WITS UNIVERSITY

School of Computer Science and Applied Mathematics
UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

COMS3008A: Parallel Computing
Exercise 3

2021-8-31

Objectives

Students should be able to

- Design parallel algorithms for problems with common computation patterns such as reduction and broadcasting.
- Identify the concurrency, task dependency, communication, and synchronization in a parallel solution.

Problems

1. Consider the multiplication of a dense $n \times n$ matrix A with a vector b to yield another vector y. The ith element $y[i]$ of the output vector is the dot-product of the ith row of A with the input vector b, i.e., $y[i] = \sum_{j=1}^{n} A[i, j]b[j]$. For this problem, design a parallel algorithm that involves decomposition, assignment of tasks, analyzing the communication pattern, and mapping.
2. Consider the problem of sorting a sequence A of n elements using quicksort algorithm. Quicksort is a divide and conquer algorithm that stars by selecting a pivot element x and then partitions the sequence A into two subsequences A_0 and A_1 such that all the elements in A_0 are smaller than x and all the elements in A_1 are greater than or equal to x. Each subsequence is then sorted by recursively calling quicksort. How would you parallelize quicksort?
3. Many parallel algorithms for distributed memory systems require a broadcast step in which one task (or process) communicates a value it holds to all of the other tasks.
    - How a broadcast would be implemented for a parallel computer with the following interconnect architecture respectively.

  - Bus
  - Ring
  - Hypercube
  - Fat tree
- What is the time complexity for broadcast with Hypercube interconnect architecture?

4. Figure 1 illustrates a task dependency graph of a decomposition for an $N \times N$ grid based iterative solver. Each node represents a task for a grid point, and a red arrows shows the dependency between a pair of tasks, i.e., a red arrow from a grid point P to another grid point Q means the task Q is dependent on task P. Analyse the concurrency, i.e., which tasks can be executed in parallel, based on this task dependency graph.

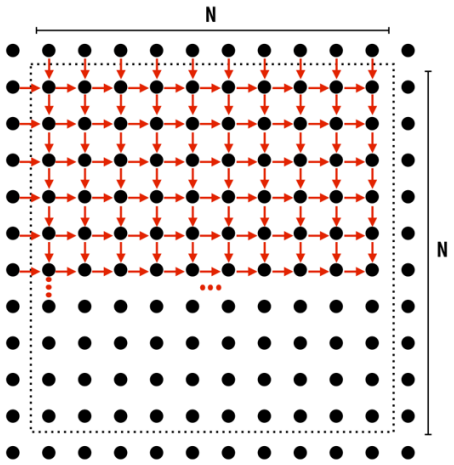

Figure 1: Each row elements depends on the element to the left; each column depends on the previous column.

5. The problem from Question 4 is modified so that it has more concurrency. As shown in Figure 2, the solution now takes two phases, where in the first phase, the algorithm updates all the red cells, and in the second phase, it updates all the black cells. The algorithm repeats this process until convergence.

   (a) Discuss the concurrency and task dependencies for this modified approach.

   (b) How would you assign the tasks to P number of processes? Assume $P = 4$.

   (c) What are the communication and synchronization like during the execution of this iterative solver?

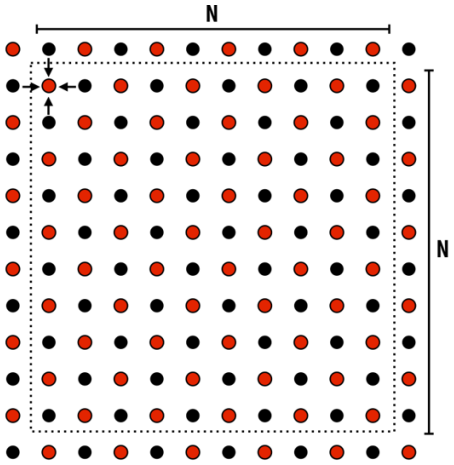

Figure 2: Update all the red cells first, then update all the black cells. The process is repeated until convergence.