

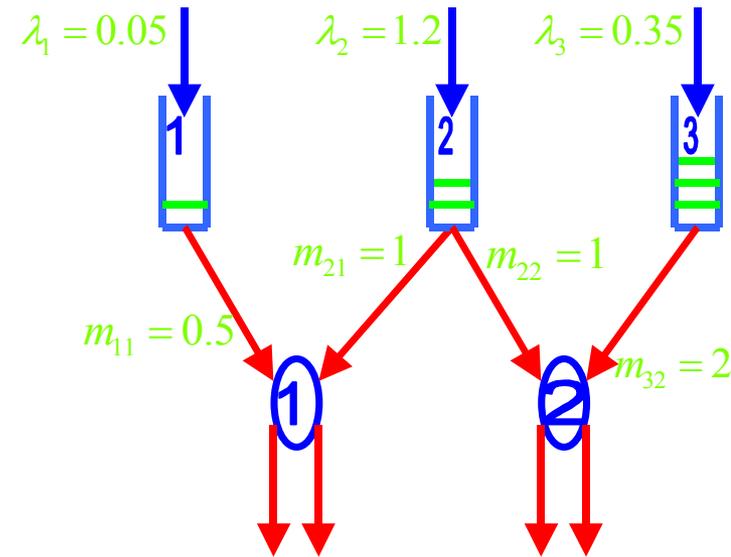
Dynamic Scheduling for Parallel Server Systems

R. J. Williams

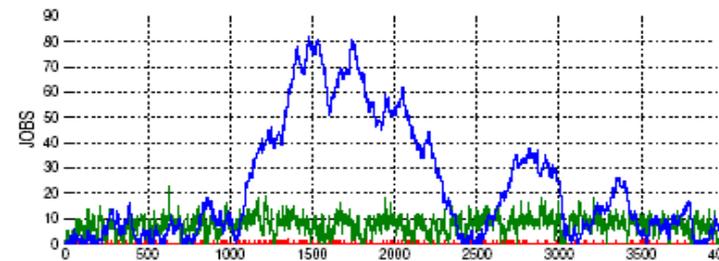
Parallel server queuing systems are stochastic systems that perform one-pass processing of jobs of multiple types using a bank of servers operating in parallel. Jobs that cannot be served immediately wait in buffers (one for each job type). Such systems arise in a variety of application contexts in computer science, manufacturing and the service industry. Often servers in these systems have overlapping capabilities or are cross-trained so that some buffers can receive service from any one of a certain subset of the servers. The terms flexible server or skill-based routing are often associated with such systems. A simple three-buffer, two-server example of such a system is shown at right with average arrival rates for the buffers indicated by $\lambda_i, i=1,2,3$ and the mean service time for processing jobs from buffer i using server j indicated by m_{ij} .

In general, for a parallel server system, it is a challenging problem to schedule waiting jobs to available servers so as to optimize a measure of performance such as the cumulative cost of holding jobs in buffers. For a diffusion analogue of this control problem, Harrison and Lopez (*Heavy traffic resource pooling in parallel-server systems*, *Queueing Systems*, 33 (1999), 339-368) showed that, assuming a condition on the average data called the complete resource pooling condition, there is a control for the diffusion problem under which the servers cooperate most efficiently to act as a single pooled processing resource or "superserver". Harrison and Lopez conjectured that there should be an interpretation of this optimal diffusion control as a scheduling policy for the original parallel server system that performs well when the system is heavily loaded. Such an interpretation is not immediate as individual jobs have become infinitesimal in the diffusion model.

Williams proposed such an interpretation in the form of a threshold policy for the original parallel server system and in joint work with her graduate student S. L. Bell proved that this threshold policy is asymptotically optimal in the heavy traffic limit under the complete resource pooling condition. At right a simulation of queue lengths as a function of time is shown when this general policy is used on the example shown above with exponential interarrival and service times and holding costs per unit time proportional to total queue length. For relevant publications, see <http://www.math.ucsd.edu/~williams/research.html>



Simulation with dynamic threshold priority policy: server 1 gives priority to buffer 1, server 2 gives priority to buffer 2, except when queue 2 goes below a threshold of size 10



Queue lengths for buffer 1 ---, buffer 2 ---, buffer 3 --- versus time