7 Priority Queueing and Capacity Planning for Server Farms

Multiserver priority queues Much of queueing theory is devoted to analyzing priority queues, where jobs (customers) are labeled and served in accordance with a priority scheme: high-priority jobs (H) preempt medium-priority jobs (M), which in turn preempt low-priority jobs (L) in the queue. Priority queueing comes up in a wide array of applications: sometimes users pay for their jobs to have higher priority; other times the priority of a job is artificially created, so as to maximize a company's profit by favoring big spenders [16, 26]. While priority queueing in a *single-server system* has been well understood since the 1950's [3], priority queueing in a *multi-server system*, see Figure 1(left), is far less tractable. Almost all papers analyzing multi-server priority queues are approximations, restricted to only two priority classes and exponential job size distributions, [14, 12, 21, 14, 12, 15, 20, 17, 20, 18, 6, 7, 5, 13]. For more than two priority classes, only coarse approximations exist, either based on approximating multi-server priority behavior by single-server priority behavior [2], or via aggregating priority classes [18, 21]. We ask:

> *What do per-class mean response times look like for a multi-server system? How do these compare with those for a single-server system?*

Difficulty/ Our approach What makes this problem so difficult is the need for a Markov chain which *grows unboundedly in m dimensions*, where m is the number of classes. Our approach is very different from all above approaches. We deploy *recursive dimensionality reduction*, RDR, which combines ideas from [27], [4], and [19]. The idea is to reduce an mD-infinite chain to a 1D-infinite chain, one dimension at a time. As each class is added, the effect of all the higher priority classes on the newly-added class is analyzed using a collection of busy periods, see [11].

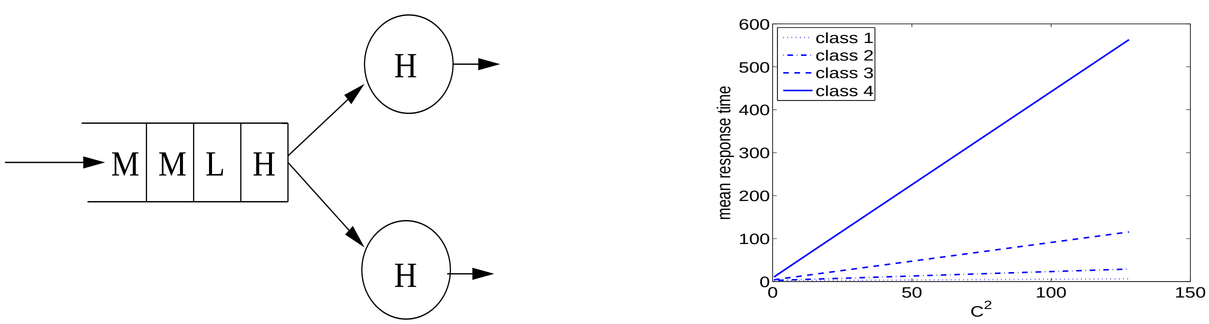


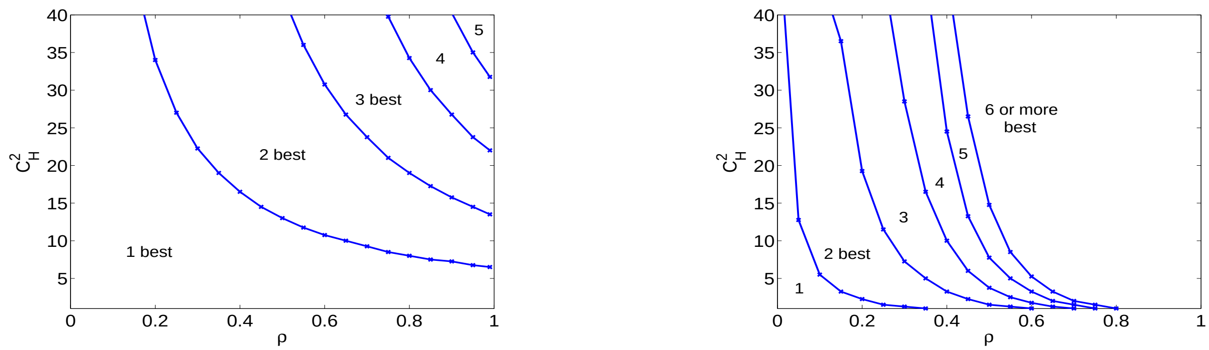
Figure 1: *(Left) Server farm where high-priority jobs served first. (Right) Two-server system, M/G/2, with 4 identical job classes and load $\rho = 0.8$: mean per-class response times as a function of job size variability (C^2).*

Our results RDR is the first technique to provide response time numbers for $m > 2$ priority classes under general job size distributions [11]. It is also a highly accurate method. Figure 1(right) shows results for per-class mean response times for 4 classes in an M/G/2, shown as a function of the variability of the job size distribution. It is interesting to note (not shown in figure) that these numbers are quite different from what would be obtained using a single server approximation of the system. A single (double-speed) server can perform far *worse* than a 2-server system when job size variability is high, since there is no way for small jobs to overtake large ones in a single-server system.

Capacity-planning problems The above observation prompts us to ask a capacity-planning question:

> *When is one fast server better than k slow servers, each running at $1/k$th the speed? It turns out that answers to questions like these depend greatly on how jobs are prioritized in the multiserver system.*

Our results In [28] we find that the optimal number of servers depends on ρ (high load implies more slow servers are better) and C^2 (more variability implies more slow servers are better). Interestingly, we find that when classes are prioritized *effectively*, with shorter jobs being given high priority so as to minimize mean response time, then the optimal solution points to fewer fast servers, see Figure 2(a), as compared with *poor prioritization*, where longer jobs are given high priority, Figure 2(b). Capacity planning is an extremely important problem in operations management. In [1] we look at the *dynamic staffing problem*, where staffers (or servers) are allowed to migrate to different queues as needed, and develop an even more general technique to handle that problem.



(a) Effective prioritization (b) Poor prioritization

Figure 2: *How many servers is best, as a function of variability of high priority jobs and load?*

Impact Dimensionality Reduction (DR), Recursive Dimensionality Reduction (RDR), and further generalizations thereof, have been applied to a long list of problems for which there was previously no way of deriving accurate performance numbers: [11, 10, 28, 1, 24, 25, 8, 23, 9, 22].

Funding This work is funded by an NSF SMA/PDOS grant CCR-0615262 (2006-2009).

References

[1] A. Bhandari, A. Scheller-Wolf, and M. Harchol-Balter. An exact and efficient algorithm for the constrained dynamic operator staffing problem for call centers. *Management Science, to appear*, 2007.

[2] A. Bondi and J. Buzen. The response times of priority classes under preemptive resume in M/G/m queues. In *ACM Sigmetrics*, pages 195–201, August 1984.

[3] A. Cobham. Priority assignments in waiting line problems. *Operations Research*, 2:70–76, 1954.

[4] J. D. P. Gaver. A waiting line with interrupted service, including priorities. *Journal of the Royal Statistical Society, Series B*, 24:73–90, 1962.

[5] W. Feng, M. Kawada, and K. Adachi. Analysis of a multiserver queue with two priority classes and (M,N)-threshold service schedule ii: preemptive priority. *Asia-Pacific Journal of Operations Research*, 18:101–124, 2001.

[6] H. Gail, S. Hantler, and B. Taylor. Analysis of a non-preemptive priority multiserver queue. *Advances in Applied Probability*, 20:852–879, 1988.

[7] H. Gail, S. Hantler, and B. Taylor. On a preemptive Markovian queues with multiple servers and two priority classes. *Mathematics of Operations Research*, 17:365–391, 1992.

[8] M. Harchol-Balter, C. Li, T. Osogami, A. Scheller-Wolf, and M. Squillante. Cycle stealing under immediate dispatch task assignment. In *15th ACM Symposium on Parallel Algorithms and Architectures*, pages 274–285, San Diego, CA, June 2003.

[9] M. Harchol-Balter, C. Li, T. Osogami, A. Scheller-Wolf, and M. Squillante. Task assignment with cycle stealing under central queue. In *23rd International Conference on Distributed Computing Systems*, pages 628–637, Providence, RI, May 2003.

[10] M. Harchol-Balter, T. Osogami, and A. Scheller-Wolf. Robustness of threshold policies in a beneficiary-donor model. *Performance Evaluation Review*, 33(2), 2005.

[11] M. Harchol-Balter, T. Osogami, A. Scheller-Wolf, and A. Wierman. Multi-server queueing systems with multiple priority classes. *QUESTA*, 51(3–4):331–360, 2005.

[12] E. Kao and K. Narayanan. Modeling a multiprocessor system with preemptive priorities. *Management Science*, 2:185–97, 1991.

[13] E. Kao and S. Wilson. Analysis of nonpreemptive priority queues with multiple servers and two priority classes. *European Journal of Operational Research*, 118:181–193, 1999.

[14] E. P. C. Kao and K. S. Narayanan. Computing steady-state probabilities of a nonpreeptive priority multiserver queue. *Journal on Computing*, 2(3):211 – 218, 1990.

[15] H. Leemans. *The Two-Class Two-Server Queue with Nonpreemptive Heterogeneous Priority Structures*. PhD thesis, K.U.Leuven, 1998.

[16] D. McWherter, B. Schroeder, N. Ailamaki, and M. Harchol-Balter. Improving preemptive prioritization via statistical characterization of OLTP locking. In *Proceedings of the 21st International Conference on Data Engineering*, San Francisco, CA, April 2005.

[17] D. Miller. Steady-state algorithmic analysis of M/M/c two-priority queues with heterogeneous servers. In R. L. Disney and T. J. Ott, editors, *Applied probability - Computer science, The Interface, volume II*, pages 207–222. Birkhauser, 1992.

[18] I. Mitrani and P. King. Multiprocessor systems with preemptive priorities. *Performance Evaluation*, 1:118–125, 1981.

[19] M. Neuts. Moment formulas for the markov renewal branching process. *Advances in Applied Probabilities*, 8:690–711, 1978.

[20] B. Ngo and H. Lee. Analysis of a pre-emptive priority M/M/c model with two types of customers and restriction. *Electronics Letters*, 26:1190–1192, 1990.

[21] T. Nishida. Approximate analysis for heterogeneous multiprocessor systems with priority jobs. *Performance Evaluation*, 15:77–88, 1992.

[22] T. Osogami. Analysis of Multi-server Systems via Dimensionality Reduction of Markov Chains. Ph.D. Thesis. Carnegie Mellon University, June 2005.

[23] T. Osogami, M. Harchol-Balter, and A. Scheller-Wolf. Analysis of cycle stealing with switching times and thresholds. In *Proceedings of ACM SIGMETRICS*, pages 184–195, San Diego, CA, June 2003.

[24] T. Osogami, M. Harchol-Balter, and A. Scheller-Wolf. Analysis of cycle stealing with switching times and thresholds. *Performance Evaluation*, 61(4):374–369, 2005.

[25] T. Osogami, M. Harchol-Balter, A. Scheller-Wolf, and L. Zhang. Exploring threshold-base policies for load sharing. In *Forty-second Annual Allerton Conference on Communication, Control, and Computing*, University of Illinois, Urbana-Champaign, October 2004.

[26] B. Schroeder, M. Harchol-Balter, A. Iyengar, E. Nahum, and A. Wierman. How to determine a good multi-programming level for external scheduling. In *Proceedings of the 22nd International Conference on Data Engineering*, Atlanta, GA, April 2006.

[27] M. Squillante, F. Wang, and M. Papaefthymiou. Stochastic analysis of gang scheduling in parallel and distributed environments. *Performance Evaluation*, 27:273–396, 1996.

[28] A. Wierman, T. Osogami, M. Harchol-Balter, and A. Scheller-Wolf. How many servers are best in a dual-priority M/PH/k system? *Performance Evaluation*, 2006.