ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

HS 2010 Prof. R. Wattenhofer / J. Smula, T. Langner

Discrete Event Systems

Exercise Sheet 9

1 Waiting Problem

In the lecture, we presented the following theorem without proof.

Theorem. *Let $X_1, \dots X_n$ be independently and exponentially distributed random variables with parameters $\lambda_1, \dots, \lambda_n$. Then $X := \min\{X_1, \dots, X_n\}$ is exponentially distributed with parameter $\lambda = \lambda_1 + \cdots + \lambda_n$.*

a) Prove the theorem for $n = 2$.

Hint: If Y is exponentially distributed with parameter λ, we have

$$\Pr[Y > t] = e^{-\lambda t} \ .$$

b) Use the result from **a)** to prove the theorem for arbitrary n.

2 Queuing Networks

Customers of the Internet Service Provider *RedWindow* who have problems with their Internet access, can call a hot-line. There, a customer must first talk to a dispatcher. The dispatcher is very moody and with probability p_d, he kicks people out of the line. However, with probability $1 - p_d$, a customer is connected to a technician. The technician can solve the problem with probability p_t. However, if he can not solve it, he claims that the problem is the fault of the monopolistic modem producer *Beep*. Thus, with probability $1 - p_t$, the customer has to call *Beep*. Unfortunately, the agent at Beep can solve the problem only with probability p_b. With probability $1 - p_b$, the customer is told that RedWindow is the source of the problem, and hence the customer is connected back to the dispatcher of RedWindow. And so on and so forth...

In the following, we assume that a customer calling RedWindow for the second time experiences exactly the same success probabilities as in the first round. Let now the arrival times of the *direct* (i.e., not reconnected) calls to RedWindow be Poisson distributed with parameter λ. Moreover, assume that the technician of RedWindow and the agent of Beep do not get additional (direct) calls. The service times of the dispatcher, the technician and the agent are exponentially distributed with parameter μ_d (dispatcher), μ_t (technician) and μ_b (Beep agent). If the dispatcher, the technician or the agent are occupied, the customer is put into the waiting line of the corresponding person.

a) Model the situation using the techniques from the lecture.

b) Describe the arrival rate of the phone calls at the technician of RedWindow as a function of p_d, p_t, p_b and λ!

c) How long is a customer in the waiting queue of the technician after he has been forwarded from the dispatcher until he is eventually served (on average)?

d) Now assume that $p_d = 1/6$, $p_t = 1/5$, $p_b = 1/4$, and $\lambda = 5$ per hour. Moreover, let $\mu_d = 20$ per hour, $\mu_t = 10$ per hour, and $\mu_b = 10$ per hour. Compute the expected number of customers in the system (of both RedWindow and Beep together)! What is the expected time a customer is in the system?

e) The technician of RedWindow is quite lazy and hence tells the dispatcher to kick customers out of the line with such a high probability that he only gets one call per hour. Compute the new p_d assuming that all other parameters stay the same!

3 Bin Packing

In order to finance their living, Jasmin and Tobias work at the assembly line of a production firm. Since they usually are mentally occupied with their intellectual duties from the PhD studies, physically stressed from selling ice cream in the sun all day and tired from the night watch shifts at the bank, their task is very simple: They have to pick the items delivered on the assembly line, put them into a bin and close the bin.

Now assume that there are n items of size $s_i \leq 1$ while the bins have size 1. Moreover, assume that the algorithm used by Tobias and Jasmin is a very simple one: The items are handled in order of their arrival and put into a bin as long as there is enough space left. If an item arrives that does not fit into the bin anymore, they close the bin and start with a new empty bin.

Calculate the competitive ratio with respect to the total number of bins Jasmin and Tobias need compared to an offline algorithm which distributes the items optimally among the bins.

4 Paging

Paging plays an important role in almost every computer system. Typically, there is a fast cache which allows fast access, but has limited space. On the other hand, access to the disk is slow, but space is plentiful.

We consider a simple system in which the cache has enough space to store three pages. Given a request for a page p_i, the system must make p_i available in the cache. If p_i is already in the cache (called a *hit*), the system does nothing. Otherwise (a *miss*), the system incurs a *page fault* and must copy the page p_i from the disk to one of the three locations in the cache. In case all three slots in the cache are already occupied with other pages, the system is faced with the problem of which page to evict from the cache in order to make space for p_i.

In our model, we have to pay a price of 1 for each page fault, while accessing a page that is already in the cache is for free. In this exercise, we analyze the competitiveness of several well known paging strategies.

a) Consider the following paging strategies. Which of them are competitive and which are not?

- FIFO (First-in/First-out): Replace the page that has been in the cache longest.
- LFU (Least Frequently Used): Replace the page that has been requested the smallest number of times since entering the fast memory.
- LIFO (Last-in/First-out): Replace the page most recently moved to the cache.
- LRU (Least Recently Used): When eviction is necessary, replace the page whose most recent request was the earliest.
- FWF (Flush When Full): Whenever there is a page fault and there is no space left in the cache, evict *all* pages currently in the cache.

Hint: An online strategy is *not competitive* if its competitive ratio is unbounded.

b) All the above strategies are deterministic. Prove a lower bound on the competitive ratio of any deterministic paging strategy.