SRPT for Multiserver Systems

Isaac Grosof* Ziv Scully* Mor Harchol-Balter*

1. INTRODUCTION

The Shortest Remaining Processing Time (SRPT) scheduling policy and variants thereof have been deployed in many computer systems, including web servers [4], networks [8], databases [3] and operating systems [1]. SRPT has also long been a topic of fascination for queueing theorists due to its optimality properties. In 1966, the mean response time for SRPT was first derived [10], and in 1968 SRPT was shown to minimize mean response time in both a stochastic sense and a worst-case sense [9]. However, these beautiful optimality results and the analysis of SRPT are only known for *single-server* systems. Almost nothing is known about SRPT in *multiserver* systems, such as the M/G/k, even for the case of just $k = 2$ servers.

The SRPT policy for the M/G/k is defined as follows: at all times, the k jobs with smallest remaining processing time receive service, preempting jobs in service if necessary. We assume a central queue, meaning any job can be dispatched or migrated to any server at any time, and a preempt-resume model, meaning preemption incurs no cost or loss of work.

It seems believable that SRPT should minimize mean response time in multiserver systems because it gives priority to the jobs which will finish soonest, which seems like it should minimize the number of jobs in the system. However, it was shown in 1997 that SRPT is not optimal for multiserver systems in the worst case [5, 6]. That is, one can come up with an adversarial arrival sequence for which the mean response time under SRPT is larger than the optimal mean response time. In fact, the ratio by which SRPT's mean response time exceeds the optimal mean response time can be arbitrarily large [5, 6].

The fact that multiserver SRPT is not optimal in the worst case provokes a natural question about the *stochastic* case.

> *Is SRPT optimal or near-optimal for minimizing mean response time in the the M/G/k?*

Unfortunately, this question is entirely open. Not only is it not known whether SRPT is optimal, but multiserver SRPT has also eluded stochastic analysis.

> *What is the mean response time for the M/G/k under SRPT?*

The purpose of this paper is to answer both of these questions in the high-load setting. Under low load, response time is dominated by service time, which is not affected by the scheduling policy. In contrast, under high load, response time is dominated by queueing time, which can vary wildly under different scheduling policies. We thus focus on the high-load setting, and specifically on the limiting behavior as load approaches capacity.

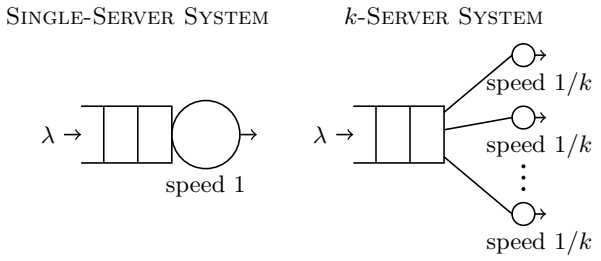


Figure 1.1: Single-server and k-server systems

Our main result is that, under mild assumptions on the service requirement distribution,

> *SRPT is an optimal multiserver policy for minimizing mean response time in the M/G/k in the limit as load approaches capacity.*

We also give the *first mean response time bound for the M/G/k under SRPT*. The bound is valid for all loads and is tight for load near capacity.

The technique by which we bound response time under SRPT-k is widely generalizable. It can also be applied to PSJF-k, RS-k, and FB-k (See [2]).

Our approach to analyzing SRPT on k servers is to compare its performance to that of SRPT on a single server which is k times as fast, where both systems have the same arrival rate λ and service requirement distribution S. Specifically, let *SRPT-k* be the policy which uses multiserver SRPT on k servers of speed $1/k$, as shown in Figure 1.1. We compare SRPT-k with SRPT-1, which is ordinary single-server SRPT. The *system load* $\rho = \lambda \mathbf{E}[S]$ is the average rate at which work enters the system. The maximal total rate at which the k servers can do work is 1, so the system is stable for $\rho < 1$, which we assume throughout.

Our main result is that in the $\rho \to 1$ limit, the mean response time under SRPT-k, $\mathbf{E}\left[T^{\text{SRPT-}k}\right]$, approaches the mean response time under SRPT-1, $\mathbf{E}\left[T^{\text{SRPT-1}}\right]$. Because SRPT-1 minimizes response time among all scheduling policies, this means that SRPT-k is asymptotically optimal among k-server policies. In particular, let OPT-k be the optimal k-server policy. Then

$$\mathbf{E}\left[T^{\text{SRPT-1}}\right] \le \mathbf{E}\left[T^{\text{OPT-}k}\right] \le \mathbf{E}\left[T^{\text{SRPT-}k}\right],$$

so showing that $\mathbf{E}\left[T^{\text{SRPT-}k}\right] \to \mathbf{E}\left[T^{\text{SRPT-1}}\right]$ as $\rho \to 1$ also shows that $\mathbf{E}\left[T^{\text{SRPT-}k}\right] \to \mathbf{E}\left[T^{\text{OPT-}k}\right]$ as $\rho \to 1$.

Our approach is inspired by two very different worlds: the stochastic world and the adversarial worst-case world. Purely stochastic approaches are difficult to generalize to the M/G/k for many reasons, including the fact that multiserver

*Carnegie Mellon University, Computer Science Department, 5000 Forbes Ave, Pittsburgh, PA 15213, USA, {igrosof, zscully, harchol}@cs.cmu.edu

systems are not work conserving. Purely adversarial worst-case analysis is easier but leads to weak bounds when directly applied to the stochastic setting.

What makes our analysis work is a strategic combination of the stochastic and worst-case techniques. We use the more powerful stochastic tools where possible and use worst-case techniques to bound variables for which exact stochastic analysis is intractable.

2. ANALYSIS OF SRPT-K

Consider a tagged job j of size x. We will call a job ℓ *relevant* to j if ℓ has smaller remaining size than j. Otherwise, we call ℓ irrelevant.

Traditional tagged job analysis cannot be applied to SRPT-k because SRPT-k is not work conserving. Our approach is to find a way to make SRPT-k appear work-conserving while the tagged job j is in the system. We do this by introducing the new concept of *virtual work*. Virtual work encapsulates all of the time that the servers spend either idle or working on irrelevant jobs while j is in the system. By thinking of these times as "virtual work", the system appears to be work-conserving while j is in the system, allowing us to bound the response time of j.

We will bound j's response time by bounding the *total amount of server activity* between j's arrival and departure. Between j's arrival and departure, each server can be doing one of four categories of work.

- *Tagged work:* serving j.
- *Old work:* serving a job which is relevant to j that was in the system upon j's arrival.
- *New work:* serving a job which is relevant to j that arrived after j.
- *Virtual work:* either idling or serving an job which is irrelevant to j.

The response time of j is exactly the total of tagged, old, new, and virtual work. The main idea behind our analysis is to bound this total by a single (work-conserving) busy period.

Definition 2.1. A *relevant busy period* for a job of remaining size x started by (possibly random) amount of work V, written $B_{\leq x}(V)$, is the time until a system starting with V total work becomes empty, where only arrivals of size at most x are admitted and the system completes work at rate 1 throughout.

We can bound of each of the four categories of work.

- Tagged work is j's size x.
- Old work is equal to the amount of relevant work seen by j upon arrival. By the PASTA property [11], this is $\texttt{RelWork}_{\leq x}^{\text{SRPT-}k}$, the steady state amount of relevant work for a job of size x.
- New work is bounded by all jobs of size at most x that arrive during a relevant busy period $B_{\leq x}(\cdot)$ started by tagged, old, and virtual work.
- Virtual work is easily shown to be at most $(k-1)x$, because virtual work is only done while j is in service, since SRPT-k prioritizes j over irrelevant jobs.

Taken together, these yield a stochastic dominance bound,

$$T^{\text{SRPT-}k}(x) \leq_{\text{st}} B_{\leq x}\big(\texttt{RelWork}_{\leq x}^{\text{SRPT-}k} + kx\big). \tag{2.1}$$

Our next task is to bound $\texttt{RelWork}_{\leq x}^{\text{SRPT-}k}$, the steady state amount of relevant work for a job of size x under SRPT-k. A purely stochastic analysis of relevant work would be very difficult. We therefore take the following hybrid approach. We consider a pair of systems which experience the same arrival sequence:

- *System* 1, which uses SRPT-1; and
- *System* k, which uses SRPT-k.

We compare the amounts of relevant work in each system, giving a *worst-case bound for the difference*. This allows us to use the previously known *stochastic analysis* of $\texttt{RelWork}_{\leq x}^{\text{SRPT-1}}$ to give a stochastic bound for $\texttt{RelWork}_{\leq x}^{\text{SRPT-}k}$.

For any time t, let $\texttt{RelWork}_{\leq x}^{(1)}(t)$ be the amount of relevant work in System 1 at t, and similarly for $\texttt{RelWork}_{\leq x}^{(k)}(t)$. Our goal is to give a worst-case bound for the difference in relevant work between Systems 1 and k,

$$\Delta_{\leq x}(t) = \texttt{RelWork}_{\leq x}^{(k)}(t) - \texttt{RelWork}_{\leq x}^{(1)}(t).$$

To bound $\Delta_{\leq x}(t)$, we split times t into

- *few-jobs intervals*, during which there are fewer than k relevant jobs at a time in System k; and
- *many-jobs intervals*, during which there are at least k relevant jobs at a time in System k.

A similar splitting was used by Leonardi and Raz [5, 6].

Lemma 2.2. *For any arrival sequence and at any time t, the difference between the relevant work in System* 1 *and the relevant work in System k is bounded by*

$$\Delta_{\leq x}(t) \leq kx.$$

Proof. If t is in a few-jobs interval, there are at most $k-1$ relevant jobs in System k, each of remaining size at most x, so $\Delta_{\leq x}(t) \leq \texttt{RelWork}_{\leq x}^{(k)}(t) \leq (k-1)x$.

If instead t is in a many-jobs interval, we argue as follows. During the many-jobs interval, $\Delta_{\leq x}(t)$ is nonincreasing. This is because System k completes relevant work at rate 1 during a many-jobs interval, which is at least as fast as System 1 completes relevant work. (Recall that the systems experience identical arrival sequences, so arrivals do not change $\Delta_{\leq x}(t)$.)

It thus suffices to bound $\Delta_{\leq x}(t)$ when t is the start of a many-jobs interval. It can be shown that System k has at most k relevant jobs at the start of a many-jobs interval, so $\Delta_{\leq x}(t) \leq \texttt{RelWork}_{\leq x}^{(k)}(t) \leq kx$ in this case, as desired. □

2.1 Response Time Bound

Recall that the *waiting time* of a job is the time between its arrival and its first instant of service. We write $W^{\text{SRPT-1}}(x)$ for the waiting time of a job of size x under SRPT-1.

Theorem 2.3. *In an M/G/k, the response time of a job of size x under SRPT-k is bounded by*

$$T^{\text{SRPT-}k}(x) \leq_{\text{st}} W^{\text{SRPT-1}}(x) + B_{\leq x}(2kx).$$

Proof. From (2.1), we know that

$$T^{\text{SRPT-}k}(x) \leq_{\text{st}} B_{\leq x}\big(\texttt{RelWork}_{\leq x}^{\text{SRPT-}k} + kx\big).$$

By plugging in Lemma 2.2, we find that

$$\begin{aligned} T^{\text{SRPT-}k}(x) &\leq_{\text{st}} B_{\leq x}\big(\texttt{RelWork}_{\leq x}^{\text{SRPT-1}} + 2kx\big) \\ &= B_{\leq x}\big(\texttt{RelWork}_{\leq x}^{\text{SRPT-1}}\big) + B_{\leq x}(2kx). \end{aligned}$$

To obtain the desired bound, we recall the waiting time in SRPT-1 [10],

$$W^{\text{SRPT-1}}(x) = B_{\leq x}\big(\texttt{RelWork}_{\leq x}^{\text{SRPT-1}}\big).$$ □

While Theorem 2.3 gives a good bound on the response time under SRPT-k, we can tighten the bound further.

Theorem 2.4. *In an M/G/k, the mean response time of a job of size x under SRPT-k is bounded by*

$$\mathbf{E}\left[T^{\text{SRPT-}k}(x)\right] \leq \frac{\int_0^x \lambda t^2 f_S(t)\,dt}{2(1-\rho_{\leq x})^2} + \frac{k\rho_{\leq x}x}{1-\rho_{\leq x}} + \int_0^x \frac{k}{1-\rho_{\leq t}}\,dt,$$

where $f_S(\cdot)$ is the probability density function of the service requirement distribution S, and $\rho_{\leq x} = \int_0^x \lambda t f_S(t)\,dt$ is the load due to jobs of size at most x.

Proof. See [2]. □

3. OPTIMALITY OF SRPT-K IN HEAVY TRAFFIC

Using Theorem 2.3, we now bound $\mathbf{E}\left[T^{\text{SRPT-}k}\right]$ in relation to $\mathbf{E}\left[T^{\text{SRPT-1}}\right]$.

Theorem 3.1. *In an M/G/k, the mean response time under SRPT-k is bounded by*

$$\mathbf{E}\left[T^{\text{SRPT-}k}\right] \leq \mathbf{E}\left[T^{\text{SRPT-1}}\right] + \frac{2k}{\lambda}\log\left(\frac{1}{1-\rho}\right)$$

Proof. Let $R(x) = \mathbf{E}[B_{\leq x}(x)]$. Taking expectations over Theorem 2.3, we find that

$$\mathbf{E}\left[T^{\text{SRPT-}k}\right] \leq \mathbf{E}\left[W^{\text{SRPT-1}}\right] + 2k\mathbf{E}[R(S)],$$

Waiting time is less than response time by definition, so

$$\mathbf{E}\left[W^{\text{SRPT-1}}\right] \leq \mathbf{E}\left[T^{\text{SRPT-1}}\right].$$

After straightforward calculus, we obtain

$$\mathbf{E}\left[R(S)\right] = \frac{1}{\lambda}\log\left(\frac{1}{1-\rho}\right),$$

implying the desired bound. □

Corollary 3.2. *In an M/G/k with service requirement distribution S which is either (i) bounded or (ii) unbounded with tail function of upper Matuszewska index less than -2,*

$$\lim_{\rho\to 1}\frac{\mathbf{E}\left[T^{\text{SRPT-}k}\right]}{\mathbf{E}\left[T^{\text{SRPT-1}}\right]} = 1.$$

Proof. Since $T^{\text{SRPT-1}}$ minimizes mean response time [9], it suffices to show that

$$\lim_{\rho\to 1}\frac{\mathbf{E}\left[T^{\text{SRPT-}k}\right]}{\mathbf{E}\left[T^{\text{SRPT-1}}\right]} \leq 1.$$

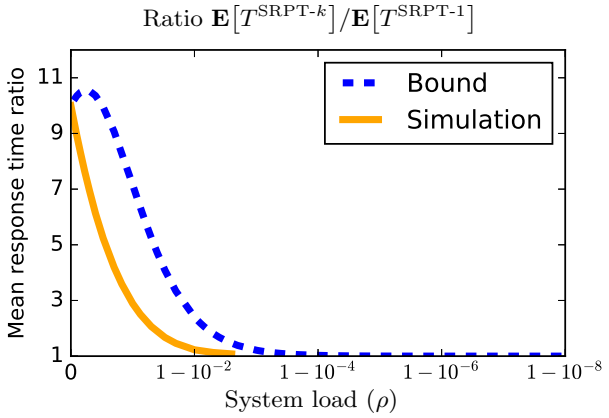
It follows immediately from the results of Lin et al. [7] that under the conditions on S assumed,

$$\lim_{\rho\to 1}\frac{\log\left(\frac{1}{1-\rho}\right)}{\mathbf{E}\left[T^{\text{SRPT-1}}\right]} = 0.$$

Applying Theorem 3.1, the desired limit follows. □

From Corollary 3.2 and the optimality of SRPT-1 [9], it also follows that SRPT-k is asymptotically optimal among all k-server policies.

As an illustration of the optimality of SRPT-k, we plot the ratio $\mathbf{E}\left[T^{\text{SRPT-}k}\right]/\mathbf{E}\left[T^{\text{SRPT-1}}\right]$ in Figure 3.1. The important feature to notice in Figure 3.1 is that as system load ρ approaches 1, both our analytic bound and the simulation converge to response time ratio 1.



The plot above shows the ratio $\mathbf{E}\left[T^{\text{SRPT-}k}\right]/\mathbf{E}\left[T^{\text{SRPT-1}}\right]$. Observe that as $\rho \to 1$, both our bound and the simulation converge to a ratio of 1. Our simulation of this ratio is the solid orange curve. Our analytic upper bound derived in Theorem 2.4 is the dashed blue curve. We use $k = 10$ servers. The service requirement distribution $S = \text{Uniform}(0, 2)$. We only simulate up to $\rho = 0.9975$ due to long convergence times.

Figure 3.1: Convergence of mean response time ratio

References

[1] Bunt, R. B. Scheduling techniques for operating systems. *Computer 9*, 10 (1976), 10–17.

[2] Grosof, I., Scully, Z., and Harchol-Balter, M. SRPT for multiserver systems. *arXiv* (2018).

[3] Guirguis, S., Sharaf, M. A., Chrysanthis, P. K., Labrinidis, A., and Pruhs, K. Adaptive scheduling of web transactions. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on* (2009), IEEE, pp. 357–368.

[4] Harchol-Balter, M., Schroeder, B., Bansal, N., and Agrawal, M. Size-based scheduling to improve web performance. *ACM Trans. Comput. Syst. 21*, 2 (May 2003), 207–233.

[5] Leonardi, S., and Raz, D. Approximating total flow time on parallel machines. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing* (1997), ACM, pp. 110–119.

[6] Leonardi, S., and Raz, D. Approximating total flow time on parallel machines. *Journal of Computer and System Sciences 73*, 6 (2007), 875 – 891.

[7] Lin, M., Wierman, A., and Zwart, B. Heavy-traffic analysis of mean response time under shortest remaining processing time. *Performance Evaluation 68*, 10 (2011), 955 – 966.

[8] Mangharam, R., Demirhan, M., Rajkumar, R., and Raychaudhuri, D. Size matters: Size-based scheduling for mpeg-4 over wireless channels. In *Multimedia Computing and Networking 2004* (2003), vol. 5305, International Society for Optics and Photonics, pp. 110–123.

[9] Schrage, L. Letter to the editora proof of the optimality of the shortest remaining processing time discipline. *Operations Research 16*, 3 (1968), 687–690.

[10] Schrage, L. E., and Miller, L. W. The queue M/G/1 with the shortest remaining processing time discipline. *Operations Research 14*, 4 (1966), 670–684.

[11] Wolff, R. W. Poisson arrivals see time averages. *Operations Research 30*, 2 (1982), 223–231.