of Data Communications Systems and Their Applications*, G. Pujolle, Ed., 1981.

[84] T. G. Kurtz, *Approximation of Population Processes*. Society for Industrial and Applied Mathematics, 1981.

[85] V. Anantharam and M. Benchekroun, "A technique for computing sojourn times in large networks of interacting queues," *Probability in the Engineering and Informational Sciences*, vol. 7, no. 04, pp. 441–464, 1993.