New Stability Results for Multiserver-job Models via Product-form Saturated Systems

Isaac Grosof
Carnegie Mellon University
Pittsburgh, PA, USA
igrosof@cs.cmu.edu

Mor Harchol-Balter
Carnegie Mellon University
Pittsburgh, PA, USA
harchol@cs.cmu.edu

Alan Scheller-Wolf
Carnegie Mellon University
Pittsburgh, PA, USA
awolf@andrew.cmu.edu

ABSTRACT

Multiserver-job (MSJ) models are increasingly common in today's datacenters. In these models, each job runs on multiple servers concurrently, for some duration. The most common service ordering for jobs is First-Come First-Served (FCFS). Unfortunately, MSJ FCFS models are hard to analyze, and even the stability region of MSJ FCFS models is not well understood.

Stability has only been analyzed in the case where all jobs have independent exponentially-distributed durations with the same mean (the "single duration" setting). This does not allow, however, for modeling the common situation where jobs with higher server need (number of servers required) also have higher expected duration. This paper provides the first analysis of stability for MSJ FCFS models in the two-class setting, where each class of jobs has its own server need and its own exponentially-distributed duration.

To analyze stability, we make use of the saturated system, whose throughput determines the stability region. While the saturated system has been useful in deriving stability regions in the past, it has never been applied in settings where jobs occupy multiple servers. By looking at the saturated system in a new light, we find that its solution has an attractive product form in the two-class setting, and a different product form in the single-duration setting, both novel results. Besides solving our problem, this product form may also serve as a gateway to analyzing other complex models.

1 Introduction

Multiserver queueing theory traditionally focuses on settings such as the $M/G/k$ where every job occupies exactly one server. These models have remained popular for decades because they capture the behavior of many important systems, including previous computing systems, while remaining amenable to theoretical analysis. However, these one-server-per-job models are no longer representative of modern computing systems. In modern data centers, jobs require a variable amount of resources by default. In Google's recently published trace of its "Borg" computation cluster, jobs vary in CPU requirement by a factor of 10^5 [6, 12].

To capture this common situation, we focus on the "multiserver-job model", in which each job occupies a fixed number of servers (its "server need"), typically more than one, throughout its time in service (its "service duration"). We specifically focus on the first-come first-served (FCFS) service ordering, which is the default policy in both cloud computing and supercomputing [2, 11].

Currently, little is known about FCFS service in MSJ models. Even the stability region is not generally understood. More is known about MSJ models with more specialized scheduling policies, both for stability and mean response time [4, 6, 7, 9].

The most general prior result on stability in the MSJ FCFS setting is due to Rumyantsev and Morozov [10]. They studied a setting in which each job has service duration $X \sim Exp(\mu)$, independent of the job's server need, which can come from a general distribution, and used a matrix-analytic approach to explicitly characterize the stability region. We call this setting the "single duration" setting.

However, in real systems, service duration is typically positively correlated with server need. For example, a significant correlation is present in the aforementioned Borg trace [5, 12]. To capture this correlation, we study a two-class MSJ FCFS model, where each class has a distinct server need and a distinct exponentially-distributed duration, $X_i \sim Exp(\mu_i)$. Note that this is not a strict generalization of the single-duration setting, as an arbitrary server-need distribution is permitted in that setting.

In this paper, we give the first characterization of the stability region of the two-class setting. This is the first stability region result for any MSJ FCFS model in which server need and service duration are correlated.

We make key use of the *saturated system* approach [1, 3] to prove our result. While this approach is a well-known method for characterizing stability, we are the first to apply it to the MSJ system. The saturated system is a closed system in which completions trigger new arrivals, so the total server need of jobs in the system is always at least the number of servers. Note that some servers may remain idle, due to the FCFS service and packing limitations. An important property of the saturated system is the *saturation rule*: The throughput of the saturated system matches the threshold of stability of the original system [1]. Thus, to determine the stability region of the original system, it suffices to determine the stationary distribution of the saturated system.

In this paper, we explicitly characterize the stationary distribution of the saturated system for the two-class setting, proving that it has a product-form structure. We also prove that in the single-duration setting, the saturated system also exhibits a product-form solution, which differs from the two-class setting's solution. We thereby provide a cleaner proof of the stability region of the single-duration setting. These product-form results are of independent interest, and open

up the potential for a new class of product-form results, going beyond prior product-form results for open systems such as Order-Independent queues [8].

In Section 2, we define the saturated system. Next, we present our results: Section 3 for the single-duration system, and Section 4 for the two-class system.

2 Saturated System

The saturated system is a closed system, meaning that completions trigger new arrivals – there is no exogenous arrival process. The number of jobs is always large enough that the total server need is at least k, the number of servers. For this reason, the system is referred to as "saturated".

There are three equivalent formulations of the saturated system:

I. A closed system in which there are always exactly k jobs. Whenever a job completes, a new job arrives. The state descriptor is a ordered list of k jobs. This formulation is simple, but the resulting system has many possible states.

II. A closed system in which there are always just enough jobs such that the total server need is at least k. Note that at most 1 job can be in the queue, in this formulation. The state descriptor tracks the job in the queue, if any, and the multiset of jobs in service. Under this formulation, there are very few possible system states. In Section 4, we simplify this formulation even further.

III. A system with infinitely many jobs in the queue, at all times. Baccelli and Foss used this formulation in their original proof of the "saturation rule" [1].

All of these formulations of the saturated system can be coupled such that the set of jobs in service is identical at all times and the same jobs complete at the same times, so we refer to them interchangeably as "the saturated system".

We use the first formulation for the single-duration setting, and the second for the two-class setting.

In the saturated system, there are two natural Markov chains to consider: The continuous-time Markov chain, and the embedded discrete-time Markov-chain that changes state once per job completion. Note that we focus on exponential job durations, so the state only changes on job completions.

The two Markov chains are closely related, and stationary distributions of each can be used to derive the stationary distribution of the other. In this paper, we focus on the embedded DTMC, which we prove has a product-form stationary distribution in each of our two settings.

3 Single Duration Results

First, let us consider a multiserver-job (MSJ) system in which all jobs require $Exp(\mu)$ duration, independent of server need. Server needs are sampled i.i.d. from an arbitrary distribution, where p_ℓ represents the probability that a job has server need ℓ. Jobs are served in FCFS order. We call this the "single-duration" setting.

For this result, we use the "exactly k jobs" formulation (I) of the saturated system. A state of the saturated system is an ordered list m of k jobs, where each job is identified by its server need ℓ. We write m_i to represent the server need of the i^{th}-oldest job in the system. The set of jobs in service is the largest initial sequence of jobs with total server need $\leq k$. Let $\sigma(m)$ denote the number of jobs served in state m.

Let π_m represent the stationary probability of state m.

THEOREM 1. *In the single-duration setting, the stationary distribution of the embedded DTMC of the saturated system is:*

$$\pi_m = \prod_{i=1}^{k} p_{m_i}.$$

PROOF. Let $P(m, m')$ represent the probability that state m transitions to state m', after a completion and the corresponding arrival. Let $\mathbb{M}$ represent the set of all possible states of the saturated system.

Consider the balance equation for a given state m:

$$\pi_m = \sum_{m' \in \mathbb{M}} \pi_{m'} P(m', m). \tag{1}$$

For each possible server need $1 \leq \ell \leq k$, a state could transition to m via the completion of a server need ℓ job. Let $m^\ell(1)$ denote one specific predecessor state to m, the state $m = [\ell, m_1, m_2, \ldots, m_{k-1}]$. From state $m^\ell(1)$, if the server need ℓ job in position 1 completes, and then a server need m_k job arrives, the system transitions to state m.

Several more states can transition to state m via the completion of a server need ℓ job. If a state has a server need ℓ job in service, and the other $k-1$ jobs in the system have server needs $[m_1, \ldots, m_{k-1}]$, the state will transition to state m if the server need ℓ job completes and a server meed m_k job arrives. In total, there are $\sigma(m^\ell(1))$ such states, because the server need ℓ job will be in service if and only if it is in one of the positions that receives service in state $m^\ell(1)$.[1] We call these states $m^\ell(j)$, for $1 \leq j \leq \sigma(m^\ell(1))$, where the index j represents the position where the server need ℓ job is inserted. Note that for all j, $\sigma(m^\ell(j)) = \sigma(m^\ell(1))$, because the same set of jobs is served in each of these states.

From any state $m^\ell(j)$, the probability $P(m^\ell(j), m)$ of transitioning to state m is $p_{m_k}/\sigma(m^\ell(1))$, because each of the $\sigma(m^\ell(j)) = \sigma(m^\ell(1))$ jobs in service are equally likely to depart, because each job has the same exponential completion rate μ. If the server need ℓ job completes, a server need m_k job must then arrive to finish the transition to state m. The stationary probability of state $m^\ell(j)$ under the proposed stationary distribution π is $\pi_m p_\ell / p_{m_k}$. Thus, the corresponding portion of the balance equation (1) is:

$$\sum_{j=1}^{\sigma(m^\ell(1))} \pi_{m^\ell(j)} P(m^\ell(j), m)$$
$$= \pi_m \frac{p_\ell}{p_{m_k}} \frac{p_{m_k}}{\sigma(m^\ell(1))} \sigma(m^\ell(1)) = \pi_m p_\ell.$$

Summing over all server needs ℓ, we see that (1) must hold. Because m was an arbitrary state, the balance equation holds for all states. Because π sums to 1, it is the stationary distribution. □

4 Two-class Results

Next, consider a MSJ system in which there are two classes of jobs: Class 1 jobs have server need ℓ_1 and duration $Exp(\mu_1)$, while class 2 jobs have server need ℓ_2 and duration $Exp(\mu_2)$, where $\ell_1 < \ell_2 \leq k$. Arriving jobs are class 1 with probability

[1] We distinguish otherwise identical states where the server need ℓ job that will complete is inserted in distinct locations.

p_1, and class 2 with probability $p_2 = 1 - p_1$, sampled i.i.d. Service is again FCFS. We call this the "two-class" setting.

For this result, we use the "just enough total server need" formulation (II) of the saturated system, with a slight modification: a completion only triggers a new arrival if there are enough servers available that a class 1 job could fit into service. After a completion, if the total server need of jobs in the system is at most $k - \ell_1$, a new arrival occurs. Arrivals continue until the total server need exceeds $k - \ell_1$. This formulation still has the same jobs in service and the same completion times as any of the formulations in Section 2.

A state of the saturated system consists of a triple $[h, a, b]$. a represents the number of class 1 jobs in service, b represents the number of class 2 jobs in service. h represents whether a class 2 job is blocking the head of the queue. If $a\ell_1 + b\ell_2 > k - \ell_1$, then no additional jobs could fit in service, so we write $h = 0$. If $a\ell_1 + b\ell_2 \leq k - \ell_1$, then there must be a class 2 job at the head of the queue, so we write $h = 1$.

There is exactly one state for each possible number a of class 1 jobs in service. To see this, imagine starting with a class 1 jobs in service, then adding class 2 jobs until the total server need threshold is reached. This must reach a unique state. We refer to this state as $s_1(a)$.

There is exactly one state with $h = 0$ for each possible number b of class 2 jobs in service. To see this, imagine starting with b class 2 jobs in service, then adding class 1 jobs until the total server need threshold is reached. This too must reach a unique state with $h = 0$. We refer to this state as $s_2(b)$.

For a given state s, a transition consists of a completion, perhaps followed by the blocking job entering service, and perhaps followed by jobs being newly generated until the total server need is again $\geq k$. For a given state s, let $f_1(s)$ denote the probability that the next completion is a class 1 job, and define $f_2(s) = 1 - f_1(s)$ correspondingly. For a given state $[\cdot, a, b]$, we have

$$f_1([\cdot, a, b]) = \frac{a\mu_1}{a\mu_1 + b\mu_2}, \quad f_2([\cdot, a, b]) = \frac{b\mu_2}{a\mu_1 + b\mu_2}.$$

Again, let π_s denote the stationary probability of state s.

THEOREM 2. *In the two-class exponential setting, the stationary distribution of the embedded DTMC of the saturated system is:*

$$\pi_{[h,a,b]} = C p_1^a p_2^{b+h} \prod_{i=1}^{a} \frac{1}{f_1(s_1(i))} \prod_{j=1}^{b} \frac{1}{f_2(s_2(j))}$$

where C is a normalization constant.

PROOF SKETCH. See [5, Sections 4 and 5] for a full proof.

To verify that π is the correct stationary distribution, we must show that the balance equations hold for each state:

$$\pi_s = \sum_{s'} \pi_{s'} P(s', s)$$

In fact, a form of local balance holds, corresponding to the class of the job completed in each transition. Let $P_1(s', s)$ be the probability of a transition from s' to s due to a completion of a class 1 job, and define $P_2(s', s)$ correspondingly.

We show that the following local balance result holds:

$$p_1 \pi_s = \sum_{s'} \pi_{s'} P_1(s', s) \quad p_2 \pi_s = \sum_{s'} \pi_{s'} P_2(s', s).$$

At this point, it remains to enumerate over all states to prove that local balance holds, splitting up states based on whether $h = 0$ or $h = 1$. See [5] for details. □

5 Conclusion

We give the first product-form solutions for the saturated system in two MSJ FCFS settings: The single-duration setting and the two-class setting. These solutions allow us to characterize the stability region of both settings. This is the first characterization of stability in the two-class setting, and a simpler proof for the single-duration setting.

An interesting direction for future work is to study whether either of these product-form results can be generalized to a larger class of settings. Specifically, we believe that one could consider any FCFS system where there is some rule determining which jobs can serve together. We believe that results in this paper generalize to any such FCFS system.

6 References

[1] F. Baccelli and S. Foss. On the saturation rule for the stability of queues. *Journal of Applied Probability*, 32(2):494–507, 1995.

[2] D. G. Feitelson, L. Rudolph, and U. Schwiegelshohn. Parallel job scheduling—a status report. In *Workshop on Job Scheduling Strategies for Parallel Processing*, pages 1–16, New York, NY, USA, 2004. Springer.

[3] S. Foss and T. Konstantopoulos. An overview of some stochastic stability methods. *Journal of the Operations Research Society of Japan*, 47(4):275–303, 2004.

[4] J. Ghaderi. Randomized algorithms for scheduling VMs in the cloud. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, Apr. 2016.

[5] I. Grosof, M. Harchol-Balter, and A. Scheller-Wolf. Stability for two-class multiserver-job systems. *arXiv preprint arXiv:2010.00631*, 2020.

[6] I. Grosof, M. Harchol-Balter, and A. Scheller-Wolf. WCFS: A new framework for analyzing multiserver systems. *Queueing Systems*, 2022.

[7] I. Grosof, Z. Scully, M. Harchol-Balter, and A. Scheller-Wolf. Optimal scheduling in the multiserver-job model under heavy traffic. *Proc. ACM Meas. Anal. Comput. Syst.*, 6(3), Dec 2022.

[8] A. E. Krzesinski. *Order Independent Queues*, pages 85–120. Springer US, Boston, MA, 2011.

[9] S. T. Maguluri, R. Srikant, and L. Ying. Stochastic models of load balancing and scheduling in cloud computing clusters. In *2012 Proceedings IEEE Infocom*, pages 702–710. IEEE, 2012.

[10] A. Rumyantsev and E. Morozov. Stability criterion of a multiserver model with simultaneous service. *Annals of Operations Research*, 252(1):29–39, 2017.

[11] L. Sliwko. A taxonomy of schedulers–operating systems, clusters and big data frameworks. *Global Journal of Computer Science and Technology*, 2019.

[12] M. Tirmazi, A. Barker, N. Deng, M. E. Haque, Z. G. Qin, S. Hand, M. Harchol-Balter, and J. Wilkes. Borg: The next generation. In *Proceedings of the Fifteenth European Conference on Computer Systems*, EuroSys '20, New York, NY, USA, 2020.