The University of Melbourne
COMP90025 Parallel and Multicore Computing
Semester 2, 2017 Final Examination

School of Computing and Information Systems
COMP90025 Parallel and Multicore Computing
Reading Time: 15 minutes
Writing Time: 3 hours

Open Book Status: Closed Book

This paper has 3 pages including this page

Identical Examination Papers: none
Common Content: none

Authorized Materials:
No materials are authorized.

Instructions to invigilators:
No papers may be taken from the exam room.

Instructions to students:
The total marks for this paper is 60.
All answers are to be written in the script book(s) provided.
Attempt all questions - partial credit is available.
Ensure your student number is written on all script books during writing time.
The examination is worth 60% of the subject assessment.

Paper to be held by Baillieu Library: yes

continued on next page

Q.1. **(a)** [**5 marks**] In lectures we sometimes talked about "processor optimal" or "optimal processor allocation". What does this mean, and what is the difference between this and the definition of optimality for PRAM algorithms?

(b) [**4 marks**] How does our notion of algorithm optimality change when, rather than shared memory, we consider algorithms that run on an architecture such as a mesh or hypercube?

(c) [**1 marks**] Provide the definition of *speed up*.

Q.2. **(a)** Consider the use of `#pragma omp parallel for` in OpenMP:

i. [**2 marks**] If each iteration of the loop takes a differing and potentially unpredictable amount of time, what can be done to avoid load imbalance over the threads?

ii. [**2 marks**] What is the purpose of the `ordered` directive in the case of a parallel for statement? Include a trivial example in your explanation.

iii. [**2 marks**] What is the purpose of the `collapse` directive in the case of a parallel for statement? Include a trivial example in your explanation.

(b) [**4 marks**] When dividing a problem space over a number of processors, e.g. an image, sometimes the space can be divided into square regions and sometimes it can be divided into columns (or strips). Assume that the time taken to send a message of k elements from one processor to another is $t_m = t_s + k\,t_d$, where t_s is startup time and t_d is the time taken per element. What is the condition on t_s for the square partitioning method to take less time than the column partitioning method? Show your working out and any further assumptions that you require.

Q.3. **(a)** [**5 marks**] Show how to sort n numbers in $\mathcal{O}(\log^2 n)$ time using an EREW PRAM.

(b) [**5 marks**] Show how to compute the prefix sum on an $n = 2^t$ node hypercube in time $\mathcal{O}(t)$.

Q.4. **(a)** [**10 marks**] Consider two processors, A and B, each having a sorted list of k integers. Consider the problem of finding the k smallest integers from the combined $2k$ integers, i.e. where the answer should be available on processor A. One trivial way to do this is for processor B to send its entire list to processor A, and then A merges them to obtain the smallest k. Assuming computation time is negligible, this would take time $t_m = t_s + k\,t_d$, where t_s is startup time and t_d is the time to send an integer. Assuming computation time is negligible, is there a faster way (on average) to obtain the result at A? Provide your reasoning and any further assumptions made.

(b) [**10 marks**] Consider an $n \times n$ mesh network, where each processor, (i, j), $i, j \in \{1, 2, \ldots, n\}$, contains an element $a_{i,j}$ of a matrix A. Assuming each edge of the mesh can send one element per time step (in both directions),

 continued on next page

describe the communication steps required to transpose the matrix over the mesh, i.e. so that processor (i, j) obtains $a_{j,i}$. How many steps in total does this take?

(**c**) [**10 marks**] Given a sequence Q of n numbers (positive and negative), the *maximum subsequence* of Q is the contiguous subsequence that has the maximum sum among all contiguous subsequences of Q. Devise an EREW PRAM algorithm for finding the maximum subsequence of Q in $\mathcal{O}(\log n)$ time. You may assume that Q contains 2^k elements where $k \geq 0$.

END OF EXAMINATION