Stochastic Bounds for Randomized Load Balancing

Antonie S. Godtschalk
Dept. of Mathematics and Computer Science
Eindhoven University of Technology
a.s.godtschalk@student.tue.nl

Florin Ciucu
T-Labs / TU Berlin
florin@net.t-labs.tu-berlin.de

ABSTRACT

Randomized load balancing is a cost efficient policy for job scheduling in parallel server queueing systems whereby, with every incoming job, a central dispatcher randomly polls some servers and selects the one with the smallest queue. By exactly deriving the jobs' delay distribution in such systems, in explicit and closed form, Mitzenmacher [5] proved the so-called 'power-of-two' result, which states that by randomly polling only two servers yields an exponential improvement in delay over randomly selecting a single server. Such a fundamental result, however, was obtained in an asymptotic regime in the total number of servers, and does do not necessarily provide accurate estimates for practical finite regimes with small or moderate number of servers. In this paper we obtain stochastic lower and upper bounds on the jobs' average delay in non-asymptotic regimes, by borrowing ideas for analyzing the particular case of the Join-the-Shortest-Queue (JSQ) policy. Numerical illustrations indicate not only that the obtained bounds are remarkably accurate, but also that the existing exact but asymptotic results can be largely misleading in some finite regimes.

Categories and Subject Descriptors

G.3 [**Mathematics of Computing**]: Probability and Statistics—*Queueing Theory*

General Terms

Theory, Performance

Keywords

SQ(d), JSQ, matrix-geometric analysis

1. INTRODUCTION

Parallel server queueing systems model a wide range of scenarios related to daily situations, e.g., toll booths, bank tellers, supermarket cashiers, etc., or to computer and communication systems, e.g., multi-processor systems, data centers, etc. Scheduling in these complex systems concerns the assignment of a single server to execute each arriving job. Existing scheduling policies reveal an interesting tradeoff between 1) the optimality of some performance metric, e.g., jobs' (average) delay, and 2) cost efficiency, e.g., in terms of minimizing the amount of overhead. At one extreme, the policy of (non-)randomly selecting a server has no feedback cost (as communication from the servers to the dispatcher) but conceivably leads to very large delays, and even to instabilities when the selection process is not adequately balanced. At the other extreme, the *Join the Shortest Queue* (JSQ) policy, whereby the dispatcher sends each job to the server with the shortest queue, minimizes delay but has a very high feedback cost because all servers must report their queue lengths for every job arrival, and thus raises a valid concern regarding practical implementations.

In order to reduce the feedback cost, and yet to keep the delay 'small', JSQ has been generalized to SQ(d), whereby the dispatcher runs JSQ only for a *subset* of d randomly sampled servers from the uniform distribution (see Mitzenmacher [5] and Luczak and McDiarid [3]). Note that SQ(d) reduces to a simple uniform random selection when $d = 1$, and to JSQ when $d = N$, where N is the total number of servers. A fundamental qualitative result is that SQ(2) yields an exponential improvement over SQ(1) in terms of delay, yet with a conceivably small overhead cost. This result is known as the 'power-of-two' result [5].

Despite its apparent simplicity, SQ(d) is very difficult to analyze in terms of the delay metric, even for a classical input with Poisson arrivals and exponential job sizes. In fact, SQ(d) can be exactly analyzed only for $d = 1$, in which case the problem reduces to the M/M/1 queue. What makes the problem particularly difficult, when $d > 1$, is that the generator matrix of an underlying N-dimensional Markov chain (representing, for instance, the number of jobs at each of the servers' queues) has an irregular structure. For this reason, solutions have been so far developed either in asymptotic regimes or in terms of bounds in particular cases.

An exact solution on the delay distribution was obtained in an asymptotic regime in the total number of servers, i.e., for $N \to \infty$ [5]; this solution was instrumental to showing the 'power-of-two' result. Lower and upper bounds on delay were obtained for the particular case when $d = N$, i.e., JSQ. The main idea is to transform the original Markov chain with the inherent irregular structure into Markov chains with some regular structure (see Adan et al. [1], Lui et al. [4], or Zhao and Grassmann [7]). To get a lower bound, for instance, the transformation consists in redirecting some transitions between the states of the original Markov chain in such a way that the new system is less loaded than the original one. Moreover, the newly formed generator matrix has a periodic structure such that its analysis becomes amenable to matrix-geometric techniques (Neuts [6]).

MAMA Workshop, 2012 London, United Kingdom

In this paper we extend such methods for computing lower and upper delay bounds to the general SQ(d) case. The extension is not straightforward, but on the contrary, because of a much more compounded transformation process needed to produce Markov chains with a regular structure. We thus provide the first non-asymptotic results for SQ(d) policy which can be applied in finite regimes with small to moderate number of servers. A drawback of the obtained bounds, however, is that they are obtained in implicit form, as they are based on matrix-geometric techniques, and are thus unable to provide qualitative insight alike the 'power-of-two' result. However, the bounds are numerically tractable, and are remarkably tight. Moreover, the bounds illustrate that the asymptotic result from [5] can be largely inaccurate (e.g., smaller than the lower bound) in regimes with small number of servers.

The rest of the paper is organized as follows. In Section 2 we summarize the main ideas for computing lower and upper bounds for the SQ(d) policy. In Section 3 we numerically test the accuracy of the obtained bounds, and in Section 4 we conclude the paper.

2. MODEL AND ANALYSIS

We consider the general SQ(d) scheduling policy with N parallel servers. Jobs arrive at a central dispatcher according to a Poisson process with rate λN, and their service times are exponentially distributed with unit mean. With every arriving job, the dispatcher randomly polls d servers according to a uniform distribution without replacement, out of the N servers. The d selected servers report the number of jobs in their systems, and the newly arriving job joins the server with the smallest number of existing jobs; ties are resolved arbitrarily. At every server, jobs are served according to the FIFO policy. We impose the stability condition $\lambda N < 1$.

The Poisson/exponential arrivals' model enables the construction of a continuous-time Markov chain to model the evolution of the SQ(d) policy. The set of states is

$$\mathcal{M} = \{\boldsymbol{m} : \boldsymbol{m} = (m_1, m_2, \ldots, m_N)\} \ ,$$

where m_1 denotes the largest number of jobs at the N servers, m_2 denotes the second largest number of jobs, and so on, such that m_N denotes the smallest number of jobs. The corresponding generator matrix is denoted by Q, and the steady-state distribution of the number of jobs in the systems $\boldsymbol{\pi} = \pi\boldsymbol{m}$ can be obtained by solving

$$\boldsymbol{\pi} Q = \boldsymbol{0}, \ \boldsymbol{\pi}\boldsymbol{e} = 1 \ .$$

For example, if $N = 3$, then $\boldsymbol{\pi} = (\pi_{(0,0,0)}, \pi_{(1,0,0)}, \ldots)$, where $\pi_{(0,0,0)}$ denotes the equilibrium probability for the state $(0, 0, 0)$ with no jobs in the whole system.

As we have pointed out earlier, the computation of the steady-state distribution $\boldsymbol{\pi}$ is hampered by the irregular structure of the generator matrix Q. In order to circumvent this problem we next borrow ideas from the JSQ analysis (see Adan et al. [1]), whereby the original Markov chain is transformed by suitably redirecting transitions such that the new generator matrix has some regular structure. Concretely, we introduce a threshold parameter T such that, in the transformed Markov chains (one for getting lower bounds and another for getting upper bounds), the following condition must hold

$$m_1 - m_N \leq T \ . \tag{1}$$

To enforce this condition we suitably redirect some transitions from the original chain. In particular, to get a stochastic lower bound, we redirect transitions according to the following two rules:

1. When a departure causes the violation of Eq. (1), then the departure occurs from (one of) the longest queue(s) instead of the shortest queue.

2. When an arrival causes the violation of Eq. (1), then the arrival is sent to (one of) the shortest queue(s) instead of the longest queue.

In turn, to get a stochastic upper bound for SQ(d), we redirect the transitions in the following way:

1. When a departure causes the violation of Eq. (1), then the departure may not occur.

2. When an arrival causes the violation of Eq. (1), then the arrival is accompanied by the addition of one extra job at each of the shortest queues.

The key advantage of these transformations is that we can partition the newly constructed state spaces (for the lower/upper bounds systems) into blocks of states with a periodic structure between adjacent blocks. Moreover, each block has a finite number of states which can be further ordered according to the total number of jobs in the system; ties are broken according to a lexicographical ordering. Overall, the newly generator matrices Q have the form

$$Q = \begin{pmatrix} R_{00} & R_{01} & 0 & 0 & 0 & \cdots \\ R_{10} & A_1 & A_0 & 0 & 0 & \cdots \\ 0 & A_2 & A_1 & A_0 & 0 & \cdots \\ 0 & 0 & A_2 & A_1 & A_0 & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots \end{pmatrix} .$$

Here, R_{00}, R_{01} and R_{10} correspond to the matrices created by transition rates within the boundary blocks, transitions from a non-boundary block to a boundary block and from a boundary block to a non-boundary block, respectively. The non-boundary blocks, i.e., A_0, A_1 and A_2, are of order $m \times m$, where m is the number of states in such a block, i.e.,

$$m = \binom{N + T - 1}{T} .$$

The generator matrix Q is irreducible and, since two states in different non-boundary blocks can reach each other via paths not passing through the boundary block R_{00}, the generator $A_0 + A_1 + A_2$ is also irreducible. We define the stationary probabilities $(\vec{p}_0, \vec{p}_1, \vec{p}_2, \ldots)$, where $\vec{p}_0$, $\vec{p}_1$ and $\vec{p}_2$ correspond to the equilibrium probability vectors of the boundary block, and the first and second non-boundary block, respectively. Then, Theorem 1.7.1 of Neuts [6] yields the solutions of $(\vec{p}_0, \vec{p}_1, \vec{p}_2, \ldots)$ for the lower and upper bound models, i.e.,

$$\vec{p}_{q+1} = \sigma^N \vec{p}_q \text{ and } \vec{p}_{q+1} = R\vec{p}_q, \ q = 1, 2, \ldots \ , \tag{2}$$

respectively. Here, σ is the unique solution, inside the unit circle, of the equation in x

$$x = \sum_{k\geq 0} x^k \beta_k \ , \tag{3}$$

where β_k is the probability that there are k departures within two arrivals, i.e.,

$$\beta_k = \int_0^\infty \frac{t^k}{k!} \mathrm{e}^{-t} dA(t) \ .$$

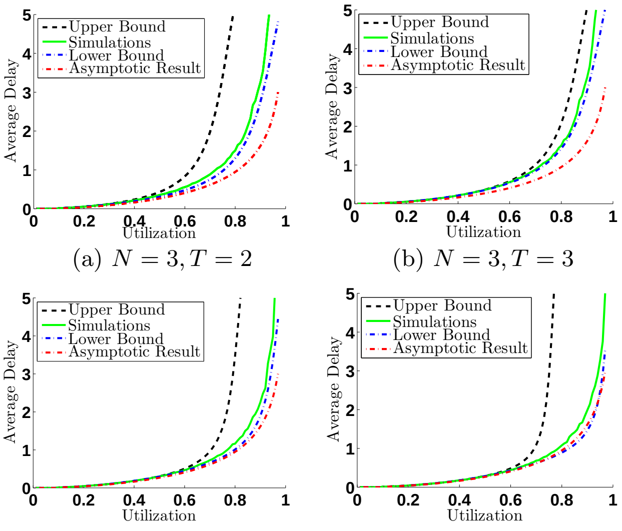
Solving for Eq. (3) in the case of the SQ(d) lower bound model yields $\sigma = \rho$, where $\rho = \lambda N$ is the system's utilization. In turn, for the SQ(d) upper bound model, the rate matrix R from Eq. (2) can explicitly be computed (see Latouche and Ramaswami [2]). Next, $(\vec{p}_0, \vec{p}_1, \vec{p}_2)$ can be obtained by solving the following system of equations for the lower/upper bound models

$$(\vec{p}_0, \vec{p}_1, \vec{p}_2) \begin{pmatrix} R_{00} & R_{01} & 0 \\ R_{10} & A_1 & A_0 \\ 0 & A_2 & A_1 + RA_2 \end{pmatrix} = (\vec{p}_0, \vec{p}_1, \vec{p}_2) \ ,$$

where $R = \rho^N$ for the lower bound model. From these equilibrium probabilities of the states, we can obtain the lower and upper bounds on the average delay. Concretely, for each state we know the number of waiting jobs at each server, i.e., server i has $\max\{(m_i - 1), 0\}$ jobs, and we can multiply this number by the equilibrium probability of the corresponding state. By doing so for all states, we can numerically compute the bounds on the jobs' average delay.

3. NUMERICAL RESULTS

To test the accuracy of our lower and upper bounds, we simulated the SQ(d) model for 10^6 transitions (arrivals and departures), of which we ignored the first 10^5 when calculating the jobs' average delay. For comparison, we also consider the exact, but asymptotic, results from [5].



(a) $N = 3, T = 2$

(b) $N = 3, T = 3$

(c) $N = 6, T = 3$

(d) $N = 10, T = 3$

Figure 1: Average delay as a function of utilization for SQ(2) for various number of servers $N = 3, 6, 10$ and threshold parameter $T = 2, 3$

In Figure 1.(a-d) we show the average delay as a function of utilization for SQ(2). The first observation is that there is a tradeoff between the accuracy of the upper bounds and the computational complexity. Indeed, (a) and (b) indicate that the upper bounds are quite loose by letting $T = 2$, and are getting significantly tighter by letting $T = 3$. However, the numerical complexity increases significantly with T because the sizes of the (non-)boundary blocks in the generator matrix Q are exponential in T. As a related remark, different values of T change the stability condition for the SQ(d) upper bound (recall the last two rules for redirecting transitions from the previous section). The second observation is that the lower bounds are accurate over all three values of N, i.e., 3, 6, and 10. Finally, the asymptotic results (which are in fact invariant to the value of N), significantly underestimate the 'true' results for small values of N, and especially at high utilizations.

4. CONCLUSIONS

In this paper we have considered the SQ(d) scheduling policy, and we have carried out matrix-analytical methods to compute stochastic lower and upper bounds on the average delay. The merit of the obtained bounds is that they hold in non-asymptotic regimes, and thus complement existing exact results obtained in asymptotic regimes. Numerical results revealed that there is an interesting tradeoff between the accuracy of the obtained upper bounds and the dimension of the computational complexity. Moreover, the lower bounds are remarkably tight, whereas the asymptotic results may be misleading in finite regimes.

Acknowledgment

The authors thank Sem Borst for many insightful comments.

5. REFERENCES

[1] I. J. B. F. Adan, G. J. van Houtum, and J. van der Wal. Upper and Lower Bounds for the Waiting Time in the Symmetric Shortest Queue System. *Annals of Operations Research*, 48(2):197–217, Apr. 1994.

[2] G. Latouche and V. Ramaswami. A Logarithmic Reduction Algorithm for Quasi-Birth-Death Processes. *Journal of Applied Probability*, 30(3):650–674, Sept. 1993.

[3] M. J. Luczak and C. McDiarmid. On the Maximum Queue Length in the Supermarket Model. *Annals of Probability*, 34(2):493–527, May 2006.

[4] J. Lui, R. R. Muntz, and D. Towsley. Bounding the Mean Response Time of the Minimum Expected Delay Routing Policy: An Algorithmic Approach. *IEEE Transactions on Computers*, 44(12):1371–1382, Dec. 1995.

[5] M. Mitzenmacher. The Power of Two Choices in Randomized Load Balancing. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1094–1104, Oct. 2001.

[6] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: an Algorithmic Approach.* John Hopkins University Press, 1981.

[7] Y. Zhao and W. K. Grassmann. Queueing Analysis of a Jockeying Model. *Operations Research*, 43(3):520–530, May-June 1995.