CS 596 Quiz #1: Introduction to Parallel Programming

(All questions carry equal points) Name: solution

Microprocessors and Memory

1. Why do modern microprocessors have caches? Why are caches small compared to main memory?

Because regular memo ry is too slow to feed the processors fast enough.
Because cache, while faster, is much more expensive to produce (and there isn't enough room on the processor or chip for lots of it).

2. What two fundamental principles should you obey when using data in order to get maximum benefit out of caches? Describe each briefly. Hint: Caches are designed with these two principles in mind.

Spatial locality and temporal locality.

3a. What is the clock period of a processor? What are the units?

Clock period, or cycle, is the smallest unit of time in which an instruction can occur. Units are s^{-1}, or Hz.

3b. What is the floating point performance of a processor? What are the units?

Floating point performance is the rate at which it can complete floating point operations. Units are floating point operations per second, of flops.

4. If two different processors (for example, a Pentium 3 and a G4) have the same clock period , do they have the same *theoretical peak performance*? Why or why not?

The theoretical peak performance is number of operations a processor can complete per second. This is a function of both the number of cycles per second (clock speed) *and* the number of operations that can be completed per cycle. The latter number is determined by the chip architecture and instruction set. It typically ranges from 1 (e.g. Intel Pentium) to 4 (e.g. IBM Power3, Motorola G4) for modern microprocessors. Hence they will not have the same theoretical peak performance.

5. If two different processors have the same *theoretical peak performance*, will they achieve the same *sustained performance* on real codes? Why or why not?

Generally not. Depends on the details of each processor's architecture and the rest of the system: amount/speed of cache, speed of bus and memory, etc.

Parallelism:

6. What are the two fundamental types of parallelism? Describe each very briefly.

Data parallelism and task parallelism.

7. Which of the above two (i.e. answers of question# 6) is employed more often in real codes? (Hint: which is easier to code, load balance, etc.)

Most codes employ both forms of parallelism to some degree, but more of the parallelism is expressed as data parallelism.

8. Can the loop below be executed safely in parallel using loop parallelism ? Why or why not?
Fortran loop :

```
      do I =1, N
         Y(I) = a*X(I-1) + b*Y(I) + c*Z(I+1)
      end do

C loop :
      for ( I=1, I< N+1 ; I++ ) {
         Y(I) = a*X(I-1) + b*Y(I) + c*Z(I+1) ;
      }
```

Yes. There are no parallel dependencies between values of Y on different iterations.

9. When you run a MPI parallel code on a parallel machine what are the two fundamental things your code needs to know ? (hint: the first two MPI calls after MPI initilization)

Total number of processors on which your code is running and what is the rank (which goes from 0 to N-1) of each processor

10. What is latency? What is bandwidth?

Latency is the time it takes to start sending (or receiving) a message. Latency can also be defined as time it takes to send a zero length message.
Bandwidth is the rate at which data is transmitted.

Amdahl's Law:
11. What is the form of Amdahl's Law that gives the speedup S on N processors in terms of the serial fraction of code, the parallel fraction, and N? (Hint: you know the speedups for $f_s = 1.0$ and $f_s = 0.0$.)

$$S = \frac{1}{f_s + (f_p/N)}$$

12. If 75% of the operations in a code can be executed in parallel, what is the theoretical maximum speedup it can achieve on 3 processors? On any number of processors?

Setting $f_s = 0.25$, $f_p = 0.75$, the theoretical speed up on 3 processors is 2. The maximum speedup on any number of processors is 4.

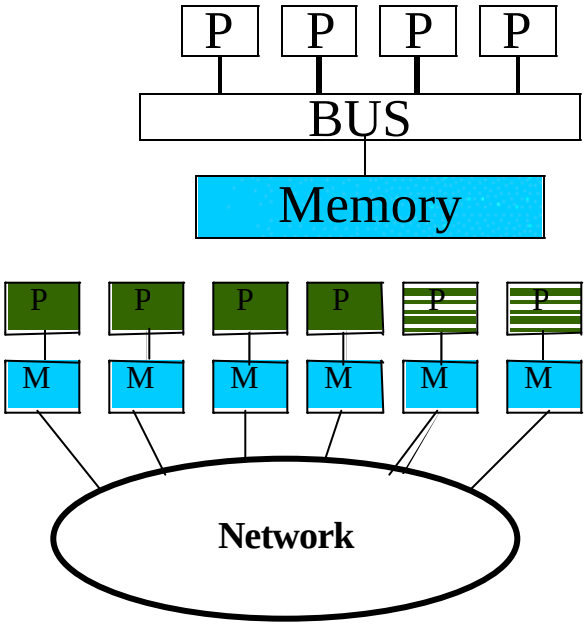
13. What are at least three factors that can cause the speedup or a real code to be *less* than that predicted by Amdahl's Law?

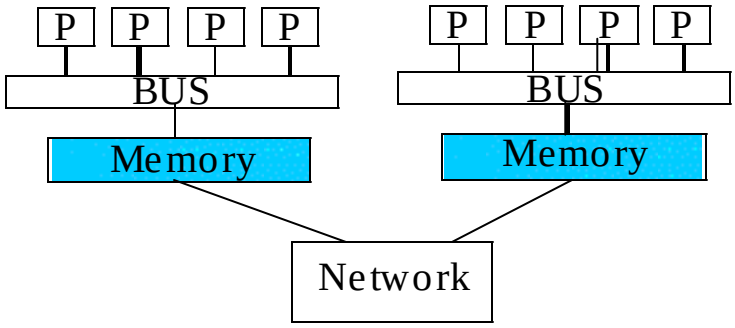
Load imbalance, communications, I/O…

Parallel Computing Architectures:

14. Draw each of these systems schematically:

a) Shared memory system
b) Distributed memory system
c) Hybrid system that links shared-memory nodes





15. What is the fundamental difference between a system that links SMP nodes and a cc-NUMA system?

Each node has a *directory* in the cc-NUMA system that tracks memory locations on other nodes and enable system to create a single logical address space.

16. Why are shared memory systems generally considered easier to program than distributed memory machines?

Because there is a global address space, so every variable name means the same thing (points to the same data) to every processor and therefore data traffic doesn't need to be managed explicitly (no messages!)

17. What is cache coherence?

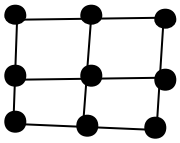
Cache coherence means letting all processors know when a value has been changed in a cache of any processor. If a processor changes a variable's value and the new value is in its cache when another processor needs the variable, it is important that the system notify *all* processors that the variable's value in memory, and in their own caches(if they had also read it from memory), is wrong and the correct value is still in the first processor's cache.

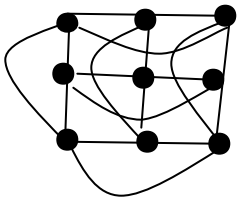
Network Interconnects

18. What are the two basic mechanisms for moving data between the processors and the memory in a shared memory system? What are the pros and cons of each?

Bus, crossbar: bus is simple but easy to saturate. Crossbar is more expensive but scales better.

19. Draw a 3 X 3 mesh network. Draw a 3 X 3 Torus or Wraparound mesh .





20. Bisection Bandwidth is defined as the minimum number of communication links that have to be removed to partition the network into two equal halves. What is the Bisection Bandwidth of a ring with even number of processors and of a hypercube with p processors.

    Bisection BW of a ring is 2. And for a hypercube it is p/2.