CS531 HW 4 Solution

Chain-Wu Lee, lee-d@cs.buffalo.edu

May 6, 1998

1. *Optimal Hypercube Parallel Prefix*
   The work is similar to the optimal mesh parallel prefix algorithm

   (1) Calculate the parallel prefix sequentially on each PE, suppose the result of the last data entry in PE_i is x_i.

   (2) Do the parallel prefix on x_i by using hypercube parallel prefix algorithm. Assume the result is y_i.

   (3) Update the data entry by y_i within each PE.

   This algorithm runs on $n/\lg n$ PEs, each PE contains $\lg n$ data. Step 1 takes $\Theta(\lg n)$ time, step 2 takes $\Theta(\lg(n/\lg n)) = \Theta(\lg n)$ time, step 3 takes $\Theta(\lg n)$ time, so total time needed is $\Theta(\lg n)$.

2. *Interval Prefix Computation*

3. *Quicksort*
   The problem is equivalent to the array packing problem with marked of whether the element is smaller than the pivot (the first element). Please see the hadout for detail.

4. *Arc Covering*
   Please refer to the class notes and see how the problem can be converted into parallel prefix problem.

5. *Optimal EREW PRAM Algorithm*
   The PRAM algorithm listed on the textbook is an EREW algorithm because no two PEs will access the same memory location at the same time.

6. *Carry-Lookahead Addition*
   The key to the algorithm is to transform the sentence "the ith carry is a 1 if and only if the leftmost non-P to the right of the ith position is a G" into a parallel prefix operator. We can defined the *carry function* as follows :

	NA	1	0
P	NA	1	0
G	1	1	0
S	0	1	0

   At beginning all the carry bit is set to be NA and then perform the parallel prefix with the *P,G, S* flags according to the function list above.

7. *q-Dimension Mesh*
   The algorithm is an extension of linear and mesh architecture. The q-dimension mesh algorithm will try to finish the 1st dimension parallel prefix, then send them to the 2nd dimension; the 2nd dimension try to send the result to the 3rd dimension and so on until all q dimension is accomplished.

   The time needed for every dimension is bounded by the communication diameter. The size of the q-dimension mesh is $\sqrt[q]{n}$ for each dimension. For the k-dimension mesh, the communication diameter is $k \times \sqrt[q]{n}$. Therefore the computation time is

$$\sqrt[q]{n} + 2\sqrt[q]{n} + 3\sqrt[q]{n} \cdots q\sqrt[q]{n} = \sum_{k=1}^{q} k\sqrt[q]{n} = \frac{q(q+1)}{2} \times \sqrt[q]{n}$$

If we consider the dimension q a constant, the time needed is $\Theta(\sqrt[q]{n})$.

8. *Skyline Visibility*
   This problem is equivalent to find the max parallel prefix on the angle of each line segment observing from the origin. For the PRAM and Mesh architecture, you can use the cost optimal algorithms listed on the textbook. For the PRAM, the number of PEs is $N/\lg N$, the time needed is $\Theta(\lg N)$ and cost is N. For the Mesh architecture with size $N^{1/3} \times N^{1/3}$, the number of PEs is $N^{2/3}$, the time needed is $\Theta(N^{1/3})$ therefore the cost is N which is optimal.

   As to the tree architecture, we assume the that each PE at the leaves of the tree holds 1 data entry, therfore there totally $2n-1$ PEs in the tree. The parallel prefix can be accomplished by simulating the PRAM algorithm. The first step each PE tries to communicate with the PE 1 position ahead (2 hops), then 2 position ahead (4 hops) and then 4 ahead (6 hops) and so on .... Therefore the time needed is equivalent to the communication hops. More generally, if a leaf node wants to communication with a PE of k position ahead, it needs to go through $2 \times (\lg k + 1)$ communicaton hops

$$T = \sum_{k=1}^{\lceil \lg n \rceil} 2 \times (\lg k + 1) = 2 \times (\sum_{k=1}^{\lceil \lg n \rceil} \lg k + \lceil \lg n \rceil) = \Theta(\sum_{k=1}^{\lceil \lg n \rceil} \lg k)$$

   For the entry $\sum_{k=1}^{\lceil \lg n \rceil} \lg k$, it is equivalent to $\lg 1 + \lg 2 + \ldots + \lg\lceil \lg n \rceil = \lg(1 \cdot 2 \cdots \lceil \lg n \rceil) = \lg(\lceil \lg n \rceil !)$. By the Stirling approximation, $\lg(\lceil \lg n \rceil !) = \Theta(\lg n \cdot \lg\lg n)$. Therefore the time is $\Theta(\lg n \cdot \lg\lg n)$ and the cost is $\Theta(n \cdot \lg n \cdot \lg\lg n)$.

9. *Pyramid Parallel Prefix*
   The whole concept is similar to the mesh parallel parallel prefix algorithm except it uses the the pyramid PEs and links as the communication path so that the running time can be reduced. For a machine with n PEs on the base and $\sqrt{n}$ on each side of the base, the parallel prefix time needed for the 1st dimension (row) is equal to

$$1 + 2 + 3 + \ldots + \lceil \lg \sqrt{n} \rceil = \sum_{k=1}^{\lceil \lg \sqrt{n} \rceil} k = \Theta(\lg^2 \sqrt{n}) = \Theta(\lg^2 n)$$

   Similarly for the 2nd dimension broadcast and then the row update, all takes $\Theta(\lg^2 n)$ time.

   The cost $= (\Theta(n) \cdot \Theta(\lg^2 n)) = \Theta(n \lg^2 n)$.

10. *Mesh of Trees Parallel Prefix*
    The asympotic running time should be equivalent to the pyramid parallel prefix algorithm, therefore the time is $\Theta(\lg^2 n)$ and the cost is $\Theta(3n - 2n^{1/2}) * \Theta(\lg^2 n) = \Theta(n \lg^2 n)$.