15-829 **HW 6 – Due start of class 11/04**

Homework is due at the start of class on 11/04. Please turn in a hardcopy. It can be handwritten. Please show all computations.

Reading from your textbook: Chapters 28-33 (you can skip the math and just concentrate on the text and intuitions).

Exercises:

1. **Analysis Problem: Comparing NP-Prio and P-Prio and FCFS**

   Assume that you have a file server that downloads files. In this problem you will analytically compare 3 policies for scheduling file downloads.

   (i) Non-preemptive Priority (NP-Prio)

   (ii) Preemptive Priority (P-Prio)

   (iii) First-Come-First-Served (FCFS).

   A "job" consists of a file download. The "size" (service requirement) of a job is best approximated by the size of the file being downloaded. Files come in 3 classes:

   - Class 1 files can have any size between 0 and ∞, but tend to be small on average: $S_1 \sim \text{Exp}(1)$.
   - Class 2 files range from moderate to larger size: $S_2 \sim \text{Uniform}(10, 60)$.
   - Class 3 files are all large, ranging in size from 50 to 1000: $S_3 \sim \text{BoundedPareto}(k = 50, p = 1000, \alpha = 1.2)$.

   The file size distribution is

   $$S = \begin{cases} S_1 & \text{w.p. } \frac{1}{3} \\ S_2 & \text{w.p. } \frac{1}{3} \\ S_3 & \text{w.p. } \frac{1}{3} \end{cases}$$

   The system load is $\rho = 0.8$ and the arrival process is a Poisson process.

   (a) What is ρ_1, the load consisting of class 1 jobs? What is ρ_2? What is ρ_3?

   (b) For each of the three policies, what is $\mathbf{E}[T(1)]$, $\mathbf{E}[T(2)]$, $\mathbf{E}[T(3)]$, and overall $\mathbf{E}[T]$? Here $\mathbf{E}[T(k)]$ denotes the mean response time of class k jobs.

   (c) How are the three policies ranked with respect to overall mean response time, $\mathbf{E}[T]$? Why intuitively does this make sense?

   (d) For which policy is $\mathbf{E}[T(1)]$ lowest? Why is this?

   (e) For which policy is $\mathbf{E}[T(3)]$ highest? Why is this?

2. **Simulation Problem: Scheduling to reduce variability**

Shashank decides to build a web server to serve file requests. Arrivals into Shashank's web server come from all over the world and are well modeled by a Poisson process. Shashank has measured his file size distribution and found that it's well-modeled by

$$S \sim \text{BoundedPareto}(k = 1333.333, p = 10^{10}, \alpha = 1.8) \ .$$

(a) In Shashank's initial attempt, he schedules requests in FCFS order. Simulate Shashank's FCFS queue for loads $\rho = 0.1, 0.2, 0.3, \ldots, 0.9$ and plot $\mathbf{E}[T]^{\text{FCFS}}$ as a function of ρ.

(b) Shashank reasons that mean response time is too high because C_S^2 is high. What is C_S^2? To remedy the problem, Shashank decides to instead schedule requests in Processor-Sharing (PS) order. Simulate Shashank's PS queue for loads $\rho = 0.1, 0.2, 0.3, \ldots, 0.9$ and plot $\mathbf{E}[T]^{\text{PS}}$ as a function of ρ. Put this on the same plot as your FCFS results.

(c) Shashank is still not pleased with his response times at high load. He decides to try scheduling requests in Shortest-Remaining-Processing-Time (SRPT) order. Simulate Shashank's SRPT queue for loads $\rho = 0.1, 0.2, 0.3, \ldots, 0.9$ and plot $\mathbf{E}[T]^{\text{SRPT}}$ as a function of ρ. Put this on the same plot as your FCFS and PS results.

(d) Unfortunately, Shashank's system only allows 3 priority levels, so he can't implement true SRPT. Instead, as an approximation to SRPT he labels:

- Files with Remaining size < 4000 are Priority 1;
- Files with $4000 <$ Remaining size < 20000 are Priority 2;
- Files with Remaining size > 20000 are Priority 3.

Note that a file of size 30000 will start out as priority 3; but after the first 10000 bytes are sent it will transition to priority 2; and when the file has only 4000 bytes remaining to be sent, it will transition to priority 1. We call this scheduling policy Bucketed-SRPT (BSRPT). Simulate Shashank's BSRPT queue for loads $\rho = 0.1, 0.2, 0.3, \ldots, 0.9$ and plot $\mathbf{E}[T]^{\text{BSRPT}}$ as a function of ρ. Put this on the same plot as your FCFS, PS, and SRPT results.

(e) How did BSRPT do? Is this what you expected? Explain.

Here's a paper that actually implements these policies for a web server and also studies issues of unfairness:
Harchol-Balter, Schroeder, Bansal, Agrawal "Size-based scheduling to improve web performance" *ACM Transactions on Computer Systems*, Vol. 21, No. 2. pp. 207-233, 2003.