Parallel Algorithms
TD 1-PRAM

Oguz Kaya
oguz.kaya@universite-paris-saclay.fr

Master 1 QDCS/2023-24

1 Sum of an array

Let $X[N]$ be an array of size N and $\otimes$ an associative operation.

Question 1

a) Design a parallel algorithm for an EREW PRAM machine, $\textsc{Sum}(X[N])$ to compute and return the sum of elements in the array, using $O(N)$ processors.

b) What is the parallel time/depth of the algorithm?

c) What is the work and cost of the algorithm?

2 Prefix scan on an array

Let $X[N]$ be an array of size N and $\otimes$ an associative operation.

Question 2

a) Design a parallel algorithm for an EREW PRAM machine, $\textsc{Scan}(X[N], \otimes)$ to compute and return the prefix scan array $X'[i] = X[1] \otimes \cdots \otimes X[i]$ for all i, using $O(N)$ processors and running in $O(\log_2 N)$ time. Verify that all reads and writes are exclusive. Do not treat the array as a linked list (Hint: Try using recursion on the array). Compute the work and cost of this algorithm. Is it work-optimal? Is it efficient? Can we reduce the the number of PUs to make it efficient?

b) This time, design an algorithm specifically for P processors that does the same computation in $O(N/P + \log_2 P)$ time, then compute its work and cost. Is it work-optimal? For what values of P does it stay efficient?

3 Read/write conflict resolution on an EREW PRAM

In the EREW PRAM model, it is forbidden to simultaneous read from/write to the same memory case by multiple PUs. However, in parallel algorithms, we often need to use a common variable (a global parameter, a flag in a loop, etc.) in all PUs. To do this, we can first create a copy of the variable for each PU, then access them in parallel without any conflicts in an EREW PRAM algorithm.

Question 3

a) Design an algorithm $\textsc{Broadcast}(x)$ for EREW PRAM that puts a copy of x in an array $px[i]$ for $1 \le i \le N$, and returns the array, so that PUs can use their corresponding copies in their subsequent computations with no read conflicts. What is the time complexity?

b) Design an algorithm $\textsc{Reduce}(px[N], \otimes)$ in EREW PRAM that merges the values $px[i]$ with the associative operation $\otimes$, i.e., $x = px[1] \otimes \cdots \otimes px[N]$, then returns x. What is the time complexity?

c) Design an algorithm $\textsc{Allreduce}(px[N], \otimes)$ that performs a reduction on the array $px[:]$, then returns an array of size N containing a copy of this value obtained from reduction. Try to aim to perform this in exactly the same number of time steps w.r.t. $\textsc{Broadcast}$ and $\textsc{Reduce}$ (not only asymptotically, but also up to the same constant factor). What is the work and cost of this algorithm? Is it work-optimal? Is it efficient?

4 Inverting a list

Let $next[N]$ be a linked list of size N.

Question 4

a) Design an algorithm $\textsc{Invert}(next[N])$ for an EREW PRAM machine which computes the inverse of this linked list, and returns it in a pointer array $prev[N]$ (e.g., if $next[3] = 7$, $prev[7] = 3$).

b) What is the execution time using N PUs?

5 Selection in a list using EREW PRAM

Let $next[N]$ represent a list with N nodes, each node i having a label $label[i] = red$ or $label[i] = blue$.

Question 5

a) Design a EREW algorithm $\textsc{SelectBlues}(next[N], label[N])$ that updates the list $next[N]$ so that in the end i) each blue node points to the next blue node in the original list, ii) each red node points to NIL, and iii) the algorithm returns the index of the head node of the updated list containing all blue nodes.

b) What is the execution time, work, and cost?

6 Separation in an array using EREW PRAM

Let $A[N]$ be an array of N objects with an associated function $\textsc{Label}(A[i]) = 0$ or $\textsc{Label}(A[i]) = 1$ for its elements.

Question 6

a) Design an EREW PRAM algorithm $\textsc{Separate01}(A[N])$ returns a permuted copy $A'[N]$ of this array containing all elements with label 0 in the beginning of the array (so that $\textsc{Label}(A'[i]) = \{0, \ldots, 0, 1, \ldots, 1\}$ for $1 \le i \le N$). Hint: Try to use the prefix scan algorithm do determine where to write each element after permutation.

b) What is the execution time, work, and cost?

7 Sorting an array of unique integers

Let $X[N]$ be an array of unique integers, meaning no two elements $X[i]$ and $X[j]$ are equal for $1 \le i \ne j \le N$.

Question 7

a) Find an algorithm $\textsc{Sort}(X[N])$ that returns the sorted array $X[N]$ using a CRCW PRAM. Specify the mode of fusion to employ for concurrent writes.

b) What is the execution time, work, and cost?

c) How can you make the algorithm work if elements in $X[N]$ are not unique?

d) How can you make the algorithm work for an CREW PRAM?

8 Parallel matrix multiplication

The multiplication of two square matrices $A, B \in \mathbb{R}^{N\times N}$ yields another square matrix $C = AB \in \mathbb{R}^{N\times N}$ whose elements C_{ij} are defined as follows:

$$C_{ij} = \sum_{k=1}^{N} A_{ik}B_{kj}, 1 \le i, j \le N$$

Question 8

a) Design a CRCW PRAM algorithm $\textsc{MatMul}(A[N][N], B[N][N])$ to carry out the multiplication of two $N \times N$ matrices $C = AB$, then return the matrix C. What is the execution time, work, and cost? To obtain this execution time, how many PUs are needed?

b) Do the same for a CREW PRAM machine. What is the execution time, work, and cost? How to make this algorithm work on a EREW PRAM machine as well?

9 Simulation of different fusion modes in CRCW

Consider a CRCW machine in consistent fusion mode, meaning simultaneous writes to a memory case are allowed if and only if the same value is written to the memory case.

Question 9

a) Prove that with a CRCW in consistent fusion mode, we can simulate the execution of a CRCW having the "minimum" fusion (meaning that it keeps the minimum value among all simultaneous writes), staying in the same time complexity as CRCW in consistent mode. You can use more PUs in the CRCW consistent fusion mode simulation than what is used in the CRCW minimum fusion mode execution.

10 Computing the depth of nodes in a tree

Let $parent[N]$ be the array of parent pointers in a tree having N nodes (such that $parent[i] = -1$ if the node i is the root of the tree, and $1 \leq parent[i] \neq i \leq N$ otherwise). The goal of this exercise is to compute the depth of all nodes in the tree in parallel. The depth of the root node is defined to be zero, and the depth of any other node is the depth of its parent plus one (i.e., $depth[i] = depth[parent[i]] + 1$).

Question 10

a) Design a CREW PRAM algorithm $\textsc{TreeDepth}(parent[N])$ to compute the depth of each node in the tree, and return it in an array. What is the execution time, work, and cost? Would your algorithm work on an EREW PRAM machine?

b) This time, assume that the tree is binary, and in addition to parent pointers, each node i also has a pointer towards its left and right children $left[i]$ and $right[i]$ (with $left[i] = -1$ if the node i has no left child, likewise for the right child). Design an EREW PRAM algorithm $\textsc{BinTreeDepth}(parent[N], left[N], right[N])$ to compute and return the depth of all nodes in the tree in $O(\log_2 N)$ time. Hint: Consider the depth-first traversal of the tree, and try to use the prefix scan algorithm on this traversal.

11 Parallel breadth-first search

The goal of this exercise is to perform a parallel breadth-first search (BFS) on a given directed graph $G(V, E)$ having $|V|$ vertices and $|E|$ edges $E[t] = (i_t, j_t)$ for $1 \leq t \leq |E|$ and $1 \leq i_t \neq j_t \leq |V|$.

Question 11

a) Design an algorithm $\text{BFS}(G(V, E), s)$ that performs a parallel BFS on a given directed graph G, starting from the vertex s ($1 \leq s \leq |V|$), on a CRCW PRAM machine. The algorithm should compute and return two arrays, $parent[i]$ which gives the parent of each node i in the BFS traversal tree, and $level[i]$ that represents the level of node i in the BFS tree (i.e., its distance from the starting vertex s). Specify the fusion mode to be employed.

b) Find the execution time, work, and cost of the algorithm.

12 Connected components of an undirected graph

In this example, we will develop a parallel algorithm to find the connected components of a given undirected graph $G(V, E)$ having $|V|$ vertices and $|E|$ edges $E[t] = (i_t, j_t)$ for $1 \leq t \leq |E|$ and $1 \leq i_t \neq j_t \leq |V|$ on a CRCW PRAM machine. In the end, we want the algorithm to i) assign a label $C[i]$ to each vertex i so that $C[i]$ holds the minimum vertex index in the component that vertex i belongs to and ii) return this label array.

Question 12

a) Develop a CRCW PRAM algorithm Connected-Components(V, E) that finds the correct component label $C[i]$ for each vertex $1 \leq i \leq |V|$, and returns this array.

b) What is the execution time?

c) What is the work of the algorithm? Is it work-optimal?

d) What is the cost of the algorithm? Is it efficient? Can it be made efficient using less PUs?

13 Sorting in EREW PRAM

Question 13

a) Develop a EREW PRAM algorithm Sort$(A[N])$ that sorts an array $A[N]$ with respect to increasing $A[i].key$ attribute in $O(\log N)$ time.

b) What is the work of the algorithm? Is it work-optimal?