The effect of local scheduling in load balancing designs

Ho-Lin Chen
Center for Mathematics of Information
California Institute of Technology
Pasadena, CA 91125

Jason R. Marden
Social Information Sciences Laboratory
California Institute of Technology
Pasadena, CA 91125

Adam Wierman
Computer Science Department
California Institute of Technology
Pasadena, CA 91125

1. INTRODUCTION

Load balancing is a common approach to task assignment in distributed architectures such as web server farms, database systems, grid computing clusters, and others. In such designs there is a dispatcher that seeks to balance the assignment of service requests (jobs) across the servers in the system so that the response time of jobs at each server is (nearly) the same. Such designs are popular due to the increased robustness they provide to bursts of traffic, server failures, etc., as well as the inherent scalability they provide. However, there is also a major drawback to load balancing designs – some performance is sacrificed. Specifically, it would be possible to reduce user response times by moving away from load balancing designs.

Our goal in this paper is to study the degree of inefficiency in load balancing designs. Further, we will show that the degree of inefficiency depends on the scheduling discipline used locally at each of the servers, i.e. the local scheduler.

Our results (see Section 3) show that the local scheduling policy has a significant impact on the degree of inefficiency in load balancing designs. In particular, the local scheduler in traditional designs is often modeled by Processor Sharing (PS), which shares the server evenly among all jobs in the system. When the local scheduler is PS, the degree of inefficiency grows linearly with the number of servers in the system. In contrast, if the local scheduler is changed to Shortest Remaining Processing Time first (SRPT), as has been suggested in a variety modern designs [7, 3, 10], the degree of inefficiency can be independent of the number of servers in the system and instead depend only on the heterogeneity of the speed of the servers.

2. MODEL DESCRIPTION

To study the efficiency of load balancing designs, we will use the queueing model pictured in Figure 1. The system consists of n parallel queues $Q_1, \ldots, Q_n$ with service rates $\mu_1, \ldots, \mu_n$ where $\mu_i \geq \mu_{i+1}$. Let X_i be a random i.i.d. job size at queue i having p.d.f. $f_i(x)$ and c.d.f. $F_i(x)$. Let $\overline{F}_i(x) = 1 - F_i(x)$ and note that $E[X_i] = 1/\mu_i$.

The arrival process to the system is Poisson with rate Λ, where $\Lambda < \sum_{i=1}^{n} \mu_i$. There is a load balancing dispatcher that probabilistically routes arrivals to queues so that the mean response time (a.k.a. sojourn time, flow time), $E[T_i]$, at each queue is the same. This model follows from the assumption that routing decisions are made without observing the queue length and that the dispatcher reacts to periodic performance measurements attained from each queue with the goal of balancing response times across the queues. This is a common design constraint in distributed web servers.

The resulting arrival rate to Q_i is Poisson with rate λ_i and the load at queue i is $\rho_i := \lambda_i/\mu_i < 1$. Further, we define the remaining service capacity, a.k.a., "gap", at Q_i as $\gamma_i := \mu_i - \lambda_i > 0$. Thus, each Q_i is a stationary M/GI/1 queue. Note that all incoming jobs are routed to one of the servers, i.e., there is no balking. For reasons that we will describe in Section 2.1, *we will be considering the heavy-traffic behavior of this system.* That is, we will analyze the response time as $\Lambda \to \sum_i \mu_i$, which also ensures that each Q_i is in heavy-traffic, i.e. $\rho_i \to 1$.

Interestingly, we can take a slightly different view of the load balancing model than we have described to this point which will prove useful. In particular, can view the stationary behavior of the load balancing system as the equilibrium point of a non-atomic routing game. This is often termed a "selfish routing game" [8]. In particular, a set of λ_i correspond to Nash equilibrium arrival rates in this non-atomic game if for all Q_i, if $\lambda_j > 0$, then $E[T_i] \leq E[T_j]$. In this setting the *price of anarchy* (PoA) is defined as the worst-case ratio between a Nash equilibrium and the global optimum. We will use this measure to characterize the inefficiency in load balancing designs.

2.1 Local scheduling

We will consider two possibilities for the local scheduler: PS and SRPT. PS is important because it is commonly used as a model for the traditional scheduling designs in many computer systems including, bandwidth scheduling in web servers, flow scheduling in routers, and CPU scheduling in operating systems. SRPT is important because it is known to minimize the mean response time in a single server queue [13] regardless of the arrival and service process. Further, it has recently been suggested as an alternative to PS in many applications [7, 3, 10].

There is a large literature studying each of these policies and we refer the reader to [19] for background on PS and to [5, 15] for background on SRPT. For our purposes, we will need only results characterizing the mean response time under these policies.

In an M/GI/1 PS queue $E[T_i]$ has the following simple form:

$$E[T_i] = \frac{1}{\mu_i - \lambda_i}$$

The mean response time under SRPT has a much more complicated form. Let $m_{ij}(x) = E[X_i^j 1_{X_i<x}] = \int_0^x t^j f_i(t)dt$ be a truncated j-th moment of job size distribution X_i. Let $\widetilde{m_{ij}}(x) = E[\min(x, X_i^j)] = i\int_0^x t^{j-1}\overline{F}_i(t)dt$ be a different truncated j-th moment. Finally, let $\rho_i(x) = \lambda_i m_{i1}(x)$. Now, we can write the

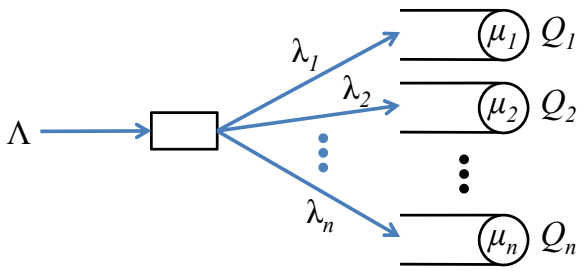

Figure 1: A diagram of the load balancing model considered in this paper.

mean response time of SRPT in an M/GI/1 queue as follows:

$$E[T_i] = \int_0^\infty \left(\int_0^x \frac{1}{1-\rho_i(t)} dt + \frac{\lambda_i \widetilde{m_{i2}}(x)}{2(1-\rho_i(x))^2} \right) dF_i(x)$$

The complicated form of $E[T_i]$ under SRPT makes it difficult to study this policy directly. Instead, we will consider the behavior of this policies in *heavy-traffic*, i.e., as $\Lambda \to \sum_i \mu_i$. In this case, recent results provide a simpler form under certain job size distributions including bounded distributions [15, 17], exponential distributions [1], and Pareto distributions [2, 16]. *We will limit ourselves to Pareto job size distributions in this paper.* The following proposition follows from combining the results in [2, 16].

PROPOSITION 1. *Consider an M/GI/1 SRPT queue, as $\rho_i \to 1$, and $X_i \sim Pareto(\alpha, x_L)$ with $\alpha > 1$, i.e., $\overline{F}_i(x) = (x/x_L)^\alpha$ for some $x_L > 0$, then*

$$E[T_i] = \begin{cases} \Theta\left(\log\left(\frac{1}{1-\rho_i}\right)\right), & \text{if } \alpha < 2 \text{ (Case A);} \\ \Theta\left(\log^2\left(\frac{1}{1-\rho_i}\right)\right), & \text{if } \alpha = 2; \\ \Theta\left(\frac{1}{(1-\rho_i)^{\frac{\alpha-2}{\alpha-1}}}\right), & \text{if } \alpha > 2 \text{ (Case B).} \end{cases}$$

Note that we will focus only on the case of $\alpha \neq 2$.

Proposition 1 only specifies the growth rate of $E[T_i]$ under SRPT in heavy-traffic, and we need a simple equation to facilitate our analysis. So, we will use the following functional form that encompasses both $E[T_i]$ under PS and an approximation for $E[T_i]$ under SRPT in heavy-traffic. This approximation has been shown using simulations to be accurate for SRPT [15]. Further, it matches the bounds on SRPT derived in [16].

$$E[T_i] := \begin{cases} \frac{1}{\mu_i} \log\left(\frac{\mu_i}{\mu_i - \lambda_i}\right), & \text{(Case A);} \\ \frac{1}{\mu_i} \left(\frac{\mu_i}{\mu_i - \lambda_i}\right)^m, & \text{(Case B).} \end{cases} \quad (1)$$

Under this formulation, we see that the contribution of Q_i to the overall response time is given by $\lambda_i E[T_i]$ which, in heavy-traffic, becomes

$$C_i(\lambda_i) := \begin{cases} \log\left(\frac{\mu_i}{\mu_i - \lambda_i}\right), & \text{(Case A)} \\ \left(\frac{\mu_i}{\mu_i - \lambda_i}\right)^m, & \text{(Case B)} \end{cases}$$

Thus, for a given set of arrival rates, $(\lambda_1, ..., \lambda_n)$, the overall average response time of the system in heavy traffic is

$$E[T; (\lambda_i, \ldots, \lambda_n)] := \sum_i C_i(\lambda_i).$$

2.2 Prior work on selfish load balancing

In their seminal work [6], Koutsoupias & Papadimitriou introduced the concept of the price of anarchy in the context of a 2 server load balancing game. Following that work, there was an explosion of research studying selfish load balancing games in both the atomic and non-atomic settings, see [8] for a survey.

Our context is the non-atomic load balancing game, which is a special case of the non-atomic routing game that was first introduced by Pigou [9] and later was formally defined by Wardrop [14]. For this reason, equilibriums in these games are often called "Wardrop equilibria." The price of anarchy was first studied in this setting by Roughgarden & Tardos [12], and the work that followed is surveyed in [8]. The fundamental result for non-atomic routing games is that the price of anarchy is independent of the network structure under a wide variety of latency (response time) functions.

However, there are very few results in the case of latency functions specified by queueing models, as we consider in this paper. In particular, the only bounds on the price of anarchy that are known to hold independently of the network structure hold only when $\Lambda < \min_i \mu_i$, see [11]. The reason for this is that outside of this light-traffic regime the network structure matters. Recently, the first result to characterize the impact of the network structure was provided by independently by Haviv & Roughgarden [4] and Wu & Starobinski [18] who proved that, in the model of this paper, the price of anarchy under PS local scheduling is $O(n)$. Further, they provide an example illustrating that this is tight. In the current paper, we extend this work to consider a different local scheduler: SRPT.

3. RESULTS

We have two sets of results that we will describe. First, we will state explicit results characterizing the globally optimal (Section 3.1) and the load balancing (Section 3.2) routing designs. Then, in Section 3.3, we will derive bounds on the inefficiency in load balancing designs, i.e., the price of anarchy.

3.1 Globally optimal routing

In this section we consider the global optimization problem where a dispatcher needs to determine the λ_i in order to minimize the overall mean response time $E[T]$. That is,

$$\min \sum_i C_i(\lambda_i)$$
$$\text{s.t.} \sum_i \lambda_i = \Lambda;$$
$$0 \leq \lambda_i < \mu_i.$$

We can solve this optimization explicitly to obtain

$$\lambda_j^{opt} = \begin{cases} \mu_i - \frac{\left(\sum_i \mu_i\right) - \Lambda}{n}, & \text{(Case A);} \\ \mu_j - \left(\frac{\mu_j^{m/(m+1)}}{\sum_i \mu_i^{m/(m+1)}}\right) \left(\left(\sum_i \mu_i\right) - \Lambda\right), & \text{(Case B).} \end{cases}$$

Note the intuition behind the form of these arrival rates – the excess service capacity $(\sum_i \mu_i - \Lambda)$ is divided up (Case A) evenly and (Case B) in proportion to $\mu_i^{m/(m+1)}$. Also, notice that Case A is behaving the same as Case B with $m = 0$. Finally, note that the mean response time can be written immediately in terms of these arrival rates.

3.2 Load balancing routing

In the case of (selfish) load balancing, we know that in heavy-traffic all queues are used and thus the arrival rates must satisfy

$$E[T_i] = E[T_j],\ \forall i, j \in \{1, ..., n\}.$$

From this condition, it is possible to derive the arrival rates explicitly, and we attain the following results. In Case A we find that λ_i^{ne} must satisfy

$$1 - \frac{\lambda_i^{ne}}{\mu_i} = \left(1 - \frac{\lambda_1^{ne}}{\mu_1}\right)^{\mu_i/\mu_1}$$

where λ_1^{ne} is the solution to

$$\left(\sum_i \mu_i\right) - \Lambda = \sum_i \mu_i \left(1 - \frac{\lambda_1^{ne}}{\mu_1}\right)^{\mu_i/\mu_1}$$

In Case B, we have that

$$\lambda_j^{ne} = \mu_j - \left(\frac{\mu_j^{(m-1)/m}}{\sum_i \mu_i^{(m-1)/m}}\right)\left((\sum_i \mu_i) - \Lambda\right)$$

At this point, it is interesting to contrast λ_i^{ne} with λ_i^{opt}. First, notice the similarity between λ_i^{ne} and λ_i^{opt} in Case B. Second, notice that λ_i^{ne} in Case B with $\mu = 1$ is equal to λ_i^{opt} under Case A. This means that the load balancing arrival rates for PS local scheduling are equal to the optimal arrival rates under SRPT when job sizes are Pareto with infinite variance. Again, it is also important to point out that the overall $E[T]$ can be written as a simple function of the arrival rates stated above.

3.3 Efficiency of selfish routing

Now that we have characterized the global optimum and load balancing arrival rates, we can begin to understand the inefficiency in load balancing designs. We will measure this inefficiency using the "price of anarchy", which we define in this setting as

$$\begin{aligned} \max & \frac{E[T; (\lambda_1^{ne}, \ldots, \lambda_n^{ne}), k]}{E[T; (\lambda_1^{opt}, \ldots, \lambda_n^{opt}), k]} \\ \text{s.t. } & 0 \le \lambda_i < \mu_i; \\ & \mu \le \mu_i \le k\mu. \end{aligned}$$

Note that we have inserted a parameter k that bounds the ratio of the server speeds. We will refer to k as the "heterogeneity" of the system and we will state bounds on the price of anarchy in terms of the number of servers, n, and the heterogeneity, k.

We will start by stating our result in Case A.

THEOREM 2. *The price of anarchy in Case A is $O(k)$.*

Additionally, this bound on the price of anarchy is achieved in the specific case when $\mu_1 = k\mu$ and $\mu_2 = \ldots = \mu_n = \mu$.

In Case B, we have the following result

THEOREM 3. *The price of anarchy in Case B is*

$$\max_{\mu \le \mu_j \le k\mu} \frac{\left(\sum_j \mu_j\right)\left(\sum_j \mu_j^{(m-1)/m}\right)^m}{\left(\sum_j \mu_j^{m/(m+1)}\right)^{m+1}}$$

We have solved the optimization problem above in a few special cases. For example, when $m = 1$, we have that the price of anarchy is $O(n)$. Similarly, when $m = 0$, we have that the price of anarchy is $O(k)$. Further, if $\mu_1 = k\mu$ and $\mu_2 = \ldots = \mu_n = \mu$, the price of anarchy is $O(k^{1-m}n^m)$, which provides a lower bound on the price of anarchy in general. We believe that this lower bound is tight and our current research is working to prove this result.

If we interpret these theorems in the context of local schedulers, what we see is that under PS the price of anarchy grows linearly with the number of servers in the system. In contrast, under SRPT the price of anarchy is affected to a smaller degree by the number of servers but the heterogeneity of the server speeds also plays a role. Further, in the extreme case of Pareto job sizes with infinite variance, the price of anarchy under SRPT is unaffected by the size of the system and instead grows linearly with the heterogeneity of the system.

It should be pointed out again that the above interpretation is founded on heavy-traffic approximations. However, intuitively, the heavy-traffic regime should provide the worst-case price of anarchy. Our results show that this is true under PS since the price of anarchy in heavy-traffic matches the overall price of anarchy.

Finally, let us return to an observation we made earlier – the load balancing arrival rates under PS λ_i^{ne} (Case B with $m = 1$) match the globally optimal arrival rates under SRPT λ_i^{opt} in Case A. This is interesting because it means that the ratio between $E[T]$ in these two settings matches the ratio between $E[T]$ under PS and SRPT in the M/GI/1 queue. This provides another interpretation of the price of anarchy in our setting: it characterizes the benefit attainable by switching from PS to SRPT.

4. REFERENCES

[1] N. Bansal. On the average sojourn time under M/M/1/SRPT. *Oper. Res. Letters*, 22(2):195–200, 2005.

[2] N. Bansal and D. Gamarnik. Handling load with less stress. *Queueing Systems*, 54(1):45–54, 2006.

[3] M. Harchol-Balter, B. Schroeder, M. Agrawal, and N. Bansal. Size-based scheduling to improve web performance. *ACM Transactions on Computer Systems*, 21(2), May 2003.

[4] M. Haviv and T. Roughgarden. The price of anarchy in an exponential multi-server. *Oper. Res. Letters*, 35:421–426, 2007.

[5] L. Kleinrock. *Queueing Systems*, volume II. Computer Applications. John Wiley & Sons, 1976.

[6] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proc of Symp. on Theor. Aspects of Comp. Sci.*, pages 404–413, 1999.

[7] D. Lu, H. Sheng, and P. Dinda. Size-based scheduling policies with inaccurate scheduling information. In *Proc. IEEE of MASCOTS*, 2004.

[8] N. Nissan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic game theory*. Cambridge University Press, New York, NY, USA, 2007.

[9] A. Pigou. *The Economics of Welfare*. Macmillan, 1920.

[10] M. Rawat and A. Kshemkalyani. SWIFT: Scheduling in web servers for fast response time. In *Symp. on Net. Comp. and App.*, 2003.

[11] T. Roughgarden. The price of anarchy is independent of the network topology. *J. Comp. Syst. Sci.*, 67(2):341–364, 2003.

[12] T. Roughgarden and E. Tardos. How bad is selfish routing. *J. ACM*, 49(2):236–259, 2002.

[13] L. E. Schrage. A proof of the optimality of the shortest remaining processing time discipline. *Operations Research*, 16:678–690, 1968.

[14] J. G. Wardrop. Some theoretical aspects of road traffic research. *Proc of Institute of Civil Engineers*, 1:325–378, 1952.

[15] A. Wierman. *Scheduling for today's computer systems: Bridging theory and practice*. PhD thesis, Carnegie Mellon University, 2007.

[16] A. Wierman, M. Harchol-Balter, and T. Osogami. Nearly insensitive bounds on SMART scheduling. In *Proc. of ACM Sigmetrics*, 2005.

[17] A. Wierman and M. Nuyens. Scheduling despite inexact job-size information. In *Proc. of ACM Sigmetrics*, 2008.

[18] T. Wu and D. Starobinski. On the price of anarchy in unbounded delay networks. In *Proc. of Game Theory for Comm. and Networks*, 2006.

[19] S. Yashkov. Mathematical problems in the theory of shared-processor systems. *J. of Soviet Mathematics*, 58:101–147, 1992.