Design and Analysis of Algorithms (COMP3001)

Tutorial 11
Parallel algorithms

Question 1

PARALLEL_ODD_EVEN_MERGE_SORT
INPUT: n data elements, each on its own PE_j , $1 \leq j \leq n$.
OUTPUT: Sorted data such that $PE_j \leq PE_{j+1}$ for $1 \leq j \leq n$.

```
1    for i ← 1 to ⌈n/2⌉
2        for all PEⱼ such that j is odd and 1 ≤ j <n
3            if PEⱼ.a > PEⱼ₊₁.a
4                swap(PEⱼ.a, PEⱼ₊₁.a)

5        for all PEⱼ such that j is even and 1 ≤ j <n
6            if PEⱼ.a > PEⱼ₊₁.a
7                swap(PEⱼ.a, PEⱼ₊₁.a)
```

a) Trace the execution of the above algorithm on the data [1, 5, 2, 3, 4, 6, 7, 4].

b) What is the time complexity of PARALLEL_ODD_EVEN_MERGE_SORT?

c) What is the work / cost complexity of PARALLEL_ODD_EVEN_MERGE_SORT?

d) Is PARALLEL_ODD_EVEN_MERGE_SORT work optimal? Is it work efficient?

e) What is the speed-up of PARALLEL_ODD_EVEN_MERGE_SORT?

f) What is the PE efficiency of PARALLEL_ODD_EVEN_MERGE_SORT?

Question 2

Algorithm Parallel_Search_CRCW (*x*, *A*[1 .. *n*])
   index ← -1   // initialized with an invalid index value
   forall P_i **do in parallel**   // $1 \le i \le n$
      if $A[i] = x$ **then**
         index ← *i*
      endif
   endfor

Algorithm Parallel_Search_EREW (*x*, *A*[1 .. *n*])
Broadcast(*x*, *B*[1 .. *n*]) // $O(\log_2 n)$; *n* PEs

// $1 \le i \le n$
forall P_i **do in parallel**   // $O(1)$; *n* PEs
   if $A[i] = B[i]$ **then**
      $B[i] ← i$
   else
      $B[i] ← \infty$
endfor

// PE with the smallest ID that finds $A[i] = x$
return $i = fan\_in (B[1 .. n])$ // $O(\log_2 n)$; *n* PEs

a)   Trace the execution of Parallel_Search_CRCW on array $A = [1, 5, 2, 3, 4, 6, 7, 4]$ and each of the following search keys:
- $x = 4$
- $x = 9$

b)   Explain why Parallel_Search_CRCW requires a CRCW model

c)   Trace the execution of Parallel_Search_EREW on array $A = [1, 5, 2, 3, 4, 6, 7, 4]$ and each of the following search keys:
- $x = 4$
- $x = 9$

Question 3

Consider a system with four distributed servers $\{S_1, S_{10}, S_{20}, S_5\}$, and 10 objects $\{x_0, x_4, x_7, x_6, x_{15}, x_{25}, x_{17}, x_{10}, x_{21}, x_3\}$ Assume consistent hashing where each identifier is using *m* = 5 bits, i.e., value [0 .. 31].

a)   What objects is each server responsible for?

b)   If server S_{10} leaves the system, which server (s) will be responsible for its objects?

c) If a new server S_{25} joins the system, what objects will the server be responsible for?

d) Describe the advantages and disadvantages of using consistent hashing as compare to the *conventional* hashing?