Assignment #6

EEL4713

Peter Borenstein

May 4th , 2012

Introduction

The focus of this assignment is to investigate the benefits of various cashe sizes. The assignment requires the use of the simulation programs SESC and Cacti to compare cashe misses and execution time for various block sizes. The cashe is fast and expensive memory located closest to a processor. Since the cashe is expensive, most of a computer's memory is stored is slower and lager disks. We assign groups of words into blocks. Many sections of memory are assigned the same block. Switching out blocks takes time. Two guiding principles of locality are if a processor uses a word, it will likely reuse it again soon, and if a processor uses a word, it will likely need a word with a nearby address next.

We worked with a split cashe. The was an instruction and a data memory cashe. Various sies were tried for the instruction cash first.

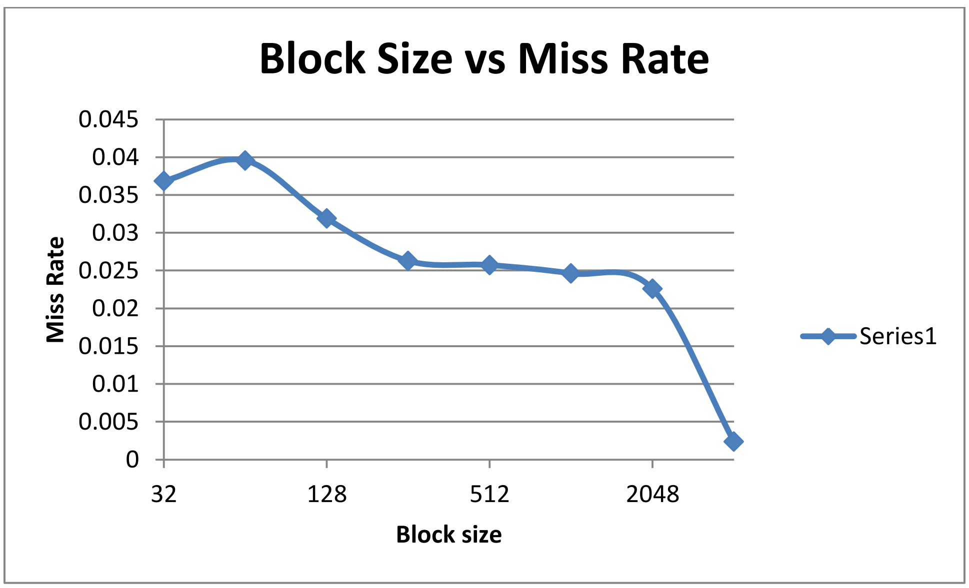

Figure 1 block size vs miss rate

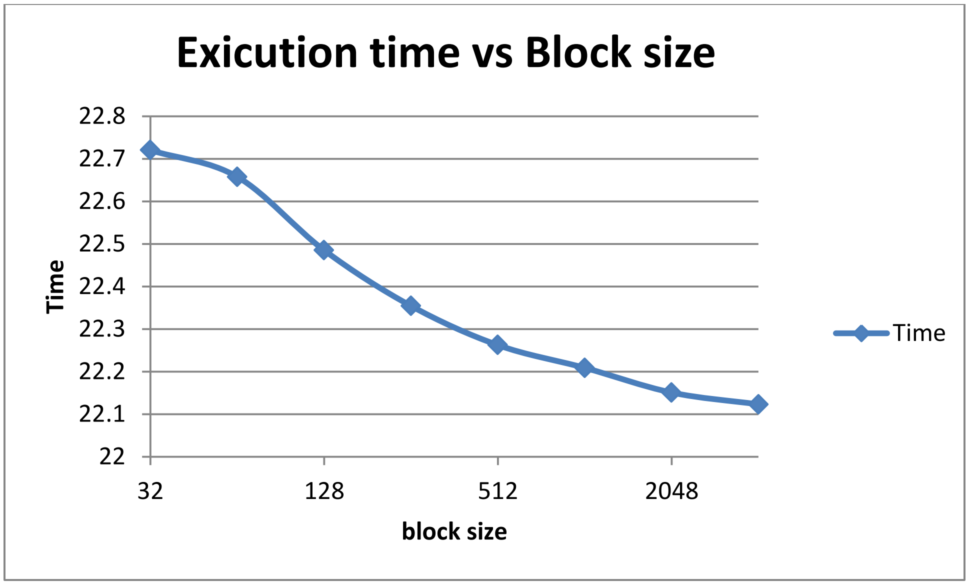

Figure 2block size vs exicution time

Block sizes 32, 64, 128, 512, 1024, 1048 and 4096 bytes were simulated. It can be observed that the miss rate decreases with the block size. This is expected. Spatial locality tells us that if we use an item of memory, we are likely to need nearby addresses. The larger block will store many nearby words. A larger block will also require a shorter index and tag address for a direct mapped cashe.

The execution time does not necessarily guarantee a faster execution time. A cashe miss for a larger block size requires more memory to be moved into the block upon a miss. Figure 2 shows us that the execution time is leveling off despite figure1 showing us that the miss rate continues to decrease.

I have the blue book.
6.3.1 a) 11ms+0.5/7200*60+1/34000+1/60000=11+4.2+0.03+0.02ms=15.25ms
b)9ms+0.5/7200*60+1/30000+1/62500=9+4.2+0.03+0.0016=13.23ms

6.3.2 a) 11ms+0.5/7200*60+2/34000+2/60000=11+4.2+0.06+0.02ms=15.28ms
b)9ms+0.5/7200*60+2/30000+2/62500=9+4.2+0.07+0.0032=13.30ms

6.3.3 The dominant factor for each disk is the seek time. It takes over half the time required in each disk. I would focus any improvements on the seek time.

6.15.1 a) FEFE Xor A387 Xor F345 Xor FF00=513C
b) AB9C Xor 0098 Xor 00FF Xor 2FFFF=8404

6.15.2a)FEFE Xor 00FF Xor 4582=BB83
b) AB9C Xor F457 Xor A387=FC4C

6.15.3 RAID 4 is more efficient. It has one less logic operation. However, RAID 4 has to read the old D0 value off the disk. RAIN 3 would be preferable if we have to do rapid writes. We would not want to spend as much time reading and writing involved with RAID 4.

6.15.4 RAID 5 spreads the parity bits across the disks. Since the parity bits are read and written to on every write, the parity disk in RAID 4 is a bottleneck. RAID 5 would be more populare on a system that does many writes at once, like on a server.

6.18.1 a) AFR=1000*8760/1000000=8.76 or 0.867%
b) AFR= 1000*10512/1500000=7.008 or 0.7%

6.18.2 This question isn't clear. There are two ways to view the problem. I will do both to ensure I get points.

First method (assume the AFR stays high after the first month):

a)0.867*3=2.6% for the first 5 years, 5.2% sixth year, 10.4% the 7th, 20.8% in the 8th, 41.6% in the 9 and 83.2% in the 10th.

(0.026*5+0.052+0.104)*1000=286 disks will need to be replace in 7 years.

(0.026*5+0.052+0.104+0.208+0.416+0.832)*1000=1742 disks will need to be replace in 10 years. This indicates some disk will be changed out twice.

b)0.7*3=2.1% for the first 5 years, 4.2% sixth year, 8.4% the 7th, 16.8% in the 8th, 33.6% in the 9th and 67.2% in the 10th.

(0.021*5+0.042+0.084)*1000=231 disks will need to be replace in 7 years.

(0. 021*5+0.042+0.084+0.168+0.336+0.672)*1000=1403 disks will need to be replace in 10 years. This

Second method (assume the AFR goes back to normal after the first month:

a)0.867*3=2.6% for the first month, 0.867% the first 4 years, 1.73% the 5th year, 3.5% the 6^{th}, 6.9% in the 7^{th}, 13.9% in the 8^{th} , 27.7% in the 9^{th} and 55.5% in the 10^{th}.

((0.026+0.00867*11)/12+0.00867*3+0.0173+0.035+0.069)*1000=157 disks will need to be replace in 7 years.

((0.026+0.00867*11)/12+0.00867*3+0.0173+0.035+0.069+0.139+0.277+0.555)*1000=1128 disks will need to be replace in 10 years. This indicates some disk will be changed out twice.

b)0.7*3=2.1% for the first month, 0.7% for the first 4 years, 1.4% the 5th year, 2.8% the 6^{th}, 5.6% in the 7^{th}, 11.2% in the 8^{th}, 22.4% in the 9^{th}, and 44.8% in the 10^{th}.

((0.021+0.007*11)/12+0.007*3+0.014+0.028+0.056)*1000=127 disks will need to be replace in 7 years.

((0.021+0.007*11)/12+0.007*3+0.014+0.028+0.056+0.112+0.224+0.448)*1000=911 disks will need to be replace in 10 years. This indicates some disk will be changed out twice.