Homework #1 (100 Points)

Due 03:00pm on Tuesday 02/02/2016 using gsubmit

Feel free to make assumptions, if you feel that such assumptions are justified or necessary. Please state your assumptions clearly. Unreasonable assumptions lead to unreasonable grades!

Read the course syllabus available on-line. Submitting a solution to this problem set (or any future problem sets) implies that you read and agree to abide by the rules and stipulations set forth in the syllabus, including rules concerning code of conduct, missed exams, and policies on late/lost homework assignments, *etc.*

1. A system consists of 1 CPU, 1 Disk, and 1 Network Interface. A web server running on this system consists of a process that waits for an HTTP request. Once the request is received, the process services it by fetching the requested file from disk (disk I/O) and then by sending the file content to the client (network I/O). Assume that serving a request consists of the following phases:
   - I. Process uses the CPU for 2 msec (parse HTTP request and start disk I/O) // CPU is busy here
   - II. Process waits for the Disk for 8 msec (fetch the file) // Disk is busy here
   - III. Process uses the CPU for 2 msec (store content in memory and start network I/O) // CPU is busy here
   - IV. Process waits for the network for 6 msec (send file content over the network to the client) // Network is busy here
   - V. Process uses the CPU for 1 msec (cleanup) // CPU is busy here

   Assuming that there can only be one process in the system, i.e., a batch processing system, with a multiprogramming level (MPL) = 1, and assuming that there are always requests for the server to service, i.e., once the process is done with a request, it can immediately start on the next, answer the following questions:
   - a. What are the utilizations of the CPU, disk, and network?
   - b. What is the capacity of the web server?

   To improve resource utilization, you decided to use multiple processes to serve more than one request at a time. In other words, you decided to go for MPL > 1. To do so, you designed the web server to consist of a set of N processes, each of which serving one of the requests coming into the system. Again, assuming that there are plenty of requests for each process to service, and that the system has been running for a long time (i.e., it reached some steady state) answer the following questions:
   - c. For N=2, what is the utilization of the CPU, disk, and network?
   - d. For N=2, what is the capacity of the web server?
   - e. Repeat parts c. and d. for N=3

   f. Assuming that you can set N to any value you want, what is the maximum capacity of this web server?
   g. What is the bottleneck that limits the capacity of the web server in your answer for part f. above?

2. Using Amdahl law, answer the following questions:
   a. Calculate the speedup in the web server (described in problem 1) if the speed of the CPU is r-fold the original speed, which means that the amount of time to execute a snippet of code is reduced by a factor of 1/r. Plot the speedup of the web server as a function of r.
   b. What is the theoretical limit for the speedup of the web server, i.e., assuming that the CPU can be infinitely fast?
   c. Repeat part a and b for the other two devices (the disk and the network).

   From a system's efficiency perspective, it is desirable to equalize the utilization of all resources used by the web server.

   d. What is the minimum speedup for the various devices that will achieve this goal?
   e. What is the resulting capacity of the web server under the improvement in part d?

3. To speed up memory access, caching is typically used. A memory cache is a small but fast memory where data recently accessed is kept in anticipation of future references. When an access is made, if the data is in the cache, then it is returned quickly. This is called a cache hit, otherwise main memory is accessed and the access is said to be a cache miss. For the purposes of this problem, assume that the latency of the main memory is eight times the latency of the cache (i.e., if an access to the cache takes one unit of time, then access to main memory would take 8 units of time). Now, consider two possible optimizations for a memory system. The first will cut the latency of the main memory by 50%, whereas the second would cut the latency of the cache by 20%. Answer the following questions.

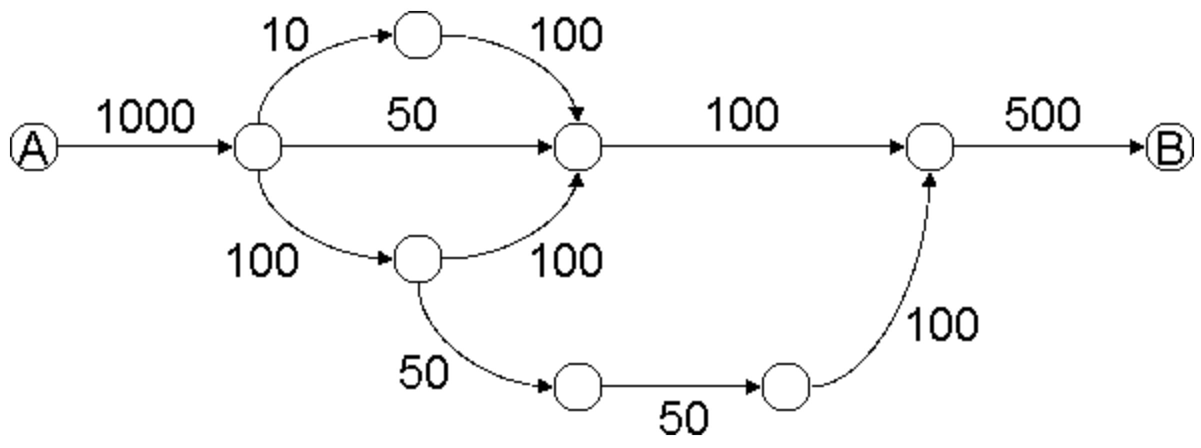
   a. If the cache hit rate is 95% what speedup is achieved under each one of the two optimizations under consideration (separately)?
   b. Under what condition on the cache hit rate would you select each one of the two optimization under consideration (separately)?
   c. What speedup is achieved if both optimizations are adopted. Your answer should be a function of the hit rate, which you should take as a variable *h*.

4. Two nodes in an ad-hoc network communicate over a path that consists of H "hops", where a hop is the link between two intermediate notes in the ad-hoc network. The capacity between any two nodes along that path was rated at 10Mbps. Data is sent over that path using "packets" each of which consisting of 1,500 bytes, with 150 of these bytes used to carry meta information such as routing information, checksums, protocol settings, etc. (i.e., they are not part of the "payload"). Due to cross traffic and power cycling on intermediate nodes, it was determined that a fraction P of all packets sent on any one link are lost.
   a. What is the effective throughput for that path? Your answer should be a function of "H" and "P".
   b. What is the effective throughput when H is 1 and when P=0.01?

c. What is the maximum loss probability on a single hop that will result in an effective bandwidth of more than 6Mbps for an average-length path L?

5. The problem of finding the capacity of general networks is known as the graph MaxFlow problem (http://en.wikipedia.org/wiki/Maximum_flow_problem), which can be solved iteratively by (i) finding a feasible "path" between the source and destination and computing the "flow" capacity of that path (e.g., the path going from A to B through the three nodes on the straight line between them has a flow capacity, f1=50) and (ii) reducing the capacities of the links on that path accordingly, eliminating any link whose capacity reaches 0, to obtain a new graph (called the residual graph) and (iii) repeating prior steps to get additional flow capacities, f2, f3, ... until all the paths between A and B are exhausted. The capacity of the original network (i.e., the solution to the maxflow problem) is simply the sum C=f1+f2+... [Hint: You can save yourself some work by first reducing the graph using the serial/parallel reductions described in some of the problems in the first set of exercises.]

   a. Apply this algorithm to compute the capacity of the network shown below -- show your work by drawing all residual graphs you come up with.
   b. You are asked to increase the capacity of the network by upgrading the capacity of a single existing link in the network. Suggest an algorithm that achieves that goal and apply it to the graph above.
   c. How much higher could the network capacity be as a result of upgrading a single network link? Explain your logic.
   d. Imagine you are engaged in a cyber warfare and you want to inflict the most performance degradation by taking down one link other than the ingress and egress links (e.g., by subjecting it to a Denial of Service attack). Which link would you choose and why?


10
100
A
1000
50
100
500
B
100
100
100
50
50