CS 620: Homework 2

Carmen St. Jean

September 12, 2012

1. *(1) Give a key advantage of the memory/storage hierarchy model.*

   The key advantage of the memory/storage hierarchy model is that it gives you efficiency. You gain fast access of cheap storage thanks to the cache and main memory rather than having to read or write directly from the disk.

2. *(1) Give a key disadvantage of the memory/storage hierarchy model.*

   The key disadvantage of the memory/storage hierarchy model is that it requires the operating system to be very complex. This makes understanding or designing an operating system non-trivial.

3. *(1) What is the key performance advantage of multiprogramming?*

   Multiprogramming increases CPU usage by organizing jobs so that the CPU is always busy since a single user cannot keep either the CPU or I/O devices busy at all times. For example, the CPU is idle while one job waits for an I/O operation to complete. With multiprogramming, the CPU can be used to do another job to prevent the CPU from being idle. Keeping several jobs in memory makes it so the CPU is never idle and increases the performance of the computer as a whole.

4. *(2) Consider the following code:*

```
for (i = 0; i < 20; i++)
   for (j = 0; j < 10; j++)
      a[i] = a[i] * j;
```

   (a) *Give 1 example of spatial locality in the code.*

   Spatial locality, also called sequential access, indicates that when one area of memory is accessed, then areas of memory nearby may soon be accessed. An example of this in the code is how the

elements of the `a` array are referenced (with the outer loop). Since arrays are usually stored so that its elements are in contiguous memory addresses, `[0]` will be next to `a[1]`, which is next to `a[2]`, and so forth. Therefore, as `i` is incremented, the area in memory where `a[i]` is located is very near to where the previous instruction looked in memory.

(b) *Give 1 example of temporal locality in the code.*

Temporal locality, also called rereference data, indicates that when one area of memory is accessed, then it is likely to be accessed again in the near future. An example of this in this code is how `a[i]` is referenced ten times in a row (with the inner loop). As `j` is incremented, the same `a[i]` is referenced repeatedly until `i` is incremented.

5. *(5) What is a distributed system? Differentiate between distributed systems and*

A distributed system is a collection of physically separate computer systems that are networked to provide users with access to the various resources that the system maintains.

(a) *clusters;*

A cluster system is usually many computers sharing storing that are linked by a local-area network (LAN). A distributed system is different because it has no shared storage. Clusters are usually made of colocated computers, while a distributed system may have machines spread far apart.

(b) *supercomputers;*

A supercomputer is usually the combined use of hundreds or thousands of processors for complex computational. A supercomputer is often many machines located in a single room dedicated to a single task or group of related tasks, but a distributed system involves distributed machines, as the name implies, for robustness or reliability.

(c) *on-line classes;*

An on-line course is an example of a distributed system. Every student most likely has to use a web interface and/or a plug-in to view the content of the course that is hosted by some web server. The teacher too must use some kind of web interface and/or a plug-in to upload content to the web server. Students may also be

submitting assignments or taking quizzes through the interface so that they also upload content to the web server, which in turn is downloaded by the teacher. In this sense, the on-line course is a distributed system where teachers, students, and the server hosting all content are nodes.

(d) *peer-to-peer system;*

A peer-to-peer system is a network where any computer can act as either a client or a server for the other computers to share resources without a central server. Distributed systems may have a central server or machines which take on a master role compared to others which may be more like slaves. Generally, peer-to-peer systems have the same abilities and can negotiate between themselves. A peer-to-peer system involves machines that are located physically far apart and the users of the system can connect or disconnect at the will, as would be the case with something like Napster, while the distributed system tends to be known designated machines that would most likely be connected at all times.

6. *(2) The cache management software (replacement policy, prefetch technique, write policy) impacts the hit rate of the cache. Give 2 other factors that impact on the performance of a cache.*

Concerning distributed systems, the replication policy of the cache is a factor that impacts the performance of a cache. Replication policy is when the data is generated once and then copied to the other machines in the distributed system. Copying data to all of the nodes of your system will ensure that every node is up-to-date, but the action of copying the data everywhere also lowers your performance in general. However, saving time by replicating the data on only some nodes is problematic in its own way if one of the nodes left out needs access to that data.

Memory fragmentation can impact the performance of the cache. For example, say you are writing a C program and call `malloc` for a very large array. If `malloc` allocates discontiguous areas of memory, then you cannot use sequential access to its fullest extent because you've lost spatial locality. After one access of the array, prefetching will store sequential areas of memory in the cache, but the sequential areas may not be part of the array since the allocation was discontiguous. Therefore the prefetching will negatively impact your performance since a lot of time was devoted to caching a large amount of data that wasn't

even part of the array.

7. *Given that cat = 10ns; mat = 200ns; dat = 5ms. Suppose a program with 500 instructions is executed. Assume that 200 instructions hit in the cache, 200 instructions hit in MM, and the rest are loaded from disk.*

   (a) *(1.5) What is the average instruction access time?*
   The average instruction time is 1000130 ns (or 1.00013 ms):

$$\begin{aligned} t &= \frac{200}{500} \cdot 10 + \frac{200}{500} \cdot (100 + 200) + \frac{100}{500} \cdot (100 + 200 + 5 \cdot 10^6) \\ &= 0.4 \cdot 10 + 0.4 \cdot 210 + 0.2 \cdot 5000210 \\ &= 4 + 84 + 1000042 \\ &= 1000130 \end{aligned}$$

   (b) *(3) What is the cache hit rate and the memory hit rate?*
   Let h_c be the cache hit rate and h_m be the memory hit rate:

$$h_c = \frac{200}{500} = 0.4 = 40\%$$
$$h_m = \frac{200}{300} = 0.6667 = 66.67\%$$

   The cache hit rate is 40% and the memory hit rate is 66.67%.

   (c) *(2) Write the equation computing average access time using the cache hit rate and the memory hit rate.*
   Let *cat* be the cache access time, *mat* be the memory access time, and *dat* be the disk access time. Then t, the average access time, can be expressed as:

$$t = h_c \cdot cat + (1 - h_c)[h_m \cdot (cat + mat) + (1 - h_m)(cat + mat + dat)]$$

   (d) *(1.5) compute the average instruction access time using the hit rates you computed above? (If your answer to a. and d. are dissimilar, youre doing something wrong.)*

$$\begin{aligned} t &= h_c \cdot cat + (1 - h_c)[h_m \dot{(}cat + mat) + (1 - h_m)(cat + mat + dat)] \\ &= 0.4 \cdot 10 + (1 - 0.4)[(0.6667)(210) + (1 - 0.6667)(5000210)] \\ &= 4 + (0.6)[140 + (0.3333)(5000210)] \\ &= 4 + (0.6)[140 + 1666736.6667] \end{aligned}$$

$$
\begin{aligned}
&= 4 + 0.6 \cdot 1666876.6667 \\
&= 4 + 1000126 \\
&= 1000130
\end{aligned}
$$

8. *(2) During a performance study of a caching system, the average access time was measured with and without prefetching with the same set of programs and data (i.e., workload). It was found that the average instruction access time increased with prefetching. What are possible reasons for the performance drop with prefetching?*

   A possible reason prefetching caused a performance drop was that the programs executed carried out very simple tasks that required very little data access and so the prefetching cached significantly more data than was even necessary for those tasks.

   Another possible reason is that the programs executed for such a long time that the data in the cache was replaced before it was used the second time (or other subsequent times), resulting in cache misses. In this case, the time that had been devoted to caching the data counts against you.

9. *(3) If the average access time is 10% greater than the cache access time, what is the hit rate h?*

   Let t be the average access time, *cat* be the cache access time, and *mat* be the memory access time. Since the average access time is 10% greater than the cache access time, $t = 1.1cat$. Therefore:

$$
\begin{aligned}
t &= h \cdot cat + (1 - h) \cdot (cat + mat) \\
1.1\ cat &= h \cdot cat + (1 - h) \cdot (cat + mat) \\
1.1\ cat &= h \cdot cat + cat + mat - h \cdot cat - h \cdot mat \\
1.1\ cat &= h \cdot cat - h \cdot cat + cat + mat - h \cdot mat \\
1.1\ cat &= cat + mat - h \cdot mat \\
0.1\ cat - mat &= -h \cdot mat \\
h &= \frac{0.1cat - mat}{-mat} \\
h &= \frac{-cat}{10\ mat} + 1
\end{aligned}
$$