Why Segregating Short Jobs from Long Jobs under High Variability is Not Always a Win

Mor Harchol-Balter and Alan Scheller-Wolf and Andrew Young

***Abstract*— This paper investigates the performance of task assignment policies for server farms as the variability of job sizes (service demands) approaches infinity. The Size-Interval-Task-Assignment policy (SITA), which separates short jobs from long jobs, has long been viewed as the panacea for dealing with high-variability job-size distributions. A very recent paper [16] showed that this common wisdom is flawed: SITA can actually be inferior to the much simpler greedy policy, Least-Work-Left (LWL), for certain common job-size distributions, including many modal, hyperexponential, and Pareto distributions.**

The above finding leads one to question whether providing isolation for short jobs from long ones is inherently bad, or whether it is just SITA's strict isolation of short jobs that sometimes leads to poor performance. To answer this question, we consider a much more flexible policy, which we call "Cycle-Stealing" (CS). The CS policy is very similar to LWL, in that short jobs can go to any queue, but it still provides short jobs isolation from longs (one server is reserved for short jobs). While CS has many of the same properties as LWL, including high utilization of both servers, we prove, surprisingly, that, for high variability job sizes, CS performs poorly whenever SITA performs poorly. This result suggests that the notion of isolating short jobs from long jobs, under high variability workloads, is sometimes simply not the right thing to do.

I. Introduction

A. Task assignment policies

One of the oldest and most fundamental questions arising in server farms is the question of which dispatching policy should be used for routing jobs to servers. This policy is known as the *task assignment policy*. A common goal of the task assignment policy is to minimize mean response time, where response time is measured from when a job arrives until it completes.

We are particularly interested in situations with high job size variability. It is well-known that empirical computer workloads such as Web file sizes, CPU process lifetimes, IP flow durations, and wireless call times have very high job size variability, with job sizes fitting Pareto or other high-variance distributions [2], [7], [13], [20], [21]. This paper studies the mean response time of task assignment policies in the limit as job size variability goes to infinity, while the mean job size stays fixed. To denote job-size variability, we use the squared coefficient of variation, $C^2 = \text{var}[X]/\mathbf{E}^2[X]$, where X is a random variable representing the job size (service requirement). Job sizes are assumed to be i.i.d. from some general distribution. We assume that jobs arrive to the server farm according to a Poisson process with rate λ. For a server farm with n servers, system load ρ is defined as: $\rho = \lambda \mathbf{E}[X]$. Note that $\rho = n$ corresponds to a fully loaded system. In our analysis, we will generally assume $n = 2$ because that suffices to make our points. Importantly, we will assume that jobs are *not preemptible*. That is, a long job cannot be preempted when a short job arrives, and then resumed later. This model is common for supercomputing farms [11], [20], manufacturing systems [17], [4], data centers, IO systems, etc., where it is expensive to preempt jobs and thus even long jobs are typically run to completion.

For our server farm model, there are many common choices of task assignment policies. The *Round-Robin* policy assigns the first job to host 1, the second to host 2, the third to host 3, the ith to host i mod n plus 1, and so forth. The *Join-the-Shortest-Queue (JSQ)* policy assigns each incoming job to the host with the fewest *number* of jobs queued there. The *Least-Work-Left (LWL)* policy assigns each incoming job to the host with the least total work remaining. Here "work" is the sum of the remaining size of the job in service plus the sizes of all the jobs in the queue at the host. The *SITA (Size-Interval Task Assignment)* assigns a size-interval to each host, so that "short" jobs are sent to the first host, "medium-length" jobs are sent to the second host, and "long" jobs to the third, etc., where the cutoffs for differentiating size classes are chosen *optimally*, so as to minimize mean response time. SITA with $n = 2$ is illustrated in Figure 1. The above task assignment policies are all dispatching policies, whereby each incoming job is immediately dispatched to a host.

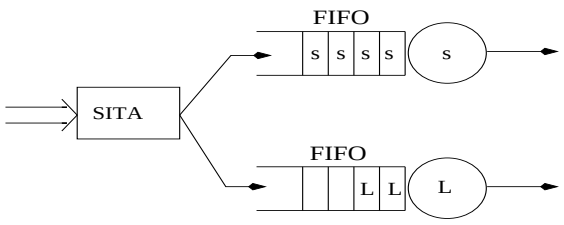


Fig. 1. *Illustration of SITA task assignment with 2 server hosts.*

Importantly, the LWL policy is *equivalent* to the classical central FIFO queue, (denoted by M/GI/n for the case of

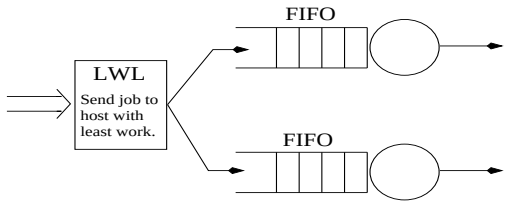
This work was supported by NSF SMA/PDOS Grant CCR-0615262.

Mor Harchol-Balter is an Associate Professor of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 15213 harchol@cs.cmu.edu

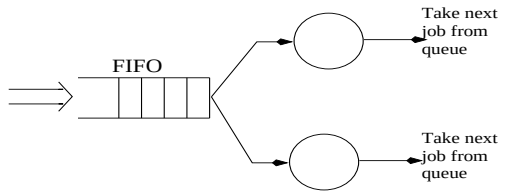
Alan Scheller-Wolf is a Professor in the Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA, 15213 awolf@andrew.cmu.edu

Andrew Young is an Executive Director at Morgan Stanley, 1585 Broadway, New York, NY, 10036 andrew.young@morganstanley.com

Poisson arrivals and n servers), where there are no queues at the hosts; instead jobs queue up in a central queue. A free host simply takes the next job from the central queue. The LWL and M/GI/2 policies are illustrated in Figure 2. Specifically, under M/GI/n, jobs go to the same host as they would have under LWL and are served there at the same time as under LWL (see [11] for an inductive proof). The response times under M/GI/n and LWL are thus identical. What's nice about this equivalence is that, while the LWL policy requires knowing the sizes of jobs, the M/GI/n policy does not.



(a) LWL



(b) M/GI/2

Fig. 2. *LWL task assignment is equivalent to M/GI/2.*

B. The advantages of SITA for high variability

While a great many papers have been written comparing the response time of different task assignment policies, e.g., [5], [6], [9], [11], [12], [19], [22], [23], all of these papers conclude (via numerical analysis, simulation, or approximation) that, for high job-size variability, the SITA policy is superior to all the other common policies mentioned above. The reason for the superiority of SITA task assignment lies in the fact that SITA allows short jobs their own "express-line," thereby giving them isolation from long jobs. Since most jobs are short jobs, the resulting mean response time is lowered. By contrast, all other policies described above *mix* short and long jobs, allowing short jobs to get "stuck" behind long jobs, greatly increasing mean response time. The SITA policy (and its variants) has been part of the common wisdom in some form or other for a long time, and has been the focus of many papers including [18], [12], [11], [23], [8], [19], [6], [22], [10], [3], [1], [9], [5], [20].

There are several papers which specifically compare the performance of SITA to LWL [3], [6], [8], [10], [11], [12], [19], [22], [23]. All of these find that as job size variability is *increased*, SITA becomes far superior to LWL (for low C^2, SITA may be worse than LWL because not all servers are utilized; however, this behavior changes quickly as C^2 is increased).

Despite these comparisons showing that SITA outperforms LWL by orders of magnitude for high job size variability, a proof of this fact has never materialized. SITA itself is difficult to analyze, even for Poisson arrivals, because in general there is no closed-form expression for the optimal size cutoff, and hence the resulting response time. Furthermore, LWL cannot be analyzed exactly, since the M/GI/n queue (equivalent to LWL) is in general only approximable. Thus, many of the existing results have used simulation to assert their claims, or have looked at phase-type job-size distributions, approximations, or heavy-traffic regimes.

C. Why SITA is not always a win for high variability

In a very recent paper [16], we show that the common wisdom about task assignment for high C^2 is wrong: We prove that SITA is not always superior to LWL as $C^2 \to \infty$; in fact SITA can be unboundedly worse than LWL. We show that both SITA and LWL can exhibit both convergent and divergent asymptotic behavior, depending on the load and job-size distribution. By convergent behavior, we mean that the mean response time approaches a constant as $C^2 \to \infty$ (while holding $\mathbf{E}[X]$ fixed) and by divergent behavior, we mean that the mean response time approaches infinity as $C^2 \to \infty$ (while holding $\mathbf{E}[X]$ fixed).

Specifically, for each box in Table I, [16] produces several examples of classes of distributions that fall within that box. This includes Box 3, which are distributions where SITA diverges and LWL converges. The examples used to illustrate these boxes are not esoteric in nature. They do not presume arcane distributions or assume very light or heavy load or a very high number of servers. Job size distributions considered include the Bimodal and Trimodal distributions, the hyperexponential (H_2) and three-phase hyperexponential (H_3), and the Bounded Pareto and Pareto job size distributions.

	Convergent LWL	Divergent LWL
Convergent SITA	BOX 1	BOX 2
Divergent SITA	BOX 3	BOX 4

TABLE I
ALL FOUR BEHAVIORS ARE COMMON.

But how can SITA be bad for high variability workloads, when it is specifically designed for those workloads? There are two things that can go wrong under SITA (for simplicity we assume just 2 servers and one cutoff differentiating short and long jobs):

- **Observation 1**: The stringent segregation of shorts and longs mandated by SITA can lead to underutilization of the servers under any job size distribution. Specifically,

there are times when the short job queue has multiple jobs and the long job server is idle. The reverse situation also occurs, although less frequently since short jobs arrive with higher frequency. By contrast LWL (or equivalently the M/GI/n queue) does not suffer from underutilized servers.

- **Observation 2**: Some job size distributions may inherently prevent the creation of two sub-distributions both with finite variance, meaning that one of the two SITA queues has infinite variance.

To see an illustration of how SITA fails, we consider an example from Box 3 in Table I, which is shown in Figure 3. Here the job size distribution is the Bounded Pareto distribution with parameter of $\alpha = 1.6$ and $\rho = 0.95$. More information about the Bounded Pareto and how we make $C^2 \to \infty$ while holding $\mathbf{E}[X]$ fixed is provided in Section IV-B. The important point to note is that as $C^2 \to \infty$, the upper limit on the Bounded Pareto also increases to infinity, meaning that the Bounded Pareto becomes a Pareto distribution. The Pareto and Bounded Pareto distributions are known to well-model empirical job size distributions for a wide variety of computing applications [2], [7], [21], [13], [20].

Figure 3 compares the mean response time under SITA to an upper bound on LWL as $C^2 \to \infty$ while holding $\mathbf{E}[X]$ fixed. For lower C^2 SITA improves upon LWL, however, there is a cross-over point, at sufficiently high C^2, after which SITA diverges, while LWL converges.[1] This cross-over point was not observed in prior work (which mostly relies on simulation, numerical methods, heavy-traffic approximations or M/GI/2 approximations). This is possibly because the prior literature didn't consider the very high C^2 regions, thus (incorrectly) concluding that SITA is always superior to LWL.

The results shown in Figure 3 are understandable in light of the above two observations. Firstly, no matter where the cutoff is placed in SITA, the long job server sees a Bounded Pareto distribution with C^2 approaching infinity (Observation 2 above). Hence the mean delay at the second server goes to infinity, even when multiplied by the fraction of long jobs. It may seem that LWL should diverge as well, since it too should suffer from the infinite C^2. However by *Observation 1* above, we see that LWL has a second server to help alleviate the situation where one job gets stuck behind another, while SITA does not always have this flexibility, if the two jobs are on the same side of the cutoff. Thus LWL's delay can be finite (if $\rho < 1$) even if $C^2 \to \infty$. See [16] for a formalization of the above argument.

D. Best of both worlds? The CS policy

While Figure 3 shows that SITA can sometimes perform poorly, one may wonder whether the issue is the particular definition of SITA, rather than the general heuristic of

[1]This cross-over point is actually lower than it appears, because the LWL curve is an upper bound. Also, in [16], examples are given with much lower cross-over points.

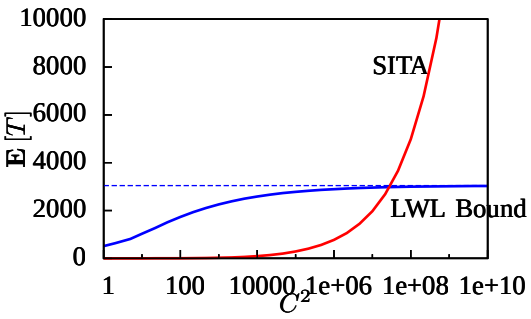


Fig. 3. *Mean response time under SITA vs. LWL for Bounded Pareto($\alpha = 1.6$) job size distribution [16]. SITA's mean response time diverges while that of LWL converges.*

providing isolation for short jobs from long ones. Ideally, one would like to still provide isolation for short jobs, but do it in a way that achieves the good utilization of LWL.

Towards this end, we introduce the Cycle-Stealing (CS) policy, depicted in Figure 4. To keep things simple, we assume that there are only two servers. We define a size cutoff ψ. Jobs of size $< \psi$ are referred to as "short" jobs, with subscript S, and jobs of size $> \psi$ are referred to as "long" jobs, with subscript L.

Under *CS*, there is one central queue only. When server 1 becomes free, it takes the *short* job closest to the head of the central queue; if there is no short job, it stays idle. When server 2 becomes free, it takes the job at the head of the central queue, regardless of whether that is a short or long job. This is in contrast to *SITA* where jobs are immediately dispatched upon arrival.

Importantly, for CS (and for SITA), the ψ cutoff is assumed to be chosen optimally, so as to minimize mean response time. The optimal ψ for CS is not typically the same as that for SITA. Also, for both policies, jobs of size exactly ψ are categorized probabilistically into short versus long.

Observe that the CS policy is designed to have (almost) all the flexibility of LWL while still providing isolation for shorts. The CS policy is discussed in more detail in [14], [15].

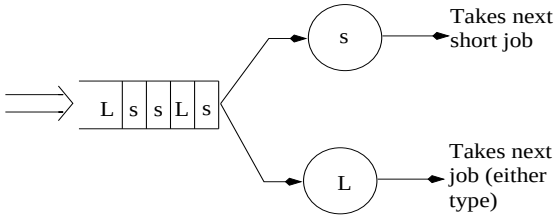


Fig. 4. *Cycle stealing task assignment.*

E. Results and Impact

Definition 1: A policy, $\mathcal{P}$, above is said to *diverge* if, for all ψ, the mean delay under $\mathcal{P}(\psi)$ goes to infinity as $C^2 \to \infty$ while $\mathbf{E}[X]$ is held fixed.

This paper proves that, surprisingly, the CS policy diverges whenever the SITA policy diverges. Thus, for any box in Table I where SITA diverges, CS does too, meaning that LWL can outperform CS under high variability.

This reaffirms the message that providing isolation for short jobs under high job-size variability may not always be the best strategy.

II. High-Level Plan

Our overall goal is to prove that:

"Whenever SITA diverges, CS also diverges"

We introduce a bit of notation: For any policy, $\mathcal{P}$, let $D^{\mathcal{P}}$ denote the delay under policy $\mathcal{P}$. The delay of a job is its response time minus its service requirement. We will sometimes write $\mathcal{P}(\psi)$ to denote policy $\mathcal{P}$ with cutoff ψ. We use p_S to denote the fraction of short jobs, relative to ψ, and p_L to denote the fraction of long jobs. We use X_S (respectively, X_L) to denote job size of short jobs (respectively long ones). Likewise, we use ρ_S to denote the load made up of short jobs, where $\rho_S = \lambda p_S \mathbf{E}[X_S]$, and likewise for long jobs.

The remainder of the paper is devoted to the proof of the following theorem, where D denotes delay:

Theorem 1: If $\mathbf{E}[D]^{SITA(\psi)} \to \infty$ for all ψ, as $C^2 \to \infty$, then $\mathbf{E}[D]^{CS(\psi)} \to \infty$ for all ψ, as $C^2 \to \infty$.

Proof:

We will use the fact that

$$\begin{aligned} \mathbf{E}[D]^{SITA(\psi)} &= p_S(\psi)\mathbf{E}[D_S]^{SITA(\psi)} \\ &\quad + p_L(\psi)\mathbf{E}[D_L]^{SITA(\psi)} \end{aligned} \tag{1}$$

Observe that whenever $C^2 \to \infty$, it must be the case that $\mathbf{E}[X_L^2] \to \infty$ (assuming that ψ is finite), which implies that $\mathbf{E}[D_L]^{SITA(\psi)} \to \infty$. But that by itself does not imply $\mathbf{E}[D]^{SITA(\psi)} \to \infty$ because there's still the $p_L(\psi)$ term in (1).

The entire first term in (1) is bounded for every finite ψ, $\forall C^2$, provided that $\rho_S(\psi) < 1$. Thus $\mathbf{E}[D]^{SITA(\psi)}$ is finite if and only if $p_L(\psi)\mathbf{E}[D_L]^{SITA(\psi)}$ is finite *and* $\rho_S(\psi) < 1$. Thus, for any given ψ, there are two possible reasons why $\mathbf{E}[D]^{SITA(\psi)} \to \infty$:

1) $p_L\mathbf{E}[D_L]^{SITA(\psi)} \to \infty$
2) $\rho_S(\psi) > 1$[2]

For a given ψ, if either of the above properties holds, then SITA's delay will be infinite for that cutoff ψ. If neither of these is true, then SITA is convergent (since it converges on at least that ψ and maybe others).

We are given that $\mathbf{E}[D]^{SITA(\psi')}$ goes to infinity for all ψ'. We now consider a given ψ. By definition, our given ψ either satisfies property 1 above or property 2 above (or both). Lemma 1 shows that if our given ψ satisfies property 1 above, then $\mathbf{E}[D]^{CS(\psi)}$ goes to infinity. Likewise, Lemma 2 shows that if our given ψ satisfies property 2 above and $\mathbf{E}[D]^{SITA(\psi')} \to \infty$ $\forall \psi'$, then $\mathbf{E}[D]^{CS(\psi)}$ goes to infinity. Since SITA diverges, every ψ must satisfy either property 1 or property 2, and thus, $\mathbf{E}[D]^{CS(\psi)}$ goes to infinity for all ψ. ■

[2]We are excluding the case $\rho_S(\psi) = 1$.

III. Analysis of CS

A. Case 1

Lemma 1: For any given ψ, if $p_L\mathbf{E}[D_L]^{SITA(\psi)} \to \infty$, then $\mathbf{E}[D]^{CS(\psi)} \to \infty$.

Proof:

Since $p_L\mathbf{E}[D_L]^{SITA(\psi)} \to \infty$, it follows that $p_L\mathbf{E}[D_L]^{CS(\psi)} \to \infty$, since server 2 under CS sees the same long jobs as SITA, plus it additionally sees some short jobs.

Hence $\mathbf{E}[D]^{CS(\psi)} \to \infty$. ■

B. Case 2

Lemma 2: For any given ψ, if $\rho_S(\psi) > 1$ and $\mathbf{E}[D]^{SITA(\psi')} \to \infty$ for all ψ', then $\mathbf{E}[D]^{CS(\psi)} \to \infty$.

Proof:

Consider a tagged short arrival. Under CS, the tagged arrival looks at server 2 and, by PASTA, with probability ρ_L, it sees a long job there. Suppose the age of that job is x. This means that, looking backwards in time, server 2 has been busy for at least x units of time. This implies that server 2 has not completed a small job for the past $\geq x$ time units. Since server 2 has been busy for the past x time units, we can argue that, with $1-\delta$ probability, where $0 < \delta < 1$, a certain (large) amount of work has built up in the (central) queue, and correspondingly, that this translates to at least a certain (large) expected delay, D, for the short tagged arrival.

To make this formal, we will need to make use of a few lemmas, provided at the end of this section. Firstly, Lemma 3 deals with the average rate that work accumulates during the time that server 2 is blocked. We expect this rate of accumulation to be $\rho_S - 1$. Lemma 3 says that, for any $\epsilon > 0$ and $\delta > 0$, we can prove that the average work accumulation rate is at least $\rho_S - 1 - \epsilon$, with probability at least $1-\delta$, provided that server 2 is blocked for a long enough time, x_0, where x_0 is some function of δ and ϵ.

Lemma 4 below relates the work buildup seen by a tagged job to its expected delay.

We are now ready to consider the probability that the delay of a short job exceeds u, given that a long job is in residence at server 2. We will derive this assuming $u > u_0$, where u_0 will be specified later.

$$\begin{aligned} &\mathbf{P}\{\text{delay of short} > u \mid \text{arrival sees long at server 2 }\} \\ &\geq \mathbf{P}\left\{\begin{array}{l}\text{accum. short work } > 2u+\psi \mid \\ \text{arriv. sees long job at serv. 2}\end{array}\right\} \quad \text{by Lemma 4} \\ &\quad \text{Let } f(u) = \frac{2u+\psi}{\rho_S - 1 - \epsilon} \\ &\geq \mathbf{P}\{\text{age of long } \geq f(u)\} \\ &\quad \cdot\mathbf{P}\{\text{work accum. at rate } > \rho_S - 1 - \epsilon \text{ during } f(u)\} \\ &\quad \text{Now, in order to make this second term exceed } 1-\delta, \\ &\quad \text{we need } f(u) > x_0(\delta,\epsilon) \text{ from Lemma 3.} \end{aligned}$$

$$\frac{2u+\psi}{\rho_S-1-\epsilon} > x_0 \iff u > \frac{x_0(\rho_S-1-\epsilon)-\psi}{2} \equiv u_0$$

$$\begin{aligned}
&\geq \mathbf{P}\{X_{Le} \geq f(u)\} \cdot (1-\delta), \qquad u > u_0 \quad \text{by Lemma 3} \\
&= \mathbf{P}\{X_{Le} \geq (2u+\psi)/(\rho_S-1-\epsilon)\} \cdot (1-\delta) \\
&= \mathbf{P}\left\{\frac{\rho_S-1-\epsilon}{2} X_{Le} - \frac{\psi}{2} > u\right\} \cdot (1-\delta) \\
&= \mathbf{P}\left\{cX_{Le} - \frac{\psi}{2} > u\right\} \cdot (1-\delta) \quad \text{where } c = \frac{\rho_S-1-\epsilon}{2} \\
&= \mathbf{P}\{Y > u\} \cdot (1-\delta) \qquad \text{(assuming } u > u_0 \text{)}
\end{aligned}$$

where we define $Y = cX_{Le} - \frac{\psi}{2}$.

At this point, we have seen that:

$$\begin{aligned}
&\mathbf{P}\{\text{delay of short } > u \mid \text{arrival sees long at server 2}\} \\
&\quad \geq \begin{cases} \mathbf{P}\{Y > u\}(1-\delta) & \text{if} \quad u > u_0 \\ 0 & \text{if} \quad u \leq u_0 \end{cases}
\end{aligned}$$

Then, integrating both sides of the above with respect to u, we have that:

$$\begin{aligned}
&\mathbf{E}\left[\text{Delay of short} \mid \text{long in service}\right] \\
&\geq \int_0^{u_0} 0du + \int_{u_0}^{\infty} \mathbf{P}\{Y > u\}(1-\delta)du \\
&= (1-\delta)\mathbf{E}[Y] - (1-\delta)\int_0^{u_0} \mathbf{P}\{Y > u\}\, du \\
&\geq (1-\delta)\left(\mathbf{E}[Y] - u_0\right) \\
&= (1-\delta)\left(c\mathbf{E}[X_{Le}] - \frac{\psi}{2} - u_0\right)
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathbf{E}\left[\text{Delay of short}\right] &\geq \rho_L(1-\delta)\left(c\mathbf{E}[X_{Le}] - \frac{\psi}{2} - u_0\right) \\
&= \text{linear in } \rho_L \mathbf{E}[X_{Le}]
\end{aligned}$$

All that's left is to show that

$$\rho_L \mathbf{E}[X_{Le}] \to \infty \text{ as } C^2 \to \infty$$

This will imply that the expected delay of the short job arrival is infinite under CS, and we are done.

The fact that

$$\rho_L \mathbf{E}[X_{Le}] \to \infty \text{ as } C^2 \to \infty$$

is proven formally in Lemma 5 at the end of the section, which states that the above equation must be true, otherwise there would be a cutoff under which SITA converges.

■

Lemma 3: Given short jobs with mean size $\mathbf{E}[X_S]$ contributing load $\rho_S > 1$ and $0 < \delta, \epsilon < 1$, there exists a finite $x_0(\delta, \epsilon)$, given by (4) such that the probability that work accumulates at an average rate exceeding $\rho_S - 1 - \epsilon$ during the time x server 2 is blocked exceeds $1-\delta$, $\forall x > x_0$.

Proof: Clearly, the amount of work *completing* during time x is no more than x. It thus suffices to prove that the work arriving during time x is at least $(\rho_S - \epsilon)x$.

Let $N(x)$ be a random variable for the number of Poisson arrivals during x, and let $X_S(i)$ be a random variable for the size of the ith short job, $1 \leq i \leq N(x)$.

$$\begin{aligned}
&\mathbf{P}\{\text{Work arriving during } x \geq (\rho_S - \epsilon)x\} \\
&= \mathbf{P}\left\{\sum_{i=1}^{N(x)} X_S(i) \geq (\rho_S - \epsilon)x\right\} \\
&= \mathbf{P}\left\{\frac{1}{x}\sum_{i=1}^{N(x)} X_S(i) \geq \rho_S - \epsilon\right\} \\
&= \mathbf{P}\left\{\frac{N(x)}{x} \cdot \frac{1}{N(x)} \sum_{i=1}^{N(x)} X_S(i) \geq \rho_S - \epsilon\right\}
\end{aligned}$$

We now condition on $\frac{N(x)}{x}$, which is the average arrival rate during time x. We define

$$\lambda^- \equiv \frac{\rho_S - \epsilon}{\mathbf{E}[X_S] - \frac{\epsilon}{2\lambda}} \tag{2}$$

Observe that $\lambda^- < \lambda$, but that $\lambda^- \to \lambda$ as $\epsilon \to 0$. We will condition on E_{λ^-}, defined as the event that $\frac{N(x)}{x} > \lambda^-$. Define $P_{\lambda^-} = \mathbf{P}\{E_{\lambda^-}\}$.

$$\begin{aligned}
&\mathbf{P}\{\text{Work arriving during } x \geq (\rho_S - \epsilon)x\} \\
&\geq \mathbf{P}\left\{\frac{N(x)}{x} \cdot \frac{1}{N(x)} \sum_{i=1}^{N(x)} X_S(i) \geq \rho_S - \epsilon \mid E_{\lambda^-}\right\} \cdot P_{\lambda^-}
\end{aligned}$$

$$\begin{aligned}
&\mathbf{P}\{\text{Work arriving during } x \geq (\rho_S - \epsilon)x\} \\
&\geq \mathbf{P}\left\{\lambda^- \cdot \frac{1}{N(x)} \sum_{i=1}^{N(x)} X_S(i) \geq \rho_S - \epsilon\right\} \cdot P_{\lambda^-} \\
&= \mathbf{P}\left\{\frac{1}{N(x)} \sum_{i=1}^{N(x)} X_S(i) \geq \mathbf{E}[X_S] - \frac{\epsilon}{2\lambda}\right\} \cdot P_{\lambda^-} \\
&\geq \left(1 - \frac{(2\lambda)^2}{\epsilon^2} \cdot \frac{1}{N(x)} \cdot \sigma^2_{X_S}\right) \cdot P_{\lambda^-} \qquad \text{by Eqn (7)}
\end{aligned}$$

But we earlier conditioned on $N(x) > x\lambda^-$, and hence $\frac{1}{N(x)} < \frac{1}{x\lambda^-}$. Thus,

$$\begin{aligned}
&\mathbf{P}\{\text{Work arriving during } x \geq (\rho_S - \epsilon)x\} \\
&\geq \left(1 - \frac{(2\lambda)^2}{\epsilon^2} \cdot \frac{1}{x} \cdot \frac{1}{\lambda^-} \cdot \sigma^2_{X_S}\right) \cdot P_{\lambda^-}
\end{aligned}$$

The goal will be to provide an x_0 such that for all $x > x_0$, the above probability exceeds the given $1-\delta$. Before we can do this, it is useful to bound P_{λ^-}, so that we can quantify its dependence on x.

$$
\begin{aligned}
P_{\lambda^-} &= \mathbf{P}\left\{\frac{N(x)}{x} > \lambda^-\right\} \\
&= \mathbf{P}\left\{\frac{N(x)}{x} > \frac{\rho_S - \epsilon}{\mathbf{E}[X_S] - \frac{\epsilon}{2\lambda}}\right\} \\
&= \mathbf{P}\left\{\frac{N(x)}{x} > \lambda - \left(\frac{\epsilon\lambda}{2\rho_S - \epsilon}\right)\right\}
\end{aligned}
$$

Now observe that $N(x) \sim$ Poisson with mean λx. Assuming that x is an integer, we can view $N(x)$ as a sum of x Poisson random variables $(N_1 + N_2 + \cdots + N_x)$ each with mean λ and variance λ.

Then

$$
\begin{aligned}
P_{\lambda^-} &\geq \mathbf{P}\left\{\frac{1}{x}\sum_{i=1}^{x} N_i > \lambda - \left(\frac{\epsilon\lambda}{2\rho_S - \epsilon}\right)\right\} \\
&\geq \mathbf{P}\left\{\frac{1}{x}\sum_{i=1}^{x} N_i - \lambda > -\left(\frac{\epsilon\lambda}{2\rho_S - \epsilon}\right)\right\} \\
&\geq 1 - \left(\frac{2\rho_S - \epsilon}{\epsilon\lambda}\right)^2 \cdot \frac{\lambda}{x} \quad \text{by Eqn (7)}
\end{aligned}
$$

Substituting in the above value of P_{λ^-}, we have that:

$$
\begin{aligned}
&\mathbf{P}\{\text{Work arriving during } x \geq (\rho_S - \epsilon)\, x\} \\
&\geq \left(1 - \frac{(2\lambda)^2}{\epsilon^2} \cdot \frac{\sigma^2_{X_S}}{x\lambda^-}\right) \cdot \left(1 - \left(\frac{2\rho_S - \epsilon}{\epsilon\lambda}\right)^2 \cdot \frac{\lambda}{x}\right) \quad (3)
\end{aligned}
$$

We now want to determine x_0 such that for all $x > x_0$, the above probability in (3) exceeds $1 - \delta$. Setting (3) $\geq 1 - \delta$ and simplifying gives:

$$
\begin{aligned}
0 \;\leq\; & \epsilon^2\delta x^2 - \left[2\lambda \cdot \frac{2\rho_S - \epsilon}{\rho_S - \epsilon} \cdot \sigma^2_{X_S} + \frac{(2\rho_S - \epsilon)^2}{\lambda}\right] x \\
& + \left[\frac{2}{\epsilon^2} \cdot \frac{(2\rho_S - \epsilon)^3}{\rho_S - \epsilon} \cdot \sigma^2_{X_S}\right]
\end{aligned}
$$

Since the rightmost term is strictly positive, we can ignore it. This yields:

$$
x_0 = \left\lceil \frac{\left[2\lambda \cdot \frac{2\rho_S - \epsilon}{\rho_S - \epsilon} \cdot \sigma^2_{X_S} + \frac{(2\rho_S - \epsilon)^2}{\lambda}\right]}{\epsilon^2\delta} \right\rceil \quad (4)
$$

■

Lemma 4: For a tagged short job, with probability 1, delay $> u$, in a 2-server CS system with cutoff ψ, if the short job sees at least $2u + \psi$ buildup of accumulated short work.

Proof: We assume that the accumulated work W can go to either server. The maximum amount of work that could be present when a server frees is ψ. Thus both servers would be busy for at least $\frac{W-\psi}{2}$ time. So the delay of a short job which sees W work is at least $\frac{W-\psi}{2}$. Now substitute in $W = 2u+\psi$ and we're done. ■

Lemma 5: If $\mathbf{E}[D]^{SITA(\psi')} = \infty$, $\forall\psi'$, then, for any ψ, $\rho_L(\psi)\mathbf{E}[X_{Le}(\psi)] \to \infty$ as $C^2 \to \infty$

Proof: Suppose, by contradiction, that $\rho_L\mathbf{E}[X_{Le}] \to \alpha < \infty$ under our cutoff ψ. That implies that $p_L\mathbf{E}[X_L^2]$ is also finite for cutoff ψ. If $\rho_S(\psi) < 1$, then $SITA(\psi)$ will converge because both the short and long components of SITA's delay converge. Therefore we assume that $\rho_S(\psi) \geq 1$. Now let's say that CS(ψ) sends $p_L(\psi)$ fraction of jobs (the long ones) to server 2, as well as $r_S > 0$ load of small jobs to server 2 to relieve the overload at server 1.

Consider now a "new" cutoff for SITA, called $\psi' < \psi$, which also sends r_S load of short jobs (the longest short jobs) to the long server (randomizing if necessary).[3]

Assume that this r_S load of small jobs corresponds to f_S fraction of small jobs. Let X_f be a random variable drawn from the job size distribution of those short jobs (those of size between ψ' and ψ) that end up serving at server 2 under cutoff ψ'.

Then, the second moment of the job sizes at server 2 under $SITA(\psi')$ is computed as follows:

$$
\begin{aligned}
\mathbf{E}[X_2^2] &= \frac{f_S}{f_S + p_L}\mathbf{E}[X_f^2] + \frac{p_L}{f_S + p_L}\mathbf{E}[X_L^2] \\
&\leq 1 \cdot \psi^2 + \frac{p_L}{f_S + p_L} \cdot \mathbf{E}[X_L^2] \\
&= \psi^2 + \frac{1}{\lambda\mathbf{E}[X_L]} \cdot \lambda \cdot \mathbf{E}[X_L] \cdot \frac{p_L}{f_S + p_L} \cdot \mathbf{E}[X_L^2] \\
&= \psi^2 + \frac{1}{\lambda\mathbf{E}[X_L]}\rho_L\mathbf{E}[X_L^2] \cdot \frac{1}{f_S + p_L} \\
&= \psi^2 + \frac{1}{\lambda 2\mathbf{E}[X_L]}\rho_L\mathbf{E}[X_L^2] \cdot \frac{2}{f_S + p_L} \\
&= \psi^2 + \frac{1}{\lambda}\rho_L\mathbf{E}[X_{Le}] \cdot \frac{2}{f_S + p_L} \\
&= \psi^2 + \frac{2\alpha}{\lambda(f_S + p_L)} \quad \text{by assumption} \quad (5) \\
&< \infty
\end{aligned}
$$

Under ψ' both servers would have load < 1 and the second server would have finite mean delay, as the second moment at server 2 is bounded by Eqn (5), and the mean delay at the first server is obviously finite. But this is in contradiction to the assumption that mean delay under SITA is infinite for all cutoffs. ■

IV. Auxiliary Lemmas and Background

A. WLLN

This section recalls the proof of the Weak Law of Large Numbers (WLLN) because we'll need an equation from here:

Theorem 2 (WLLN): Let X_1, X_2, X_3, …, be i.i.d. with finite mean $\mathbf{E}[X]$ and finite variance σ^2. Then

$$
\lim_{n\to\infty} \mathbf{P}\left\{|\frac{S_n}{n} - \mathbf{E}[X]| \geq \epsilon\right\} = 0
$$

Proof:

Markov's Inequality tells us that, if X is non-negative then:

[3] If conceivably $\rho_S(\psi)^{CS} = 1$, then we would choose ψ' to send $f_S + \epsilon$ short jobs to the long queue, ensuring $\rho_S(\psi')^{SITA} < 1$ and $\rho_L(\psi')^{SITA} < 1$.

$$\mathbf{P}\{X > t\} \leq \frac{\mathbf{E}[X]}{t}, \forall t \geq 0$$

This can be used to prove Chebyshev's Inequality which says that if Y is a random variable with finite mean $\mathbf{E}[Y]$ and finite variance σ_Y^2. Then,

$$\mathbf{P}\{|Y - \mathbf{E}[Y]| \geq t\} \leq \frac{\sigma_Y^2}{t^2}$$

Using the above, let

$$Y_n = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad \sigma_{Y_n}^2 = \frac{1}{n}\sigma_{X_1}^2$$

Thus

$$\mathbf{P}\{|Y_n - \mathbf{E}[Y]| \geq t\} \quad \leq \quad \frac{1}{t^2}\frac{1}{n}\sigma_{X_1}^2 \tag{6}$$

Letting $n \to \infty$ we obtain the Weak Law of Large Numbers.

■

Note: (6) also implies the following useful fact:

$$\mathbf{P}\{Y_n - \mathbf{E}[Y] \geq -t\} \quad \geq \quad 1 - \frac{1}{t^2}\frac{1}{n}\sigma_{X_1}^2 \quad \forall t > 0 \tag{7}$$

B. Background on Pareto and Bounded Pareto

The Bounded Pareto(k, p, α) distribution, where $0 < \alpha < 2$ and $0 < k < p$, has the following density function:

$$f(x) = \begin{cases} \frac{\alpha k^\alpha}{1-\left(\frac{k}{p}\right)^\alpha} x^{-\alpha-1} & k \leq x \leq p \\ 0 & \text{otherwise} \end{cases}$$

As $p \to \infty$, the Bounded Pareto distribution converges to the Pareto with density function:

$$f(x) = \alpha k^\alpha x^{-1-\alpha} \qquad x \geq k > 0$$

For $1 < \alpha < 2$, the Pareto distribution has finite mean, but infinite variance.

The following two Lemmas from [16] describe what happens to the Bounded Pareto when we increase C^2 while holding $\mathbf{E}[X]$ fixed:

Lemma 6: For any $\mathbf{E}[X]$, C^2, and $\alpha > 1$, we can specify a Bounded Pareto(k, p, α).

Lemma 7: Keeping $\mathbf{E}[X]$ constant, as $C^2 \to \infty$, for the Bounded Pareto distribution, $p \to \infty$ and $k \to \frac{\alpha-1}{\alpha}\mathbf{E}[X]$ (from above for $\alpha > 1$).

REFERENCES

[1] Eitan Bachmat and Hagit Sarfati. Analysis of size interval task assigment policies. *Performance Evaluation Review*, 36(2), 2008.

[2] Paul Barford and Mark Crovella. Generating representative Web workloads for network and server performance evaluation. In *Proceedings of the ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems*, pages 151–160, July 1998.

[3] James Broberg, Zahir Tari, and Panlop Zeephongsekul. Task assignment with work-conserving migration. *Parallel Computing*, 32:808–830, 2006.

[4] John Buzacott and George Shanthikumar. *Stochastic Models in Manufacturing Systems*. Prentice Hall, 1993.

[5] Valeria Cardellini, Emiliano Casalicchio, Michele Colajanni, and Philip Yu. The state of the art in locally distributed web-server systems. Technical report, 2001.

[6] Gianfranco Ciardo, Alma Riska, and Evgenia Smirni. Equiload: a load balancing policy for clustered web servers. *Performance Evaluation*, 46:101–124, 2001.

[7] Mark Crovella and Azer Bestavros. Self-similarity in World Wide Web traffic: Evidence and possible causes. pages 160–169, May 1996.

[8] Muhammad El-Taha and Bacel Maddah. Allocation of service time in a multiserver system. *Management Science*, 52(4):623–637, 2006.

[9] Hanhua Feng, Vishal Misra, and Dan Rubenstein. Optimal state-free, size-aware dispatching for heterogeneous M/G-type systems. *Performance Evaluation*, 62:475–492, 2005.

[10] Bin Fu, James Broberg, and Zahir Tari. Task assignment strategy for overloaded systems. In *Proceedings of the Eighth IEEE International Symposium on Computers and Communications*, 2003.

[11] Mor Harchol-Balter. Task assignment with unknown duration. *Journal of the ACM*, 49(2):260–288, March 2002.

[12] Mor Harchol-Balter, Mark Crovella, and Cristina Murta. On choosing a task assignment policy for a distributed server system. *IEEE Journal of Parallel and Distributed Computing*, 59:204–228, 1999.

[13] Mor Harchol-Balter and Allen Downey. Exploiting process lifetime distributions for dynamic load balancing. In *Proceedings of ACM SIGMETRICS*, pages 13–24, Philadelphia, PA, May 1996. Best Paper Award for Integrating Systems and Theory.

[14] Mor Harchol-Balter, Cuihong Li, Takayuki Osogami, Alan Scheller-Wolf, and Mark Squillante. Cycle stealing under immediate dispatch task assignment. In *15th ACM Symposium on Parallel Algorithms and Architectures*, pages 274–285, San Diego, CA, June 2003.

[15] Mor Harchol-Balter, Cuihong Li, Takayuki Osogami, Alan Scheller-Wolf, and Mark Squillante. Task assignment with cycle stealing under central queue. In *23rd International Conference on Distributed Computing Systems*, pages 628–637, Providence, RI, May 2003.

[16] Mor Harchol-Balter, Alan Scheller-Wolf, and Andrew Young. Surprising results on task assignment in server farms with high-variability workloads. In *ACM Sigmetrics 2009 Conference on Measurement and Modeling of Computer Systems*, 2009.

[17] Wallace Hopp and Mark Spearman. *Factory Physics*. McGraw Hill/Irwin, 2 edition, 2000.

[18] Steven Hotovy, David Schneider, and Timothy O'Donnell. Analysis of the early workload on the Cornell Theory Center IBM SP2. 1996.

[19] Kazumasa Oida and Kazumasa Shinjo. Characteristics of deterministic optimal routing for a simple traffic control problem. In *Performance, Computing and Communications Conference, IPCCC*, February 1999.

[20] Bianca Schroeder and Mor Harchol-Balter. Evaluation of task assignment policies for supercomputing servers: The case for load unbalancing and fairness. *Cluster Computing: The journal of Networks, Software Tools, and Applications*, 7(2):151–161, April 2004.

[21] Anees Shaikh, Jennifer Rexford, and Kang Shin. Load-sensitive routing of long-lived IP flows. In *Proceedings of ACM SIGCOMM*, September 1999.

[22] Zahir Tari, James Broberg, Albert Zomaya, and Roberto Baldoni. A least flow-time first load sharing approach for distributed server farm. *Journal of Parallel and Distributed Computing*, 65:832–842, 2005.

[23] Nigel Thomas. Comparing job allocation schemes where service demand is unknown. *Journal of Computer and System Sciences*, 74:1067–1081, 2008.