Queueing Systems (2022) 100:201–203
https://doi.org/10.1007/s11134-022-09762-x

The multiserver job queueing model

Mor Harchol-Balter[1]

Received: 24 January 2022 / Accepted: 28 February 2022 / Published online: 29 March 2022

1 Introduction

A great deal of queueing theory is devoted to studying multiserver models, such as the M/G/n. A key feature of such models is that each job runs on a *single server*. Unfortunately this one-server-per-job model is *not* a good representation of today's data centers. Almost all of today's data center jobs occupy *multiple servers simultaneously* [12]. We refer to such jobs that run on multiple servers as *multiserver jobs*. A recent trace from Google's Borg scheduler [12] shows that the number of servers utilized by a single job can vary by five orders of magnitude across jobs. Understanding the performance of systems with multiserver jobs is therefore of paramount importance.

Figure 1 shows the *multiserver job queueing model*. Jobs arrive with average rate λ into a system with n homogeneous servers, where they are served in FCFS order. A job is of class i with probability p_i. A job of class i requires (any) n_i servers, which it occupies in parallel for S_i hours, where S_i is a random variable. Importantly, the *size* of a job of class i is $n_i \cdot S_i$ and is specified in units of server-hours.

Some related models: While almost nothing is known about the performance of multiserver job queueing models, there is a cousin of this model, which we call the *dropping model*, which is analytically tractable under very general settings. In the dropping model, jobs which cannot immediately receive service are dropped. The dropping model exhibits a beautiful product form when job durations (the S_i's) are exponentially distributed, as shown in Arthurs and Kaufman [1]. Whitt [15] generalized the model to allow jobs to demand multiple resource types, while van Dijk [13] allowed durations to be generally-distributed. Related to dropping models are *streaming models*, which come up in communication networks. Here the resource being shared is bandwidth in the network. The "jobs" are audio or video flows which require a fixed bandwidth reservation to run (akin to needing a fixed number of servers). The goal is to schedule flows to minimize a cost related to dropping probabilities [4, 10].

✉ Mor Harchol-Balter
harchol@cs.cmu.edu

1 Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

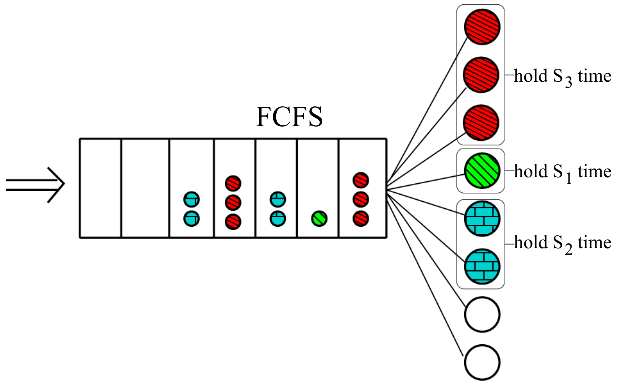

Fig. 1 The multiserver job queueing model with $n = 8$ servers. An arriving job of class i requests any n_i servers and occupies these servers (in parallel) for S_i time. In this particular illustration, $n_i = i$

2 Problem Statement

Our goal is to derive very basic metrics for the multiserver job queueing model.

Stability: We ask what is the highest value of λ such that the system is stable? Ideally (if all n servers are always occupied) our stability region would be:

$$\lambda \leq \frac{n}{\sum_i p_i n_i \mathbf{E}\left[S_i\right]}. \tag{1}$$

However the *true* stability region can be far smaller than (1). For example, consider a system with $n = 10$ servers, where all jobs request 3 or 4 servers. Clearly, there will always be some unused servers.

Response time: The *response time* of a job is the time from when it arrives until it completes service. What is the average response time across jobs?

The above questions are challenging even assuming exponentially-distributed job durations and Poisson arrivals. In addition to tracking the total number of jobs, we need to also track how many servers are occupied by each class of jobs.

3 Discussion

We present some recent work as well as avenues for future exploration.

Stability: In 2017 Rumyantsev and Morozov [11] used QBD methods to derive the stability region for the multiserver job model where all jobs have the *same* exponential service duration $S_i \sim \text{Exp}(\mu), \forall i$. This result was extended in 2020 by Grosof et al. [7] to allow for 2 classes of jobs, each with *different* exponential service time durations. Some works have explored removing exponentiality assumptions in stability analysis, e.g., [2, 3]. Particularly useful is the *saturation rule*, [3], which considers systems with an infinite number of jobs at all times.

Steady-state distribution: Brill and Green [5] and Filippopoulos and Karatza [6] consider the steady-state distribution of the number of jobs in a multiserver job system with only $n = 2$ servers, where all jobs have the same service duration $S_i \sim \text{Exp}(\mu)$, $\forall i$. However, these solutions are too complex to be practical. Another possible direction is a multiserver job model where there is a finite (but nonzero) queue capacity.

Scaling regimes: Very recently, Wang et al. [14] consider a scaling regime where *both* the number of servers requested by a job and the system load scale with the total number of servers. This allows them to obtain results on stability and the probability that an arriving job has to queue. Hong and Wang [9] establish the first bounds on mean waiting time in this same asymptotic regime.

Other scheduling policies: While our definition of the multiserver job queueing model assumes FCFS scheduling, it is common in practice to slightly relax this to allow for *backfilling*, which increases the stability region. Under backfilling, if the job at the head of the queue does not fit, then it may be jumped by newer jobs that require fewer servers, provided that running these newer jobs will not delay the older jobs, based on user-provided estimates of job durations. Unfortunately, the multiserver job model is no easier to analyze under backfilling. However, a new development by Grosof et al. [8] in 2021 considers a scheduling policy called *ServerFilling*, which goes further by ensuring that *no server is idle* provided that there are at least n jobs in the system (ServerFilling requires the assumption that the n_i's are all divisors of n). Under ServerFilling scheduling, Grosof et al. prove additively tight bounds on the mean response time of the multiserver job system, with generally-distributed job durations and Poisson arrivals.

References

1. Arthurs, E., Kaufman, J.: Sizing a message store subject to blocking criteria. In: IFIP Performance Conference, pp. 547–564 (1979)
2. Baccelli, F., Courcoubetis, C.A., Reiman, M.I.: Construction of the stationary regime of queues with locking. Stoch. Process. Appl. **26**, 257–265 (1987)
3. Baccelli, F., Foss, S.: On the saturation rule for the stability of queues. J. Appl. Prob. **32**(2), 494–507 (1995)
4. Bean, N.G., Gibbens, R.J., Zachary, S.: Asymptotic analysis of single resource loss systems in heavy traffic, with applications to integrated networks. Adv. Appl. Prob. **27**(1), 273–292 (1995)
5. Brill, P.H., Green, L.: Queues in which customers receive simultaneous service from a random number of servers: a system point approach. Manag. Sci. **30**(1), 51–68 (1984)
6. Filippopoulos, D., Karatza, H.: An M/M/2 parallel system model with pure space sharing among rigid jobs. Math. Comput. Modell. **45**(5), 491–530 (2007)
7. Grosof, I., Harchol-Balter, M., Scheller-Wolf, A.: Stability for two-class multiserver-job systems. arXiv:2010.00631, October (2020)
8. Grosof, I., Harchol-Balter, M., Scheller-Wolf, A.:The finite-skip method for multiserver analysis. arXiv:2109.12663, September (2021)
9. Hong, Y., Wong, W.: Sharp waiting-time bounds for multiserver jobs. arXiv:2109.05343, Sept. (2021)
10. Hunt, P.J., Kurtz, T.G.: Large loss networks. Stoch. Process. Appl. **53**(2), 363–378 (1994)
11. Rumyantsev, A., Morozov, E.: Stability criterion of a multiserver model with simultaneous service. Ann. Oper. Res. **252**(1), 29–39 (2017)
12. Tirmazi, M., Barker, A., Deng, N., Haque, M. E., Qin, Z. G., Hand, S., Harchol-Balter, M., Wilkes, J.: Borg: The next generation. In: Proceedings of the Fifteenth European Conference on Computer Systems (EuroSys '20), pp. 1–14, Greece, April (2020)
13. Van Dijk, N.M.: Blocking of finite source inputs which require simultaneous servers with general think and holding times. Oper. Res. Lett. **8**(1), 45–52 (1989)
14. Wang, W., Xie, Q., Harchol-Balter, M.: Zero queueing for multi-server jobs. Proc. ACM Measur. Anal. Comput. Syst. (POMACS/SIGMETRICS) **5**(1), 1–25 (2021). (**Article 7**)
15. Whitt, W.: Blocking when service is required from several facilities simultaneously. AT&T Bell Labor. Tech. J. **64**, 1807–1856 (1985)