5 Routing for Server Farms with Highly-Variable Job Sizes

The popularity of the server farm architecture stems from its price advantage (many slow servers are far cheaper than one fast one) and its flexibility (it is easy to scale capacity up and down). A server farm typically consists of a collection of host machines (servers) and a front-end high-speed router. Each incoming job is immediately dispatched via the router to one of the hosts. In supercomputing and manufacturing settings, the queue at each host is commonly served in First-Come-First-Served (FCFS) order, see Figure 1(a). The FCFS ordering stems from the fact that it is not easy to preempt jobs in these settings. By contrast, in the case of a Web server farm, the incoming HTTP requests are fully preemptible, and the scheduling of jobs at the hosts is best modeled by Processor-Sharing (PS), see Figure 1(b).

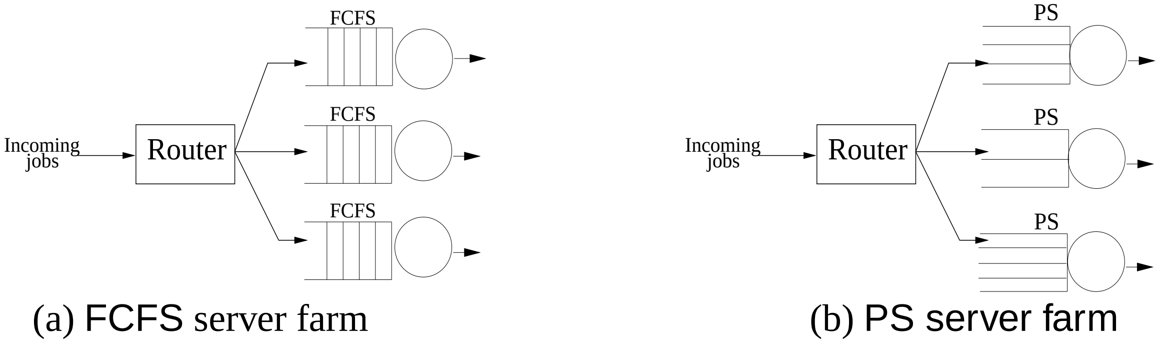

Figure 1: *Two server farm models.*

The problem The performance of any server farm depends critically on the *routing policy*, also known as the *task assignment policy*. This is the algorithm/rule for determining how to assign jobs to hosts. In this section we ask:

What is a good routing policy for server farms for minimizing mean response time?

Workload characterization The decision of which routing policy is best often depends on the *workload* (job size distribution). For many computer science applications the workload is highly variable with a heavy tail. In an award-winning paper [17, 18], we show that UNIX CPU lifetimes exhibit a highly variable Pareto distribution, as do supercomputing jobs [25, 26]. This distribution also holds for Web file sizes [3, 6] and IP flow durations [27].

Surprising results: FCFS server farms In [15, 16], we analyze different routing policies for FCFS server farms in the setting of highly-variable job size distributions, characteristic of the above computing workloads. We find several interesting results: First, common policies like Join-the-Shortest-Queue (JSQ) are poor performers, and even greedy policies, like routing jobs to the host with the Least-Work-Left (LWL), are not great performs. By contrast, a policy like our SITA (Size-Interval Task Assignment) policy [16], which segregates short jobs and long jobs into different queues, can outperform LWL by orders of magnitude. Also we find that, counter to common wisdom, it is preferable to purposely *unbalance load* rather than balance it, but the direction for unbalancing load (whether to underload or overload the short job host), is far from trivial and changes as a function of the α-parameter of the Pareto distribution [10]. In the case where *job sizes are unknown*, we can achieve almost the same level of performance as SITA by using our

TAGS (Task Assignment by Guessing Size) algorithm [13], which exploits statistical properties of the job size distribution. This work now appears in a patent [14].

Surprising results: PS server farms In [12] we study PS server farms. These perform very differently from FCFS farms. For PS farms, there is *no advantage* to isolating short jobs from long jobs, and *load balancing* is desirable. Also, JSQ is now an extremely good routing policy. We uncover a remarkable property of PS server farms: Unlike their FCFS counterparts, PS server farms with JSQ routing are *nearly insensitive to the variability of the job size distribution*. This property is unique to JSQ and does not hold for routing policies like LWL. We also provide the *first analysis of* JSQ *for PS server farms*. All prior analysis of JSQ, e.g. [2, 4, 5, 7, 8, 9, 11, 20, 22, 21, 19, 23, 24, 28, 29], assumes the FCFS server farm model, and even there very little is known beyond 2 servers and exponential job sizes. The intractability of JSQ stems from the fact that its analysis requires tracking the number of jobs at each server, which means that the Markov chain representation is unbounded in multiple dimensions. In [12], we introduce new analysis approach: *Single Queue Approximation* (SQA), whereby, we analyze a server farm with any number of servers by looking at just one queue of the server farm, in isolation from all the other queues, but where the arrival rate into that queue is conditional on the number of jobs at that queue. We prove that SQA is actually exact, for any exponential or degenerate hyperexponential job size distribution, with any variability.

Impact/Funding: I have given several keynote talks about my work on server farms. This work has also inspired several workshops at CMU, co-chaired with Alan Scheller-Wolf, such as the *WORkshop on Multiserver Scheduling (WORMS04)* [1], which was attended by famous researchers from around the globe, including Ward Whitt, Ed Coffman, Daryl Daley, Ruth Williams, Peter Glynn, Bill Massey, Ernst Biersack, R. Srikant, John Lehoczky, Sem Borst, Mark Squillante, Balaji Prabhakar, etc. This work is funded by NSF SMA/PDOS grant CCR-0615262 (2006-2009).

References

[1] WORkshop on Multiserver Scheduling (WORMS04). http://www.cs.cmu.edu/∼harchol/WORMS04/.

[2] I. Adan, J. Wessels, and W. Zijm. Analysis of the symmetric shortest queue problem. *Stochastic Models*, 6:691–713, 1990.

[3] 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.

[4] O. Boxma and J. Cohen. *Boundary value problems in queueing system analysis*. North Holland, 1983.

[5] B. Conolly. The autostrada queueing problem. *J. Appl. Prob.*, 21:394–403, 1984.

[6] M. E. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic: Evidence and possible causes. In *Proceedings of the 1996 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 160–169, May 1996.

[7] A. Ephremides, P. Varaiya, and J. Walrand. A simple dynamic routing problem. *IEEE Transac. on Auto. Cont.*, AC-25(4):690–693, 1980.

[8] L. Flatto and H. McKean. Two queues in parallel. *Communication on Pure and Applied Mathematics*, 30:255–263, 1977.

[9] G. Foschini and J. Salz. A basic dynamic routing problem and diffusion. *IEEE Trans. Comm.*, 26(3):320–328, 1978.

[10] P. Glynn, M. Harchol-Balter, and K. Ramanan. Optimal Cutoffs for Size-based Task Assignment in Heavy Traffic. Work in progress. Available from authors, 2007.

[11] W. Grassmann. Transient and steady state results for two parallel queues. *Omega*, 8:105–112, 1980.

[12] V. Gupta, M. Harchol-Balter, K. Sigman, and W. Whitt. Analysis of join-the-shortest-queue routing for web server farms. In *PERFORMANCE 2007 Conference. IFIP WG 7.3 International Symposium on Computer Modeling, Measurement and Evaluation*, Cologne, Germany, October 2007.

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

[14] M. Harchol-Balter and M. Crovella. Method and Apparatus for Assigning Tasks in a Distributed Server System. U.S. Patent Serial No. 6/223/205. Issued January, 2001.

[15] M. Harchol-Balter, M. Crovella, and C. Murta. On choosing a task assignment policy for a distributed server system. In *Lecture Notes in Computer Science, No. 1469: 10th International Conference on Modeling Techniques and Tools for Computer Performance Evaluation*, pages 231–242, 1998.

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

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

[18] M. Harchol-Balter and A. Downey. Exploiting process lifetime distributions for dynamic load balancing. *ACM Transactions on Computer Systems*, 15(3):253–285, August 1997.

[19] J. Kingman. Two similar queues in parallel. *Biometrika*, 48:1316–1323, 1961.

[20] C. Knessl, B. Matkowsky, Z. Schuss, and C. Tier. Two parallel $M/G/1$ queues where arrivals join the system with the smaller buffer content. *IEEE Trans. Comm.*, 35(11):1153–1158, 1987.

[21] H. Lin and C. Raghavendra. An analysis of the join the shortest queue (JSQ) policy. In *Proc. 12th Int'l Conf. Distributed Computing Systems*, pages 362–366, 1992.

[22] J. Lui, R. Muntz, and D. Towsley. Bounding the mean response time of the minimum expected delay routing policy: an algorithmic approach. *IEEE Trans. Comp.*, 44(12):1371–1382, 1995.

[23] R. Nelson and T. Philips. An approximation to the response time for shortest queue routing. *ACM Perf. Eval. Review*, 17:181–189, 1989.

[24] B. Rao and M. Posner. Algorithmic and approximation analyses of the shorter queue model. *Naval Research Logistics*, 34:381–398, 1987.

[25] B. Schroeder and M. Harchol-Balter. Evaluation of task assignment policies for supercomputing servers. In *Proceedings of 9th IEEE Symposium on High Performance Distributed Computing (HPDC '00)*, 2001.

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

[27] A. Shaikh, J. Rexford, and K. G. Shin. Load-sensitive routing of long-lived ip flows. In *Proceedings of SIGCOMM*, September 1999.

[28] R. W. Weber. On the optimal assignment of customers to parallel servers. *Journal of Applied Probability*, 15:406–413, 1978.

[29] W. Winston. Optimality of the shortest line discipline. *Journal of Applied Probability*, 14:181–189, 1977.