Homework Number

Erik Brakke

February 2, 2016

Answer 1

a. CPU utilization: $\frac{5ms\text{Busy Time}}{19ms\text{Total time}} = .263$ 26.3%
   Disk utilization: $\frac{8ms}{19ms} = .421$ 42.1%
   Network utilization: $\frac{6ms}{19ms} = .316$ 31.6%

b. $1req/.019s = 52.63req/second$

c. Total time: 28ms (Assuming first process wins in ties)
   CPU utilization: $\frac{12ms}{28ms} = .429$ 42.9%
   Disk utilization: $\frac{21ms}{28ms} = .750$ 75.0%
   Network utilization: $\frac{12ms}{28ms} = .429$ 42.9%

d. $2.26req/.028s = 80.71req/second$ The .26 came from the 7 seconds of a new process that is handled on the first thread

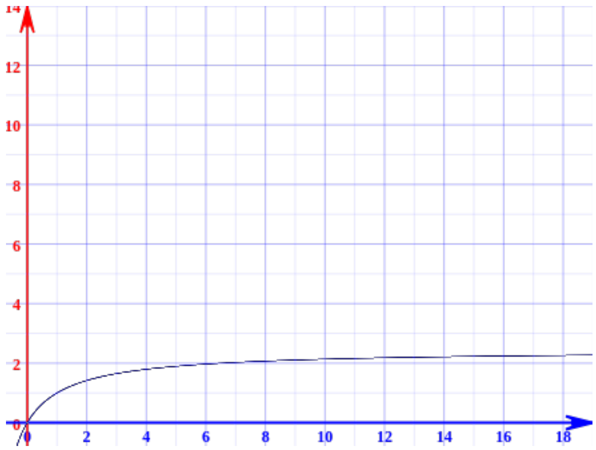
e. Total time: 35ms (Same assumption as above)
   CPU utilization: $\frac{20ms}{35ms} = .571$ 57.1%
   Disk utilization: $\frac{32ms}{35ms} = .914$ 91.4%
   Network utilization: $\frac{18ms}{35ms} = .514$ 51.4%
   Capacity: $3.63req/.035s = 103.71req/second$

f. The max capacity is 110 requests per second. (4 threads, and 4.84 requests being processed)

g. The bottleneck is the disk

Answer 2

a. MPL = 4, CPU utilization = 26/44, $speedup = \frac{1}{1-\frac{26}{44}+\frac{26/44}{r}}$



b. Take $r \rightarrow \inf$, $\frac{1}{1-(26/44)} = 2.44$

c. Disk util = 42/44, Network util = 24/44, $speedup_{disk} = \frac{1}{1-\frac{42}{44}} = 22$, $speedup_{network} = \frac{1}{1-\frac{24}{44}} = 2.2$

Answer 3

a. cache util = .95
Scenario 1 $speedup = \frac{1}{1-.05+.05/1.5} = 1.01$
Scenario 2 $speedup = \frac{1}{1-.95+.95/1.2} = 1.19$

b. We want to find the intersection of the two speedups
$\frac{1}{1-h+h/1.2} = \frac{1}{1-(1-h)+(1-h)/1.5}$ This happens when $h = .667$
So when hit rate ¡ 66.7%, take the memory speedup, else take the cache speedup

c. $speedup = \frac{1}{(h/1.2)+(h/1.5)}$

Answer 4

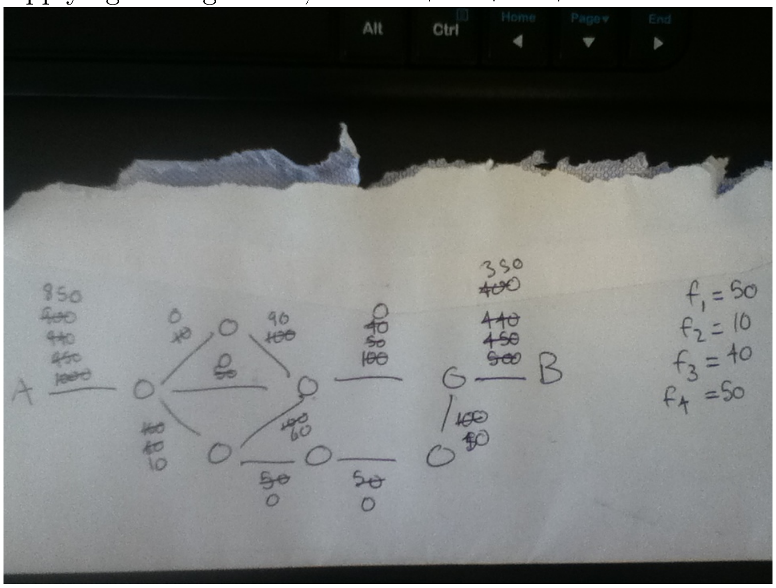
a. $150/15000 = .1$ wasted for overhead $T(H,P) = 9*(1-P)^H$

b. $T(1,.01) = 9*(.99) = 8.91Mpbs$

c. $6 < 9*(1-P)^H$
$.66 < (1-P)^H$
$\sqrt[H]{.66} < 1-P$
$P < 1-\sqrt[H]{.66}$

Answer 5

a. Applying the algorithm, $C = 50 + 10 + 40 + 50 = 150$



b. $b =$ all the bottlenecks of each path
Run the max flow algorithm again.
$u =$ the first edge not in b to reach 0
If u does not exist, then upgrading any of the bottlenecks will increase the capcity of of the system
If u does exist, then upgrading u will increase the capacity of the system Applying this above, we can see that u is the edge connecting the node with 3 arrows in, to the one with 2 arrows in (cap = 100)
We can increase that capcity of u to 110 for a total capacity of 160 for the network

c. The maximum capacity of upgrading a single link in this system is 160. The reason is that the only link that is limiting the network capcity is the link found above with the algorithm. Because there are 3 paths through that link, it's capcity is used up, while there are still other paths with capacity that could use that link. Incrasing this link will allow for other links with useable capacity to use this link.

d. I would take down the link that was found by the algorithm above. Taking this link down would create only 1 useable link, giving the network a capcity of 50 (compared to the previous 150). Because a this is the internal link that the most paths rely on, it has the most damage to being taken out of the system