Research Statement

Zhuoyao Zhang
http://www.cis.upenn.edu/∼zhuoyao

1 Introduction

My research centers around performance modeling, optimization and resource management for MapReduce workflows with completion time constrains. My work is motivated by (1) the popularity of MapReduce framework and its open source implementation Hadoop that provides an economically compelling alternative for efficient analytics over ”Big Data” in the enterprise; and (2) the recent technological trend shift towards using MapReduce frameworks in support of latency-sensitive applications, e.g., personalized advertising, spam and fraud detection, real-time event log analysis, etc. The key challenge is how to efficiently determine the optimal configuration and resource provision strategies for these applications to achieve their performance goals in a shared cloud environment with execution nondeterminisms.

To address the problem, my dissertation work aims to develop a performance evaluation framework that enables automatic resource management and configuration optimization for MapReduce workflows. Specifically, it explores the following research questions: (1) Given certain input datasets, amount of computing resources and the workflow setting, to estimate the completion time of workflow on a given Hadoop cluster; (2) Given a completion time target, to determine the amount of resources that should be allocated to the workflow to achieve it; and (3) Determine the settings for the jobs within the workflow that optimize both the workflow completion time and the resource usage.

My approach to research involves a combination of mathematical analysis, benchmarking, simulation, implementation, deployments and empirical measurements on both private cluster and public cloud platform. I actively involve researchers from diverse backgrounds, and take an inter-disciplinary approach to solve problems. My work has been published at conferences in a variety of research domains, such as autonomic computing (ACM/USENIX ICAC [3, 10]), performance management (ACM/SPEC ICPE [6], IFIP/IEEE IM [4]) and cloud computing (IEEE CLOUD [8]). I won the best student paper award in the 9th International Conference on Autonomic Computing (ACM ICAC ’12).

2 Dissertation Work

My dissertation research aims to propose a performance evaluation framework that enables automatic resource management and performance optimization for latency sensitive MapReduce workflows represented as a DAG of MapReduce jobs. It consists of the following components: (1) A performance modeling framework that efficiently estimates the workflow completion time executed on a given Hadoop cluster according to the input data size, the workflow settings and the allocated resources; (2)A solution for resource provision problem that estimates the appropriate amount of resources for a MapReduce workflow to achieve its (soft) deadline; and (3) An optimization strategy to automatically determine the optimal job settings along the workflow for efficient execution and resource usage.

Performance modeling framework. The performance modeling framework [6, 11, 12, 5] contains: (1) A platform performance model [6, 5] to predict the completion time of different MapReduce phases as a function of processed data. Specifically, we used a set of microbenchmarks to profile the *generic* phases of the MapReduce processing pipeline of a given Hadoop cluster and derived the platform performance model based on the benchmark results using linear regression techniques; (2) A bounds based analytic model in estimating a single MapReduce job completion time as a function of allocated resources. It is based on a general makespan model for computing performance bounds of a given set of n tasks that are processed by k servers and uses a compact *job profile* that contains the *average* and *maximum* task durations which extract from past execution of the same job on small sample dataset. With such job profile, we are able to estimate the *upper* and *lower* bounds of the MapReduce job completion times. The average of these bounds based estimates shows good approximate of the actual performance; (3) A MapReduce workflow model [11, 12] that combines the previous two pieces of work to estimate the completion time of a given workflow with both sequential and concurrent branches. Specifically, it iterates through the workflow DAG and estimates the input size for each job according to the job characteristics, e.g., map/reduce task selectivity and the job setting, i.e., number of reduce tasks. Based on the estimated input size, the platform performance model is used to estimate the task duration which will be passed to bounds based analytical model to estimate the jobs' duration that consist the entire workflow completion time. We evaluated the accuracy of the proposed performance modeling framework with a variety of workloads created using Pig, a high-level SQL-like abstraction on top of Hadoop for writing MapReduce programs. The evaluation is preformed on both a 66-node private cluster in HP Labs and a public cluster created with instances provided by Amazon EC2.

Resource management for MapReduce workflows. Based on the performance modeling framework, we proposed an efficient strategy to estimate the minimal resource requirement for a workflow to achieve its completion time target [10, 7]. The estimated resource allocation is then passed to a deadline aware scheduler to decide the execution of the next job and assign resources to it accordingly.

We started as a building block, the resource management for a single MapReduce job according to the bounds based model using Lagrange's multipliers, and then extended it for MapReduce workflows. We first proposed a simple *basic* resource allocation approach which works efficiently for workflows with sequential jobs but is conservative for workflows with concurrent jobs because of the pipelined execution among the concurrent jobs. In improving the basic approach, we identified that the execution order of the concurrent jobs could significantly affect the workflow completion time and also brings nondeterminism in the workflow execution. To address this issue, we first optimized a MapReduce workflow execution by enforcing the optimal execution order of its concurrent jobs using the classic Johson's algorithm. We then provided a *refined* resource allocation approach that starts from the results got from the basic approach and then step-by-step reduces the resource allocation until we reach the point that any reduction of the resource allocation will violate the completion time constraints even with the execution overlap among the concurrent jobs.

We also extended our work in public cloud environment where customers lease resources from the service provider and pay for the time these resources were used. We proposed approaches to determine in such environments, the optimal choice and the amount of resources that a user should lease from the service provider under different performance and cost constrains by comparing the predicted performance and monetary cost of different platform choices based on our performance modeling framework [9].

Workflow setting optimization. This work aims to determine the optimal job setting (number of reduce tasks) for minimizing the MapReduce workflow completion time [4, 3]. The choice of the right number of reduce tasks depends on the Hadoop cluster size, the size of the input dataset of the job, and the amount of resources available for processing this job and could significantly affect the job completion time. The effect of the job settings also propagates through the worklfow due to the data dependency: the output of the previous job becomes the input of the next job, and therefore, the number of reduce tasks in the previous job defines the number (and size) of input files of the next job, and affect its processing efficiency. There is also performance trade-offs that should be taken into consideration: depending on the application property and the input data size, a nearly optimal completion time might be achieved with a relatively small amount of the cluster resources.

To address the problem, we designed and implemented an automatic performance optimization tool, called *AutoTune*, that automates the user efforts of tuning the numbers of reduce tasks along the MapReduce workflow based on the performance modeling framework we proposed. It adopts two optimization strategies to achieve trade-offs between the workflow completion time and the resource usage for its execution: a local one with trade-offs at the job level, and a global one that makes the optimization trade-off decisions at the workflow level.

3 Future Work

Below, I briefly outline my future research agenda, describing short-term extensions to my dissertation work as well as my long-term research plan.

3.1 Extensions to Dissertation

As extensions to my dissertation work, I intend to generalize the framework especially in extending it for managing resource in heterogeneous cluster and handling data skews during processing.

Resource management in heterogeneous environment. Practical clusters typically grow incrementally over time, which result in heterogeneous cluster that contains different types of nodes. We have validated that our bounds based analytical performance model also works for heterogeneous environments [8].

An interesting follow up is to enhance our resource management strategy in such heterogeneous environment. Since different nodes in a heterogeneous cluster have different CPU/memory/network capacity, the performance of the same MapReduce job could vary significantly when executed on different nodes, e.g., computing intensive job could benefit more when executed on nodes with more powerful CPUs. To address the new challenge, I plan to extend our resource management strategy to determine the right type of nodes that should be allocated to a given MapReduce job as well as the minimal amount of the nodes to achieve the completion time target. The strategy will be leveraged in our job scheduler to make the right resource assignment decisions in a heterogeneous cluster according to the job characteristics.

Handling data skew. Data skew occurs commonly in enterprise data which results in unbalanced work and execution time for different map/reduce tasks. It causes the "stragglers" problem during MapReduce job execution that results in longer job completion time and slows down the workflow processing. It also leads to inaccuracy of our performance model. To handle the case of skewed data, I plan to enhance the existing approach by profiling the distribution of reduce keys in the map outputs and improve the accuracy of the completion time prediction with the estimated input data size for each task based on the distribution we profiled. I am also going to investigate the possibility to mitigate the unbalance work through tuning the job settings.

3.2 Looking Beyond

Looking beyond my dissertation work, my long-term research goal is to enable automated resource management and optimization for complex parallel computing in large-scale distributed systems. My dissertation work is the first step towards this vision, and I would like to enhance the current framework to further explore the following areas.

Dynamic resource management framework. Building on the current performance modeling framework, I hope to extend it towards a more general resource management and optimization framework which dynamically allocates different types of resources according to the characteristics of MapReduce jobs and different service level objectives (e.g., completion time, cost, energy consumption). The resources considered could be defined in details by its computing, communication and storage capacity and provided by different service providers. The framework should also be able to adapt to the change of workloads and system utilization by dynamically adding or removing available resources in an elastic computing environment.

Generalizing the framework for parallel data processing in distributed systems. Our performance modeling framework is built on the MapReduce and Hadoop architecture. However, the methodology we provided should not be restricted to this specific platform. As a future work, I plan to extend the existing approaches on different data-parallel middleware platforms in distributed systems such as Dryad and Spark and explore the possibility to generalize the framework to support different platforms.

Performance modeling in public cloud with virtualization. Today's public cloud platforms make extensive use of virtualization across computing storage, and network resources. An interesting trend that has emerged in recent years is the virtualization of the network layer, first demonstrated by the use of the OpenFlow API as part of the Software Defined Networking (SDN) stack, and more recently, the proposed use of Network Function Virtualization (NFV) to virtualize network services. These new innovations aim at making cloud service deployment easier, but also introduce a new set of challenges related to SLA guarantees in a multi-tenant setting. An interesting avenue of work that I plan to explore, is to develop novel performance models and resource allocation strategies that can take into considerations the high degrees of variance in highly virtualized environments.

4 Other Work

In addition to the research work around the performance modeling and optimization for MapReduce workflows, I also worked on a variety of other projects in both academic and industrial settings. I provide here a brief summary of each project:

Real time scheduling for MapReduce jobs. The emergence of cloud-based data analytics with timeliness requirements has necessitated a strong need for real-time support on the clouds which dose not provided by the current Hadoop scheduler. To address the problem, I first performed empirical experiments to understand the factors that are important to the map/reduce task completion time and its predictability, and then adapted existing multiprocessor scheduling techniques from the realtime systems domain (e.g., Earliest Deadline First (EDF)) to the cloud setting and evaluated their abilities in satisfying application time requirements [2, 1].

Job rescheduling on the Intel distributed computing platform. This project is based on Intel's NetBatch system, an Internet-scale computing infrastructure developed for running concurrently tens of thousands of chip simulations One important challenge that NetBatch faces today is the need to accommodate jobs with varying priorities and goals while keeping both the system utilization high and latency low. To address the challenge, I first analyzed job execution traces collected from tens of thousands of machines over a year-long period to identify performance bottlenecks and then performed trace-driven simulations to study the impact of various job scheduling algorithms on their platform. I also proposed an online job rescheduling strategy which reduces the average job completion time in data centers with unbalanced utilizations [13].

References

[1] Linh T. X. Phan, **Zhang, Zhuoyao**, Qi Zheng, Boon Thau Loo, and Insup Lee. An empirical analysis of scheduling techniques for real-time cloud-based data processing. In *Proceedings of the 2011 IEEE International Conference on Service-Oriented Computing and Applications*, SOCA '11, pages 1–8, 2011.

[2] Linh T.X. Phan, **Zhuoyao Zhang**, Boon Thau Loo, and Insup Lee. Real-time MapReduce Scheduling. Technical Report MS-CIS-10-32, Philadelphia, PA, 2010.

[3] **Zhang, Zhuoyao**, Ludmila Cherkasova, and Boon Thau Loo. Autotune:optimizing execution concurrency and resource usage in mapreduce workflows. In *Proceedings of the 10th USENIX International Conference on Autonomic Computing*, ICAC '13.

[4] **Zhang, Zhuoyao**, Ludmila Cherkasova, and Boon Thau Loo. Getting more for less in optimized mapreduce workflows. In *Proceedings of IFIP/IEEE international symposium on integrated network management*, IM '13.

[5] **Zhang, Zhuoyao**, Ludmila Cherkasova, and Boon Thau Loo. Parameterizable benchmarking framework for designing a mapreduce performance model (under submission). *Concurrancy and Computation: Practice and Experience.*

[6] **Zhang, Zhuoyao**, Ludmila Cherkasova, and Boon Thau Loo. Benchmarking approach for designing a mapreduce performance model. In *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering*, ICPE '13, pages 253–258, 2013.

[7] **Zhang, Zhuoyao**, Ludmila Cherkasova, and Boon Thau Loo. Performance modeling and optimization of deadline-driven pig programs. *ACM Transactions on Autonomous and Adaptive Systems (ACM TAAS)*, 2013.

[8] **Zhang, Zhuoyao**, Ludmila Cherkasova, and Boon Thau Loo. Performance modeling of mapreduce jobs in heterogeneous environments. In *Proceedings of the 6th IEEE International Conference on Cloud Computing*, CLOUD '13, 2013.

[9] **Zhang, Zhuoyao**, Ludmila Cherkasova, and Boon Thau Loo. Optimizing cost and performance trade-offs for mapreduce job processing in the cloud (under submission). In *Proceedings of IEEE/IFIP Netwok Operations and Management Symposium*, NOMS '14, 2014.

[10] **Zhang, Zhuoyao**, Ludmila Cherkasova, Abhishek Verma, and Boon Thau Loo. Automated profiling and resource management of pig programs for meeting service level objectives. In *Proceedings of the 9th international conference on Autonomic computing*, ICAC '12, pages 53–62, 2012.

[11] **Zhang, Zhuoyao**, Ludmila Cherkasova, Abhishek Verma, and Boon Thau Loo. Meeting service level objectives of pig programs. In *Proceedings of the 2nd International Workshop on Cloud Computing Platforms*, CloudCP '12, pages 8:1–8:6, 2012.

[12] **Zhang, Zhuoyao**, Ludmila Cherkasova, Abhishek Verma, and Boon Thau Loo. Optimizing completion time and resource provisioning of pig programs. In *Workshop on Cloud Computing Optimization*, CCOPT '12, pages 811–816, 2012.

[13] **Zhang, Zhuoyao**, Linh T. X. Phan, Godfrey Tan, Saumya Jain, Harrison Duong, Boon Thau Loo, and Insup Lee. On the feasibility of dynamic rescheduling on the intel distributed computing platform. In *Proceedings of the 11th International Middleware Conference Industrial track*, Middleware '10, pages 4–10, 2010.