Homework on Parallel Program Design: PCAM

Problem 1. In section 2.3.2 (Global Communication) of *Designing and Building Parallel Programs* (`http://www.mcs.anl.gov/~itf/dbpp/`) Foster discusses various approaches to summing a set of numbers that is distributed among the tasks. This is an example of a *reduction* operation. Figure 2.8 shows how a binary tree can be used to diagram how a set of n numbers distributed among p tasks can be summed with a degree of parallelism. Computations at each level of the tree can occur concurrently and the arrows indicate communication.

Suppose we want to compute a reduction sum on a hypercube. Each processor initially contains a numeric value and the goal is to compute the sum of these values and have the sum reside in a single processor.

a. Draw a $p = 4$ processor hypercube (a square) with circles as nodes, each representing a processor-switch pair. Inside each circle write a number that will be part of a reduction sum. Let this drawing represent the initial state. Now construct a sequence of drawings of the cube representing communication and compute cycles as the sum is computed. Your final drawing should show the final sum in exactly one of the processors.

b. Draw a binary tree (as in Figure 2.8 of Foster) diagramming the sum operation. Do you really need all of the communication arrows shown on Foster's diagram? Why or why not?

c. Repeat the last two steps but with a 8 processor hypercube.

d. Suppose p is a power of 2 and that $n = p$ so that each processor contains exactly one of the numbers that must be summed. Explain why the entire sum can be completed with $\log_2 p$ communication cycles.

e. Now suppose that $n > p$ but that p divides evenly into n and each processor has n/p numbers to be summed. Describe how the reduction sum can be adapted to this situation. How many communication cycles are required?

Problem 2. Suppose you are given a two-dimensional array (dimensions are M and N) containing various numbers. Work through Foster's four design steps (partitioning, communication, agglomeration, and mapping) to devise a parallel algorithm to find the maximum value stored in the array. Assume that M and N are each much larger than the number of processors p and that you can use the function `max(a,b)` to find the maximum of the two numbers a and b. Discuss your reasoning in each of the four PCAM steps, indicating why you made the choices you made. (See 2.6.2, 2.7.2, and 2.8.2 of Foster for examples). Include diagrams as needed to help explain your reasoning.