

03s2 COMP3211/COMP9211 Computer Architecture

Week 8 Tutorial Solution

Newton Cheung
ncheung@cse.unsw.edu.au
16-9-2003

Q01. [P7.32] Consider a virtual memory system with the following properties:

- * 40-bit virtual byte address
- * 16-KB pages
- * 36-bit physical byte address

What is the total size of the page table for each process on this machine, assuming that the valid, protection, dirty, and use bits take a total of 4 bits and that all the virtual pages are in use? (Assume that disk addresses are not stored in the page table.)

*The number of page table entries = $2^{40}/2^{14} = 2^{26}$
Size of page table = 2^{26} page table entries * (36-14+4)/8 bytes per entry = 208MB*

Q02. [P7.37] Some programs, such as complex simulations of weather patterns, are loaded into a computer where they will run, uninterrupted, for long periods of time. Expensive supercomputers are often purchased for these applications. Discuss some of the reasons why virtual memory may or may not be desirable for machines designed for these types of applications. What would you propose using instead of virtual memory if you decide it is not appropriate? In any case, would a cache be necessary?

Virtual memory may not be desirable for the following reasons:

- Increase memory access time
- Decrease CPU time share (OS needs more time to look at complex structures and replacement policies)

Those applications usually do one big read and many little writes mostly, they use an array like structures (consecutive memory words). A cache is necessary as the register file may not be large enough to hold all the data. There is a large temporal and spatial localities.

Q03. [P8.7] A certain government agency simultaneously monitors 100 cellular phone conversations and multiplexes the data onto a network with a bandwidth of 1 MB/sec and an overhead latency of 350 micro-sec per 1-KB message. Calculate the transmission time per message and determine whether there is sufficient bandwidth to support this application. Assume that the phone conversation data consists of 2 bytes sampled at a rate of 4 KHz.

*One phone call data (per second)
= 2 Bytes * 4KHz*

= 8KBytes

Total load for this application (per second)

= $100 * 8\text{KBytes}$

= 800KBytes

Transmission time per 1KB message

= *transmission time + overhead latency*

= $1\text{KB}/1\text{MB/sec} + 350 * 10^{-6}$

= $1.35 * 10^{-3} \text{ sec}$

Total time required to monitor

= $1.35 * 10^{-3} \text{ sec}/1\text{KB} * 800\text{KB}$

= 1.08 sec

There is not sufficient bandwidth to support this application. The application requires to transmit 800KB of data (per second). For 1MB bandwidth, the required time is 1.08 second (with the overhead latency).

Q04. [P8.18] Here are a variety of building blocks used in an I/O system that has a synchronous processor-memory bus running at 200 MHz and one or more I/O adapters that interface I/O buses to the processor-memory bus.

- Memory system: The memory system has a 32-bit interface and handles four-word transfers. The memory system has separate address and data lines and, for writes to memory, accepts a word every clock cycle for 4 clock cycles and then takes an additional 4 clock cycles before the words have been stored and it can accept another transaction.
- DMA interfaces: The I/O adapters use DMA to transfer the data between the I/O buses and the processor-memory bus. The DMA unit arbitrates for the processor-memory bus and sends/receives 4 word blocks from/to the memory system. The DMA controller can accommodate up to 8 disks. Initiating a new I/O operation (including the seek and access) takes 1 ms, during which another I/O cannot be initiated by this controller (but outstanding operations can be handled).
- I/O bus: The I/O bus is a synchronous bus with a sustainable bandwidth of 10 MB/s; each transfer is one word long.
- Disks: The disks have a measured average seek plus rotational latency of 12 ms. The disks have a read/write bandwidth of 5 MB/s when they are transferring.

Find the time required to read a 16-KB sector from a disk to memory, assuming this is the only activity on the bus.

*To initiate the disk using DMA and read 16KB from disk:
1ms + 12ms + 16KB/5MB = 1ms + 12ms + 3.2ms = 16.2ms*

*To send 16KB through the I/O:
16KB/10MB = 1.6ms*

*To write 16KB to the memory:
32bit every 4 clock cycles and wait 4 clock cycles between 32 bit data.*

*It can take 4 bytes of data every 2 clock cycle and data rate is 400MB/sec.
So time required is 16KB/400MB = 40us*

*The bottleneck of system is on DMA and disk transfer,
Time required to read 16KB sector from disk to memory is 16.2ms*

Q05. [P8.29] Assume that we have the following two magnetic disk configurations: a single disk and an array of 4 disks. Each disk has 64 sectors per track, each sector holds 1000 bytes, and the disk revolves at 7200 RPM. Assume that the seek time is 6 ms. The delay of the disk controller is 1 ms per transaction, either for a single disk, or for the array. Assume that the performance of the I/O system is limited only by the disks and the controller. Remember that the consecutive sectors on the single disk will be spread one sector per disk in the array. Compare the performance in I/Os per second of these two disk organisations, assuming that the requests are random reads, half of which are 4 KB and half of which are 16 KB of data from sequential sectors. The sectors may be read in any order; for simplicity, assume that the rotational latency is one half the revolution time for the single disk read of 16 sectors and the disk array read of 4 sectors. Challenge: Can you work out the actual average rotational latency in these two cases?

*Each sector is 1KB
Each track has 64KB*

For 4KB access:

single disk,

*ave. rotational access time + time to access 4 sector + seek and control time
= $((0.5) \cdot (60/7200)) + ((60/7200) \cdot (4/64)) + (1\text{ms} + 6\text{ms})$
= 4.17ms + 0.52ms + 7ms
= 11.7ms*

disk array,

*4.17ms + 0.13ms + 7ms
= 11.3ms*

For 16KB access:

single disk,

4.17ms + 2.08ms + 7ms = 13.25ms

disk array,

4.17ms + 0.52ms + 7ms = 11.7ms

for half 4KB access and half 16KB access,
single disk requires $(11.7\text{ms} + 13.25\text{ms}) * (0.5) = 12.48\text{ms}$
so in 1 sec, 80.2 I/O can be done.
disk array requires $(11.3\text{ms} + 11.7\text{ms}) * (0.5) = 11.5\text{ms}$
so in 1 sec, 87.0 I/O can be done.