Asymptotic Convergence of Scheduling Policies with Respect to Slowdown

Mor Harchol-Balter[1] Karl Sigman[2] Adam Wierman[3]

April 2002

CMU-CS-02-118

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We explore the performance of an M/GI/1 queue under various scheduling policies from the perspective of a new metric: the *slowdown* experienced by largest jobs. We consider scheduling policies that bias against large jobs, towards large jobs, and those that are fair, e.g., Processor-Sharing. We prove that as job size increases to infinity, all work conserving policies converge almost surely with respect to this metric to no more than $1/(1-\rho)$, where ρ denotes load. We also find that the expected slowdown under any work conserving policy can be made arbitrarily close to that under Processor-Sharing, for all job sizes that are sufficiently large.

[1] Carnegie Mellon University, Computer Science Department. Email: harchol@cs.cmu.edu.
[2] Columbia University, Department of Industrial Engineering and Operations Research. Email: sigman@ieor.columbia.edu
[3] Carnegie Mellon University, Computer Science Department. Email: acw@cs.cmu.edu.

This work was supported by NSF Career Grant CCR-0133077 and by Pittsburgh Digital Greenhouse Grant 01-1.

Keywords: scheduling; service discipline; M/G/1; slowdown; large jobs; convergence; limiting; work conserving; SRPT; shortest remaining processing time; PS; processor sharing; LRPT; longest remaining processing time

1 Introduction

It is well-known that choosing the right scheduling algorithm can have a big impact on performance, both in theory and in practice. For example, changing the scheduling algorithm in a CPU from Processor-Sharing (PS) to a scheduling policy that biases towards small jobs, such as Shortest-Remaining-Processing-Time-First (SRPT), or a scheduling policy that biases towards young jobs, such as Least-Attained-Service (LAS), can improve mean response time (a.k.a. sojourn time) dramatically.

However, less well understood is the performance impact of different scheduling policies on large jobs. For example, how does a policy that biases towards small jobs, such as SRPT, compare against a policy that biases towards large jobs, such as Longest-Remaining-Processing-Time-First (LRPT), when the performance metric is the response time of the large jobs?

In this paper we limit our discussion to an M/GI/1 queue. For the M/GI/1/PS queue with load ρ, all jobs (large or small) are slowed down by the same factor, $\frac{1}{1-\rho}$, in expectation. Because the slowdown (response time divided by job size) is the same for all job sizes, the PS policy is often referred to as the *fair policy*.

We will show that all *work conserving* scheduling policies have the same performance as PS with respect to large jobs. In particular, we show that the slowdown as job size tends to infinity under any work conserving policy is at most $\frac{1}{1-\rho}$; even for policies that clearly bias against large jobs. We also consider the expected slowdown for jobs that are not the very largest. We show that all "sufficiently-large" jobs have slowdown arbitrarily close to that of PS, where the definition of "sufficiently-large" depends on ρ and includes most jobs provided ρ is not too high.

2 Previous work

Ever since the discovery that SRPT has the lowest mean response time of any scheduling policy (for any sequence of arrival times and job sizes) [18, 22, 19], the evaluation of various scheduling policies has intrigued system designers and queueing theorists. There exist over a hundred survey papers to date on the analysis of scheduling policies, as well as many wonderful books such as [6, 11, 15, 5].

The SRPT policy in particular has received much attention. Schrage and Miller first derived the expressions for the response times in an M/G/1/SRPT queue [19]. This was further generalized by Pechinkin *et al.* to disciplines where the remaining times are divided into intervals [13]. The steady-state appearance of the M/G/1/SRPT queue was obtained by Schassberger [17]. Rajaraman et al. showed further that the mean slowdown under SRPT is at most twice the optimal mean slowdown for any sequence of job arrivals [8].

Though analytical formulas for the M/G/1 queue with various scheduling policies have been known for a long time, they are difficult to evaluate numerically, due to their complex form (many nested integrals). Hence, there was little work on the relative comparison of different scheduling policies.

More recently, papers have appeared in the literature that try to compare the performance of scheduling policies. The following papers have compared the *mean response times* of various scheduling policies under specific job size distributions and specific loads, by plotting the known formulas: [14, 20, 19, 10, 16]. A 7-year long study at University of Aachen under Schreiber [14, 20] involved extensive evaluation of SRPT for various job size distributions and loads. The survey paper by Schreiber [20] summarizes the results. They show that SRPT has significant mean response time improvements compared to other policies like FCFS, LFCS and PS.

The above mentioned results are all plots for *specific* job size distributions and loads. Hence it is not clear whether the conclusions based on these plots hold for more general job size distributions and loads. Furthermore the above studies examined *mean* response time and did not raise the problem of possible *unfairness* to long jobs.

It has often been cited that the superior performance of scheduling policies that bias towards small jobs may come at the cost of starving large jobs [3, 23, 24, 21]. Usually, examples of adversarial arrival sequences where a particular job starves are given to justify this. However, such worst case examples do not reflect the behavior of these policies in the average case. The term “starvation” is also used by people to indicate *unfairness*. It is often thought that policies that favor small jobs should result in worse expected performance for long jobs than policies that are ”fair,” like PS. The argument given is that if a scheduling policy manages to reduce the response time of small jobs, then the response times for the large jobs would have to increase considerably. This argument is valid for scheduling policies that do not make use of size, see the famous Kleinrock Conservation Law [11, Page 197].

Very recently, several papers have appeared that try to evaluate the problem of *unfairness* analytically, and thus consider the behavior of scheduling policies as a function of the job size. Bender et al. consider the metric *max slowdown* of a job as an indication of unfairness [3]. They show, with an example, that SRPT can have an arbitrarily large *max slowdown*. However, *max slowdown* is not an appropriate metric to measure unfairness. A large job may have an exceptionally long response time in some case, but it might do well most of the time.

Bansal and Harchol-Balter [2] compare the SRPT policy and the PS policy analytically for an M/G/1 queue on a per-job-size basis. They prove that if the load ρ is less than $\frac{1}{2}$, then every job, including the very largest jobs, have a lower expected response time under SRPT than under PS, for every job size distribution. They also prove that for arbitrary load ρ, the expected response time of a job of size x under SRPT is no more than c times that under PS, where c is a function of $\frac{1}{1-\rho}$. This result nicely complements the result in this paper (Theorem 5.3) which states that for all ρ, for every job size distribution, all sufficiently large jobs have expected response time (and slowdown) under SRPT which is *arbitrary close* to that under PS.

There has also been work in the area of proposing new SRPT-like policies [4, 12] that try to reduce the problem of unfairness, while still favoring the short jobs. These usually prioritize based on *both* the time a job has waited so far, and its remaining size. These policies are usually analytically intractable and have been evaluated by simulation only. However simulations show that they are promising.

To the best of our knowledge, no prior work has compared scheduling policies with respect to just their performance on large jobs.

3 The slowdown metric, the fairness metric, and some initial notation

We will throughout be considering a stable M/GI/1 queue. The average arrival rate will be λ. A job's *size* (service requirement) will be denoted by the random variable X and will be chosen i.i.d. from a continuous distribution with *finite mean* and *finite variance.* The probability density function (pdf) of the job size distribution is $f(x)$, and the cumulative distribution function (cdf) is $F(x) = P(X \leq x),\ x \geq 0$. We will denote the tail, $1 - F(x)$, by $\overline{F}(x)$. We assume that $f(x) > 0,\ x > 0$; service times can be arbitrarily large. Throughout we distinguish between the "size of a job" and the "remaining size of a job." The former denotes the service requirement upon time of arrival (original size chosen from F). The latter denotes the leftover (remaining) service time at the time in question. The load (utilization), ρ, of the server is

$$\rho \overset{\text{def}}{=} \lambda E[X] = \lambda \int_0^\infty xf(x)dx.$$

We always will assume that $\rho < 1$; the queue is stable. The load made up by the jobs of size less than or equal to x, $\rho(x)$, is

$$\rho(x) \overset{\text{def}}{=} \lambda \int_0^x tf(t)dt.$$

We will use T to denote the steady-state response time (a.k.a. sojourn time) and $T(x)$ to denote the steady-state response time for a job of size x; a customer arriving in steady-state bringing a service time of length x has a response time $T(x)$. By definition, T has the same distribution as $T(X)$, and

$$E[T] = \int_0^\infty E[T(x)]f(x)dx$$

where X is chosen independent of T throughout this paper. Note that $\{T(x) : x \geq 0\}$ is a stochastic process. Formally, at time $t = 0$ we initially start the system in steady-state, and then for each x, we construct each $T(x)$ using the same initial state and future service and interarrival times (along each sample path).

Definition 3.1 *For any given policy, the slowdown, S, is defined as response time divided by job size, namely,*

$$S = \frac{T(X)}{X}.$$

The slowdown for a job of size x, $S(x)$, is thus given by

$$S(x) = \frac{T(x)}{x}.$$

The expected slowdown for a job of size x, $E[S(x)]$, is given by

$$E[S(x)] = \frac{E[T(x)]}{x}.$$

The overall mean slowdown is given by

$$E[S] = \int_0^\infty E[S(x)]f(x)dx.$$

Our **primary metric of interest** in this paper is **slowdown**. Mean slowdown is often used as a measure of system performance as opposed to the more traditional mean response time for two reasons [7, 1, 9]. First, it is desirable that a job's response time be correlated with its size (processing requirement). We'd like small jobs to have small response times and big jobs to have big response times. By bringing down mean response time, Markov's inequality tells us that we're also dropping the fraction of jobs with really high slowdowns.

A second reason why we care about mean slowdown is that it is more representative of the performance of a large fraction of jobs. Observe that mean response time tends to be representative of the performance of just a few jobs – the bigger ones – since they count the most in the mean because their response times tend to be highest. An improvement in mean response time could just indicate that the performance of a few big jobs has improved. By contrast, mean slowdown can only be improved significantly if you affect the slowdown of a larger fraction of all jobs. Thus to improve mean slowdown, you have to touch that large set of small jobs.

It is well known that for an M/GI/1/PS queue,

$$E[S(x)]^{PS} = \frac{1}{1-\rho}. \tag{1}$$

This says that for any given load $\rho < 1$, under PS scheduling, all jobs have the same expected slowdown; hence PS is **"fair"**.

In this paper we will consider policies that significantly improve upon PS with respect to mean slowdown by giving priority to short jobs, or to young jobs. We will ask whether the large jobs suffer as a consequence. Specifically, we will be interested in the slowdown for large jobs.

Definition 3.2 *For any given scheduling policy, the slowdown for large jobs is defined (when it exists) by*

$$\lim_{x\to\infty} S(x)$$

whereby the convergence is almost sure (a.s.) convergence, by which we mean with probability 1*. The expected slowdown for large jobs is defined (when it exists) by:*

$$\lim_{x\to\infty} E[S(x)]$$

4 Brief review of common scheduling policies

In this section we define several common scheduling policies and summarize known results for these policies under an M/GI/1 queue, with respect to the mean response time for a job of size x.

PS: Processor-Sharing

Under the PS policy the processor is shared fairly among all jobs currently in the system [25]:

$$\begin{aligned} E[T(x)]^{PS} &= \frac{x}{1-\rho} \\ E[T] &= \frac{E[X]}{1-\rho} \end{aligned}$$

SRPT: Shortest-Remaining-Processing-Time-First

Under the SRPT policy, at every moment of time, the server is processing that job with the shortest remaining processing time. The SRPT policy is well-known to be optimal for minimizing mean response time [19]. The mean response time for a job of size x, $E[T(x)]^{SRPT}$, can be decomposed into a sum:

$$E[T(x)]^{SRPT} = E[W(x)]^{SRPT} + E[R(x)]^{SRPT}$$

where $E[W(x)]^{SRPT}$ is the expected waiting time for the job (the expected time for a job of size x from when it first arrives to when it receives service for the first time) and $E[R(x)]^{SRPT}$ is the expected residence time (the time it takes for a job of size x to complete service once it begins execution) [19].

$$\begin{aligned} E[W(x)]^{SRPT} &= \frac{\frac{\lambda}{2}\int_0^x t^2 f(t)dt + \frac{\lambda}{2}x^2\overline{F}(x)}{(1-\rho(x))^2}, && (2) \\ E[R(x)]^{SRPT} &= \int_0^x \frac{dt}{1-\rho(t)}. && (3) \end{aligned}$$

P-LCFS: Preemptive-Last-Come-First-Served

Under P-LCFS, whenever a new arrival enters the system, it immediately preempts the job in service. Only when that arrival completes does the preempted job get to resume service. This policy is easy to understand since a new arrival can be thought of as starting its own busy period, where the new arrival can't leave until this busy period completes. Letting B denote the length of a busy period, and X denote a service requirement as usual, we have [11]:

$$\begin{aligned} E[T(x)]^{P-LCFS} &= E[B] = \frac{x}{1-\rho} && (4) \\ E[T] &= \frac{E[X]}{1-\rho} && (5) \end{aligned}$$

LAS: Least-Attained-Service

Under LAS, the job with the least attained service gets the processor to itself. If several jobs all have the least attained service, they time-share the processor via PS. This is a very practical policy, since a job's *age* (attained service) is always known, although it's size may not be known. This policy improves upon PS with respect to mean response time and mean slowdown when the job size distribution has decreasing failure rate.

Both $E[T(x)]$ and the Laplace transform of $T(x)$ under LAS are known [11]. We need some preliminary notation.

For $x \geq 0$, let

$$X_x = \min\{x, X\}.$$

Then

$$E[X_x] = \int_0^x yf(y)dy + x\overline{F}(x)$$

$$E[X_x^2] = \int_0^x y^2 f(y)dy + x^2\overline{F}(x)$$

Observe that X_x is similar to the R.V. X, except that all job sizes have been capped at a maximum of x.

Given the above definitions, we have:

$$E[T(x)]^{LAS} = \frac{x(1-\rho_x) + \frac{\lambda}{2}E[X_x^2]}{(1-\rho_x)^2} \tag{6}$$

where

$$\rho_x = \lambda E[X_x].$$

LRPT: Longest-Remaining-Processing-Time

Under the LRPT policy, at every moment of time, the server is processing the job with the longest remaining processing time. If multiple jobs in the system have the same remaining processing time, they time-share the processor via PS. Since the LRPT policy biases towards the *longest* jobs, it is of little practical value.

We couldn't locate an analysis of this policy for the M/GI/1 queue anywhere, although analyzing LRPT isn't difficult, and we do so later in the paper.

SJF: Shortest-Job-First

SJF is the non-preemptive variant of SRPT. Under SJF, when the server is free it chooses to run the shortest job [6]:

$$E[T(x)]^{SJF} = x + \frac{\rho E[X^2]}{2E[X]} \cdot \frac{1}{(1-\rho(x))^2}$$

Other policies not mentioned above

There are many other scheduling policies that we haven't mentioned.

All non-preemptive policies that don't make use of a job's size, for example, FCFS (First-Come-First-Served), LCFS (non-preemptive Last Come First Served), or RANDOM (random) will have the same mean response time, $E[T]$, and thus for all such policies,

$$E[T(x)] = E[T] - E[X] + x = \frac{\lambda E[X^2]}{2(1-\rho)} + x$$

where X is the service time. Since these have the same performance with respect to $E[T(x)]$, we will discuss them as a group.

5 Convergence of scheduling policies in expectation

In this section, we evaluate the *expected slowdown* for the largest jobs under different scheduling policies. In Section 5.1 we consider 5 particular scheduling policies and show that they have the same expected slowdown as PS for the largest job. In Section 5.2 and Section 5.3 we generalize these results to all work conserving scheduling policies. Finally, in Section 5.4 we consider the broader problem of expected slowdown as a function of job size, for all job sizes. We find that for any work conserving policy, for sufficiently large jobs, the expected slowdown can be shown to be arbitrarily close to that of PS, where our definition of sufficiently large will typically include most jobs.

5.1 Convergence of 5 scheduling policies in expectation

This section will prove the following theorem:

Theorem 5.1 *As $x \to \infty$, expected slowdown for SRPT, P-LCFS, LAS, and LRPT is the same as for PS:*

$$\lim_{x\to\infty} E[S(x)]^{SRPT} = \lim_{x\to\infty} E[S(x)]^{P-LCFS} = \lim_{x\to\infty} E[S(x)]^{LAS} = \lim_{x\to\infty} E[S(x)]^{LRPT} = \frac{1}{1-\rho}.$$

That is, the expected slowdown for the largest job is the same under policies that bias towards short jobs, policies that bias towards long jobs, and policies that treat all jobs fairly.

Proof for SRPT

We start by looking at the waiting time component of SRPT:

$$\begin{aligned} E[W(x)]^{SRPT} &= \frac{\frac{\lambda}{2}\int_0^x t^2 f(t)dt + \frac{\lambda}{2}x^2\overline{F}(x)}{(1-\rho(x))^2} \\ &= \frac{\lambda\int_0^x t\,\overline{F}(t)dt}{(1-\rho(x))^2} \\ \lim_{x\to\infty} E[W(x)]^{SRPT} &= \frac{\lambda\int_0^\infty t\,\overline{F}(t)dt}{(1-\rho)^2} < \infty \end{aligned}$$

where finiteness follows since the service time distribution F is assumed to have finite second moment.[1]

Thus we have

$$\lim_{x\to\infty} \frac{E[W(x)]^{SRPT}}{x} = 0$$

Next consider the residence time component of SRPT:

[1] Recall $\int_0^\infty y\overline{F}(y)dy = \int_0^\infty y\int_y^\infty f(x)dxdy = \int_0^\infty f(x)\int_0^x ydydx = \int_0^\infty f(x)\frac{x^2}{2}dx = \frac{1}{2}E[X^2]$

$$\begin{aligned}
\lim_{x\to\infty} \frac{E[R(x)]^{SRPT}}{x} &= \lim_{x\to\infty} \frac{1}{x}\int_0^x \frac{dt}{1-\rho(t)} \\
&= \lim_{x\to\infty} \frac{1}{1-\rho(x)} \text{ (by L'Hopital)} \\
&= \frac{1}{1-\rho}
\end{aligned}$$

Combining waiting time and residence time, we have:

$$\lim_{x\to\infty} E[S(x)] = \lim_{x\to\infty} \frac{E[T(x)]^{SRPT}}{x} = \frac{1}{1-\rho}$$

Proof for LAS

We start with a lemma showing that for all job sizes x and for all load, the performance of LAS is worse than or equal to that of SRPT:

Lemma 5.1 *In an M/G/1, for all x and for all ρ,*

$$E[T(x)]^{SRPT} \leq E[T(x)]^{LAS}$$

Proof : The proof is simply algebraic:

$$\begin{aligned}
E[T(x)]^{LAS} &= \frac{x(1-\rho_x) + \frac{1}{2}\lambda E[{X_x}^2]}{(1-\rho_x)^2} \\
&= \frac{x}{1-\rho_x} + \frac{\frac{1}{2}\lambda\left(\int_0^x y^2 f(y)dy + x^2\overline{F}(x)\right)}{(1-\rho_x)^2} \\
&\geq \frac{x}{1-\rho(x)} + \frac{\frac{1}{2}\lambda\left(\int_0^x y^2 f(y)dy + x^2\overline{F}(x)\right)}{(1-\rho(x))^2} \\
&= \frac{x}{1-\rho(x)} + \frac{\frac{1}{2}\lambda \int_0^x y^2 f(y)dy + \frac{1}{2}\lambda x^2\overline{F}(x)}{(1-\rho(x))^2} \\
&\geq E[T(x)]^{SRPT}
\end{aligned}$$

■

The limiting slowdown of large jobs, however, is the same under LAS and SRPT as shown below:

$$\begin{aligned}
\rho_x &= \lambda\int_0^x yf(y)dy + \lambda x\overline{F}(x) = \lambda\int_0^x \overline{F}(y)dy \\
\lim_{x\to\infty} \rho_x &= \lambda\int_0^\infty \overline{F}(y)dy = \lambda E[X] = \rho
\end{aligned}$$

$$
\begin{aligned}
E[T(x)]^{LAS} &= \frac{x}{1-\rho_x} + \frac{\frac{\lambda}{2}\left(\int_0^x y^2 f(y)dy + x^2\overline{F}(x)\right)}{(1-\rho_x)^2} \\
&= \frac{x}{1-\rho_x} + \frac{\lambda \int_0^x \overline{F}(y)dy}{(1-\rho_x)^2} \\
\lim_{x\to\infty} E[S(x)]^{LAS} = \lim_{x\to\infty} \frac{E[T(x)]^{LAS}}{x} &= \lim_{x\to\infty} \frac{x}{1-\rho_x}\cdot\frac{1}{x} + \lim_{x\to\infty} \frac{\lambda\int_0^x \overline{F}(y)ydy}{(1-\rho_x)^2}\cdot\frac{1}{x} \\
&= \frac{1}{1-\rho} + \frac{\lambda\int_0^\infty \overline{F}(y)ydy}{(1-\rho)^2} \lim_{x\to\infty}\frac{1}{x}
\end{aligned}
$$

Again, by the finiteness of the second moment of F, we have:

$$
\lim_{x\to\infty} E[S(x)]^{LAS} = \frac{1}{1-\rho}
$$

Proof for LRPT

We will use the following notation in this section and throughout the rest of the paper: B will denote the length of a busy period. $B(x)$ will denote the length of a busy period started by a job of size x (an exceptional first service busy period). $B(x)|_{\lambda'}$ will denote the length of a busy period started by a job of size x where the arrival rate is λ'.

We begin by noticing that a job of size x enters either a busy or an idle system. If the job enters an idle system, $T(x) = B(x)$, since LRPT has the property that all jobs finish at the end of the busy period they arrive into under LRPT.

If the job enters a busy system, then we can again take advantage of the above property to see that $T(x) = B(x + V|busy)$, where V is the amount of work in the system seen by an arbitrary arrival and $V|busy$ is the work in the system seen by an arrival which finds the system busy. Now, since LRPT is work conserving, we know that:

$$
\begin{aligned}
E[V] &= E[W(x)]^{FCFS} = \frac{\lambda E[X^2]}{2(1-\rho)}, \text{ and} \\
E[V|busy] &= \frac{E[W(x)]^{FCFS}}{\rho}
\end{aligned}
$$

where X is the service time and $W(x)^{FCFS}$ is the waiting time in a FCFS queue.

It is well known that $E[B(Y)] = \frac{E[Y]}{1-\rho}$ for any exceptional first service time Y. Thus, it holds for $Y = x$ and $Y = x + (V|busy)$. Using this we obtain:

$$
\begin{aligned}
E[S(x)]^{LRPT} &= \rho\frac{E[B(x+V|busy)]}{x} + (1-\rho)\frac{E[B(x)]}{x} \\
&= \rho\frac{1 + \frac{1}{x}\frac{\lambda E[X^2]}{2\rho(1-\rho)}}{1-\rho} + (1-\rho)\frac{1}{1-\rho}
\end{aligned}
$$

Thus,

$$
\lim_{x\to\infty} E[S(x)]^{LRPT} = \frac{1}{1-\rho} \qquad (7)
$$

Proof for P-LCFS

For the P-LCFS policy it trivially follows from (4) that:

$$\lim_{x\to\infty} \frac{E[T(x)]^{P-LCFS}}{x} = \frac{1}{1-\rho}$$

5.2 Convergence of all work conserving scheduling policies in expectation

This section extends the analysis of the previous section. The goal is to to bound convergence in expectation of slowdown under *any work conserving policy*. We prove the following theorem:

Theorem 5.2 *For any work conserving scheduling policy*

$$\lim_{x\to\infty} E[S(x)] \leq \frac{1}{1-\rho}.$$

If the policy is also non-preemptive, then $E[S(x)] \to 1$ *as* $x \to \infty$.

Proof :

The proof of the $\frac{1}{1-\rho}$ bound stems from the observation that LRPT provides an upper bound on $T(x)^P$ for any work conserving policy P. That is, under LRPT, every job finishes the moment the busy period the job arrived into ends, which is the last possible completion moment for any work conserving policy. So, the result follows from Equation 7. For any work conserving policy P:

$$\lim_{x\to\infty} E[S(x)]^P \leq \lim_{x\to\infty} E[S(x)]^{LRPT} = \frac{1}{1-\rho}.$$

This proves the first half of the theorem.

Now we limit our discussion to non-preemptive work conserving policies. For a job of size x arriving into the system:

$$T(x) = W(x) + x$$

where $W(x)$ is the waiting time for a job of size x. Let V denote the amount of work in the system when job x arrives. Observe that $W(x)$ is less than the length of a busy period started by a job of size equal to V. That is, for all sample paths,

$$W(x) \quad \leq \quad B(V) \tag{8}$$

where $B(y)$ denotes the length of a busy period started by a job of size y. So,

$$E[W(x)] \leq \frac{E[V]}{1-\rho}$$

Thus, letting X be the service time distribution, we have

$$\begin{aligned} E[S(x)] &= \frac{E[T(x)]}{x} = \frac{E[W(x)]}{x} + \frac{x}{x} \\ &\leq \frac{E[V]}{1-\rho} \cdot \frac{1}{x} + 1 \\ &= \frac{\frac{\lambda E[X^2]}{2(1-\rho)}}{1-\rho} \cdot \frac{1}{x} + 1 \\ &\rightarrow 1 \text{ as } x \rightarrow \infty \end{aligned}$$

■

5.3 Followup remarks on convergence in expectation

A few followup observations are in order regarding Theorem 5.2.

Remark 5.1 *Theorem 5.2 does not extend to policies that are not work conserving. In fact, for every* $z \in [1, \infty)$ *there is a non work conserving policy such that* $\lim_{x\to\infty} E[S(x)] = z$.

To see this, consider the policy that makes each job wait $(z-1)x$ time before it is allowed to enter the queue of a non-preemptive, work conserving system.

Remark 5.2 *The* $\frac{1}{1-\rho}$ *bound in Theorem 5.2 is tight. In fact, For every* $z \in [1, \frac{1}{1-\rho}]$ *there is a work conserving policy such that* $E[S(x)] \rightarrow z,$ *as* $x \rightarrow \infty$.

Proof : Consider a linear combination of the FCFS and P-LCFS policies. More specifically, consider the following scheduling policy, P: with probability q an arriving job preempts the job being serviced, and with probability $1-q$ an arriving job is placed at the back of a FCFS queue to await service.

We can quickly analyze this policy to find $E[S(x)]^P$. Consider an arrival that gets placed at the front of the queue. This arrival can only be bothered by other jobs that are allowed to preempt. Thus, for this job $T(x) = B(x)|_{\lambda'}$, where $\lambda' = q\lambda$ for $q \in [0,1]$. That is, $T(x)$ is the length of a busy period started by a job of size x where the arrival rate is λ'.

Now consider a job that gets placed in the back of the queue. If the system is idle when the job arrives, we again see that $T(x) = B(x)|_{\lambda'}$. However, if the system is busy at the time of the arrival $T(x) = B(x + V|busy))|_{\lambda'}$, where V is the amount of work in system seen by an arbitrary arrival, and $V|busy$ is the work seen by an arrival which finds the system busy. As in the analysis of LRPT, we know that

$$E[V|busy] = \frac{E[W(x)^{FCFS}]}{\rho} = \frac{\lambda E[X^2]}{2\rho(1-\rho)}.$$

Let $\rho' = \frac{\lambda'}{\mu}$. Then, putting these three pieces together, we see that as $x \rightarrow \infty$:

$$\begin{aligned} E[S(x)]^P &= q\frac{E[B(x)]|_{\lambda'}}{x} + (1-q)\left[\rho\frac{E[B(x)]|_{\lambda'}}{x} + (1-\rho)\frac{E[B(x+V|busy)]|_{\lambda'}}{x}\right] \\ &= q\frac{1}{1-\rho'} + (1-q)\left[\rho\frac{1}{1-\rho'} + (1-\rho)\frac{1+\frac{1}{x}\frac{\lambda E[X^2]}{2\rho(1-\rho)}}{1-\rho'}\right] \rightarrow \frac{1}{1-\rho'} \end{aligned}$$

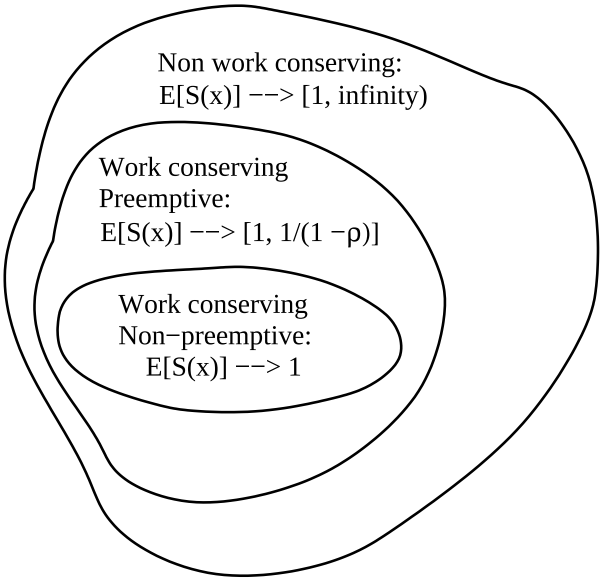

Figure 1: *Taxonomy of scheduling policies defined by the metric* $\lim_{x\to\infty} E[S(x)]$.

Notice that since ρ' is an arbitrary number in $[0, \rho]$, we can make $\frac{1}{1-\rho'}$ any number in $[1, \frac{1}{1-\rho}]$. ■

The above remarks show that the metric $\lim_{x\to\infty} E[S(x)]$ defines a taxonomy on all scheduling policies, as shown in Figure 1. Non work conserving policies have a value in $[1, \infty)$ under this metric. Preemptive work conserving policies have a value in $[1, \frac{1}{1-\rho}]$ under this metric. Non-preemptive work conserving policies all have a value of 1 under this metric. Each class is complete in that for each value in the range, there exists a policy with that value.

5.4 Bounding all work conserving policies for sufficiently-large job sizes

Until now we have concentrated on the limiting behavior as the job size $x \to \infty$. We now show that we can easily prove an upper bound of $(1+\varepsilon)\frac{1}{1-\rho}$ for the expected slowdown of all "sufficiently large" jobs under all work conserving scheduling policies for any $\varepsilon > 0$.

Let V be the amount of work in the system when a job arrives. Recall that this is the same under all work conserving policies and for jobs of any size. In fact, $E[V] = E[W(x)]^{FCFS}$.

Theorem 5.3 *Fix $\varepsilon > 0$. Then under any work conserving scheduling policy P, if $x \geq \frac{1}{\varepsilon}E[V]$, then*

$$E[S(x)]^P \leq (1+\varepsilon)E[S(x)]^{PS} = (1+\varepsilon)\frac{1}{1-\rho}.$$

If the policy is also non-preemptive and $x \geq \frac{1}{\varepsilon(1-\rho)}E[V]$, then

$$E[S(x)]^P \leq 1+\varepsilon$$

Before we state the proof, observe that provided ρ is not too high, the above theorem says that in fact *most* jobs are sufficiently large, since $E[W(x)]^{FCFS}$ will be low.

Proof :

Recall that LRPT provides an upper bound on $S(x)^P$ for any work conserving policy P. That is, every job finishes at the last possible moment under LRPT, and so the slowdown of any other policy must be bounded by that of LRPT . Thus, we need simply show that for sufficiently large x, $E[S(x)]^{LRPT} \leq \frac{1+\varepsilon}{1-\rho}$.

Observing that $T(x)^{LRPT}$ has the same distribution (hence mean) as $B(x+V)$, we have

$$\begin{aligned} E[S(x)]^{LRPT} &= \frac{1}{x} E[T(x)]^{LRPT} \\ &= \frac{1}{x} \cdot \frac{x+E(V)}{(1-\rho)} \\ &= \frac{E[V]}{x(1-\rho)} + \frac{1}{1-\rho} \end{aligned}$$

Letting $x \geq \frac{1}{\varepsilon} E[V]$ gives us

$$E[S(x)]^P \leq E[S(x)]^{LRPT} \quad \leq \quad \frac{1+\varepsilon}{1-\rho}.$$

Further, we can obtain a similar bound on convergence for non-preemptive, work conserving policies. Recall from the proof of Theorem 5.2 that for any non-preemptive, work conserving policy P, we have

$$E[S(x)]^P \leq \frac{E[V]}{1-\rho} \frac{1}{x} + 1$$

Thus, letting $x \geq \frac{1}{\varepsilon(1-\rho)} E[V]$ gives us

$$E[S(x)]^P \leq 1 + \varepsilon$$

■

6 Almost sure convergence of scheduling policies

In this section, we extend the analysis of Theorem 5.2 in order to show that under any work conserving policy the performance of the largest jobs will be at most that of PS almost surely. Recall that:

Definition 6.1 *The sequence of random variables* $\{Y_n, n = 1, 2, \ldots\}$ *is said to converge almost surely to a random variable* Y*, written* $Y_n \overset{a.s.}{\rightarrow} Y$ *as* $n \to \infty$*, if*

$$P(\lim_{n\to\infty} Y_n = Y) = 1.$$

We equivalently say that Y_n *converges to* Y *with probability* 1 *(w.p.1.).*

Theorem 6.1 *Under work conserving scheduling policies it holds a.s. (assuming the limit exists) that*

$$\lim_{x\to\infty} S(x) \leq \frac{1}{1-\rho}.$$

If the policy is also non-preemptive, then the limit does exists and $S(x) \overset{a.s.}{\to} 1$ *as* $x \to \infty$.

Proof : The proof for *non-preemptive,* work conserving policies is quick: Start with the observation that

$$P(S(x)^P \geq 1) \quad = 1 \qquad \forall x, \forall \text{ policies P}$$

This follows simply by definition of slowdown. Thus by taking limits, a.s. it holds that

$$\liminf_{x\to\infty} S(x)^P \geq 1, \forall \text{ policies P}$$

Now, recall from Equation (8) that we have a.s. that

$$S(x)^P \quad \leq \quad 1 + \frac{B(V)}{x} \quad \forall x, \forall \text{work conserving, non-preemptive policies P}$$

Taking limits we have a.s. that:

$$\limsup_{x\to\infty} S(x)^P \leq 1, \forall \text{work conserving, non-preemptive policies P}$$

It follows that for all work conserving, non-preemptive policies P the limit does exists and

$$S(x) \overset{a.s.}{\to} 1 \text{ as } x \to \infty.$$

The remainder of the proof will concentrate on work conserving policies that may allow for *preemption*.

We know that a.s.

$$T(x) \leq B(x+V),$$

where $B(y)$ is used to denote the length of a busy period started by a job of size y.

Thus

$$\lim_{x\to\infty} T(x)/x \leq \lim_{x\to\infty} \frac{B(x+V)}{x}.$$

We will complete the proof by showing that

$$\lim_{x\to\infty} \frac{B(x+V)}{x} =_{a.s.} \frac{1}{1-\rho} \tag{9}$$

If we let $\{B_i : i \geq 1\}$ denote an i.i.d. sequence of regular busy periods (non-exceptional), then $B(x)$ can be expressed as

$$B(x) = x + \sum_{i=1}^{N(x)} B_i$$

where $\{N(x) : x \geq 1\}$ is a Poisson process of rate λ independent of $\{B_i : i \geq 1\}$. We conclude that this version of $\{B(x) : x \geq 0\}$ is a compound Poisson process with a linear x term added on, so it has stationary and independent increments. Thus, almost surely,

$$\begin{aligned} \lim_{x \to \infty} \frac{B(x)}{x} &= E[B(1)] \quad \text{(by S.L.L.N)} \\ &= \frac{1}{1-\rho} \end{aligned}$$

Notice that replacing x by $x + V$ does not change this limit.

■

7 Conclusion

In this paper we consider the performance metric "slowdown for the largest job" and we show that under this metric the performance of all work conserving scheduling policies is bounded by $\frac{1}{1-\rho}$ almost surely.

This metric is also interesting for another reason; it allows us to categorize all scheduling policies into 3 classes. We find that for *non work conserving policies*, the expected slowdown of the largest job can range from 1 to infinity (and in fact every value in between is achieved by some non work conserving policy). For *preemptive work conserving policies*, the expected slowdown of the largest job can range from 1 to $\frac{1}{1-\rho}$ (and again each value in between is achieved by some preemptive work conserving policy). Lastly, for non-preemptive work conserving policies, the expected slowdown of the largest job is always 1.

This paper also raises the question of how scheduling policies compare with respect to slowdown on job sizes other than the very largest. We find that for all "sufficiently large" jobs, the expected slowdown of these jobs under any work conserving policy can be made arbitrarily close to $\frac{1}{1-\rho}$, where the definition of "sufficiently large" depends on the degree of closeness and on the system load. When the system load is not too high, "sufficiently large" ends up including most jobs. The behavior of scheduling policies on jobs other than the largest job is an interesting question which will surely generate further research.

The proofs in this paper are varied, but all surprisingly simple, which should help others in extending this work. The proofs rely on a few key observations about subdividing busy periods and on some alternative formulations of scheduling formulas. Perhaps the most useful observation is that the Longest-Remaining-Processing-Time policy can be used to bound all other work conserving policies, and that it suffices to therefore to concentrate on this one policy.

References

[1] Baily, Foster, Hoang, Jette, Klingner, Kramer, Macaluso, Messina, Nielsen, Reed, Rudolph, Smith, Tomkins, Towns, and Vildibill. Valuation of ultra-scale computing systems. White Paper, 1999.

[2] Nikhil Bansal and Mor Harchol-Balter. Analysis of SRPT scheduling: Investigating unfairness. In *Proceedings of* Sigmetrics '01, 2001.

[3] M. Bender, S. Chakrabarti, and S. Muthukrishnan. Flow and stretch metrics for scheduling continous job streams. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[4] L. Cherkasova. Scheduling strategies to improve response time for web applications. In *High-performance computing and networking: international conference and exhibition*, pages 305–314, 1998.

[5] E.G. Coffman and L. kleinrock. Computer scheduling methods and their countermeasures. In *AFIPS conference proceedings*, volume 32, pages 11–21, 1968.

[6] Richard W. Conway, William L. Maxwell, and Louis W. Miller. *Theory of Scheduling*. Addison-Wesley Publishing Company, 1967.

[7] Allen B. Downey. A parallel workload model and its implications for processor allocation. In *Proceedings of High Performance Distributed Computing*, pages 112–123, August 1997.

[8] J.E. Gehrke, S. Muthukrishnan, R. Rajaraman, and A. Shaheen. Scheduling to minimize average stretch online. In *40th Annual symposium on Foundation of Computer Science*, pages 433–422, 1999.

[9] M. Harchol-Balter and A. Downey. Exploiting process lifetime distributions for dynamic load balancing. *ACM Transactions on Computer Systems*, 15(3), 1997.

[10] Mor Harchol-Balter, M. Crovella, and S. Park. The case for srpt schduling in web servers. Technical Report MIT-LCS-TR-767, MIT Lab for Computer Science, October 1998.

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

[12] E. Modiano. Scheduling algorithms for message transmission over a satellite broadcast system. In *Proceedings of IEEE MILCOM '97*, pages 628–634, 1997.

[13] A.V. Pechinkin, A.D. Solovyev, and S.F. Yashkov. A system with servicing discipline whereby the order of remaining length is serviced first. *Tekhnicheskaya Kibernetika*, 17:51–59, 1979.

[14] R. Perera. The variance of delay time in queueing system M/G/1 with optimal strategy SRPT. *Archiv fur Elektronik und Uebertragungstechnik*, 47:110–114, 1993.

[15] M. Pinedo. *On-line algorithms, Lecture Notes in Computer Science.* Prentice Hall, 1995.

[16] J. Roberts and L. Massoulie. Bandwidth sharing and admission control for elastic traffic. In *ITC Specialist Seminar*, 1998.

[17] R. Schassberger. The steady-state appearance of the M/G/1 queue under the discipline of shortest remaining processing time. *Advances in Applied Probability*, 22:456–479, 1990.

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

[19] Linus E. Schrage and Louis W. Miller. The queue M/G/1 with the shortest remaining processing time discipline. *Operations Research*, 14:670–684, 1966.

[20] F. Schreiber. Properties and applications of the optimal queueing strategy SRPT - a survey. *Archiv fur Elektronik und Uebertragungstechnik*, 47:372–378, 1993.

[21] A. Silberschatz and P. Galvin. *Operating System Concepts, 5th Edition*. John Wiley & Sons, 1998.

[22] D.R. Smith. A new proof of the optimality of the shortest remaining processing time discipline. *Operations Research*, 26:197–199, 1976.

[23] W. Stallings. *Operating Systems, 2nd Edition*. Prentice Hall, 1995.

[24] A.S. Tanenbaum. *Modern Operating Systems.* Prentice Hall, 1992.

[25] Ronald W. Wolff. *Stochastic Modeling and the Theory of Queues.* Prentice Hall, 1989.