

PART I: OPERATING SYSTEM

1. (13 pts) Your friend implemented a multithreaded banking software using Semaphore as synchronization primitive. Assume all semaphores are initialized to 1. Below is her code snippet.

```
Semaphore locks[N];
int amounts[N];
// transfer money from acct1 to acct2
void transfer_money(int acct1, int acct2, int amount)
{
    locks[acct1].p();
    locks[acct2].p();
    amounts[acct1] -= amount;
    amounts[acct2] += amount;
    locks[acct2].v();
    locks[acct1].v();
}
```

- a. [3 pts] Explain how this can deadlock.
 - b. [10 pts] Propose a solution that is based on removing "hold-and-wait" condition. That is, a thread needs to acquire either both locks at once or not at all. Please (1) write in plain language what your solution is, then (2) write pseudo-code to implement your solution. You may add new variables and/or new semaphores. Points will be given only if your solution remove "hold-and-wait" condition and more points will be given to more efficient implementation.
2. (12 pts) Consider a file system that uses a hierarchical directory structure with an inode (index node) for each file and directory. The file system you choose uses the following file structure:
- Non-uniform depth 2-level indexed file inodes:** the inode contains 10 direct pointers and 2 indirect pointers. The direct pointers point to data blocks. An indirect pointer points to one disk block (called indirect block) that contains pointers to data blocks. Assume that the disk block size is 2000 bytes. Also assume size of each pointer in the indirect block is 4 bytes. Show your intermediate work and explain clearly for each question to receive full credits. No point will be given if you only write a number as your solution.
- a. [3 pts] What is the largest file size in bytes that can be supported by this system?
 - b. [3 pts] If the disk block size were changed to 4000 bytes, would the new

- maximum file size be twice as much? If not, why not?
- c. [3 pts] What is the number of disk blocks that need to bring into memory from memory before reading the first block in `/var/log`? Assume that nothing is currently cached in memory, and that each inode fits in one disk block.
 - d. [3 pts] What is the number of disk blocks that need to bring into memory to serve a read request of 1 byte from offset 25,000 in `/var/log` after the request in c? Assume all disk blocks are cached once they have been fetched.
3. (15 pts) Now the 淡水-新店 line of Taipei MRT(台北捷運) passes through 淡水, 台北車站, 中正紀念堂, and 新店 stations without transfer. People complain that commute between 淡水 and 新店 will need to transfer at 中正紀念堂 to distribute the load of 台北車站. Note that MRT stations are usually passed by few lines.
- a. [5 pts] Would the new setup always increase the average time spent for people commute between 淡水 and 新店? Why?
 - b. [10 pts] What two distributed computer system technologies we might apply to reduce the complaint? Why?
4. (10 pts) Traditional disk scheduling uses "SCAN" to reduce seek time. However, if the disk access requests involve real-time constraints, a "SCAN-EDF", earliest deadline first, scheduling algorithm might be used to be both real-time and efficient.
- a. [5 pts] In what scenarios the "SCAN-EDF" method does not work as it claims?
 - b. [5 pts] Would the method be applicable to memory access to reduce the time spent for page fault with respect to the seek time spent in disk access while still maintaining reasonable real-time constraints? Why?

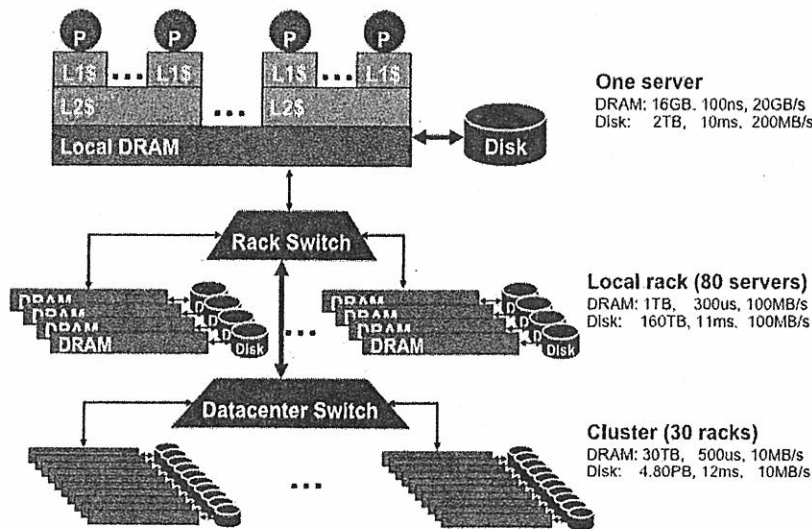
PART II: COMPUTER ARCHITECTURE

1. (10 pts) [Overview] Please explain the following terms in English. For each term, please spend no more than 100 English words.
 - (a) [2 pts] basic block
 - (b) [2 pts] non-uniform memory access architecture
 - (c) [2 pts] GPGPU
 - (d) [2 pts] cloud computing
 - (e) [2 pts] virtual machine monitor

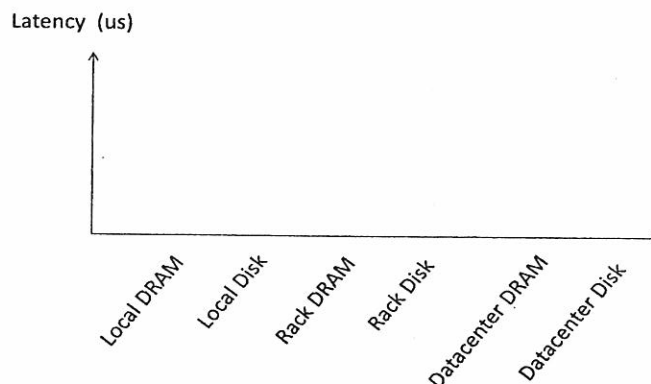
2. (12 pts) [CPU Performance] Calculate the performance of a processor taking into account stalls due to data cache and instruction cache misses. The data cache has a 92% hit rate and a 2-cycle hit latency. Assume that latency to memory and the cache miss penalty together is 100 cycles. The instruction cache has a hit rate of 90% with a miss penalty of 50 cycles. Assume the load never stalls a dependent instruction and assume the processor must wait for stores to finish when they miss the cache. Finally, assume that instruction cache misses and data cache misses never occur at the same time.
 - (f) [3 pts] Calculate the average memory access latency for the data cache.
 - (g) [3 pts] Assume the base CPI using a perfect memory system is 1.0. Calculate the additional CPI of the pipeline due to the instruction cache stalls.
 - (h) [3 pts] Same as (g), but calculate the additional CPI due to the dcache stalls.
 - (i) [3 pts] Assume 30% of instructions are loads and stores. Calculate the overall CPI for the machine.

見背面

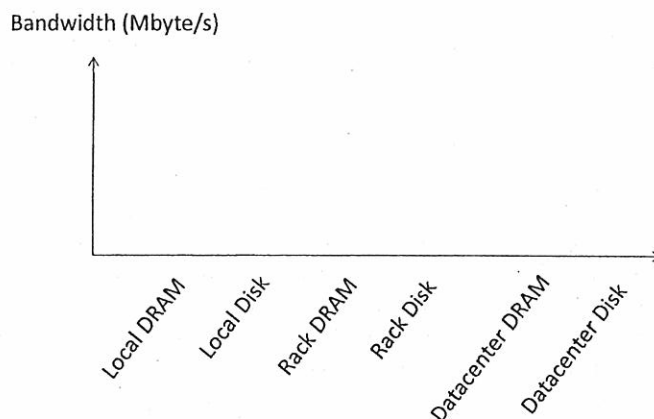
3. (13 pts) [Large Systems] The figure below shows a programmer's view of storage hierarchy of a typical datacenter. A server consists of a number of processor sockets, each with a multicore CPU and its internal cache hierarchy, local shared and coherent DRAM, and a number of directly attached disk drives. The DRAM and disk resources within the rack are accessible through the first-level rack switches, and all resources in all racks are accessible via the cluster-level switch.



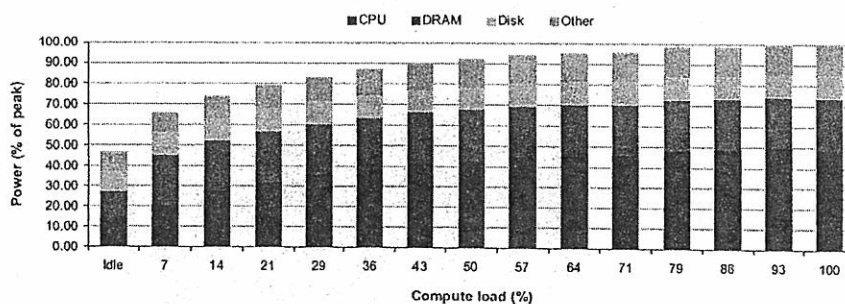
- (j) [3 pts] How would you write a program to utilize the resources in the entire system?
- (k) [3 pts] For a server, the latency required to access a data item in the datacenter depends on the location of the data. Assume the latency for a network packet to pass through one switch is 100us. ($1\text{us} = 10^{-6}\text{ s}$). Complete the chart below to characterize the relationship between the latency and the data location.



- (l) [3 pts] Assume each group of servers is connected through a 1-Gbps link to a rack-level switch that has an additional eight 1-Gbps ports used for connecting the rack to the cluster-level switch. Complete the figure below to quantify the bandwidth of the datacenter system.



- (m) [4 pts] Energy consumption is important to the operation cost of a datacenter. The figure below shows the power usage of the main subsystems for a recent Google server as the compute load varies from idle to full activity levels. Dynamic Voltage and Frequency Scaling (DVFS) is one way to save energy use on a server. How much saving can DVFS achieve? In addition to DVFS, how would you further reduce the energy cost?

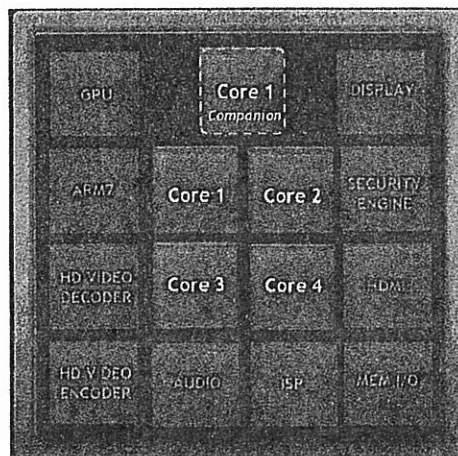


4. (15 pts) [Multicore Systems-on-Chips] Please read the following story from NVIDIA's whitepaper before answering the questions.

In February 2011, NVIDIA introduced and demonstrated its Project Kal-El mobile processor, the world's first quad core mobile processor. Project Kal-El will enable new mobile applications, new experiences, more robust multitasking, higher-quality gaming, and faster web browsing. In addition, Project Kal-El will further extend battery life by operating its CPU cores at lower frequency, yet still be able to complete more work than dual core or single core processors.

NVIDIA's Project Kal-El processor implements a so called Variable Symmetric Multiprocessing (vSMP) technology, which includes a fifth CPU core (the "Companion" core) built using a special low power silicon process that executes tasks at low frequency for active standby mode, music playback, and even video playback, as shown in the figure below. The four main "quad" cores are built using a standard silicon process to reach higher frequencies, while consuming lower power than dual core solutions for many tasks. All five CPU cores are identical ARM Cortex A9 CPUs, and are individually enabled and disabled (via aggressive power gating) based on the work load. The "Companion" core is OS transparent, unlike current Asynchronous SMP architectures, meaning the OS and applications are not aware of this core, but automatically take advantage of it. This strategy saves significant software efforts and new coding requirements.

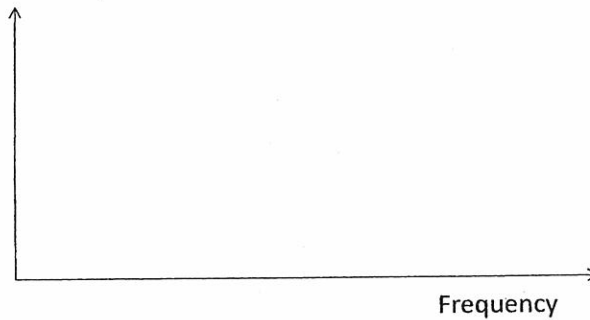
Power consumption of a silicon device is equal to the sum of leakage power and dynamic power. Leakage power is mainly determined by the silicon process technology, and dynamic power is determined by silicon process technology and by operating voltages and frequencies. The main cores build on a fast silicon Process, so they consume high leakage power under idle or active standby conditions, but are capable of running at higher frequency ranges without requiring significant increases in operating voltage. The Companion core on a low power process technology has low leakage power but has slower switching times at normal voltage levels, and to enable it to switch faster (for high frequency operations) it needs higher than normal voltage ranges.



(n) [3 pts] Draw the power-frequency curves of the Kal-El processor in the following 5 settings:

- (1) the Companion core is ON, and the main cores are OFF;
- (2) the Companion core is OFF, and one of the main cores is ON;
- (3) the Companion core is OFF, and 2 of the main cores are ON;
- (4) the Companion core is OFF, and 3 of the main cores are ON;
- (5) the Companion core is OFF, and 4 of the main cores are ON;

Power Consumption

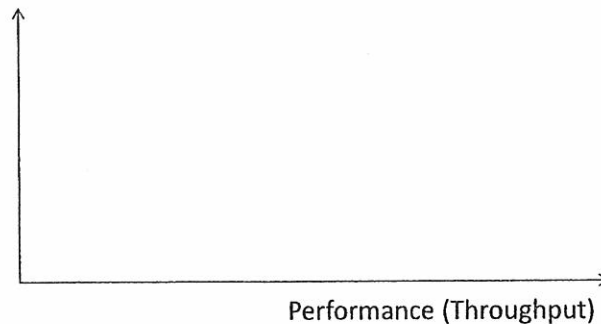


(o) [3 pts] With the power-frequency curves, how would you maximize power-performance ratio for the system when there is only ONE single-thread application running on the system, but the workload increases?

(p) [3 pts] How would you maximize power-performance ratio for the system when the number of tasks running on the system increases from 1 to 8?

(q) [3 pts] Draw the ideal power-performance curve for the system when your strategy in (p) works perfectly. Please briefly explain your curve.

Power Consumption



(r) [3 pts] The Kal-El processor is a system-on-chip (SoC) which also include intellectual properties (IP) other than CPU cores. Why are these IP's included in the SoC? Why not have more or bigger general-purpose cores, similar to x86-based desktop or server processors?