



3-3-2018

Efficient Redundancy Techniques in Cloud and Desktop Grid Systems using MAP/G/c-type Queues

Srinivas R. Chakravarthy
Kettering University, schakrav@kettering.edu

Alexander Rummyantsev
Institute of Applied Mathematical Research, Karelian Research Centre of RAS, ar0@krc.karelia.ru

Follow this and additional works at: https://digitalcommons.kettering.edu/industrialmanuf_eng_facultypubs

 Part of the [Industrial Engineering Commons](#)

Recommended Citation

Chakravarthy, Srinivas R. and Rummyantsev, Alexander, "Efficient Redundancy Techniques in Cloud and Desktop Grid Systems using MAP/G/c-type Queues" (2018). *Industrial & Manufacturing Engineering Publications*. 79.
https://digitalcommons.kettering.edu/industrialmanuf_eng_facultypubs/79

This Article is brought to you for free and open access by the Industrial & Manufacturing Engineering at Digital Commons @ Kettering University. It has been accepted for inclusion in Industrial & Manufacturing Engineering Publications by an authorized administrator of Digital Commons @ Kettering University. For more information, please contact digitalcommons@kettering.edu.



Srinivas R. Chakravarthy* and Alexander Rumyantsev*

Efficient Redundancy Techniques in Cloud and Desktop Grid Systems using *MAP/G/c*-type Queues

<https://doi.org/10.1515/eng-2018-0004>

Received October 4, 2017; accepted November 16, 2017

Abstract: Cloud computing is continuing to prove its flexibility and versatility in helping industries and businesses as well as academia as a way of providing needed computing capacity. As an important alternative to cloud computing, desktop grids allow to utilize the idle computer resources of an enterprise/community by means of distributed computing system, providing a more secure and controllable environment with lower operational expenses. Further, both cloud computing and desktop grids are meant to optimize limited resources and at the same time to decrease the expected latency for users. The crucial parameter for optimization both in cloud computing and in desktop grids is the level of redundancy (replication) for service requests/workunits. In this paper we study the optimal replication policies by considering three variations of Fork-Join systems in the context of a multi-server queueing system with a versatile point process for the arrivals. For services we consider phase type distributions as well as shifted exponential and Weibull. We use both analytical and simulation approach in our analysis and report some interesting qualitative results.

Keywords: queueing, desktop grids, distributed computing, cloud computing, matrix-analytic method, Markovian arrival process, phase type distribution, heavy tails, simulation

1 Introduction and Model Description

During early stages of growth, small and medium enterprises face the problem of procuring the required computing power. The main alternatives to the most expensive option of owning and running datacenter are the cloud computing (CC) and desktop grids (DG).

According to *NIST* [25], CC “is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.” According to Murugesan and Bojanova [26], CC “offers huge computing power, on-demand scalability, and utility-like availability at low cost”. While there are many types of CC [7], in the public cloud the users are provided requisite computing resources in a dynamic way over the Internet with the help of Web services. These Web services are provided generally by third-party vendors so as to have low cost access to users. Thus, the service providers are required to optimize the limited resources and at the same time decrease the expected latency for its users.

DG is a specifically designed, inexpensive, and powerful option for tasks that (a) require huge computational resources and (b) may be split into a large number of loosely coupled subtasks (known as *workunits*). The computational resources are harvested from the desktops, tablets, GP-GPUs, and servers owned by volunteers (in case of the so-called volunteer computing [33]) or by enterprise itself (the Enterprise DG [14]), utilizing the idle times of the aforementioned hosts. The diversity of computational resources makes time to complete the task highly unpredictable. Further, computation is only one of many steps involved in solving applied research problems and hence reduction of the expected latency is an important aspect of DG computing.

A common mechanism used in the CC as well as DG is *replication*. Each workunit is processed by multiple hosts until the *quorum* (required number of valid results from these hosts) is obtained. Replication reduces

*Corresponding Author: Srinivas R. Chakravarthy: Department of Industrial and Manufacturing Engineering, Kettering University, Flint, MI 48504, USA, E-mail: schakrav@kettering.edu

*Corresponding Author: Alexander Rumyantsev: Institute of Applied Mathematical Research, Karelian Research Centre of RAS, 11 Pushkinskaya Str., Petrozavodsk, 185910, Russia, Petrozavodsk State University, 33 Lenina Pr., Petrozavodsk, 185910, Russia, E-mail: ar0@krc.karelia.ru

expected latency [13, 15], whereas quorum reduces the probability of malicious activity and increases application turnaround [35]. Note that replication significantly complicates the model and for this reason only recently researchers have started applying queueing theory to study CC [8, 9, 18] and DG [6].

One of the possible models to study the concepts of replication and quorum may be seen in the class of Fork-Join (FJ) systems. In a classical c -server FJ system an arriving job is split into c tasks and each of these is sent to one of the c (homogeneous) servers, each having a distinct queue. Service of a job is completed as soon as all c tasks of the job have completed their services (see e.g., [2, 8, 20, 27]). In [17] a generalization of the FJ queueing model is considered (referred to as (c, k) -FJ system), wherein an arriving job, split into c tasks and sent to each of c servers, is said to have completed the service when any k out of c tasks are completed. Note that (c, c) -FJ is the classical FJ system. In [17], the Fork-Early-Cancel variation of the FJ system (referred to as (c, k) -FEC) is considered, wherein an arriving job, split into c tasks and sent to each of c queues, waits until any k of c tasks start the service, and at that moment the remaining $(c - k)$ redundant tasks are canceled. A detailed comparison of FJ and FEC models is performed in [18]. Summarizing the important results from [16–18]),

- (a) $(c, 1)$ -FJ system is equivalent to a single server system with first-come-first-served (FCFS) discipline, in which the service times are given by the minimum of c iid. service times;
- (b) $(c, 1)$ -FEC system is equivalent to a c -server system with FCFS discipline;
- (c) (c, k) -FJ is upper-bounded by (c, k) -Split-Merge (SM) system (in which the replicas of a job are not allowed to start service before the previous job completes its service), equivalent to a single server system, in which the service times are obtained as the k -th order statistics of c iid. random variables.

In [37] the authors study a multiserver FJ-system with Markovian arrival process (MAP), phase type (PH) services, task vacations and intercommunication times (modeled as PH random variables). An approximate solution based on stochastic decomposition is derived, and examples are presented comparing the approximation with simulation results under four dynamic scheduling policies for a few scenarios. We also note an interesting overview of FJ and related systems [39]. Specifically, in this survey paper, the author reviews the queueing systems related to FJ systems, the analysis of certain Markovian queues, methods to estimate the expected value of the maximum (which

play an important role in CC and DG models) of different distributions, and approximations for the mean delay.

The focus of our paper is to study the three FJ systems: (i) $(c, 1)$ -FJ system, (ii) $(c, 1)$ -FEC system, and (iii) (c, k) -SM systems, $1 \leq k \leq c$, under the assumptions that the jobs arrive according to a MAP.

The MAP, introduced as a versatile Markovian point process by Neuts [28] and later simplified using less notations by Lucantioni, et.al. [23], is described by an underlying (continuous-time) Markov chain with generator, say, D , of dimension m such that $D = D_0 + D_1$. Note that the matrix D_0 governs transitions without arrivals and the matrix D_1 governs transitions inducing arrivals to the system. It should be pointed out that MAP is a rich class of point processes and includes many classical processes such as Poisson, PH-renewal processes, and Markov-modulated Poisson process. For more details on MAP and their applications in stochastic modelling, we refer to [22, 23, 30, 31]. Further, we refer the reader to [1, 4, 5] for a review and recent work on MAP.

We assume that all c servers are homogeneous and that the services they provide are PH with representation (β, S) of order n . In simulations, however, we consider general service time distribution.

In the sequel we need the following notations. By \mathbf{e} we denote a column vector of 1's; \mathbf{e}_i , a unit column vector with 1 in the i^{th} position and 0 elsewhere; and I , and identity matrix (of appropriate dimensions). Should there be a need to emphasize the dimension, we will do so like I_m rather than I , and similarly for the others. The notation \mathbf{S}^0 is such that $\mathbf{S}\mathbf{e} + \mathbf{S}^0 = \mathbf{0}$. The symbols, \otimes and \oplus , respectively, stand for the Kronecker product and Kronecker sum of matrices. For details and properties on Kronecker products and Kronecker sums we refer the reader to [11, 24, 38].

Suppose λ and μ , respectively, denote the arrival and service rates. It is easy to verify that $\lambda = \delta D_1 \mathbf{e}$ and $\mu = [\beta(-S)^{-1} \mathbf{e}]^{-1}$, where δ is the stationary probability vector of the irreducible generator D and is obtained as the unique (positive) probability vector satisfying $\delta D = \mathbf{0}$, $\delta \mathbf{e} = 1$. The expected latency $E(\mathcal{L})$ (defined as the average sojourn time of a job in the system) and the expected cost of computing $E(\mathcal{C})$ (which is taken to be the average service time of a job) are two performance measures that play a key role in identifying an optimum (replication) strategy.

2 The steady-state analysis

In this section we will study the FJ, SM and FEC systems as described in Section 1 in steady-state.

2.1 (c, 1)-FJ system

In this system, recall, an arriving job is split into c tasks and served by each of c servers until the earliest service completion of the task, when all other tasks of the job in service are removed from the system. This system is known to be equivalent to MAP/PH/1 queue [18], where the service time, obtained as the minimum of c identical (β, S) PH-distributions, has PH-distribution (α, T) of order n^c , given as follows [29]

$$\alpha = \underbrace{\beta \otimes \cdots \otimes \beta}_c, \quad T = \underbrace{S \oplus \cdots \oplus S}_c.$$

Let $\mu_T = [\alpha(-T)^{-1}\mathbf{e}]^{-1}$ denote the service rate. In order to study the model in this section as a continuous-time Markov chain (CTMC), we need to keep track of the number of jobs, $N(t)$ in the system, the phase of the service times, $J_r, 1 \leq r \leq c$, if any, and the phase of the arrival process, $M(t)$ at time $t \geq 0$. The state space of CTMC $\{(N(t), J_1(t), \dots, J_c(t), M(t)) : t \geq 0\}$ is given by

$$\tilde{\Omega} = \{\mathbf{i}, i \geq 0\},$$

where

- set $\mathbf{0} = \{k, 1 \leq k \leq m\}$ of dimension m corresponds to the case where the system is idle and the MAP process is in one of m phases;
- set $\mathbf{i} = \{(i, j_1, \dots, j_c, k), 1 \leq i, 1 \leq j_r \leq n, 1 \leq k \leq m\}$ of dimension mn^c corresponds to the case when a job is at service, $i - 1$ jobs are in the queue (if any), the service time at r^{th} server ($1 \leq r \leq c$) is in one of the n phases, the MAP process in one of the m phases.

The generator (see e.g., [30]) of the CTMC governing the MAP/PH/1 queue is of the form

$$\tilde{Q} = \begin{bmatrix} D_0 & \alpha \otimes D_1 & & & & & \\ \mathbf{T}^0 \otimes I & \tilde{A}_1 & \tilde{A}_0 & & & & \\ & \tilde{A}_2 & \tilde{A}_1 & \tilde{A}_0 & & & \\ & & \tilde{A}_2 & \tilde{A}_1 & \tilde{A}_0 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \ddots & \ddots & \ddots \end{bmatrix}, \quad (1)$$

where

$$\tilde{A}_0 = I_{n^c} \otimes D_1, \quad \tilde{A}_1 = T \oplus D_0, \quad \tilde{A}_2 = \mathbf{T}^0 \alpha \otimes I_m.$$

2.1.1 The steady-state probability vector

Towards this end, we define $\tilde{\mathbf{x}}$, partitioned as $\tilde{\mathbf{x}} = (\tilde{\mathbf{x}}(0), \tilde{\mathbf{x}}(1), \dots)$, to be the steady-state probability vector of generator \tilde{Q} given in (1). Note that \tilde{Q} is of QBD-type and thus possesses matrix-geometric steady-state probability vector [29] (for details on QBD processes see [1, 4]). Specific structure of the generator given in (1), allows to obtain the vector $\tilde{\mathbf{x}}$ as follows.

Theorem 1. Under the stability condition $\lambda < \mu_T$, the steady-state probability vector $\tilde{\mathbf{x}}$ is of modified matrix-geometric type. Specifically, we have

$$\begin{aligned} \tilde{\mathbf{x}}(0)D_0 + \tilde{\mathbf{x}}(1)(\mathbf{T}^0 \otimes I) &= \mathbf{0}, \\ \tilde{\mathbf{x}}(0)(\alpha \otimes D_1) + \tilde{\mathbf{x}}(1)[\tilde{A}_1 + \tilde{R}\tilde{A}_2] &= \mathbf{0}, \\ \tilde{\mathbf{x}}(i) &= \tilde{\mathbf{x}}(1)\tilde{R}^{i-1}, \quad i \geq 1, \end{aligned}$$

where \tilde{R} is the minimal non-negative solution to the matrix-quadratic equation:

$$\tilde{R}^2\tilde{A}_2 + \tilde{R}\tilde{A}_1 + \tilde{A}_0 = \mathbf{0}. \quad (2)$$

and the normalizing equation is given by

$$\tilde{\mathbf{x}}(0)\mathbf{e} + \tilde{\mathbf{x}}(1)(I - \tilde{R})^{-1}\mathbf{e} = 1.$$

Proof: The proof follows by applying the matrix-geometric results as seen in [29].

2.1.2 Expected latency and the cost of computing

The expected latency, $E(\mathcal{L})$ and the expected cost of computing, $E(\mathcal{C})$, for the FJ system are given by [29]

$$E(\mathcal{L}) = \frac{1}{\lambda} \tilde{\mathbf{x}}(1)(I - \tilde{R})^{-2}\mathbf{e}, \quad E(\mathcal{C}) = \frac{c}{\mu_T}.$$

Remark 1. In a MAP/M/1-type system, the expected cost equals $E(\mathcal{C}) = \mu^{-1}$ since the minimum of c exponentials with rate μ is exponential with rate $\mu_T = c\mu$.

We note that in general $E(\mathcal{C})$ depends on c and the type of dependence is related to the so-called log-concavity (log-convexity) of the service time distribution, namely, the following lemma holds [18]:

Lemma 1. If the service time of a task has log-concave (log-convex) distribution, then $E(\mathcal{C})$ is non-decreasing (non-increasing) in c .

2.1.3 Explicit solution for MAP(2)/M/1-type system

In general, the matrix \tilde{R} has to be obtained numerically with any of the iterative procedures (see e.g. [12]). However, in the case when m is small, one can obtain \tilde{R} explicitly using complex variable approach as presented in [10]. Suppose that we consider the arrival process to be a MAP with two phases ($m = 2$) and the service distribution of a single task of the job is exponential with rate μ . Thus, the system is equivalent to MAP/M/1-type system, with the service time of the job being exponentially distributed with rate $\mu_T = c\mu$ (see Remark 1), and the stability criterion reduces to $\lambda < c\mu$. In this case, \tilde{R} may be obtained explicitly and the details are as follows.

First, note that for the current special case $n = 1$, $\beta = 1$, $S = -\mu$, which implies $\alpha = 1$, $T = -c\mu$, and $T^0 = c\mu$. This simplifies the matrices \tilde{A}_i , $i = 0, 1, 2$ as follows:

$$\tilde{A}_0 = D_1, \quad \tilde{A}_1 = D_0 - c\mu I_2, \quad \tilde{A}_2 = c\mu I_2.$$

Now we briefly outline the procedure of obtaining \tilde{R} . The necessary details may be found in [10].

1. Write down the determinantal polynomial $\det(A(\xi)) := \det(\tilde{A}_0 + \xi\tilde{A}_1 + \xi^2\tilde{A}_2)$ for ξ complex.
2. Using trigonometric solution, obtain the greatest root ξ_3 of the third degree polynomial $\det(A(\xi))/(\xi - 1) = a_3\xi^3 + a_2\xi^2 + a_1\xi + a_0$, with roots known to be real.
3. Find $b_0 = -a_0/(a_3\xi_3)$, $b_1 = a_2/a_3 + \xi_3$.
4. Find \tilde{R} as follows:

$$\tilde{R} = [b_0\tilde{A}_2 - \tilde{A}_0][\tilde{A}_1 - b_1\tilde{A}_2]^{-1}.$$

After obtaining \tilde{R} , following (2), it is easy to obtain the explicit solution for the steady-state probability vector \tilde{x} . Thus, the value $E(\mathcal{L})$ may also be obtained exactly, while $E(\mathcal{C}) = \mu^{-1}$ (see Remark 1). It allows to evaluate the system performance for relatively large values of c .

To illustrate this approach, we evaluate the value $E(\mathcal{L})$ for $c = 1, \dots, 1000$, with an example of MAP(2) with parameter matrices (D_0, D_1) given by

$$D_0 = \begin{pmatrix} -4 & 2 \\ 2 & -5 \end{pmatrix}, \quad D_1 = \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix}.$$

In this case, verify that the stationary vector $\delta = (4/7, 3/7)$ and the fundamental rate $\lambda = 2\frac{4}{7}$. We also take $\mu = 3$ so that the system is stable for any $c \geq 1$. Then, we obtain \tilde{R} following the procedure above, and depict the dependence $E(\mathcal{L})$ on c . The results are displayed in Fig. 1, with logarithmic y-axis. It can be seen, that the $E(\mathcal{L})$ decreases rapidly with increasing number of replicas c .

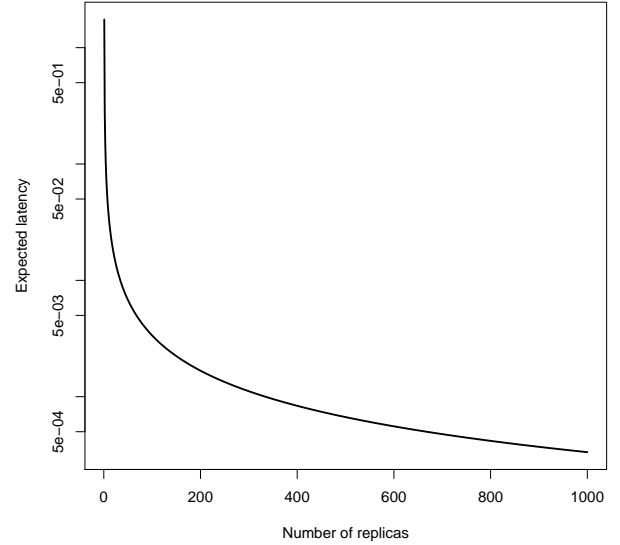


Figure 1: Reduced expected latency in MAP(2)/M/1-type system: the expected latency $E(\mathcal{L})$ (log. scale) vs. number of servers (or replicas), c .

2.2 FEC system

Here, we consider a system in which an arriving job is split into c tasks and each task is sent to each of the c servers. As soon as any one of the c tasks starts service, all other $(c - 1)$ redundant tasks waiting in the queue are canceled. The system corresponds to a queueing model of MAP/PH/ c -type. To obtain the two steady-state performance measures, we reuse a few notations from the FJ system. Let $N(t)$, J_r , $1 \leq r \leq c$, and $M(t)$ denote, respectively, the number of jobs in the system, the phase of service of the r^{th} server (ignored if that server is idle), and the phase of the arrival process, at time t . The process $\{(N(t), J_1(t), \dots, J_c(t), M(t)) : t \geq 0\}$ is a CTMC with the state space given by

$$\Omega_2 = \{\star\} \cup \{\hat{\mathbf{i}}, 1 \leq i \leq c - 1\} \cup \{\mathbf{i}, i \geq 0\},$$

where the set of states is defined as follows:

- The set $\star = \{k, 1 \leq k \leq m\}$ of dimension m corresponds to the case where the system is idle and the MAP process is in one of m phases.
- $\hat{\mathbf{i}} = \{(i, j_1, \dots, j_i, k), 1 \leq j_r \leq n, 1 \leq k \leq m\}$ of dimension mn^i corresponds to the case where i , $1 \leq i \leq c - 1$, servers are at service, with r^{th} ($r \leq i$) server busy in one of the n phases, and the MAP process is in one of the m phases.
- $\mathbf{i} = \{(i, j_1, \dots, j_c, k), 1 \leq j_r \leq n, 1 \leq k \leq m\}$ of dimension mn^c corresponds to the case where all c servers are busy with $i \geq 0$ jobs waiting in the queue; the MAP process is in one of m phases, the r^{th} ($r \leq c$) server busy in one of n phases.

is again a PH-distribution. However, the dimension of the PH-distribution of the k^{th} order statistics grows exponentially with k and hence in this paper we will resort to simulation for the steady-state analysis of the model. We refer the reader to [36] for an extended discussion of these type of models.

3 Illustrative Examples

The purpose of this section is to discuss a few illustrative examples to bring out the qualitative nature of the models under study. Towards this end, we consider five arrival processes and three service time distributions. These five MAPs and three PH-representations are as follows.

1. Erlang (ERLA):

$$D_0 = \begin{pmatrix} -2 & 2 \\ 0 & -2 \end{pmatrix}, D = \begin{pmatrix} 0 & 0 \\ 2 & 0 \end{pmatrix}$$

2. Exponential (EXPA):

$$D_0 = \begin{pmatrix} -1 \end{pmatrix}, D = \begin{pmatrix} 1 \end{pmatrix}$$

3. Hyperexponential (HEXA):

$$D_0 = \begin{pmatrix} -1.90 & 0 \\ 0 & -0.19 \end{pmatrix}, D = \begin{pmatrix} 1.71 & 0.19 \\ 0.171 & 0.019 \end{pmatrix}$$

4. MAP with negative correlation (MNCA):

$$D_0 = \begin{pmatrix} -1.00222 & 1.00222 & 0 \\ 0 & -1.00222 & 0 \\ 0 & 0 & -225.75 \end{pmatrix},$$

$$D = \begin{pmatrix} 0 & 0 & 0 \\ 0.01002 & 0 & 0.9922 \\ 223.4925 & 0 & 2.2575 \end{pmatrix}$$

5. MAP with positive correlation (MPCA):

$$D_0 = \begin{pmatrix} -1.00222 & 1.00222 & 0 \\ 0 & -1.00222 & 0 \\ 0 & 0 & -225.75 \end{pmatrix},$$

$$D = \begin{pmatrix} 0 & 0 & 0 \\ 0.9922 & 0 & 0.01002 \\ 2.2575 & 0 & 223.4925 \end{pmatrix}.$$

Note first that the first three arrival processes, namely, *ERLA*, *EXPA*, and *HEXA*, have zero correlation for two successive inter-arrival times. The arrival processes labeled *MNCA* and *MPCA*, respectively, have negative and

positive correlation (with values -0.4889 and 0.4889) for two successive inter-arrival times. The ratio of the standard deviation of the inter-arrival times of these five arrival processes with respect to *ERLA* are, respectively, 1, 1.41421, 3.17451, 1.99336, and 1.99336. The above MAP processes will be normalized so as to have a specific arrival rate.

For the service times we consider the following three (β , S) PH-distributions. These distributions will be normalized so as to arrive at a desired value for μ .

A. Erlang (ERLS) :

$$\beta = (1, 0), S = \begin{pmatrix} -2 & 2 \\ 0 & -2 \end{pmatrix}.$$

B. Exponential (EXPS) :

$$\beta = (1), S = \begin{pmatrix} -1 \end{pmatrix}.$$

C. Hyperexponential (HEXS) :

$$\beta = (0.9, 0.1), S = \begin{pmatrix} -10 & 0 \\ 0 & -1 \end{pmatrix}.$$

Example 1: The purpose of this example is to see the behavior of $E(\mathcal{L})$ and $E(\mathcal{C})$ under different arrival processes and service times. Towards this end, we fix $\lambda = 0.8$, $\mu = 1.0$, vary $c = 1, \dots, 5$ and plot the measures $\ln(E(\mathcal{L}))$ and $E(\mathcal{C})$ under various scenarios, in Fig. 2.

First, we want to point out that in [18] for the case of Poisson arrivals, it was shown that, as c increases, (a) for log-concave type service distribution, $E(\mathcal{L})$ decreases and $E(\mathcal{C})$ increases; (b) for exponential $E(\mathcal{L})$ decreases at no additional cost, that is, $E(\mathcal{L})$ decreases while $E(\mathcal{C})$ remains constant; (b) log-convex type service distribution $E(\mathcal{L})$ and $E(\mathcal{C})$ both decrease.

We notice from Fig. 2, that (for all arrival processes considered) for Erlang (log-concave) services, $\ln(E(\mathcal{L}))$ decreases and $E(\mathcal{C})$ increases with increasing c . We also notice that *MPCA* arrivals appears to produce the largest $E(\mathcal{C})$ as compared to other arrivals. In the case of hyperexponential (log-convex) services both $\ln(E(\mathcal{L}))$ and $E(\mathcal{C})$ decrease with increasing c . Note that the results agree with observed in [18]. It is also worth pointing out that among renewal arrivals, the one with the largest coefficient of variation, namely, hyperexponential, yields the largest $E(\mathcal{L})$.

Example 2: The purpose of this example is to compare FJ and FEC systems. In order to do this, we vary both λ and c and plot $E(\mathcal{L})$ in Fig. 3 and 4, respectively, for *ERLS* and *HEXS* under various scenarios.

Looking at these two figures, we observe the following key points.

- Obviously, having a larger variability in services (*HEXS*) yields a higher value for $E(\mathcal{L})$ as compared to the ones with smaller variability (*ERLS*).
- When considering Erlang services (see Fig. 3), we notice that for both Erlang and hyperexponential arrivals, a pattern (similar to the one observed in [18]) with regard to $E(\mathcal{L})$ when comparing the two systems, FJ and FEC. Namely, starting from some input rate (respectively, with increasing load) the FJ system gives smaller $E(\mathcal{L})$ compared to FEC. The crossing points of the two curves for these systems depend on the load (and hence, on the ratio of λ and c). Also, the crossing points for hyperexponential arrivals are less than the corresponding ones for Erlang arrivals. However, for positively correlated arrivals (*MPCA*) we see that FJ system appears to yield a smaller value compared to those of FEC system, and furthermore, for large λ values the difference is even more apparent.
- When looking hyperexponential services (see Fig. 4), we notice that for all combinations, FJ system appears to yield a much smaller expected latency as compared to the corresponding FEC system. This is intuitively clear, since having a highly variable services such as hyperexponential will yield a smaller service time for FJ system compared to FEC system. While this has been observed in ([18]) for the case of Poisson arrivals and for log-concave and log-convex type services, we notice here the same phenomenon for other types of arrival processes. But more than this observation, the fact that for non-Poisson arrivals and non-exponential services the measure $E(\mathcal{L})$ appears to differ significantly for the two systems: FJ and FEC.

It is worth mentioning here that when the services are exponential, we noticed that FJ system yields a higher value for $E(\mathcal{L})$ when compared to the corresponding FEC system for all combinations and for all values of λ . When λ is increased (keeping the stability) $E(\mathcal{L})$ appears to approach the same value for both systems. To save space, we do not provide the figures here, since they do not provide additional insight.

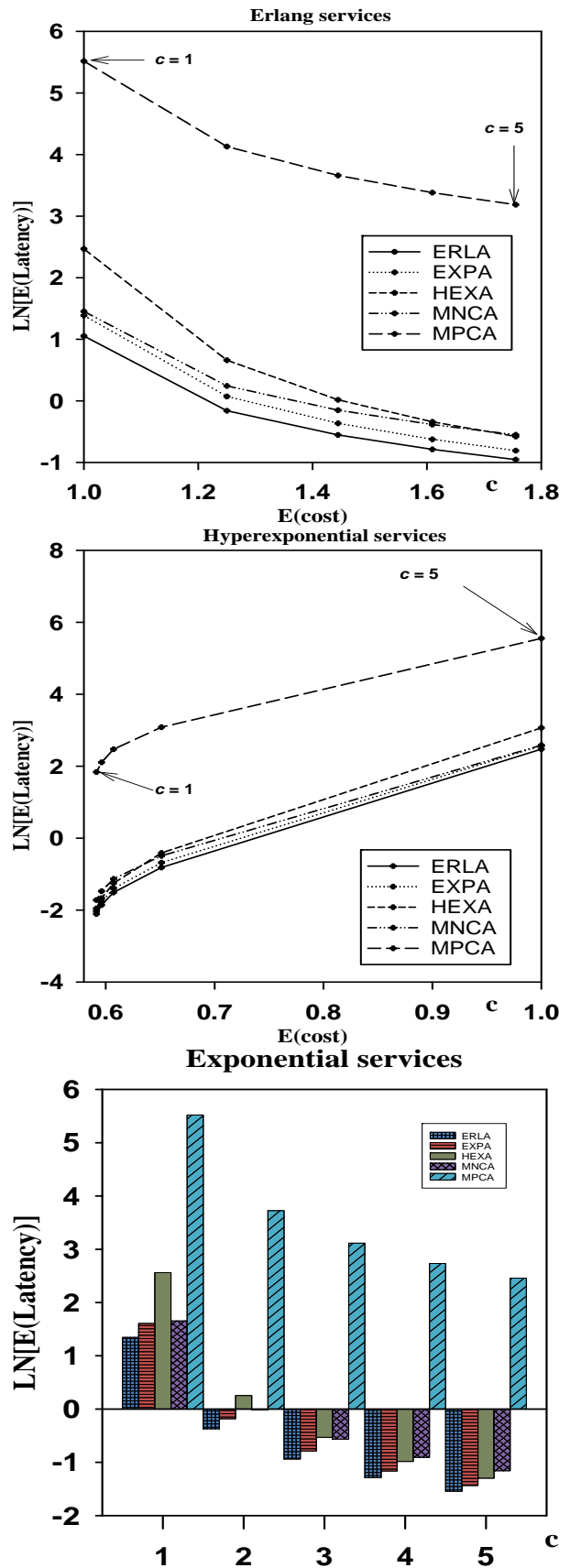


Figure 2: $\ln(E(\mathcal{L}))$ vs $E(c)$ as c is varied under different scenarios

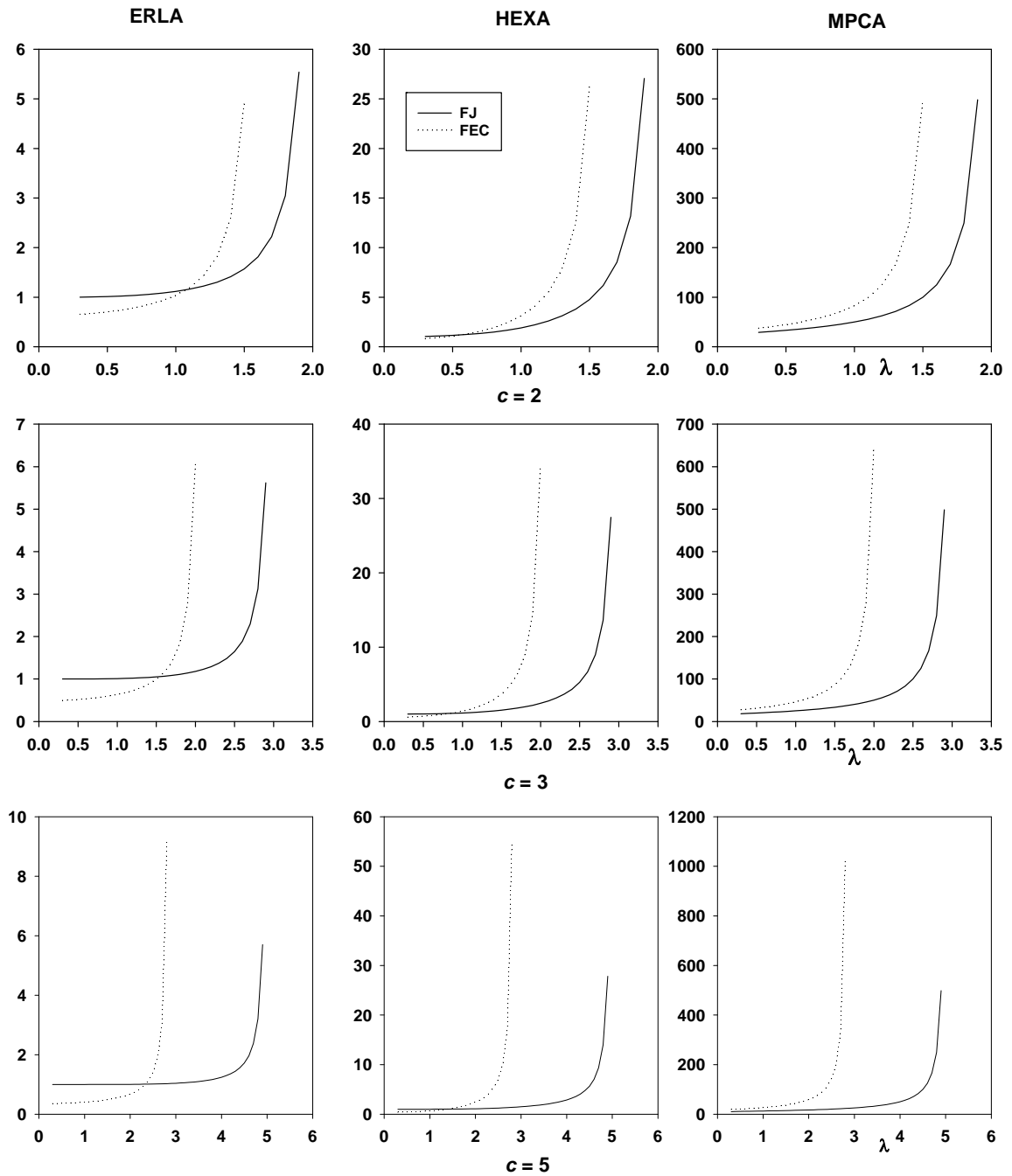


Figure 3: Plot of $E(\mathcal{L})$ under various scenarios for Erlang services

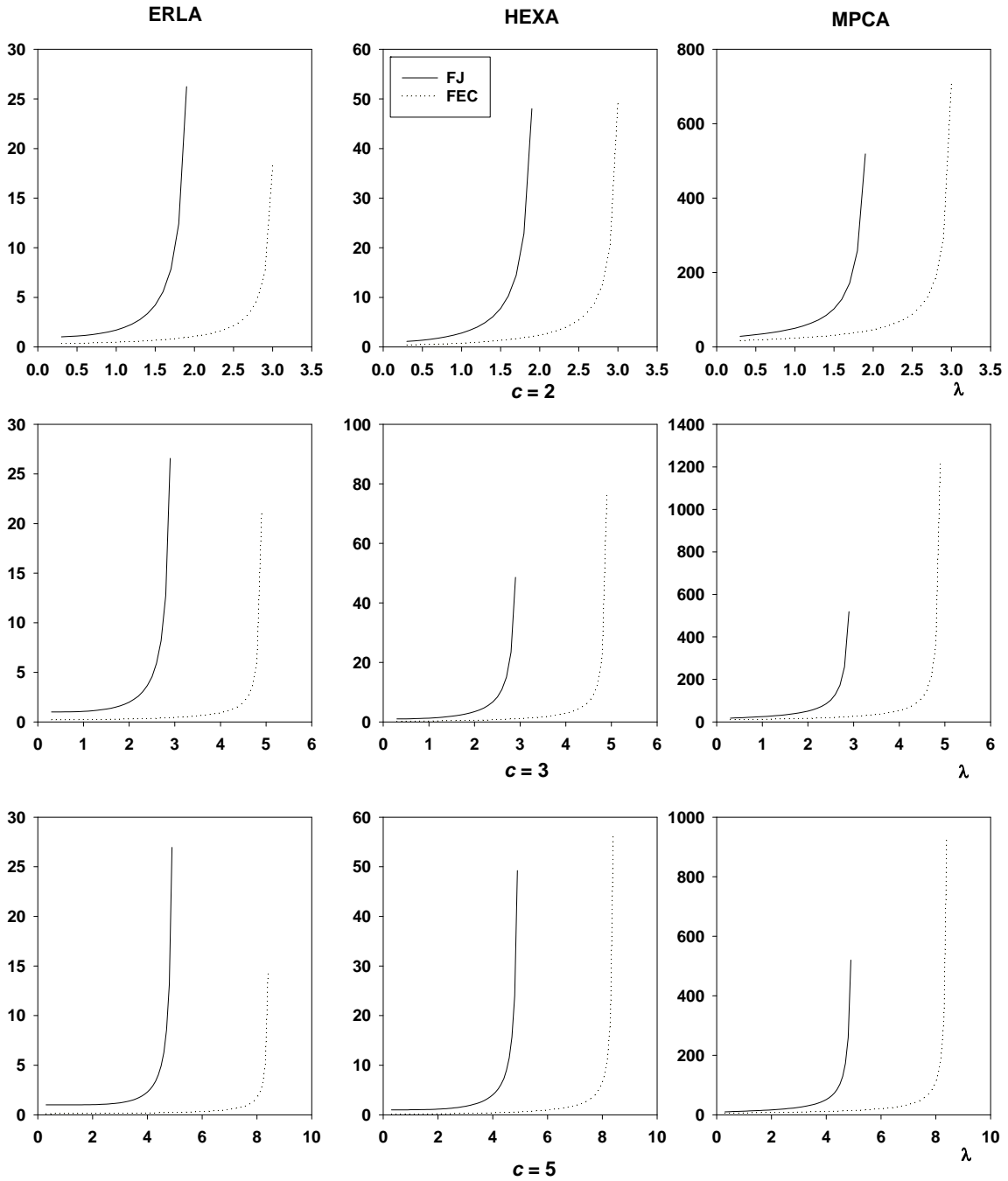


Figure 4: Plot of $E(\mathcal{L})$ under various scenarios for hyperexponential services

4 Simulation

So far, we studied FJ and FEC systems analytically. However, the state space of the CTMC governing the system under study grows exponentially with c , and hence we turn to simulation to study the systems for large values of c . To complete the experiments, we validated our simulation models with the analytical ones. Towards this end, we consider the arrival processes and service time distributions defined in Section 3. We fix $\lambda = 0.8$, $\mu = 1.0$, and vary c from 1 to 5. We compare the analytical results for FJ and FEC systems with the simulated results, for the same set of parameter values. The simulation models were implemented using ARENA [19], a powerful software used in applied stochastic modeling and other areas. We made five simulation runs with each run lasting 1,000,000 units of model time.

In Tables 1 and 2, respectively, we display the error percentages (i.e., $error = |analytical - simulated|/analytical$) for $E(\mathcal{L})$ and $E(\mathcal{C})$ (in parentheses), for the FJ and FEC systems. Generally the simulated results and the corresponding analytical results agree closely, the error percentages are less than 1%. We also denoted the scenarios with error percentages from 1.09% to 6.65%, and re-run the simulations with 10,000,000 units of model time, which provided a significant error reduce (e.g. for MPCA arrivals, EXPS services, and $c = 1$, the error percentage dropped from 6.65% to 0.92%).

Having validated the simulated models with the analytical models for the FJ and FEC systems, we now simulate these two models when dealing with large values of c .

Example 3: We fix $\lambda = 0.9$, $\mu = 1.0$, take $c = 1, \dots, 10, 15, 20, 50$ and plot the measures $E(\mathcal{C})$ and $E(\mathcal{L})$ under various scenarios, in Fig. 5 and Fig. 6, respectively. It is clear from these figures that

- for the FJ system, $E(\mathcal{C})$ increases as the number of servers (or replication parameter) is increased only for the case of ERLS. It decreases to a constant for the HEXS, and we stress the fast decrease for low values of c . Note that we plot the theoretical value $1/\mu$ for the EXPS, since the result follows analytically. In general, $E(\mathcal{C})$ depends on the type of service time distribution, which agrees to Remark 1 and Lemma 1, since the ERLS is log-concave, while HEXS is log-convex.
- for the FEC system, $E(\mathcal{C})$ is given by $\frac{1}{\mu}$ confirming the theoretical result (see Section 2.2.2).
- we see an interesting pattern among different scenarios for the measure $E(\mathcal{L})$. This measure appears to be larger for FEC compared to that of FJ system for all scenarios except for MPCA arrivals and ERLS services.

In this case (i.e., for MPCA with ERLS) we see that FJ system yields a higher $E(\mathcal{L})$. This indicates that when services have a coefficient of variation less than one and for positively correlated type of arrivals, one may need a larger number of servers as compared to other scenarios.

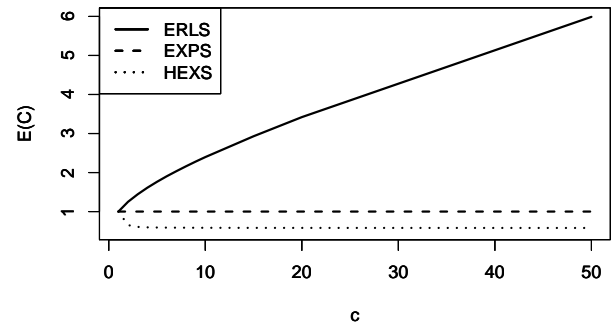


Figure 5: Plot of $E(\mathcal{C})$ under various scenarios

4.1 Simulation of (c, k) -SM System

In this section we concentrate on simulating (c, k) -SM system under various scenarios. In addition to the exponential service time distribution EXPS mentioned earlier, we also consider a log-concave and a heavy-tailed one. The cumulative distribution functions (CDF) follow:

D. Shifted Exponential (SEXS): The shifted exponential with a shift of magnitude $\Delta > 0$ is one with CDF

$$F_{SE}(x) = \begin{cases} 1 - e^{-\mu(x-\Delta)}, & x \geq \Delta, \\ 0, & x < \Delta. \end{cases}$$

E. Weibull (WEIB): The 2-parameter Weibull considered here has the CDF

$$F_{WB}(x) = \begin{cases} 1 - e^{-(2x)^{0.5}}, & x \geq 0, \\ 0, & x < 0. \end{cases}$$

First, note that the simulated mean (μ'_S) and standard deviation (σ_S) when using the three service distributions are given in Table 3 below. In the example below, we simulated the models using three simulation runs and each run is done to cover 500,000 jobs leaving the system.

Example 4: In this example we look at (c, k) -SM system by considering $c = 3, 4, 5$, vary $k = 1, \dots, c$ and vary $\rho = \lambda\mu'_S = 0.2, 0.5, 0.9, 0.95$, by choosing λ appropriately using the values of μ'_S displayed in Table 3 above. Note that the values of λ will be dependent of the type of arrival process as well as the service distribution. However, using ρ as a common parameter we can compare various scenarios to bring out the qualitative aspects of the system under

Table 1: Error percentages for $E(\mathcal{L})(E(\mathcal{C}))$ under various scenarios for FJ system

$PH - D$	c	ERLA	EXPA	HEXA	MNCA	MPCA
ERLS	1	0.45%(0.03%)	0.08%(0.07%)	0.28%(0.00%)	0.32%(0.03%)	6.30%(0.02%)
	2	0.15%(0.06%)	0.33%(0.10%)	0.06%(0.06%)	0.10%(0.06%)	3.02%(0.10%)
	3	0.20%(0.01%)	0.21%(0.01%)	0.43%(0.07%)	0.09%(0.02%)	3.74%(0.06%)
	4	0.18%(0.15%)	0.09%(0.16%)	0.16%(0.17%)	0.03%(0.18%)	2.20%(0.11%)
	5	0.05%(0.47%)	0.20%(0.47%)	0.10%(0.44%)	0.41%(0.48%)	2.93%(0.43%)
EXPS	1	0.56%(0.00%)	0.21%(0.04%)	0.58%(0.10%)	0.13%(0.02%)	6.65%(0.13%)
	2	0.19%(0.09%)	0.05%(0.03%)	0.11%(0.00%)	0.03%(0.01%)	3.83%(0.10%)
	3	0.14%(0.01%)	0.01%(0.01%)	0.09%(0.07%)	0.07%(0.06%)	0.48%(0.00%)
	4	0.10%(0.11%)	0.16%(0.12%)	0.24%(0.08%)	0.07%(0.04%)	0.64%(0.04%)
	5	0.24%(0.02%)	0.09%(0.03%)	0.09%(0.01%)	0.02%(0.03%)	1.31%(0.05%)
HEXS	1	1.31%(0.03%)	1.63%(0.14%)	0.30%(0.08%)	0.50%(0.04%)	0.91%(0.01%)
	2	0.41%(0.03%)	0.37%(0.01%)	0.16%(0.06%)	0.05%(0.01%)	0.49%(0.03%)
	3	0.23%(0.01%)	0.03%(0.01%)	0.00%(0.03%)	0.09%(0.03%)	0.02%(0.05%)
	4	0.40%(0.09%)	0.07%(0.09%)	0.14%(0.05%)	0.00%(0.00%)	1.39%(0.04%)
	5	0.46%(0.08%)	0.03%(0.04%)	0.03%(0.04%)	0.12%(0.09%)	0.60%(0.02%)

Table 2: Error percentages for $E(\mathcal{L})(E(\mathcal{C}))$ under various scenarios for FEC system

$PH - D$	c	ERLA	EXPA	HEXA	MNCA	MPCA
ERLS	1	0.45%(0.00%)	0.08%(0.00%)	0.28%(0.00%)	0.32%(0.00%)	6.30%(0.00%)
	2	0.02%(0.00%)	0.03%(0.00%)	0.14%(0.00%)	0.07%(0.00%)	0.38%(0.00%)
	3	0.01%(0.00%)	0.00%(0.00%)	0.02%(0.00%)	0.05%(0.00%)	0.72%(0.00%)
	4	0.02%(0.00%)	0.01%(0.00%)	0.01%(0.00%)	0.04%(0.00%)	0.12%(0.00%)
	5	0.02%(0.00%)	0.02%(0.00%)	0.01%(0.00%)	0.03%(0.00%)	0.96%(0.00%)
EXPS	1	0.56%(0.00%)	0.21%(0.00%)	0.58%(0.00%)	0.13%(0.00%)	6.65%(0.00%)
	2	0.01%(0.00%)	0.08%(0.00%)	0.29%(0.00%)	0.14%(0.00%)	1.62%(0.00%)
	3	0.02%(0.00%)	0.02%(0.00%)	0.10%(0.00%)	0.03%(0.00%)	1.09%(0.00%)
	4	0.03%(0.00%)	0.02%(0.00%)	0.00%(0.00%)	0.01%(0.00%)	0.83%(0.00%)
	5	0.03%(0.00%)	0.04%(0.00%)	0.01%(0.00%)	0.03%(0.00%)	1.44%(0.00%)
HEXS	1	1.31%(0.00%)	1.63%(0.00%)	0.30%(0.00%)	0.50%(0.00%)	0.91%(0.00%)
	2	0.28%(0.00%)	0.24%(0.00%)	0.13%(0.00%)	0.40%(0.00%)	0.40%(0.00%)
	3	0.16%(0.00%)	0.20%(0.00%)	0.06%(0.00%)	0.08%(0.00%)	2.84%(0.00%)
	4	0.11%(0.00%)	0.09%(0.00%)	0.02%(0.00%)	0.19%(0.00%)	3.22%(0.00%)
	5	0.09%(0.00%)	0.09%(0.00%)	0.08%(0.00%)	0.15%(0.00%)	0.97%(0.00%)

study. The results of experiment are presented on Fig. 7. It is clear from the results, that the measure $E(\mathcal{L})$ appears

- to decrease with increasing c for all scenarios and for all k (where comparison is valid). For example, we can compare k up to 3 when dealing with $c = 3, 4, 5$, we can compare k up to 4 when dealing with $c = 4, 5$.
- to exhibit an interesting pattern. As k increases to c , we see the expected latency for heavy-tailed services like Weibull appears to go from least value (compared to other service distributions) to the highest. This seems to be the case for higher c . This phenomenon can be explained as follows. For heavy-tailed distribu-

tion, when k is closer to c , the mean service time will be much higher as compared to other distributions, resulting in a higher $E(\mathcal{L})$.

5 Concluding Remarks and future research work

In this paper we studied queuing models useful in the study of efficient redundancy techniques in CC and DG systems. We combined both analytic and simulation ap-

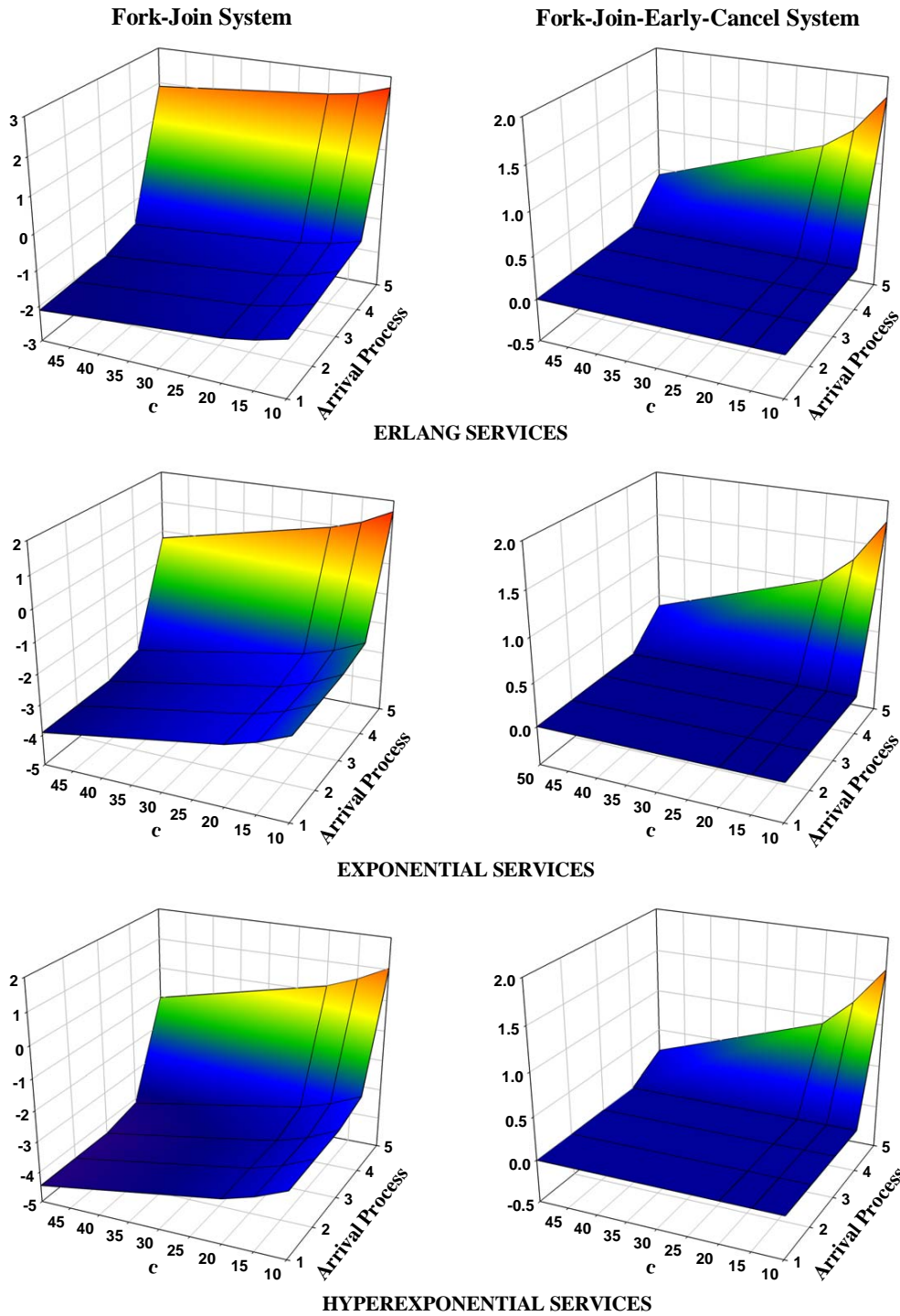


Figure 6: Plot of $E(\mathcal{L})$ under various scenarios

proach in the study of such queuing models under the assumption that the arrivals occur according to a versatile Markovian point process. Through simulation and numerical methods we showed the effect of redundancy on the ex-

pected latency as well as on the expected cost. The models studied in this paper can be extended in a number of ways. For example, heavy-tailed distributions for services [36] play a significant role in CC and DG areas. So, it will be of

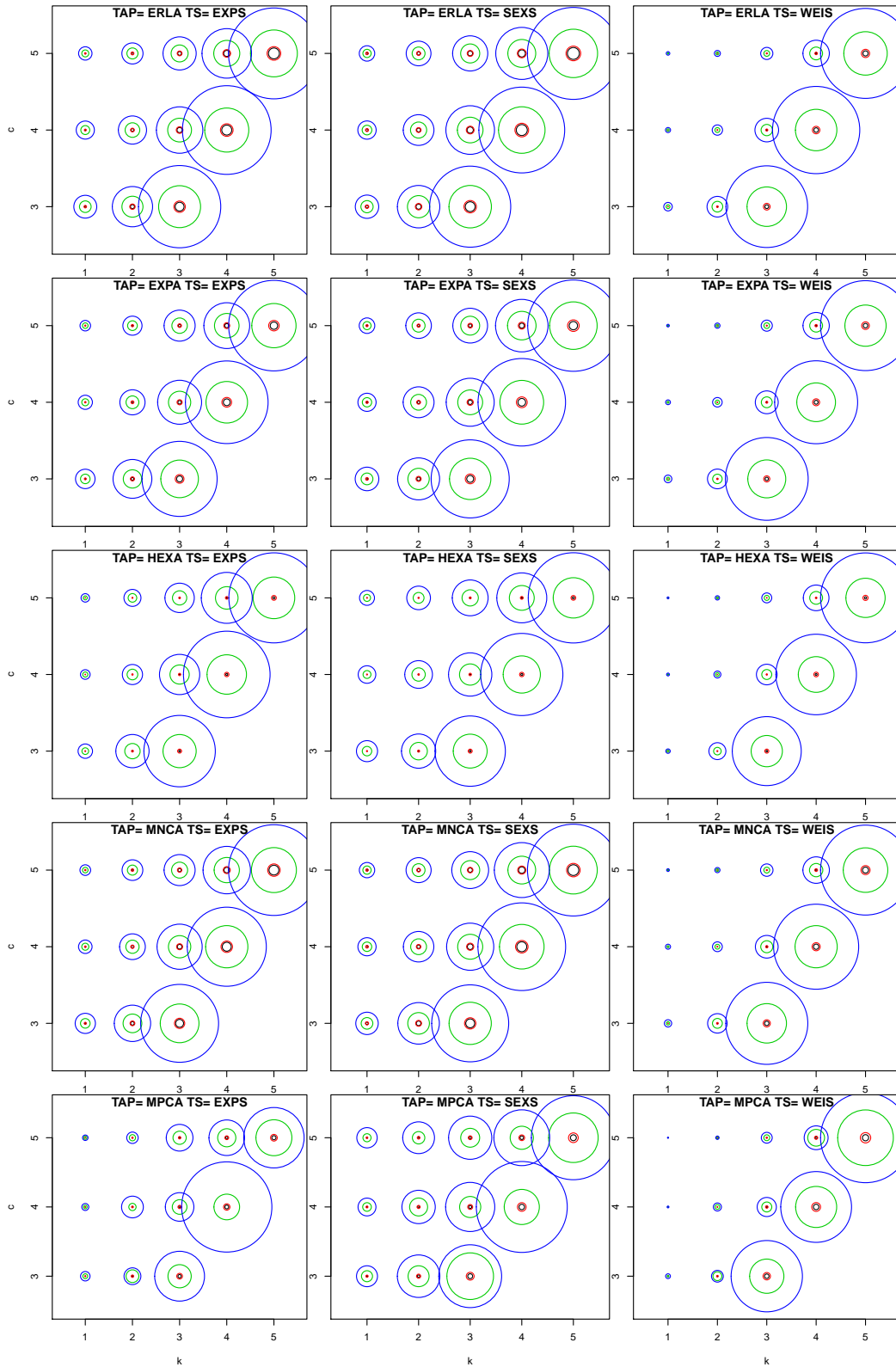


Figure 7: Plot of $E(L)$ under various scenarios when $c = 3, 4, 5$ and $k = 1, \dots, c$ for $\rho = 0.2, 0.5, 0.9, 0.95$ (black, red, green, blue circles correspondingly). Radius of the circle is proportional to the $E(L)$.

Table 3: Simulated μ'_S (σ_S) for (c, k) systems when $\mu = 1$

c	k	<i>EXPS</i>	<i>SEXP</i> (0.2)	<i>WEIB</i> (0.5, 0.5)
3	1	0.3330 (0.3324)	0.4668 (0.2666)	0.0041 (0.0644)
	2	0.8327 (0.5996)	0.8662 (0.4797)	0.0842 (0.7196)
	3	1.8310 (1.1630)	1.6671 (0.9315)	2.8885 (28.7177)
4	1	0.2500 (0.2495)	0.4001 (0.1999)	0.0009 (0.0138)
	2	0.5831 (0.4154)	0.6667 (0.3340)	0.0134 (0.1118)
	3	1.0826 (0.6491)	1.0665 (0.5204)	0.1566 (1.0599)
5	4	2.0805 (1.1896)	1.8652 (0.9514)	3.7994 (33.1038)
	1	0.2002 (0.2000)	0.3599 (0.1597)	0.0003 (0.0040)
	2	0.4497 (0.3189)	0.5599 (0.2561)	0.0036 (0.0297)
5	3	0.7825 (0.4607)	0.8269 (0.3707)	0.0284 (0.1808)
	4	1.2825 (0.6799)	1.2271 (0.5451)	0.2402 (1.3591)
	5	2.2780 (1.2052)	2.0251 (0.9653)	4.6818 (36.7680)

interest to explore this in the context of classical FJ system as well as extensions of FJ and SM systems. This will shed additional light similar to [18] but for more versatile arrival process. It is also worthy of consideration for further extensions both from theoretical and algorithmic points of view to compare the results with the model presented in [37]. It should be pointed that some preliminary results for multi-server queues with log-convex type services are available in [36].

Acknowledgement: The work of AR is partially supported by RFBR, projects 15-07-02341, 15-07-02354, 15-29-07974, 16-07-00622, and by President RF's grant No.MK-1641.2017.1. The authors express their sincere thanks to the anonymous reviewers and the editor for their suggestions and paying attention to some key references, that improved the presentation of the paper.

References

- [1] Artalejo J.R., Gomez-Correl A., He Q.M., Markovian arrivals in stochastic modelling: a survey and some new results, *Statistics and Operations Research Transactions*, 2010, 34, 2, 101–144.
- [2] Baccelli F., Makowski A.M., Shwartz A., The fork-join queue and related systems with synchronization constraints: stochastic ordering and computable bounds, *Adv. Appl. Prob.*, 1989, 21, 629–660, DOI: 10.2307/1427640
- [3] Bladt M., Nielsen B. F., *Matrix-Exponential Distributions in Applied Probability*, Probability Theory and Stochastic Modelling, 81, Springer US, Boston, MA, 2017, DOI: 10.1007/978-1-4939-7049-0
- [4] Chakravarthy S.R., The batch Markovian arrival process: A review and future work, In: A. Krishnamoorthy et al.(Eds.), *Advances in Probability Theory and Stochastic Processes*, Notable Publications Inc., NJ, 2001, 21–39.
- [5] Chakravarthy S.R., Markovian arrival processes. In: *Wiley Encyclopedia of Operations Research and Management Science*, Published Online: 15 JUN 2010, DOI: 10.1002/9780470400531.eorms0499
- [6] Chernov I., Nikitina N., Optimal Quorum for a Reliable Desktop Grid, In: *Proceedings of the Second International Conference BOINC-based High Performance Computing: Fundamental Research and Development (BOINC:FAST 2015)*, CEUR Workshop Proceedings, Vol-1502, 2015, 31–36.
- [7] Furht B., Escalante A.(Eds), *Handbook of Cloud Computing*, Springer, New York, USA, 2010, DOI 10.1007/978-1-4419-6524-0.
- [8] Gardner K., Zbarsky Z., Doroudi S., Harchol-Balter M., Hyytiä E., Scheller-Wolf A., Queueing with redundant requests: exact analysis, *Queueing Syst.*, 2016, 83, 227–259, DOI: 10.1007/s11134-016-9485-y
- [9] Gardner K., Harchol-Balter M., Scheller-Wolf A., Van Houdt B., A Better Model for Job Redundancy: Decoupling Server Slowdown and Job Size, *IEEE/ACM Transactions on Networking*, 2017, PP, 99, 1–15, DOI: 10.1109/TNET.2017.2744607
- [10] Rama Murthy Garimella, Rumyantsev A. On An Exact Solution Of The Rate Matrix Of Quasi-Birth-Death Process With Small Number Of Phases, In: *ECMS 2017 Proceedings. European Council for Modeling and Simulation*, 2017, 713–719, DOI: 10.7148/2017-0713
- [11] Graham A., *Kronecker Products and Matrix Calculus with Applications*, Ellis Horwood, Chichester, UK, 1981.
- [12] He Qi-Ming, *Fundamentals of Matrix-Analytic Methods*, Springer, New York, 2014, DOI: 10.1007/978-1-4614-7330-5
- [13] Heien E. M., Anderson D. P., Hagihara K. Batches with Unreliable Workers in Volunteer Computing Environments, *Journal of Grid Computing*, 2009, 7, 4, 501–518, DOI: 10.1007/s10723-009-9131-6
- [14] Ivashko E., Enterprise Desktop Grids, In: *Proceedings of the Second International Conference BOINC-based High Performance Computing: Fundamental Research and Development (BOINC:FAST 2015)*, CEUR Workshop Proceedings, Vol-1502, 2015, 16–21.
- [15] Joshi G., Soljanin E., Wornell G., Efficient replication of queued tasks for latency reduction in cloud systems, In: *Proceedings of 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2015, 107–114, DOI: 10.1109/ALLERTON.2015.7446992
- [16] Joshi G., Soljanin E., Wornell G., Efficient redundancy techniques for latency reduction in cloud systems, *arXiv:1508.03599 [cs.DC]*, Aug. 2015.
- [17] Joshi G., Soljanin E., and Wornell G., Queues with redundancy: Latency-Cost analysis, *Performance Evaluation Review*, 2015, 43, 2, 54–56, DOI: 10.1145/2825236.2825258
- [18] Joshi G., *Efficient Redundancy Techniques to Reduce Delay in Cloud Systems*, Ph.D. Thesis, MIT, June 2016.
- [19] Kelton W.D., Sadowski R.P., Swets N.B., *Simulation with ARENA*, Fifth ed., McGraw-Hill, New York, 2010.
- [20] Kim C., Agrawala A.K., Analysis of the fork-join queue, *IEEE Trans. Comput.*, 1989, 38(2), 1041–1053, DOI: 10.1109/12.16501
- [21] Latouche G., Ramaswami V., *Introduction to matrix analytic methods in stochastic modeling*, SIAM, 1999, DOI: 10.1137/1.9780898719734
- [22] Lucantoni D., Meier-Hellstern K.S., Neuts M.F., A single-server queue with server vacations and a class of nonrenewal arrival

- processes, *Advances in Applied Probability*, 1990, 22, 676–705, DOI: 10.2307/1427464
- [23] Lucantoni D.M., New results on the single server queue with a batch Markovian arrival process, *Stochastic Models*, 1991, 7, 1–46.
- [24] Marcus M., Minc H., *A Survey of Matrix Theory and Matrix Inequalities*, Allyn and Bacon, Boston, MA, 1964.
- [25] Mell P., Grance T., *The NIST Definition of Cloud Computing*, Special publication 800-145, National Institute of Standards and Technology: U.S. Department of Commerce, 2011, DOI: 10.6028/NIST.SP.800-145
- [26] Murugesan S., Bojanova I., *Encyclopedia of cloud computing*, John Wiley & Sons, Ltd., UK, 2016.
- [27] Nelson R., Tantawi A.N., Approximate analysis of fork/join synchronization in parallel queues, *IEEE Trans. Comput.*, 1988, 37, 6, 739–743, DOI: 10.1109/12.2213
- [28] Neuts M.F., A versatile Markovian point process, *Journal of Applied Probability*, 1979, 16, 764–779, DOI: 10.1017/S0021900200033465
- [29] Neuts M.F., *Matrix-geometric solutions in stochastic models: An algorithmic approach*, The Johns Hopkins University Press, Baltimore, MD, 1981.
- [30] Neuts M.F., *Structured stochastic matrices of $M/G/1$ type and their applications*, Marcel Dekker, NY, 1989.
- [31] Neuts M.F., Models based on the Markovian arrival process, *IEICE Transactions on Communications*, 1992, E75B, 1255–1265.
- [32] Neuts M.F., *Algorithmic Probability: A collection of problems*, Chapman and Hall, NY, 1995.
- [33] Nouman Durrani M., Shamsi J. A., Volunteer computing: requirements, challenges, and solutions, *Journal of Network and Computer Applications*, 2014, 39, 369–380, DOI: 10.1016/j.jnca.2013.07.006
- [34] Qiu Z., Perez J.F., Harrison P.G., Tackling latency via replication in distributed systems, In: *Proceedings of ICPE'16*, March 12-18, 2016, Delft, Netherlands, 2016, DOI: 10.1145/2851553.2851562
- [35] Rood B., Lewis M.J., Grid Resource Availability Prediction-Based Scheduling and Task Replication, *Journal of Grid Computing*, 7, 2009, 479–500, DOI: 10.1007/s10723-009-9135-2
- [36] Rumyantsev A., Chakravarthy S. R., Split-Merge Model of Workunit Replication in Distributed Computing, In: *Proceedings of the Third International Conference BOINC-based High Performance Computing: Fundamental Research and Development (BOINC:FAST 2017)*, CEUR Workshop Proceedings, Vol. 1973, 2017, 27–34.
- [37] Squillante M. S., Zhang Y., Sivasubramaniam A., Gautam N., Generalized parallel-server fork-join queues with dynamic task scheduling, *Annals of Operations Research*, 2008, 160, 227–255.
- [38] Steeb W-H., Hardy Y., *Matrix Calculus and Kronecker Product*, World Scientific Publishing, Singapore, 2011.
- [39] Thomasian A., Analysis of Fork/Join and Related Queueing Systems, *ACM Comput. Surv.*, 2014, 47, 2, Article 17, 71 p., DOI: <http://dx.doi.org/10.1145/2628913>
- [40] Voas J., Zhang J., Cloud computing: New wine or just a new bottle? *IEEE ITPro*, 2009, 15–17.
- [41] Vulimiri A., Godfrey P.B., Mittal R., Sherry J., Ratnasamy S., Shenker S., Low latency via redundancy, In: *Proceedings of CoNEXT'13*, Santa Barbara, California, USA, 2013, 283–294, DOI: 10.1145/2535372.2535392