A Note on the Determination of the Processing Capacity in a Multiserver Job System as a Model of Cloud Datacenters

Alexandre Brandwajn, Thomas Begin

A Note on the Determination of the Processing Capacity in a Multiserver Job System as a Model of Cloud Datacenters

Alexandre Brandwajn* and Thomas Begin†
*Baskin School of Engineering, University of California Santa Cruz, California, USA
†Univ Lyon, UCB Lyon 1, ENS Lyon, Inria, CNRS, LIP UMR 5668 - Lyon, France
Email: alexb@soe.ucsc.edu, thomas.begin@ens-lyon.fr

***Abstract*—Systems with multiserver jobs, inspired by modern datacenters, present a challenge in terms of the analysis of their performance and, in particular, the determination of their processing capacity as some servers may remain idle even though there are jobs queued for service. We consider a generalization of the multiserver jobs model to a resource of not necessarily integer quantity and jobs of different classes requiring arbitrary fractions of that resource. We present a simple approach to the analysis of such a system in steady state. Our approach relies on a suitably chosen state description and the use of conditional probabilities. The limiting values of these conditional probabilities allow us to determine the asymptotic maximum job processing capacity without having to obtain the full solution for the system.**

***Index Terms*—Maximum processing capacity, Multiserver queue, Multiserver jobs, Cloud datacenters.**

I. Introduction

In a recent paper devoted to important open problems in queueing models inspired by modern datacenters, Mar Harchol Balter [3] lists the multiserver jobs. While in traditional queueing models (e.g., [5], [2], [4]) a single job (request, customer) occupies a single server in a multiserver center, with modern applications, jobs of different classes, sharing the same First Come First Served (FCFS) queue, may require different numbers of servers to start service. Thus, it is possible for some servers to be idle while there is a queue of requests blocked behind a job needing more servers than currently available. Hence, it is not easy in general to determine the processing capacity of such a system. We present a simple approach to analyzing a system with multiserver jobs and to determining its maximum processing capacity. We refer the reader to Harchol Balter [3] for a recent state of the art. The next section outlines the model considered and our approach.

II. Model and Approach to its Solution

Consider the system depicted in Figure 1. It consists of a total resource of size B shared by L classes of jobs. Each class of jobs is characterized by the amount of resource it needs to proceed and the mean time the resource is used by the job, denoted by b_ℓ and $t_\ell = 1/\mu_\ell$, respectively, for jobs of class ℓ ($\ell = 1, \ldots, L$). The jobs arrive to the system from a Poisson source with overall rate λ and are served in the order of their arrivals (FCFS). We denote by p_ℓ the probability that an arriving job is of class ℓ. Note that our model is a generalization of the multiserver jobs model and it maps onto the latter if the values B and b_ℓ for $\ell = 1, \ldots, L$ are all integers.

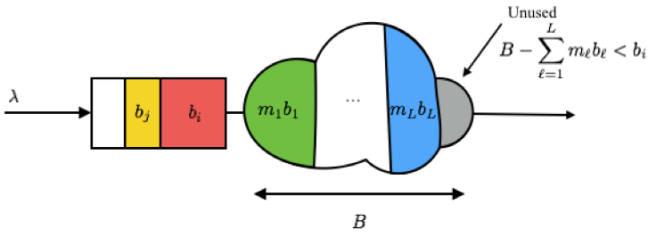

Fig. 1: Resource of size B shared by L classes of jobs.

To describe such a system in steady state, when it exists, we use the numbers of jobs of each class currently in service (denoted by m_ℓ, $\ell = 1, \ldots, L$), the class of the job at the head of the queue (if any), (denoted by i), and the total number of jobs in the system (denoted by n). Let $p(m_1, \ldots, m_L, i, n)$ be the corresponding steady-state probability, assuming it exists. We let $m = \sum_{\ell=1}^{L} m_\ell$ denote the total current number of jobs in service. For clarity, we use the value $i = 0$ when there are no jobs in the queue, i.e., when $n = m$.

To enumerate system states, we note that we must have $m_1 = 0, \ldots, [B/b_1]$, $m_2 = 0, \ldots, [(B - m_1 b_1)/b_2]$, and so on for increasing class numbers. Feasible states must satisfy the condition that $\sum_{\ell=1}^{L} m_\ell b_\ell \leq B$. Additionally, when there are jobs queued, i.e., when $n > m$, we must have $B - \sum_{\ell=1}^{L} m_\ell b_\ell < b_i$ where $i = 1, \ldots, L$ is the class of the job at the head of the queue. With these points in mind, and assuming exponentially distributed service times for all job classes, it is a straightforward matter to generate the balance equations for the steady-state probabilities $p(m_1, \ldots, m_L, i, n)$.

Although, with jobs waiting in the queue in the system considered, a single job departure can trigger the start of service for a variable number of jobs (depending on their resource requirements and the residual amount of resource left by the departure), the steady-state probability for the total number of jobs in the system, denoted by $p(n)$, can be expressed simply as

$$p(n) = \frac{1}{G} \prod_{j=1}^{n} \frac{\lambda}{u(n)}, \quad n = 0, 1, \ldots \tag{1}$$

In formula (1), G is normalizing constant such that $\sum_n p(n) = 1$, empty products are equal to 1, and $u(n)$ is the conditional rate of job completions given n, which can be written as

$$u(n) = \sum_{S(n)} (m_1\mu_1 + \cdots + m_L\mu_L) p(m1, \ldots, m_L, i|n) \tag{2}$$

where $S(n)$ denotes the set of states feasible for a given value of n jobs in the system.

Formula (2) involves the rates of job completion for each class and the conditional probability that the system is in the state $(m_1, \ldots, m_L, i)$ given the value of the total number of jobs in the system, $p(m_1, \ldots, m_L, i|n)$. Note that, from the definition of conditional probability, we have

$$p(m_1, \ldots, m_L, i, n) = p(n)p(m_1, \ldots, m_L, i|n) \tag{3}$$

Using (1) and (3) in the balance equations for our system, we readily obtain the set of equations for the conditional probabilities $p(m_1, \ldots, m_L, i|n)$. We can solve the latter using a fixed-point iteration together with formula (2) and the normalizing condition $\sum_{S(n)} p(m_1, \ldots, m_L, i|n) = 1$.

Let $\tilde{p}(m_1, \ldots, m_L, i) = \lim_{n\to\infty} p(m_1, \ldots, m_L, i|n)$ and, correspondingly, $\tilde{u} = \lim_{n\to\infty} u(n)$. It is clear from (1), that, for the steady state to exist, these limits must exist and we must have $\lambda < \tilde{u}$ since $p(n)$ is asymptotically geometric (cf. Takahashi [6] for a related discussion). By taking the limit for $n \to \infty$ in the equations for the conditional probabilities $p(m_1, \ldots, m_L, i|n)$, we easily obtain a set of equations for the limiting probabilities $\tilde{p}(m_1, \ldots, m_L, i)$. To determine the maximum job processing rate of the system, it then suffices to solve this set of equations for a value of arrival rate $\lambda = \tilde{u}$. This is easily accomplished using a fixed-point iteration with

$$\tilde{u} = \sum_{S(\infty)} (m_1\mu_1 + \cdots + m_L\mu_L) \tilde{p}(m_1, \ldots, m_L, i) \tag{4}$$

and

$$\sum_{S(\infty)} \tilde{p}(m_1, \ldots, m_L, i) = 1. \tag{5}$$

All feasible states $S(\infty)$ here are states $(m_1, \ldots, m_L, i)$ such that $\sum_{\ell=1}^{L} m_\ell b_\ell \le B$ and $B - \sum_{\ell=1}^{L} m_\ell b_\ell < b_i$, where i is the class of the job at the head of the queue ($i = 1, \ldots, L$). Thus, we can determine the processing capacity of the system without having to obtain its steady-state solution.

If we wish to analyze the system beyond its processing capacity, the solution of the set of equations for the conditional probabilities $p(m_1, \ldots, m_L, i|n)$ together with formula (2), allows us to determine $p(n)$, the steady-state probability that there are n jobs in the system, from formula (1). Hence, it is straightforward to obtain the expected sojourn time for a job, $E[W]$ using Little's law [7] as $E[W] = E[N]/\lambda$, where $E[N] = \sum_n np(n)$ is the mean number of jobs in the system. Clearly, in the context of datacenter performance, it is important to assess not only the mean response time but also its higher order properties (cf. [3]). Extensions of Little's law relating higher moments of the number in a system to higher moments of sojourn time cannot be used directly for the whole system because job overtaking is possible during service. However, they can be used for the FCFS queue of jobs waiting for service to start. Denote by $n_q(m_1, \ldots, m_L, i, n)$ the number of jobs waiting for service given that the system state is $(m_1, \ldots, m_L, i, n)$. Denote by $E[N_q^k]$ the k-th moment of the number of jobs in the queue. We have $E[N_q^k] = \sum_n p(n) \sum_{S(n)} n_q(m_1, \ldots, m_\ell, i, n)^k p(m_1, \ldots, m_L, i|n)$. Let $E[W_q^k]$ be the k-th moment of the time a job spends queued waiting for service. Looking specifically at the second moments, we get from a generalization of Little's law [7]

$$E[W_q^2] = \frac{E[N_q^2] - E[N_q]}{\lambda^2}. \tag{6}$$

Denote by $E[S_\ell^k]$ the k-th moment of the service time for jobs of class ℓ. Then the k-th moment of the overall service time is given by $E[S^k] = \sum_{\ell=1}^{L} p_\ell E[S_\ell^k]$. We assume that the time a job has to wait for service and the job's actual service time are independent (this seems to be true for each job class considered separately). This allows us to readily obtain the second moment of the sojourn time in system as

$$E[W^2] = E[W_q^2] + E[S^2] + 2E[W_q]E[S]. \tag{7}$$

In our case, $[S_\ell^k] = k!t_\ell^k$, and we have $E[W_q] = E[N_q]/\lambda$ from Little's formula. The variance of the sojourn time for a job is given by $\text{Var}[W] = E[W^2] - (E[W])^2$. Hence, we can use the one-sided inequality [1] to obtain an upper bound on the probability that the sojourn time exceeds some value t,$t > E[W]$

$$\text{Prob}\{W > t\} \le \frac{\text{Var}[W]}{\text{Var}[W] + (t - E[W])^2}. \tag{8}$$

How useful this bound may be depends on the specific values of the quantities involved, but, in any case, we have the variance of the job sojourn time. This gives us crucial information about the variability of the job response time. Higher moments of the latter can be obtained in an analogous manner.

A simple example in the next section illustrates this approach.

III. Simple Example

Consider a system with a total resource of size $B = 6$ and two job classes with respective resource requirements $b_1 = 2$ and $b_2 = 3$. Feasible states with jobs waiting in the queue $(n > m)$ can be generated from $m_1 = 0, \ldots, [B/b_1]$ and $m_2 = [(B - m_1b_1)/b_2]$: (0,2,1), (0,2,2), (1,1,1), (1,1,2), (2,0,2),

(3,0,1), (3,0,2). State (2,0,1) is not feasible since $B-2b_1 \geq b_1$. As an example, for the states (2,0,2) and (3,0,2), from the corresponding balance equations together with formulas (1) and (3), we obtain the following equations for the conditional probabilities.

$$p(2,0,2|n)(2\mu_1+\lambda) = p(2,0,2|n-1)u(n) + [p(1,1,1|n+1)\mu_2 p_2 + p(3,0,2|n+1)3\mu_1]\lambda/u(n+1)$$

$$p(3,0,2|n)(3\mu_1+\lambda) = p(3,0,2|n-1)u(n) + [p(1,1,1|n+1)\mu_2 p_1 p_2 + p(3,0,1|n+1)3\mu_1 p_2]\lambda/u(n+1)$$

Taking the limit for $n \to \infty$, we obtain

$$\tilde{p}(2,0,2)(2\mu_1+\lambda) = \tilde{p}(2,0,2)\tilde{u} + [\tilde{p}(1,1,1)\mu_2 p_2 + \tilde{p}(3,0,2)3\mu_1]\lambda/\tilde{u}$$

and

$$\tilde{p}(3,0,2)(3\mu_1+\lambda) = \tilde{p}(3,0,2)\tilde{u} + [\tilde{p}(1,1,1)\mu_2 p_1 p_2 + \tilde{p}(3,0,1)3\mu_1 p_2]\lambda/\tilde{u}$$

Similar equation can be obtained for the remaining limiting probabilities. Their solution using a fixed point iteration together with (4) and (5) yields a value for the limiting job processing rate $\tilde{u}$ that depends on the rate of arrivals λ. The asymptotic maximum value for the job processing rate is obtained when the arrival rate becomes equal to the processing rate, i.e., for $\lambda = \tilde{u}$. This latter relationship can be injected into the fixed-point iteration.

An alternative is to use $\lambda = \tilde{u}$ directly in the equations for $\tilde{p}(m_1, m_2, i)$ simplifying them even further down to $\tilde{p}(2,0,2)(2\mu_1) = \tilde{p}(1,1,1)\mu_2 p_2 + \tilde{p}(3,0,2)3\mu_1$ and $\tilde{p}(3,0,2)(3\mu_1) = \tilde{p}(1,1,1)\mu_2 p_1 p_2 + \tilde{p}(3,0,1)3\mu_1 p_2$. For other feasible states we get $\tilde{p}(3,0,1)(3\mu_1) = \tilde{p}(1,1,1)\mu_2 p_1^2 + \tilde{p}(3,0,1)3\mu_1 p_1$, $\tilde{p}(1,1,1)(\mu_1+\mu_2) = \tilde{p}(0,2,1)2\mu_2 p_1 + \tilde{p}(1,1,1)\mu_1 p_1 + \tilde{p}(2,0,2)2\mu_1 p_1 + \tilde{p}(1,1,2)\mu_2 p_1$, $\tilde{p}(1,1,2)(\mu_1+\mu_2) = \tilde{p}(0,2,1)2\mu_2 p_2 + \tilde{p}(1,1,1)\mu_1 p_2 + \tilde{p}(2,0,2)2\mu_1 p_2 + \tilde{p}(1,1,2)\mu_2 p_2$, $\tilde{p}(0,2,1)(2\mu_2) = \tilde{p}(0,2,2)2\mu_2 p_1 + \tilde{p}(1,1,2)\mu_1 p_1$ and $\tilde{p}(0,2,2)(2\mu_2) = \tilde{p}(0,2,2)2\mu_2 p_2 + \tilde{p}(1,1,2)\mu_1 p_2$.

The solution of this system of linear equations (subject to the normalizing condition (5)) yields the asymptotic maximum job processing rate via formula (4).

In Figure 2 we show the limiting processing rate $\tilde{u}$ as a function of the arrival rate λ for our example system with $\mu_1 = \mu_2 = 1$ and $p_1 = p_2 = 0.5$.

We observe that between arrival rate values of $\lambda = 2$ and $\lambda = 2.1$ the limiting job completion rate approaches its maximum value for which we still have $\tilde{u} > \lambda$. This is the maximum job processing rate for the given set of parameters.

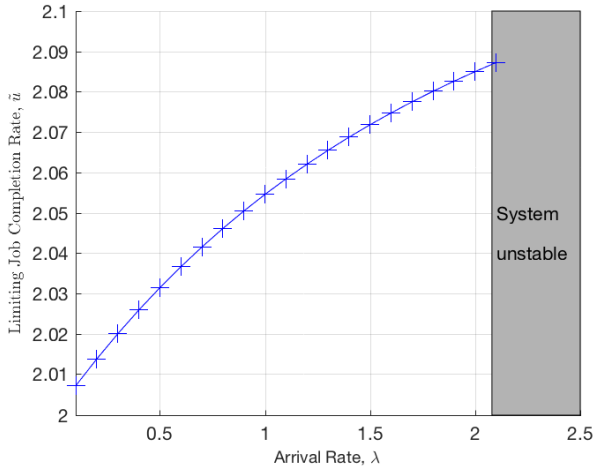

Fig. 2: Limiting job completion rate as a function of arrival rate.

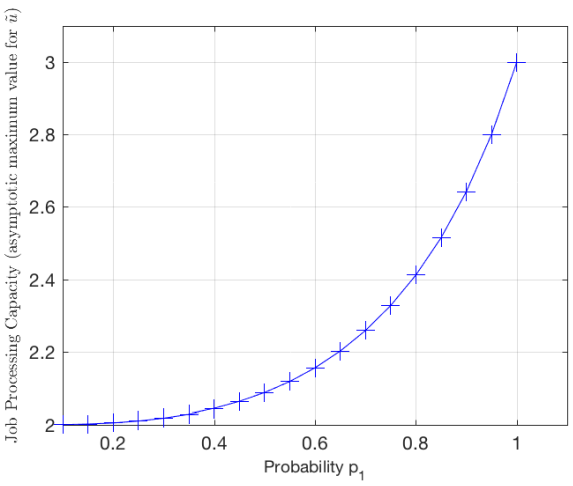

Fig. 3: Job processing capacity as a function of job mix.

Figure 3 shows the values of the asymptotic maximum job processing rate as a function of p_1, the probability that an arriving job is of class 1.

Clearly, when all jobs belong to a single class, we can easily determine the maximum job completion rate. It is interesting to see that the asymptotic maximum job completion rate is, in our example, a strongly non-linear function of the probabilities p_ℓ so that even a small proportion of more resource demanding jobs can have an important effect on the job processing capacity of a multiserver job queue (and, thus, a cloud datacenter).

To illustrate the accuracy of the results obtained for the second moment of the job sojourn time, we consider again a system with a total resource of size $B = 6$ and two job classes with respective resource requirements $b_1 = 2$ and $b_2 = 3$. We assume that we have $p_1 = p_2 = 0.5$ but each job class has a different mean service time $t_1 = 1/\mu_1 = 0.4$ and $t_2 = 1/\mu_2 = 1.6$) so that the overall mean service time is 1.

TABLE I: Example of results for second moments of sojourn time and queueing time.

Arrival rate, λ	$E[W]$	$E[W^2]$	$E[W_q^2]$	Prob$\{W > 10\}$
1.0	1.438	4.763	1.167	0.036
	(1.438)	(4.843)	(1.245)	(0.003)
1.2	1.743	6.758	2.553	0.052
	(1.741)	(6.814)	(2.610)	(0.006)
1.4	2.271	11.120	5.858	0.091
	(2.273)	(11.216)	(5.946)	(0.016)
1.6	3.341	23.416	16.013	0.217
	(3.347)	(23.513)	(16.090)	(0.056)
1.8	6.471	85.707	72.044	0.779
	(6.467)	(86.367)	(72.702)	(0.216)

We show in Table 1 a few values of the first two moments of the sojourn time, as well as the second moment of the time in queue obtained using the approach described in Section II. As a "sanity check", the values in parenthesis give the corresponding moments estimated in a discrete-event simulation of the system considered. The simulation used 7 independent replications of 5,000,000 job completions each. We notice the generally good agreement between the values computed from the solution of our model and those estimated in the simulation.

We also show the upper bound values obtained for the probability Prob$\{W > t = 10\}$ from the one-sided inequality. The values in parenthesis in this column give the values for the fraction of sojourn times that exceeded the given target value in the simulation runs. Clearly, the one-sided inequality leads to rather conservative bounds in our case, and, for $\lambda = 1.8$, a better upper bound for Prob$\{W > 10\}$ can be obtained from the basic Markov's inequality Prob$\{W > t\} \leq E[W]/t = 0.6471$. Clearly, this a very simple example and the number of feasible states may grow very rapidly with the number of jobs classes and the size of the resource. A potential approach to explore in such cases might be class aggregation in which a few dominant job classes are represented explicitly while others are aggregated. This the subject of future work.

IV. Conclusions

We have considered a generalization of the multiserver jobs model and presented a simple approach to its analysis in steady state. Of particular interest in such a system is the determination of its job processing capacity. Our approach is based on a suitably chosen state description and the use of conditional probabilities. Studying the limiting values of these conditional probabilities allows us to determine the asymptotic maximum job processing capacity without having to obtain the full solution for the system. Additionally, we show that with our state description generalized Little's law can be used in the systems considered to obtain higher moments of the time jobs spend in the queue and in the system as a whole.

References

[1] Arnold O. Allen. *Probablity, Statistics and Queueing Theory with Computer Science Applications, Second Edition*. Elsevier, 1990.

[2] Tulin Atmaca, Thomas Begin, Alexandre Brandwajn, and Hind Castel-Taleb. Performance evaluation of cloud computing centers with general arrivals and service. *IEEE Transactions on Parallel and Distributed Systems*, 27(8):2341–2348, 2016.

[3] Mor Harchol-Balter. Open problems in queueing theory inspired by datacenter computing. *Queueing Systems*, 97(1):3–37, 2021.

[4] Hamzeh Khazaei, Cornel Barna, Nasim Beigi-Mohammadi, and Marin Litoiu. Efficiency analysis of provisioning microservices. In *2016 IEEE International conference on cloud computing technology and science (CloudCom)*, pages 261–268. IEEE, 2016.

[5] Hamzeh Khazaei, Jelena Misic, and Vojislav B Misic. Performance analysis of cloud computing centers using M/G/m/m+r queuing systems. *IEEE Transactions on parallel and distributed systems*, 23(5):936–943, 2011.

[6] Yukio Takahashi. Asymptotic exponentiality of the tail of the waiting-time distribution in a ph/ph/c queue. *Advances in Applied Probability*, 13(3):619–630, 1981.

[7] Ward Whitt. A review of $L = \lambda W$ and extensions. *Queueing Systems*, 9(3):235–268, 1991.