1 "All Can Win" Theorems – Why Biased Scheduling is Fair

The problem: The SYNC (Scheduling Your Network Connections) project is largely motivated by a single question:

> *Is it possible to reduce the expected response time of* **every** *job/request in a Web server, simply by changing the order in which we schedule the requests?*

In this section we focus on *static* requests, of the form "Get me a file," while Section 3 deals with requests involving *dynamic* content.

Our idea Our idea is simple: Traditionally, requests at a Web server are scheduled independently of their size. The requests are time-shared (processor-sharing, PS), with each request receiving a *fair share* of the web server resources. We propose to modify existing Web servers to implement *biased scheduling*, in which priority is given to *short* requests, or those requests which have *short remaining time*, in accordance with the well-known algorithm Shortest-Remaining-Processing-Time-first (SRPT).

The controversy It has long been known that SRPT has the lowest mean response time of any scheduling policy, for any arrival sequence and job sizes [38, 40]. Despite this fact, applications have shied away from using this policy for fear that SRPT "starves" big jobs [7, 41, 42, 39]. It is often stated that the huge average performance improvements of SRPT over other policies stem from the fact that SRPT *unfairly penalizes* the large jobs in order to help the small jobs. Conservation laws are often quoted in arguments that the performance of small jobs *cannot* be improved without hurting the large jobs.

The truth – analysis In [5] we show that this fear that SRPT penalizes large jobs as compared with PS is unfounded in many common situations, particularly heavier-tailed workloads. Consider for example an M/G/1 queue, where jobs sizes (service requirements) are distributed like Web requests, according to a Bounded-Pareto job size distribution with α-parameter near 1, exhibiting very high variability [6]. In this case, we find that *every single job*, including a job of the maximum possible size, prefers SRPT to PS in expectation (unless the load, ρ, is very close to 1), see Figure 1. This surprising result provably extends to *all* job size distributions when $\rho < 0.5$ [5], and, in the case of unbounded Pareto job size distributions, it extends to all values of load ρ [9, 24].

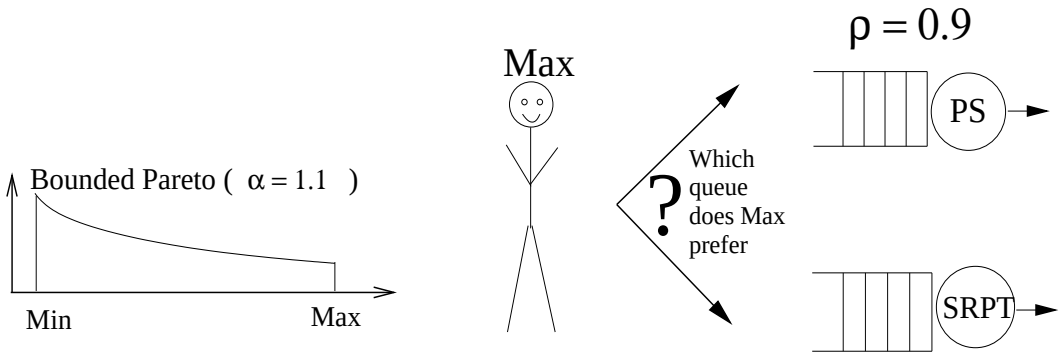

Figure 1: *All-can-win theorem. Jobs of all sizes prefer* SRPT *to* PS.

Furthermore, SRPT is not the only scheduling policy with good fairness properties. In [44] (award

paper) we develop the first theoretical framework for studying the fairness of all scheduling policies, and classify all common scheduling policies with respect to their fairness properties when compared with PS.

The truth – implementation Motivated by our theoretical results, in [23] we implement an approximation of SRPT scheduling of HTTP requests at an Apache web server. We modify the Linux kernel to change the order that the server's socket buffers are drained onto the server's access link (uplink); a priority is associated with each socket and this priority is increased dynamically as the remaining size (number of bytes left) in the file being retrieved goes down. We show that our SRPT server implementation significantly outperforms the unmodified (PS) server, under both a LAN setting and a WAN setting (network loss and delay), and under both open and partly open system configurations, using trace-based workloads. Figure 2(a) shows the significant improvement over SRPT over the unmodified server with respect to mean response time, as a function of load, where *load* is the ratio of the bandwidth needed by files requested to the total bandwidth available on the uplink. To evaluate unfairness, Figure 2(b) shows the mean response time as a function of request size. SRPT scheduling improves the mean response times of most requests by a factor of close to 10, while the mean response time for the largest size file only increases negligibly under SRPT scheduling (due to overhead in the socket switching implementation). These performance benefits come at no loss in byte throughput or job throughput. Practically, this says that a web server employing SRPT scheduling should be able to retrieve text files (smaller requests) 10 times faster under SRPT than under the unmodified, PS scheduling, while images (large requests) will not take longer than in the unmodified server because of the heavy-tailed property of web workloads.

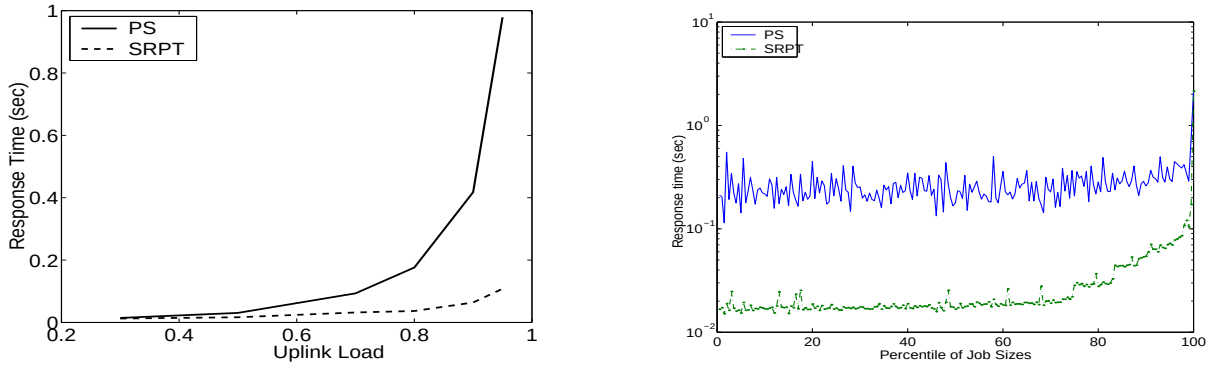

Figure 2: *(Left) Mean response time as a function of load for a web server in a LAN environment. Shown under SRPT modification, and under the unmodified, PS, server. (Right) Mean response time as a function of the size of the file requested, for load* 0.8*.*

Extensions/Generalizations In implementing SRPT scheduling, it became apparent that any real-world implementation is only an *approximation* of the idealized SRPT scheduling policy studied in queueing theory books. Real implementations sometimes only allow for coarse differentiation between jobs, or need to work without always knowing the remaining size. This discrepancy between theoretical SRPT and practical implementations motivated us to invent the notion of a *class of policies*, which we call the SMART policies, which is broad enough to include any policy following the general heuristic of biasing towards short jobs, and particularly, encompasses approximations implemented in practice. In [46] we prove that all SMART policies are *2-competitive*, and in [47], we show that the tail of response time of SMART policies matches that of SRPT. Our

implementation work has also shown us that *unpredictability in response times* is as important to users as *unfairness in response times*. To address this point, we extend our classification of fairness in scheduling policies [44] to a classification of predictability in response times, [45].

Impact – followup of others Until a few years ago, there were almost no scheduling papers at conferences like Sigmetrics. Stochastic scheduling analysis was considered by many to be "finished" around the time of the Conway, Maxwell, Miller book [11]. Our counterintuitive *theoretical* results in 2001 on fairness ([5]), which were quoted by many as "defying all conservation laws," backed by our kernel-level *implementation* of connection scheduling in web servers ([21, 23, 12, 22]), launched an entire industry of new theoretical and applied investigations into the power of scheduling, with a focus on fairness. By 2003 Sigmetrics devoted an entire session to the analysis of scheduling algorithms. In 2004 Sigmetrics devoted a whole session to "Scheduling and Unfairness," and there has been an entire session devoted to scheduling at Sigmetrics every year since. Recently, I was asked to put together a special issue for *Performance Evaluation Review* overviewing the new trends in this explosion in scheduling research, see [20].

Our SRPT-based scheduling for Web servers has been extended in many papers including, [10, 15, 18, 28, 27, 29, 30, 48], as well as in the SWIFT project [36]. Ernst Biersack's group has done considerable work on porting ideas from our SYNC project to routers, where he employs the LAS algorithm (Least-Attained-Service) which favors "young" flows (those which have sent few bytes thus far) [35, 34, 33]. Similar ideas involving favoring short or young flows are employed in [13, 49, 4, 16]. There have also been many papers studying the idea of multi-level age-based scheduling to favor short jobs, including papers by Aalto et al. [3, 1, 2] and Misra et al. [14].

Our definition of fairness has been applied to many new policies and in various computer systems designs, e.g., Rai, Biersack, et al. [35, 34], Gong and Williamson [18, 19], Misra and Rubenstein [15], Kherani and Nunez-Queija [25], and Friedman and Henderson [17]. Expanded definitions of fairness have also been developed, e.g., Levy and Raz [26] and Sandmann [37]. There have also been a great many new theoretical papers investigating SRPT and other SMART scheduling policies, e.g., [8, 32, 31, 43].

Funding This research was supported by (i) NSF CAREER - CCR-0133077 "The Impact of Resource Scheduling on Improving Server Performance"; (ii) Pittsburgh Digital Greenhouse grant from 2000-2001, "Connection Scheduling in Web Servers"; (iii) a gift from EMC^2 Corporation; (iv) IBM graduate student fellowship; (v) financial support from Cisco Systems, Network Appliance, and IBM.

References

[1] S. Aalto. M/G/1/MLPS compared with M/G/1/PS within service time distribution class imrl. *Mathematical Methods in Operations Research*, 64:309 – 325, 2006.

[2] S. Aalto and U. Ayesta. Mean delay analysis of multi-level processor-sharing disciplines. In *Proceedings of IEEE Infocom '06*, 2006.

[3] S. Aalto, U. Ayesta, and E. Nyberg. Two-level processor sharing scheduling disciplines: Mean delay analysis. In *Proceedings of the ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, June 2004.

[4] K. Avrachenkov, U. Ayesta, P. Brown, and N. Nyberg. Differentiation between short and long TCP flows: Predictability of the response time. In *Proc. of IEEE INFOCOM*, March 2004.

[5] N. Bansal and M. Harchol-Balter. Analysis of SRPT scheduling: Investigating unfairness. In *Proceedings of ACM SIGMETRICS*, 2001.

[6] P. Barford and M. Crovella. The surge traffic generator: Generating representative web workloads for network and server performance evaluation. In *In Proc. of the ACM SIGMETRICS*, 1998.

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

[8] S. C. Borst, O. Boxma, R. Nunez-Queija, and A. Zwart. The impact of the service discipline on delay asymptotics. *Performance Evaluation*, 54:175–206, 2003.

[9] P. Brown. Comparing FB and PS scheduling policies. *Performance Evaluation Review*, 34(3), 2006.

[10] H. Cha, J. Bang, and R. Ha. Static document scheduling with improved response time in http/1.1. In *Lecture Notes in Computer Science, vol. 2343*, pages 384–393. Springer-Verlag Heidelberg, August 2003.

[11] R. W. Conway, W. L. Maxwell, and L. W. Miller. *Theory of Scheduling*. Addison-Wesley Publishing Company, 1967.

[12] M. Crovella, B. Frangioso, and M. Harchol-Balter. Connection scheduling in web servers. In *USENIX Symposium on Internet Technologies and Systems*, pages 243–254, Boulder, CO, October 1999.

[13] S. Deb, A. Ganesh, and P. Key. Resource allocation between persistent and transient flows. *IEEE/ACM Trans. Netw.*, 13(2), 2005.

[14] H. Feng and V. Misra. Mixed scheduling disciplines for network flows (the optimality of FBPS). In *Proceedings of the Fifth Workshop on Mathematical performance Modeling and Analysis (MAMA 2003)*, June 2003.

[15] H. Feng, V. Misra, and D. Rubenstein. PBS: A unified priority-based scheduler. In *Proceedings of the ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, June 2007.

[16] D. Ferrero and G.Urvoy-Keller. Size-based scheduling to improve fairness and performance in 802.11 networks. Technical Report Technical Report RR 2125, Institut Eurecom, France, December 2006.

[17] E. Friedman and S. B. Henderson. Fairness and efficiency in web server protocols. In *Proceedings of the ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, June 2003.

[18] M. Gong and C. L. Williamson. Quantifying the properties of SRPT scheduling. In *11th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 2003)*, October 2003.

[19] M. Gong and C. L. Williamson. Simulation evaluation of hybrid srpt scheduling policies. In *12th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 2004)*, October 2004.

[20] M. Harchol-Balter. New perspectives on scheduling. *Performance Evaluation Review, Special Issue*, 35(1), April 2007.

[21] M. Harchol-Balter, N. Bansal, B. Schroeder, and M. Agarwal. SRPT scheduling for web servers. In *7th International Job Scheduling Strategies for Parallel Processing*, pages 11–20, 2001.

[22] M. Harchol-Balter and B. Schroeder. Kernel-level connection scheduling implementation source code. http://www-2.cs.cmu.edu/∼bianca/srpt/srpt.html/.

[23] M. Harchol-Balter, B. Schroeder, N. Bansal, and M. Agrawal. Size-based scheduling to improve web performance. *ACM Transactions on Computer Systems*, 21(2):207–233, May 2003.

[24] M. Harchol-Balter, K. Sigman, and A. Wierman. Asymptotic convergence of scheduling policies with respect to slowdown. In *PERFORMANCE 2002 Conference. IFIP WG 7.3 International Symposium on Computer Modeling, Measurement and Evaluation*, 2002.

[25] A. Kherani and R. Nunez-Queijia. TCP as an implemenentation of age-based scheduling: fairness and performance. In *Proceedings of* INFOCOM '06, 2006.

[26] H. Levy, D. Raz, and B. Avi-Itzhak. A resource-allocation queueing fairness measure. In *Proceedings of the ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, June 2004.

[27] D. Lu, P. Dinda, Y. Qiao, and H. Sheng. Effects and implications of file size/service time correlation on web server scheduling policies. In *Proc. of IEEE MASCOTS'05*, 2005.

[28] D. Lu, H. Sheng, and P. A. Dinda. Size-based scheduling policies with inaccurate scheduling information. In *Proc. of IEEE MASCOTS'04*, 2004.

[29] R. Mangharam, M. Demirhan, R. Rajkumar, and D. Raychaudhuri. Size matters: Size-based scheduling for MPEG-4 over wireless channels. In *SPIE & ACM Proceedings in Multimedia Computing and Networking*, pages 110–122, 2004.

[30] C. D. Murta and T. P. Corlassoli. Fastest connection first: A new scheduling policy for web servers. In *Proceedings of 18th International Teletraffic Congress (ITC)*, September 2003.

[31] M. Nuyens, A. Wierman, and A. Zwart. Preventing large sojourn times with SMART scheduling. *Operations Research, to appear*, 2007.

[32] M. Nuyens and A. Zwart. A large deviations analysis of the GI/G/1 SRPT queue. *Queueing Systems*, 54:85–97, 2006.

[33] I. Rai, E. W. Biersack, and G. Urvoy-Keller. Size-based scheduling to improve the performance of short TCP flows. *IEEE Network Magazine*, 19(1):12 – 17, 2005.

[34] I. Rai, M. Vernon, G. Urvoy-Keller, and E. W. Biersack. Performance models for las based scheduling disciplines ina packet switched network. In *Proceedings of the ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, June 2004.

[35] I. A. Rai, G. Urvoy-Keller, and E. Biersack. Analysis of las scheduling for job size distributions with high variance. In *Proceedings of the ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, June 2003.

[36] M. Rakwat and A. Kshemkayani. SWIFT: Scheduling in web servers for fast response time. In *Second IEEE International Symposium on Network Computing and Applications*, April 2003.

[37] W. Sandmann. A discrimination frequency based queueing fairness measure with regard to job seniority and service requirement. In *Next Generation Internet Networks*, April 2005.

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

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

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

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

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

[43] A. Wierman. On the effect of inexact size information in size based policies. *Performance Evaluation Review*, 34(3):21–23, 2006.

[44] A. Wierman and M. Harchol-Balter. Classifying scheduling policies with respect to unfairness in an M/GI/1. In *Proceedings of ACM SIGMETRICS*, San Diego, CA, June 2003. Best Student Paper Award.

[45] A. Wierman and M. Harchol-Balter. Classifying scheduling policies with respect to higher moments of conditional response time. In *ACM Sigmetrics 2005 Conference on Measurement and Modeling of Computer Systems*, 2005.

[46] A. Wierman and M. Harchol-Balter. Nearly insensitive bounds on SMART scheduling. In *ACM Sigmetrics 2005 Conference on Measurement and Modeling of Computer Systems*, 2005.

[47] C. W. Yang, A. Wierman, S. Shakkottai, and M. Harchol-Balter. Tail asymptotics for policies favoring short jobs in a many-flows regime. In *ACM Sigmetrics 2006 Conference on Measurement and Modeling of Computer Systems*, 2006.

[48] S. Yang and G. de Veciana. Size-based adaptive bandwidth allocation:optimizing the average QoS for elastic flows. In *Proceeedings of IEEE Infocomm '02*, 2002.

[49] J. Zhou and T. Yang. Selective early request termination for busy internet services. In *Proc. of WWW'06*, 2006.