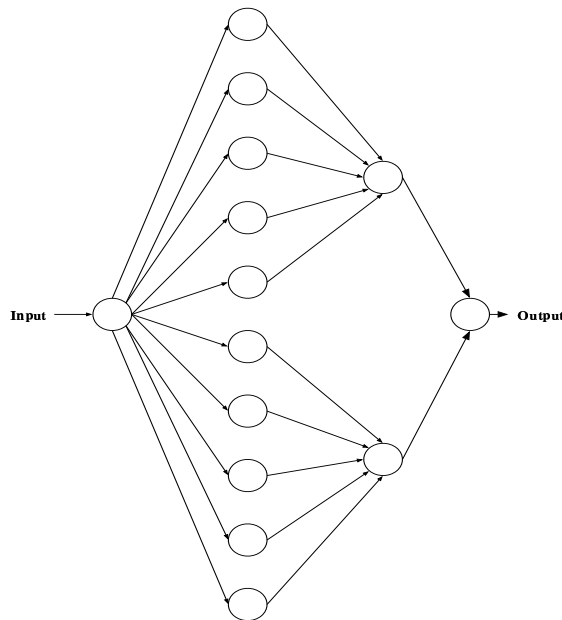


# CS 635 Fall 04

## Homework 1

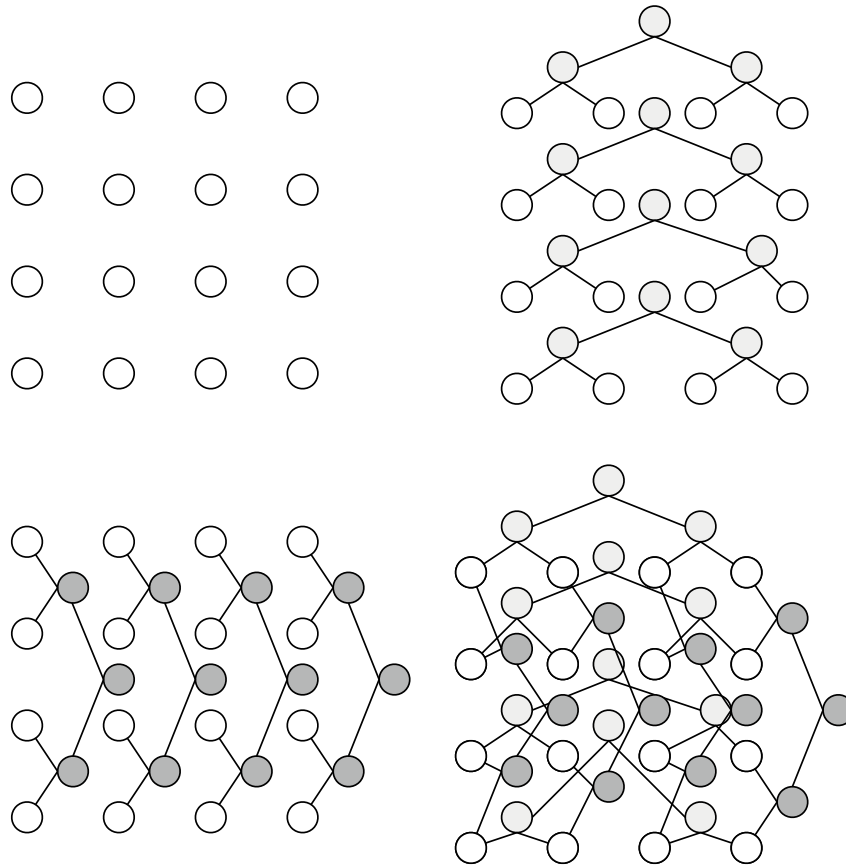
Please turn in your solutions to the following problems at the beginning of class on *Tuesday, September 29, 2004*.

1. The figure below is a *task graph* that corresponds to an algorithm. Each node represents a routine that is inherently sequential and cannot be further subdivided. There are 14 tasks: one for input, one for output, and 12 computation tasks. The arrows in the graph indicate that the input task must be completed before any of the computational tasks can start. Similarly, 10 tasks must finish before two additional tasks can be executed; the completion of the two tasks permits the final output task to begin.



- (a) Assuming that each of the tasks takes 1 unit of sequential time, what is the maximum speedup that can be achieved if this problem is solved on 2 processors?
  - (b) What is the upper bound on the speedup if this problem is solved with  $p$  parallel processors?
  - (c) What is the smallest number of processors sufficient to achieve the speedup in part (b)?
2. Let  $A$  and  $B$  be any two processors in a  $d$ -dimensional hypercube.
    - (a) What is the distance, in terms of communication links, between processors  $A$  and  $B$ ?
    - (b) Find the average distance between any two processors in the hypercube.
    - (c) It is claimed that the number of distinct paths connecting processors  $A$  and  $B$  in a  $d$ -dimensional hypercube is equal to  $d$ . Give an example with  $d = 3$  to support or refute this claim.
  3. Give the interconnection network functions that describe the links in a pyramid topology.
  4. A *permutation* maps a set of  $p$  processor ID's  $\{0, 1, 2, \dots, p - 1\}$  onto itself. Consider an Omega network with  $p = 8$  processors.

- (a) How many different permutations exist for this network assuming that the switches can be set to straight-through or crossover?
- (b) What is the minimum number of these permutations that must be applied to achieve a linear shift of the data (i.e. where node  $i$ 's data is sent to node  $i + 1$  for all  $i$ )?
5. A *mesh of trees* is a network that imposes a tree interconnection on a grid of processors. Beginning with a  $\sqrt{p} \times \sqrt{p}$  grid, a complete binary tree is first imposed on each row of the grid. Then a complete binary tree is imposed on each column of the grid. An example of a  $4 \times 4$  mesh of trees is shown below:



Assuming that the nodes at the intermediate levels are switches, determine the bisection width, diameter, and total number of switching elements in a  $\sqrt{p} \times \sqrt{p}$  mesh of processors.

6. Write a CREW PRAM algorithm that finds the summation of  $n$  elements in polylogarithmic time on a polynomial number of processors.
7. Write a CREW PRAM algorithm that finds the product of two  $n \times n$  matrices in polylogarithmic time on a polynomial number of processors.
8. Consider the nearest neighbor averaging problem studied in class, and recall that we looked at several data parallel (SIMD) programs for solving the problem.
- (a) Describe, in words, an approach you might use to perform this computation on  $p$  MIMD processors for an  $n \times n$  matrix when  $p$  is much smaller than  $n$  (e.g.  $p = 16$  and  $n = 1024$ ).

(b) What speedup and efficiency do you expect to observe for your method?

9. Regarding speedup measures:

- (a) If 25% of the computations in a program must be performed sequentially, what is the maximum speedup achievable according to Amdahl's law?
- (b) Suppose a parallel application requires 220 seconds to complete on a 64 processor system. Further analysis reveals that 5% of the time is spent on serial computation on a single processor. What is the scaled speedup of the application according to Gustafson's Law?
- (c) A scientist claims that a 20 million dollar system with 16,384 processors can achieve a scaled speedup of 15,000 for her application. What is the maximum fraction of the execution time that can be devoted to inherently sequential operations if this is to happen?

10. Code, compile, and run  $p$  copies of the "Hello World" MPI program on  $p$  nodes of the ITE cluster (one copy per node), where  $1 < p \leq 8$  is of your choosing. Turn in a copy of the output that is produced.

A Simple SPMD MPI program:

```
#include "mpi.h"
#include <stdio.h>

int main(argc, argv)
    int argc;
    char *argv[];
{
    MPI_Init (&argc, &argv); /* Initialize MPI state */
    printf ("Hello world\n");
    MPI_Finalize()           /* Clean up MPI state */

    return 0;
}
```

To run an MPI (LAM) program, you should do something like this:

```
Step 0: set up your LAMHOME and PATH variables if needed
        set up ssh keys for the nodes
Step 1: hcc -o foo foo.c -lmpi % Compile program
Step 2: vi hostfile % Make a list of compute nodes (schema)
Step 3: recon -v hostfile % Test the schema
Step 4: lamboot -v hostfile % Start LAM for the schema
Step 5: tping n0... % to ping node(s) (if desired)
Step 6: mpirun -v -c xxx foo % Run xxx copies of the code
Step 7: mpitask or mpimesg % for tracing (if desired)
Step 8: lamclean -v % Kill orphan processes
Step 9: wipe -v hostfile % Terminate LAM
```