Lecture 20

Final Exam Review
Announcements

• Final Exam:
  Thursday, December 7th  3:00PM to 6:00 PM
• Open book,  but no notes
  Text, handouts, lecture slides
• Office hours during exam week by appointment
Terms and concepts

- Know the definition and significance of ....
- Parallel speedup and efficiency, super-linear speedup, scaled speedup,
- Scalability, cost, cost-efficient, isoefficiency
- Amdahl’s law, serial bottlenecks
- Granularity, surface-to-volume effect
- Loop carried dependence
Terms and concepts

• CRCW and CREW PRAM
• SPMD, MIMD, SIMD
• Multiprocessors and multicomputers; shared memory, message passing
• Interconnection networks: hypercube, ring, mesh, k-ary d-cube, broadcast and reduction algorithms; diameter and bisection bandwidth
• Message startup, half power point $n^{1/2}$, peak bandwidth
• Processor geometry and blocked data decompositions
Shared Memory Terms and Concepts

• Cache coherence, cache consistency
• False sharing
• Snooping protocol, bus based coherence
• UMA (SMP) vs. CC-NUMA designs
• Distributed shared memory, directories
• Race conditions, non-determinacy, critical sections, atomicity
• Locks and barriers
• Pthreads and OpenMP interfaces
Other techniques

• Data parallel programming
  – forall semantics, data layouts, implicit communication

• Load balancing:
  – blocked data decompositions, cyclic decomposition, self-scheduling, client/server computation
  – Irregular decompositions, recursive coordinate decomposition

• Performance modeling and performance measurement techniques
Algorithms

• Know the purpose of the following algorithms, and the significant implementation issues affecting performance.
• Be familiar with performance models for each and be prepared to analyze performance and scalability
• **Sorting**
  – Enumeration sort, odd-even sort, odd-even transposition sort, bitonic sort, bucket sort, sample sort, shell sort
• **Stencil methods**
  – **ODE solver:** 1D case
  – Multidimensional cases: 2D and 3D
  – “Curse of dimensionality:” surface to volume ratios
  – Convergence and how to check it
  – Ghost cells, partitioning, performance models
  – Dealing with Loop carried dependences
• **Matrix multiplication**
  – Canon’s algorithm, SUMMA
• **LU Decomposition (Gaussian elimination)**
  – Cyclic decompositions
• **Collective communication:** ring and hypercube algorithms for reduction/broadcast, allReduce, gather/scatter
Practice problems

- Barrier synchronization
- Why are two lock variables needed, arrival and departure
- Initialization?

```c
Void barrier(. . . ){ 
    LOCK(arrival);
    Count++;
    If (Count < n$proc) UNLOCK(arrival);
    Else UNLOCK(departure);
    LOCK(departure);
    Count--;
    If (count > 0) UNLOCK(departure);
    Else UNLOCK(arrival);
    Return;
}
```
Communication and data parallelism

- Hand translate to message passing code
- What communication operations are needed?
- CSHIFT circulates the data
- Which data parallel program is faster?

Program 1

(1) for ( i=0; i<n; i++ ){
   (2) XI[:] = X[i];
   (3) D[:] = XI[:] - X[:];
   (4) F += abs( D[:] );
}

Program 2

(1) T = X;
(2) for (i=0; i<n; i++){
   (3) F += abs( X-T );
   (4) T = CSHIFT(T,1);
   (5) }

Hand translate to message passing code

What communication operations are needed?

CSHIFT circulates the data

Which data parallel program is faster?
Data parallelism

• Can we parallelize the inner loops as shown?

LOOP #1
for j = 0 to n-1
  for i = 0 to n-1
    A[i, j+1] = A[i, j];
  for i = 0 to n-1
    A[i, 1:n] = A[i,0:n-1];

LOOP #2
for j = 0 to n-1
  for i = 0 to n-1
    A[i, j+1] = A[i, j];
for j = 0 to n-1
  A[0:n-1, j+1] = A[0:n-1, j];
Networks

- Diameter, bisection bandwidth
- Map a 1D ring onto a hypercube, mesh, k-ary d-cube
- How many parallel/unique paths between any two nodes in a hypercube?
- Sort on a 2D network
- Sort on a hypercube