

## Sample Question and Answer

1. You are experimenting with the original and optimized version of a computer and got the following results:
  - The clock rate of the unoptimized version is 5% higher
  - 30% of the instructions in the unoptimized version are loads and stores
  - The optimized version executes two thirds as many loads and stores as the unoptimized version. For all other instructions the dynamic execution counts are unchanged.
  - All instructions (including load and store) take one clock cycle

Considering that the optimized version only reduces the loads and stores, normally associated with procedure calls and returns, which version is faster? Justify your decision quantitatively.

### Solution

#### ClockRate:

**Rate(unoptimized) = 1.05**

**Rate(optimized) = 1**

**Load/Store in the unoptimized version: 30%**

**The optimized version only executes 2/3 of load/store in unoptimized version.**

$$\frac{\text{Execution Time (optimized)}}{\text{Execution Time (unoptimized)}} = \frac{(1 * 0.3 * 2/3 + 0.7) / 1}{1 / 1.05} = \frac{0.945}{1}$$

**Therefore, optimized version is faster.**

2. Simulate a 16 word, 2-way set associate cache with 2 word cache lines and LRU replacement policy; assume the cache is initially invalid. Show the cache after the following sequence of address references (the addresses are given in hexadecimal), where all references are instruction or data fetches (no stores):

62 63 64 98 65 99 66 99 67 68 69 6A 63 64 108 65 109 66 109 67 68 69 6B

#### Solution:

**The cache will look like:**

<b>Set 0</b>	<b>108</b>	<b>109</b>
	<b>68</b>	<b>69</b>
<b>Set 1</b>	<b>62</b>	<b>63</b>
	<b>6A</b>	<b>6B</b>
<b>Set 2</b>	<b>64</b>	<b>65</b>
<b>Set 3</b>	<b>66</b>	<b>67</b>

- 3 Consider three machines with different cache organizations, and measured instruction and data miss rates:

	Instruction Miss Rate	Data Miss Rate	Organization
Cache 1:	4%	8%	Direct mapped with one-word blocks.
Cache 2:	2%	5%	Direct mapped with four-word blocks.
Cache 3:	2%	4%	Two-way set associative with four-word blocks.

Assume each instruction takes one reference to memory for the instruction, and one-third of the instructions take one reference to memory for data (the other two-thirds have no data references). The cache miss penalty is  $6+W$ , where  $W$  is the number of words in a cache block. Compute the aggregate miss rate and the average memory access time for each machine. Find which machine spends the least and most cycles on cache misses.

### Solution:

Each instruction takes one reference to memory, and 1/3 of instructions also require a data reference; thus, for every four memory references, 3 are instruction and 1 is data, or 3/4 of memory references are for instructions, 1/4 for data. If the instruction miss rate is  $X$  and the data miss rate is  $Y$ , the aggregate miss rate is simply  $.75X+.25Y$ .

Cache 1:  $.75*4\% + .25*8\% = 5\%$

Cache 2:  $.75*2\% + .25*5\% = 2.75\%$

Cache 3:  $.75*2\% + .25*4\% = 2.5\%$

The average memory access time is computed as hit-time + miss-rate\*miss-penalty. We use the aggregate miss-rate here, and assume hit-time is one.

Cache 1:  $1+.05*7 = 1.35$

Cache 2:  $1+.0275*10 = 1.275$

Cache 3:  $1+.025*10 = 1.25$

Note that Cache 3 has a miss rate 1/2 that of Cache 1, but the average memory access time is very close.

Every 7500 instructions will include 2500 data references, for 10000 memory references. Cache 1 will have 500 misses in those references, using 3500 cycles to fill the cache. Cache 2 will have 275 misses, using 2750 cycles to fill the cache line. Cache 3 will have 250 misses, using 2500 cycles to fill the cache line. Thus, Cache three uses the least bandwidth for cache fills.

The common mistake here is to compute the aggregate as:  $X+Y/3$ , which gives the average likelihood of a cache miss per instruction, but not the average miss rate from the point of view of the cache.

- 4 Which gives better performance:

1. A machine with a base CPI of 1, overall cache miss rate of 3%, cache penalty of 10 cycles, and clock cycle time of  $C$  (frequency =  $1/C$ ), or
2. A machine with base CPI of 1, overall cache miss rate of 5%, cache penalty of 16 cycles, and clock cycle time of  $0.6C$  (frequency =  $5/(3C)$ ).

**Solution:**

For IC instructions, the execution time is  $IC * CPI * C$ , where C is the cycle time; to figure in the cache miss rates, we must know how many memory references per instruction. This data was not given, so we use the data from the previous question: 1 instruction reference for each instruction and 1 data reference for every 3 instructions. The cache miss rate will cause an extra MP (miss penalty) cycles for every MR (miss rate) memory references, which occur at a rate of MRI (memory references per instruction). The execution time formula is  $IC * (CPI + MRI * MR * MP) * C$ . The two machines are:

machine 1:  $IC * (1 + 1.3333 * 0.03 * 10) * C$

machine 2:  $IC * (1 + 1.3333 * 0.05 * 16) * 0.6C$

comparing the two, the IC and C terms drop out

machine 1 =  $IC * (1 + 4/3 * 0.3) * C = 1.4 * IC * C$

machine 2 =  $IC * (1 + 4/3 * 0.8) * 0.6 * C = 1.24 * IC * C$

So, machine 2 gives better performance.

5. Suppose a disk transfers data at 5MB/s, sustained rate. If the disk controller must interrupt the processor for every 4-Byte word, how often will the processor be interrupted for continuous operations?

**Solution:**

5MB/s is 1.25 million words per second, so 1.25 million times per second, or every .8 microseconds.

6. Suppose the disk transfers data at 5MB/s, as before. If the disk controller interrupts the processor for every sector, where a sector is 512 Bytes, how often will the processor be interrupted for continuous operation?

**Solution:**

5MB/s is 10000 sectors per second (roughly), so 10000 times per second, or every .1 millisecond.

7. Suppose the disk transfers data at 5MB/s, as above. If the disk controller uses the backplane bus in the first problem to transfer every 8 Bytes as a single transaction, what fraction of the bus bandwidth will be used by continuous IO operation?

**Solution:**

5MB/s is .625 million 64-bit transactions per second. A one (64-bit) word transaction takes 4 cycles. So this is 2.5 million cycles; out of 50 million cycles, this is 5% of the available bus bandwidth.

A simpler calculation is to divide the needed bandwidth, 5MB/s, by the bus bandwidth for one-word transfers from problem one, 100MB/s, to get 5%.

Common mistake: the bus as described in problem one transfers 8 data bytes in a single cycle; thus, only 4 cycles are needed for a one-word (8-byte) transaction. A common mistake is to assume a 4-byte data transfer bus, which would require 5 cycles for a one-word (8-byte) transaction.

8. What is the average time to read or write a 512 byte sector from a disk rotating at 5,400 RPMs, if the seek time is 12 ms, the transfer rate is 4 MB/sec, and the controller overhead is 8 ms?

**disk acc = seek time + rot. delay + transfer time + control overhead**

**rot. delay =  $0.5 / (5400 / 60) = 5.6$  ms**

**transfer time =  $512 / 4\text{MB} = 0.128$  ms**

**disk acc =  $12 + 5.556 + 0.128 + 8$  ms = 25.684 ms**

**If the transfer rate increases to 8 MB/sec and the control overhead decreases to 6 ms, how much faster is the new disk system?**

**disk acc =  $12 + 5.556 + 0.064 + 6$  ms = 23.62 ms**

**The new disk system is  $25.684 / 23.62 = 1.09$  times faster.**