CpSc-418 **Midterm** Oct. 23, 2001

HINT: Brief answers are good. Show your work for calculations. Sentence form answers should be short and to the point.

1. (5 points) What is your name? **Mark Greenstreet**

2. (35 points) **AFRAID**

   (a) (5 points) What problem does the paper address?

   RAID-5 arrays have poor performance for writing small blocks because each write requires four disk accesses (two reads followd by two writes). The paper describes a variant on RAID that achieves performance comparable to non-RAID for small-writes.

   (b) (10 points) What are three key observations that lead to the solution presented in the paper?

   i. Disks are sufficiently reliable that RAID-5 is over kill.

   ii. File system traffic is bursty.

   iii. Users accept the idea that data may be lost if the file was written just before a crash.

   (c) (5 points) What is the solution presented in the paper?

   For small writes, just the data is written initially. This makes the data in that stripe temporarily unredundant. The parity is updated during the next idle time of the disk array.

   (d) (5 points) What is "parity lag"?

   Parity lag is the time between when data is written and when the parity for its stripe is updated. AFRAID uses parity lag to trade-off slightly reduced reliability for better performance. Reliability is reduced because data can be lost if a crash occurs between writing the data and updating the parity. Performance is increased because small-writes only have the latency of one disk access.

   (e) (10 points) What is Amdahl's Law? Amdahl's Law is usually used to describe performance. State an equivalent law about reliability. What does this have to do with the AFRAID paper?

   Amdahl's Law (original):

   Amdahl's Law describes how the overall performance of a system increases if the performance of one aspect is increased. Amdahl's Law is expressed by the formula:

$$\text{Speed-Up}_{overall} = \frac{1}{f/s + (1-f)} = \frac{1}{1 + f*(1-1/s)}$$

Where

f = *fraction of the total time of the original system that is affected by the improvement*
s = *the factor by which the improved part is sped-up*

Amdahl's Law for Reliability:

$$\text{MTTF}_{\text{new}} = \frac{\text{MTTF}_{\text{old}}}{f/s + (1-f)}$$

Where

f = *fraction of the total failure rate of the original system that is affected by the improvement*
s = *the factor by which the failure rate of the improved part is reduced*

What this has to do with AFRAID:

For large values of s, MTTF_{new} is determined mainly by f. For disk arrays, disks are only one source of failures: failures of the controller, power supply, cabling, etc. are also significant. RAID-5 produces a large value of s, and the MTTF is determined by the other components. By relaxing RAID-5 a little (making s a little smaller), the performance of the array can be improved with little impact on the overall reliability.

3. (20 points) **Moore's Law**

(a) (5 points) What is Moore's Law?
Give the original statement and the current version.
Give a one sentence explanation of what it means.

Original statement: *The number of transistors on a chip doubles every year.*

Original statement: *The number of transistors on a chip doubles every one-and-a-half years.*

What it means: *The rapid, exponential growth of integrated circuit capability drives many aspects of computer architecture and software.*

(b) (5 points) What implications does Moore's Law have for hardware design?
(Your answer should have one or two sentences.)

Chips get faster and memories get bigger just because of technology improvements. This makes increasingly complicated and high-performance hardware designs possible.

(c) (5 points) What implications does Moore's Law have for software design? (Your answer should have one or two sentences.)

Because computers get faster and have larger memories at an exponential rate, each release of a software product can have more features, need more CPU cycles, and need more memory than the previous release.

(d) (5 points) How long has Moore's Law been valid?
State two limitations that may affect Moore's law in the future. (You can just name them or write one or two sentences.)

How long has Moore's Law been valid: *Since 1965 (when Moore's paper was written, 36 years ago).*

Limitation 1: *Physical limits including leakage currents and hot-electron effects.*

Limitation 2: *Economic limits: the cost of fabrication facilities is also growing exponentially (but not as fast as the number of transistors per chip).*

4. (25 points) **Caching**
Consider the following program fragment (we won't consider *why* anyone would write such an inefficient piece of code):

```
double a[2500];
double sum = 0.0;
for(int i = 0; i < 1000; i++)
    for(int j = 0; j < 2500; j++)
        sum += a[j];
```

Assume that this fragment runs on a machine with a 16Kbyte, direct mapped, L1 data cache, with 64 byte cache lines and a LRU replacement policy. Assume a 10 ns. miss penalty. A double is eight bytes. Assume sum is held in a register. Assume that the L1 cache is empty at when the fragment starts executing.

(a) (10 points) How many data cache misses occur while executing this code?
Classify these misses.

Number of cold-start misses:
The total size of the array is

$$2500\,\text{doubles} \times \frac{8\,\text{bytes}}{\text{double}} = 20000\,\text{bytes}$$

Each cache line holds 64 cache lines. Thus, each element of the array into the cache for the first time incurs:

$$\left\lceil 20000\,\text{bytes} \times \frac{1\,\text{block}}{64\,\text{bytes}} \times \frac{1\,\text{cold-start-miss}}{\text{block}} \right\rceil = 313\,\text{cold-start-misses}$$

Executing the program incurs 313 cold start misses.

Number of capacity misses:

The array occupies 4000 more bytes than the cache can hold. Each iteration of the outer loop after the first incurs

$$999\,\text{iterations} \times \left\lceil 4000\,\frac{\text{bytes}}{\text{iteration}} \times \frac{1\,\text{block}}{64\,\text{bytes}} \times \frac{1\,\text{capacity-miss}}{\text{block}} \right\rceil = 62937\,\text{capacity-misses}$$

Number of conflict misses:

The number of conflict misses is equal to the number of capacity misses. Because the array is accessed sequentially, each miss causes the next block that will be accessed at that index to be evicted. There 62937 conflict misses.

Total number of cache misses misses:

	313	cold-start misses
+	62937	capacity misses
+	62937	conflict misses
=	126187	total misses

(b) (5 points) What is the total miss penalty when executing this fragment?

$$126187\,\text{total misses} \times \frac{10\,\text{ns}}{\text{miss}} = 1.26\,\text{milliseconds}$$

(c) (5 points) Now, add a 16 entry victim cache.
The L1 miss penalty is 2ns when the data is available in the victim cache.
What is the total miss penalty with a victim cache?

There are 126 misses per iteration of the outer loop. This is greater than the capacity of the victim cache. Thus, the victim cache won't remove any misses. The total miss menalty remains 1.26 milliseconds.

(d) (5 points) Now, add four stream prefetch buffers (and no victim cache).
The L1 miss penalty is 2ns when the data is available in a stream prefetch buffer. Assume that the prefetch buffers are not limited by prefetch bandwidth.
What is the total miss penalty with a stream prefetch buffers?

All of the misses come from one stream; so, a single prefetch buffer would be sufficient. With the prefetch buffers, the number of L1 misses remains the same. In the first iteration of the outer loop, the first L1 miss has a 10 nanosecond miss penalty, and the other 312 incur 2 nanosecond penalties. In the remaining 999 iterations of the outer loop, there are two L1 misses with 10 nanosecond miss penalties, and the remaining 124 have 2 nanosecond penalties. The total penalty is 0.268 milliseconds.

5. (20 points) **Terms of the Midterm**
Define each term below. Write one or two sentences per definition **and** cite a paper from the reading list where the term was defined (the title, or the first few words of the title are an adequate citation).

   (a) (5 points) Slip sparing.

   Slip sparing is a mechanism for avoiding bad sectors on a disk. All sectors at and after the bad sector are mapped to one sector (or track) later on the disk. From the Ruemmler and Wilkes paper on modeling disks.

   (b) (5 points) TPC-C.

   A benchmark for transaction processing (i.e. database) workloads. From the IBM paper on the S-80.

   (c) (5 points) Virtual output queueing.

   A method for improving the throughput of network switches. Each input maintains separate queues for each output port. From McKeown's paper on gigabit switches.

   (d) (5 points) Write once.

   A method for maintaining cache coherence where the first write to a cache block is broadcast and any other caches holding the block invalidate their copies. From Goodman's paper on reducing memory traffic.