

Size-based Scheduling Policies with Inaccurate Scheduling Information

Dong Lu* Huanyuan Sheng[†] Peter Dinda*
{donglu,h-sheng,pdinda}@northwestern.edu

*Department of Computer Science,

[†]Department of Industrial Engineering and Management Sciences
Northwestern University

Abstract

Size-based scheduling policies such as SRPT have been studied since 1960s and have been applied in various arenas including packet networks and web server scheduling. SRPT has been proven to be optimal in the sense that it yields—compared to any other conceivable strategy—the smallest mean value of occupancy and therefore also of waiting and delay time. One important prerequisite to applying size-based scheduling is to know the sizes of all jobs in advance, which are unfortunately not always available. No work has been done to study the performance of size-based scheduling policies when only inaccurate scheduling information is available. In this paper, we study the performance of SRPT and FSP as a function of the correlation coefficient between the actual job sizes and estimated job sizes. We developed a simulator that supports both M/G/1/m and G/G/n/m queuing models. The simulator can be driven by trace data or synthetic data produced by a workload generator we have developed that allows us to control the correlation. The simulations show that the degree of correlation has a dramatic effect on the performance of SRPT and FSP and that a reasonably good job size estimator will make both SRPT and FSP outperform PS in both mean response time and slowdown.

1. Introduction

In a queuing system, job requests continuously arrive to be serviced by one or several servers or stations. A request requires a certain service time to be completed. A request is queued when it arrives and remains in the system until it is complete, with the total time from arrival to completion being called the sojourn time or response time. Scheduling policies determine which requests in the queue are serviced

at any point in time, how much time is spent on each, and what happens when a new request arrives. Common goals of the scheduling policy are to minimize the mean sojourn time (response time of the request), the average slowdown (the ratio of its response time to its size), and to act fairly to all requests.

Many policies are widely used due to their simplicity or perceived fairness. First Come First Served (FCFS) is a non-preemptive policy in which the requests are run to completion in the order in which they were received. A more common policy is Processor Sharing (PS), which is preemptive. In PS all requests in the queue are given an equal share of the server's resources and all requests share roughly same slowdown, thus PS is considered to be fair. Generalized Processor Sharing (GPS) generalizes PS with priorities. Often, FCFS can be combined with PS or GPS, with FCFS dispatching of requests from the queue to a pool of processes or threads that are collectively scheduled using PS or GPS. These policies ignore the service time or job size of the requests.

Sized-based scheduling policies, such as Shortest Remaining Processing Time (SRPT) and the Fair Sojourn Protocol (FSP) incorporate the service time or the job size of the request into their scheduling decisions, and thus can achieve shorter mean response time than the scheduling policies that ignore the job size information, such as FCFS and PS.

The primary concern with SRPT is the fear that large jobs may starve under SRPT [23], that the average performance improvements of SRPT over other policies stem from SRPT unfairly penalizing large jobs in order to help small jobs. Recent research [2, 11] has shown that the performance gains of SRPT over PS in fact do not usually come at the expense of large jobs.

These results make size-based scheduling more practical. However, size-based scheduling policies require a priori knowledge of job sizes, which is not always available. This is another reason for the lack of broad application of these policies [2]. All previous research work has targeted ideal size-based policies where the job sizes are assumed to be accurately known in advance. As a result, the behavior

Effort sponsored by the National Science Foundation under Grants ANI-0093221, ACI-0112891, ANI-0301108, EIA-0130869, and EIA-0224449. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation (NSF).

of size-based scheduling policies with inaccurate scheduling information is largely unknown. This paper is the first work to address the question.

The rest of the paper is organized as follows. We talk about related work in Section 2. In Section 3, we describe our simulation setup and introduce a useful random number generator algorithm that allows us to control the correlation between two random number series. Next, we show simulation results on performance in Section 4, and fairness results in Section 5. We conclude with two new applications of size-based scheduling policies in Section 6 that would be made possible given reasonably accurate job size estimators.

2. Related work

SRPT has been studied since the 1960s. Schrage first derived the expression for the response time in an M/G/1 queue [19]. For a general queuing system (G/G/1) Schrage proved in 1968 that SRPT is optimal in the sense that it yields—compared to any other conceivable strategy—the smallest mean value of occupancy and therefore also of waiting and delay time [18]. Schassberger obtained the steady state appearance of the M/G/1 queue with SRPT policy in 1990. Perera studied the variance of delay time in M/G/1/SRPT queuing systems and concluded that the variance is lower than FIFO and LIFO [16]. Bux introduced the SRPT principle into packet networks [4] in 1983.

Recently, SRPT [2, 21, 12, 11, 10] and FSP [10] have received much attention in the context of connection scheduling at web servers. Bansal, et al proved theoretically that the degree of unfairness under SRPT is surprisingly small assuming an M/G/1 queuing model and heavy-tailed job size distribution [2]. Gong, et al further investigated the fairness issues of SRPT through simulation [11] and confirmed the theoretical results regarding the asymptotic convergence of scheduling policies with respect to slowdown [13]. Harchol-Balter, et al prototyped SRPT scheduling on Apache web server and their evaluation showed the superiority of SRPT over PS [12] in terms of mean response time. To further improve the fairness of SRPT scheduling, Friedman, et al proposed Fair Sojourn Protocol (FSP) that combined SRPT with PS to trade off fairness with performance [10]. They concluded that FSP is both efficient in a strong sense (similar to SRPT), and fair, in the sense of guaranteeing that it weakly outperforms processor sharing (PS) for every job on any sample path.

All the previous research on size-based scheduling assumes accurate knowledge of job sizes a priori, which is not obtainable in many cases. To the best of our knowledge, no work has been done to characterize size-based policies with inaccurate scheduling information. Furthermore, most theoretical work assumes the M/G/1 queuing model due to its analytical simplicity. However, many computer systems

are better modeled with a G/G/n/m queuing model, where both job arrival and job size distribution are not Poisson, and there are n servers serving a queue with limited capacity m . A web server is an example. Previous research [15, 9] has shown that Poisson processes are valid only for modeling the arrival of user-initiated TCP sessions such as the arrival of TELNET connections and FTP connections. HTTP arrivals are not Poisson. Previous work [9] pointed out that the aggregated interarrival times of HTTP requests can be modeled with a heavy-tailed Weibull distribution.

There has been significant work on the G/G/n queuing model, and only the most closely related papers are listed here. Tabet-Aouel, et al gave analytic approximations for the mean sojourn time of P ($P \geq 2$) priority classes in a stable G/G/c/PR queue with general class interarrival and service time distributions and c ($c \geq 2$) parallel servers under pre-emptive resume (PR) scheduling [22]. Boxma, et al considered a G/G/1 queue in which the service time distribution and/or the interarrival time distribution has a heavy tail, i.e., a tail behavior like t^{-v} with $1 \leq v \leq 2$, such that the mean is finite but the variance is infinite. Depending on whether the service time distribution is heavier than that of the interarrival time distribution, they concluded that the stationary waiting time can be modeled as either a Kovalenko distribution or a negative exponential distribution [3]. However, we are unaware of any analytical results on G/G/n/m for SRPT or FSP scheduling in regimes where interarrival times and service times are heavy-tailed.

To characterize size-based policies with inaccurate scheduling information under more realistic queuing models such as G/G/n/m, we developed a simulator that can support PS, FSP, and SRPT in both M/G/1/m and G/G/n/m. The simulator operates on a trace of request arrivals, which can come either from real world traces or from a trace generator. The trace contains the request arrivals, the actual job sizes, and the estimated job sizes. Our trace generator allows us to control the correlation coefficient R between actual job size and estimated job size in a trace. Using the simulator and the trace generator, we study the mean response time and slowdown for SRPT and FSP scheduling policies with estimated job size information. Our simulation experiments with generated traces show that the performance of size-based policies is strongly related to the degree of correlation (R) between estimated job size and actual job size. For low values of R , these scheduling policies perform *worse* than PS, but given a reasonably good job size estimator, SRPT and FSP can outperform PS in both mean response time and slowdown.

3. Simulation setup

In this section, we describe our performance metrics, simulator validation, synthetic trace generation and simu-

Scheduling Policy	Description
PS	Processor Sharing scheduling policy.
SRPT	Ideal Shortest Remaining Processing Time scheduler, job sizes are known accurately a priori. The scheduler always choose the job with the shortest remaining size to serve first.
SRPT-E	Shortest Remaining Processing Time scheduler that uses estimated job sizes as scheduling information. The scheduler always chooses a job with the estimated shortest remaining size to serve first.
FSP	Ideal Fair Sojourn Protocol
FSP-E	Fair Sojourn Protocol that uses estimated job sizes as scheduling information.

Figure 1. Scheduling policies used in the paper.

Queuing Model	Description
$M/G/1/m$	Poisson arrival process; General job size distribution (Pareto and Weibull); Single server ; Limited queue capacity m .
$G/G/n/m$	General arrival process (Pareto and Weibull); General job size distribution (Pareto); n servers ; Limited queue capacity m .

Figure 2. Queuing models studied in the paper.

lation parameters.

Throughout the paper, we refer to the scheduling policies as listed in Figure 1, and the queuing models used as listed in Figure 2.

We set up the simulations driven by our synthetic traces to investigate how the degree of the correlation (R) between actual job size and estimated job size affects the performance of SRPT-E and FSP-E, where estimated job size is used as scheduling information, and compare them with a size-oblivious policy (PS), ideal SRPT, and FSP where actual job sizes are assumed to be known a priori.

Using the standard definition [19], we define the *load* on the queuing system as mean arrival rate divided by mean service rate throughout the rest of the paper. Unless otherwise stated, we fixed the load to be 0.9, making the queuing system reasonably heavily loaded.

3.1. Performance metrics

Our performance metrics are the mean response time and slowdown.

- Mean response time: Response time refers to the time span between a job’s arrival at and departure from the server. It is also known as sojourn time or turn around time. Mean response time has been used as a primary performance metric in queuing theory [19, 2, 11].
- Slowdown: Using the definition introduced by Bansal and Harchol-Balter [2], we define slowdown of a job as the ratio of its response time to its size (or service time). Slowdown is also referred to as normalized response time [2]. This metric is important because it reflects how long a job waits in the system relative to its

size, thus helps to evaluate unfairness. Under a fair policy like PS, all jobs experience the same slowdown.

3.2. Simulator

Our simulator supports both $M/G/1/m$ and $G/G/n/m$ queuing systems. It is driven by a trace in which each request contains the arrival time, actual job size, and estimated job size. We use synthetic traces generated with interarrival times from exponential and bounded Pareto distributions, actual job sizes from bounded Pareto distributions, and estimated job sizes also from bounded Paretos. In the synthetic traces, we directly control the correlation, R , between actual size and estimated size, as described later. Throughout the rest of the paper each simulation is repeated 20 times and we present the average.

Similar to Bansal and Harchol-Balter’s work [2], we concentrate on $M/G/1/m$ queuing model for the simulations presented in this paper. Simulations with $G/G/n/m$ queuing model show similar trends and the details can be found in our technical report [14]. Figure 4 shows the parameters of the bounded Pareto distributions used for the simulations shown in the rest of the paper. We used identical bounded Pareto distributions for both estimated job size and actual job size distributions as shown in Figure 2.

We validated our simulator by:

- Assuring that Little’s law is never violated on each run, using effective arrival rate as appropriate for limited queue capacity.
- Repeating the simulations described in Friedman and Henderson’s FSP paper [10]. We got nearly identical results under FSP, SRPT, and PS policies.

- Comparing our simulation results with the analytic results of Bansal and Harchol-Balter’s SRPT fairness paper [2]. Our simulation results are qualitatively consistent with theirs.

3.3. Controlling R in synthetic traces

Given some parametric distribution, e.g. exponential, and a target correlation coefficient R , we generate pairs of random numbers where each number of the pair is chosen from its required distribution and where the two numbers of the pair are correlated to degree R . To do this, we use a simplified Normal-To-Anything (NORTA) method. The basic ideas and proofs behind NORTA were developed by Cario and Nelson [5]. Given the distributions $dis_{estimatedsize}$ and $dis_{actualsize}$, our target correlation coefficient R and our sample size N , the following algorithm generates N pairs:

- 1 Set $\rho = R$
- 2 Generate two independent random numbers $x_1, x_2 \sim N(0, 1)$.
- 3 let $y_1 = x_1, y_2 = \rho \times x_1 + \sqrt{(1 - \rho^2)} \times x_2$
- 4 let $u_1 = NormCDF(y_1, 0, 1)$, $u_2 = NormCDF(y_2, 0, 1)$ where $NormCDF(y_i, 0, 1)$ is the CDF value of a standard normal distribution at y_i for $i = 1, 2$. It can be shown that $u_i \sim U[0, 1], i = 1, 2$
- 5 let $estimatedsize = F_{dis_{estimatedsize}}^{-1}(u_1)$, $actualsize = F_{dis_{actualsize}}^{-1}(u_2)$ where $F_{dis_{estimatedsize}}$, $F_{dis_{actualsize}}$ are the CDFs of our desired distributions for estimated size and actual size respectively. F_{dis}^{-1} is the inverse of F_{dis} .
- 6 Repeat steps 2-5 N times generating N pairs $\{(estimatedsize_j, actualsize_j)\}$. $\{estimatedsize_j, j = 1, \dots, N\}$ and $\{actualsize_j, j = 1, \dots, N\}$ are two correlated random numbers each following their own distributions.
- 7 Compute the correlation coefficient of $\{estimatedsize_j\}$, $\{actualsize_j\}$ and call it ρ_{temp} . If $\rho_{temp} > R$, then decrease ρ and go to step 2. If $\rho_{temp} < R$, then increase ρ and go to step 2. If $\rho_{temp} \approx R$ then stop.

Figure 3 gives some examples of estimated size/actual size pairs generated for different values of R .

To show the correctness of this algorithm, we can try following analysis: First, it is easy to see that $y_1, y_2 \sim N(0, 1)$ and $u_1, u_2 \sim U[0, 1]$, thus $estimatedsize \sim dis_{estimatedsize}$ and $actualsize \sim dis_{actualsize}$. Second, it can be shown that y_1 is correlated with y_2 and thus so is u_1 with u_2 . Intuitively it follows that $\{estimatedsize_j\}$ and $\{actualsize_j\}$ are correlated as well. Cario and Nelson showed that (1) ρ_{temp} is a nondecreasing continuous function of ρ , and (2) ρ_{temp} and ρ share the same sign. These properties guarantee the termination of the above simplified NORTA algorithm and let us bound the values of R that can be achieved by NORTA. If we sample ρ from 0 to 1, we can estimate the range of ρ_{temp} , producing a set of sets of pairs, ordered with increasing R as a side effect. This is exactly

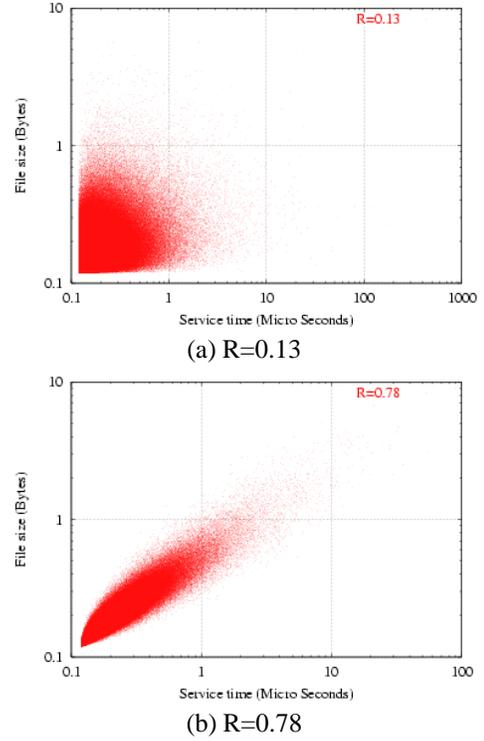


Figure 3. Examples of generated estimated size/actual size pairs.

how we generated correlated random pairs of file size and service time. Depending on the structures of different distributions, ρ_{temp} may not always take a full range of $[0, 1]$, which is why some of the results we show here have a restricted range of R .

In Section 4, we show the simulation results on mean response time. We study slowdown as a function of job size and correlation coefficient R in Section 5.

4. Simulation results on mean response time

To study the effects of the correlation R between actual job size and estimated job size on the performance of SRPT-E and FSP-E, we generated traces with controlled correlation as described in the previous section. We used bounded Pareto distributions for both actual job size and estimated job size. For the arrival process, we consider Poisson arrivals (exponential interarrival times), heavy tailed Pareto arrivals, and heavy tailed Weibull arrivals. For all the simulations of this section, the load (mean arrival rate divided by mean service rate) is 0.9, and the queue capacity is 5000. A single server is assumed. Multiple servers are similar to the single server case, hence the results are not shown here.

The scheduling policies used (SRPT, SRPT-E, FSP, FSP-E and PS) are described in Figure 1. Each graph data point

	α	Lower bound	Upper bound
Job size	2.0	0.1188	10^5

Figure 4. Parameters for Bounded Pareto Distribution.

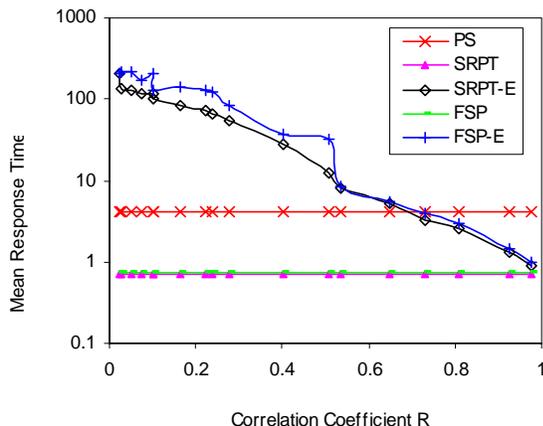


Figure 5. Mean sojourn time versus R , synthetic traces, M/G/1/m, Pareto service times, Poisson arrivals.

represents the average of 20 simulations, each of which has processed 0.5 million job requests.

Figure 5 shows the effects of R on the mean response time of different scheduling policies with a Poisson arrival process, corresponding to the M/G/1/m queuing model. The interarrival mean used to generate Poisson process is 0.264. Note that the Y axis is in log scale. Heavy-tailed Pareto and Weibull arrival process with the same job size distributions, corresponding to the G/G/1/m queuing model, show similar results and thus are omitted here.

As shown in Figure 5, SRPT-E results in lower mean response time than FSP-E does in most cases. The performance of SRPT-E and FSP-E increases quickly with increasing R . When R is very small, SRPT-E and FSP-E essentially behaves like a random scheduling policy, and it is worse than PS in mean response time. When R exceeds a threshold, SRPT-E and FSP-E performance exceed that of PS in both M/G/1/m and G/G/1/m. The threshold is about 0.7 in our simulations. We believe this threshold is a function of R and both distributions of job size and estimated job size. Beyond this point, SRPT-E and FSP-E’s performance increases quickly with increasing R . The figure clearly shows that SRPT performance is strongly tied to R , even at high values of R . Improvements in estimating actual job size can dramatically improve SRPT and FSP for a wide range of R .

The lack of accurate job size information has been an important reason why SRPT is not widely deployed [2, 20]. Our simulations show that a reasonably good job size estimator is required for SRPT-E and FSP-E to outperform PS in terms of mean response time.

5. Simulation results on slowdown

Fairness is one major concern when applying size-based scheduling policies such as SRPT in practice. Theoretical [2] and simulation [11] work has shown that the degree of unfairness under SRPT is very small. However, no work has been done to study the fairness issue when the scheduler doesn’t have accurate job size information. Like previous work [2, 11, 10], we use slowdown as fairness metric.

We evaluate the slowdown of SRPT-E and FSP-E as a function of job size and the correlation coefficient R . Figures 6 through 11 show the slowdown versus the percentile of the job size distribution, with R increasing from 0.0224 to 0.9778. Note that the Y axis is in log scale in all these figures. The job sizes are categorized into 100 bins with each bin containing one percentile of the job size distribution. Again the load of system is 0.9.

From Figures 6 and 7, we can clearly see that both SRPT-E and FSP-E perform very poorly compared to PS when correlation is weak. This comes as no surprise because poor estimation of job sizes would render these policies almost equivalent to random scheduling. Much longer delays are imposed on jobs across the board. However, as the estimates improve, i.e. the increase of R values, the SRPT-E curve moves downward. It begins to outperform PS at $R = 0.4022$ for small jobs. For SRPT-E at the level of $R = 0.5366$, jobs below the 30 percentile have lower slowdown than with PS. For FSP-E, at the level of $R = 0.5366$, the slowdown is close to that caused by PS.

When R increases to 0.7322, both SRPT-E and FSP-E perform better than PS in general. For SRPT-E, close to 93% of the jobs have slowdown smaller than that of PS, while for FSP-E, it appears that all jobs have lower slowdown than that of PS. When estimated size is highly correlated with actual job size, SRPT-E and FSP-E’s performance closely resembles that of ideal SRPT and FSP (Figure 11). Note that in most cases, the performance of ideal SRPT and FSP is very close and not clearly distinguished in the figures.

Based on the simulation results in Section 4 and this section, it is clear that both the mean response time and slowdown of size-based policies heavily depend on the correlation between actual job sizes and estimated job sizes. Although SRPT-E has a lower mean response time than FSP-E, it causes larger slowdowns when the correlation coefficient is larger than about 0.7.

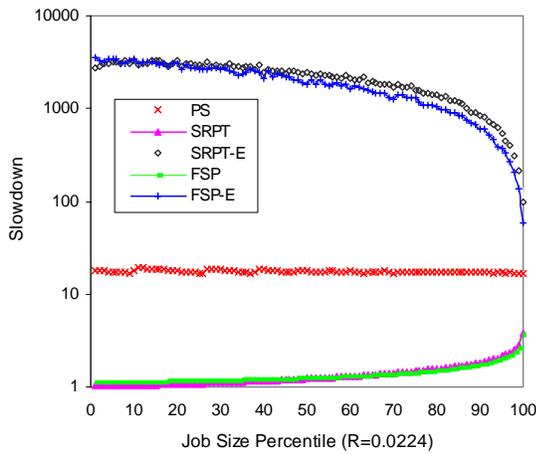


Figure 6. Slowdown as a function of the percentile of the job size distribution. Synthetic traces, $R = 0.0224$, $M/G/1/m$, Pareto service times, Poisson arrivals.

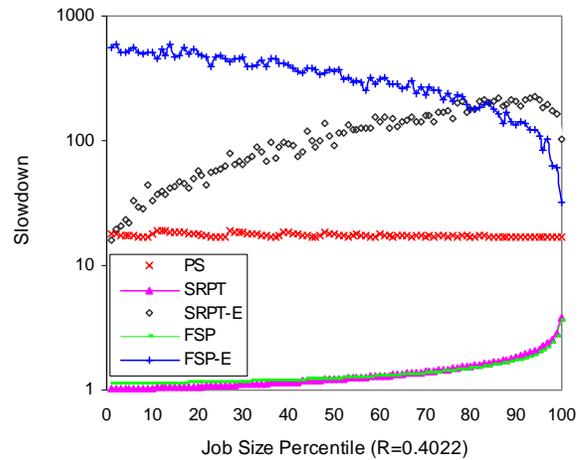


Figure 8. Slowdown as a function of the percentile of the job size distribution. Synthetic traces, $R = 0.4022$, $M/G/1/m$, Pareto service times, Poisson arrivals.

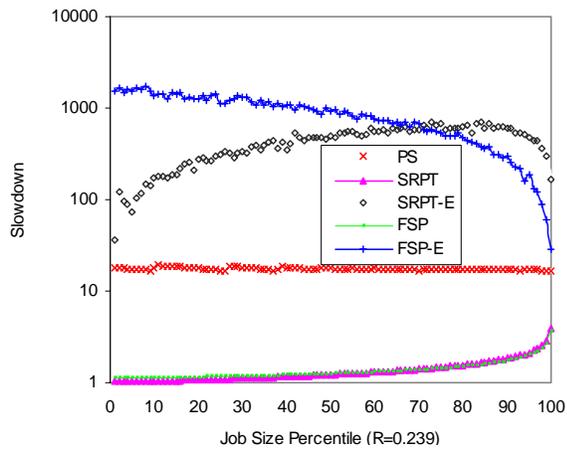


Figure 7. Slowdown as a function of the percentile of the job size distribution. Synthetic traces, $R = 0.239$, $M/G/1/m$, Pareto service times, Poisson arrivals.

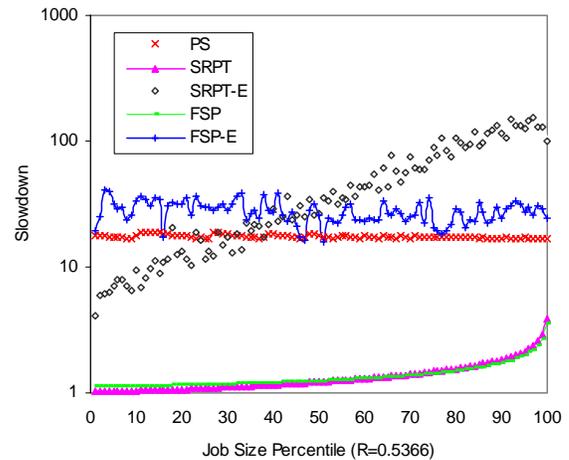


Figure 9. Slowdown as a function of the percentile of the job size distribution. Synthetic traces, $R = 0.5366$, $M/G/1/m$, Pareto service times, Poisson arrivals.

The simulations have clearly shown that high correlation between actual job sizes and estimated job sizes not only helps SRPT-E and FSP-E to reduce the mean response time, but also helps to achieve smaller slowdowns across various job sizes. More important than an accurate prediction for each job size is the order of the jobs in the queue—they need only be ordered by their size. A reasonably good estimator will enable SRPT-E and FSP-E to outperform PS in both mean response time and slowdown.

6. New applications

In this section, we describe two applications where the size-based policy could be successfully applied with a reasonably good job size estimator.

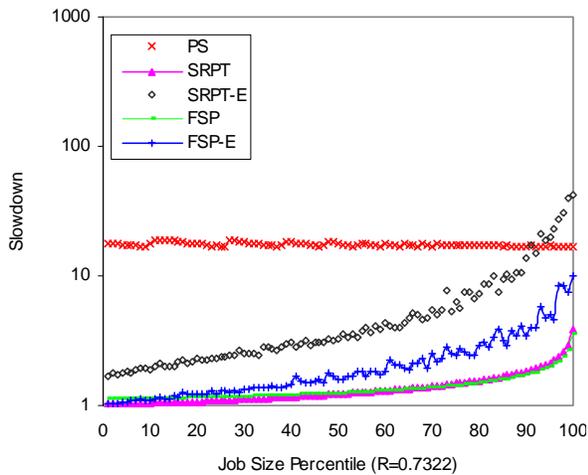


Figure 10. Slowdown as a function of the percentile of the job size distribution. Synthetic traces, $R = 0.7322$, $M/G/1/m$, Pareto service times, Poisson arrivals.

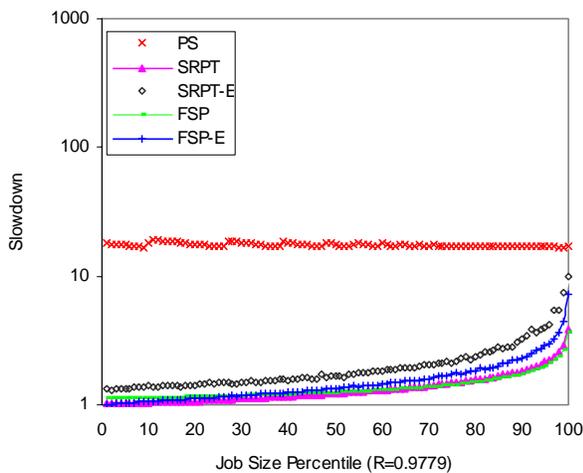


Figure 11. Slowdown as a function of the percentile of the job size distribution. Synthetic traces, $R = 0.9779$, $M/G/1/m$, Pareto service times, Poisson arrivals.

6.1. P2P server side scheduling

Peer-to-Peer systems have grown significantly in popularity over the last few years. In the context of peer-to-peer file sharing, research efforts have focused on routing, search, incentives, and a few other topics. But no one has looked at the server side scheduling problem except our own ongoing work [17].

The server side of current file sharing P2P applications such as Kazaa [1] is similar to a web server in that they both accept requests for files and send back the requested files. But there are significant differences between them from a scheduler's point of view. Requests to P2P nodes are typically a small chunk of the complete file, and the amount of data actually served is often a fraction of the request size. Therefore, the job sizes cannot be known a priori for the P2P application. Furthermore, while web servers can reasonably assume full control over resources, P2P applications are commonly configured with quite conservative upper bounds for each thread's resource consumption to minimize their impact on other applications.

To apply the SRPT-E or FSP-E scheduling policies in current P2P file sharing applications, we have to use estimated job sizes as scheduling information. Our work [17] shows that by using the requested data chunk size as the scheduling metric, we can get a mean response time that is only 30% of that of FCFS and 50% of that of PS. We are currently working on better job size estimators to further enhance the performance of the P2P applications.

6.2. Network backup system scheduling

Backups protect file systems from user errors, disk or other hardware failures, software errors that may corrupt the file system. The most common uses of backups are to restore files accidentally deleted by users and to recover from disk failures. As more and more data needs reliable and efficient backup, the backup techniques are becoming increasingly important.

Chervenak, et al [6] has shown that as the capacities of new storage devices continue to increase at a rate that is much faster than the speed of disk and tape access, it will take increasing long to read the contents of a disk drive and write them to a backup device. Therefore, it is very important to improve the efficiency of backup systems.

Most current backup systems such as Amanda [8, 7] allow concurrent backups in a networked computing environment, where multiple file systems are backed up in parallel to one or more backup systems. A network backup system is similar to a web server in that they both transfer a large number of files between multiple machines via network. However, unlike a web server, a network backup system doesn't know the job sizes a priori because of the use of the incremental backup scheme, which copies only those files that have been created or modified since a previous backup. Furthermore, incremental backup schemes that compute and store file differences are the extreme case, where job size is not known until after completion.

If SRPT or FSP scheduling can be applied on network backup systems, the mean response time could be lowered. However, because the job sizes cannot be obtained in advance, estimated job sizes have to be used for scheduling.

We speculate that history-based time series predictors can be applied to estimate the backup size for each machine, and job priority can be assigned accordingly. We are currently studying this possibility.

7. Conclusions and future work

Through simulations, we have evaluated the performance of size-based scheduling policies (SRPT and FSP, with PS for comparison), as a function of the correlation between actual job size and estimated job size. We found that SRPT and FSP's performance strongly depends on the correlation. When provided with weak correlation, SRPT and FSP can actually perform worse than PS, but given a reasonably good job size estimator, they can outperform PS in both mean response time and the slowdown. We also described two new applications of SRPT and FSP.

To generate correlated trace data for the simulation, we introduced a new random number pair generation technique, where each number of the pair is chosen from its required distribution and they are correlated to degree R . This technique can be very useful for simulation related research in this and other areas.

To the best of our knowledge, this work is the first to address the performance of size-based policies with inaccurate scheduling information. However, we believe that we have by no means completely addressed this area. More simulations and theoretical work are necessary for a better understanding. We are very interested in analytical results for the impact of inaccurate job size information, and simulation results for a wider range of workload characteristics. We are studying the impact of inaccurate job size information on web server performance [14]. Finally, we are studying SRPT-scheduled P2P systems [17].

References

- [1] Kazaa homepage. <http://www.kazaa.com>.
- [2] BANSAL, N., AND HARCHOL-BALTER, M. Analysis of SRPT scheduling: investigating unfairness. In *Proceedings of SIGMETRICS/Performance* (2001), pp. 279–290.
- [3] BOXMA, O., AND COHEN, J. Heavy-traffic analysis for the G/G/1 queue with heavy-tailed distributions. *Queueing Systems* 33 (1999), 177–204.
- [4] BUX, W. Analysis of a local-area bus system with controlled access. *IEEE Transactions on Computers* 32, 8 (1983), 760–763.
- [5] CARIO, M. C., AND NELSON, B. L. Numerical Methods for Fitting and Simulating Autoregressive-to-Anything Processes. *INFORMS Journal on Computing* 10, 1 (1998), 72–81.
- [6] CHERVENAK, A., VELLANKI, V., AND KURMAS, Z. Protecting file systems: A survey of backup techniques. In *Proceedings of the Joint NASA and IEEE Mass Storage Conference* (1998).
- [7] DA SILVA, J., AND GUDMUNDSSON, O. The amanda network backup system manager. In *Proceedings of the USENIX Systems Administration conference* (1993).
- [8] DA SILVA, J., GUDMUNDSSON, O., AND MOSSE, D. Performance of a parallel network backup manager. In *Proceedings of USENIX* (1992), pp. 17–26.
- [9] DENG, S. Empirical model of WWW document arrivals at access links. In *Proceedings of the IEEE International Conference on Communication* (June 1996).
- [10] FRIEDMAN, E. J., AND HENDERSON, S. G. Fairness and efficiency in web server protocols. In *Proceedings of SIGMETRICS/Performance* (2003).
- [11] GONG, M., AND WILLIAMSON, C. Quantifying the properties of srpt scheduling. In *Proceedings of IEEE MASCOTS* (2003).
- [12] HARCHOL-BALTER, M., SCHROEDER, B., BANSAL, N., AND AGRAWAL, M. Size-based scheduling to improve web performance. *ACM Transactions on Computer Systems (TOCS)* 21, 2 (May 2003).
- [13] HARCHOL-BALTER, M., SIGMAN, K., AND WIERMAN, A. Asymptotic convergence of scheduling policies with respect to slowdown. *Performance Evaluation* 49, 1/4 (2002).
- [14] LU, D., SHENG, H., AND DINDA, P. Effects and implications of file size/service time correlation on web server scheduling policies. Tech. Rep. NWU-CS-04-33, Northwestern University, Computer Science Department, April 2004.
- [15] PAXSON, V., AND FLOYD, S. Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking* 3, 3 (1995), 226–244.
- [16] PERERA, R. The variance of delay time in queueing system M/G/1 with optimal strategy SRPT. *Archiv fur Elektronik und Uebertragungstechnik* 47, 2 (1993), 110–114.
- [17] QIAO, Y., LU, D., BUSTAMANTE, F., AND DINDA, P. Looking at the server side of peer-to-peer systems. Tech. Rep. NWU-CS-04-37, Department of Computer Science, Northwestern University, March 2004.
- [18] SCHRAGE, L. E. A proof of the optimality of the shortest remaining processing time discipline. *Operations Research* 16 (1968), 678–690.
- [19] SCHRAGE, L. E., AND MILLER, L. W. The queue M/G/1 with the shortest remaining processing time discipline. *Operations Research* 14 (1966), 670–684.
- [20] SCHREIBER, F. Properties and applications of the optimal queueing strategy srpt - a survey. *Archiv fur Elektronik und Uebertragungstechnik* 47 (1993), 372–378.
- [21] SCHROEDER, B., AND HARCHOL-BALTER, M. Web servers under overload: How scheduling can help. Tech. Rep. CMU-CS-02-143, Carnegie Mellon School of Computer Science, June 2002.
- [22] TABEL, A. N., AND KOUVATSOS, D. D. On the approximation of the mean response times of priority classes in a stable G/G/C/PR queue. *Journal of the Operational Research Society* 43 (1992), 227–239.
- [23] TANENBAUM, A. *Modern Operating Systems*. Prentice Hall, 1995.