DEPARTMENT OF COMPUTER SCIENCE

COMP-417 Parallel and Distributed Computing

Midterm Exam Study Guide

- Lectures 1 – 6.
- Quizzes mainly the theoretical questions.

Similar questions may appear on the exam

Structure of the exam

- **Part 1: Theoretical questions (about 60 points)**
    a. Questions *similar (but can cover ANY topics from the slides that may not have been asked in the quizzes)* to the quizzes, but you will be asked to elaborate the answer like:
        i. *compare and contrast* various terms/concepts
        ii. *justify your answer* (i.e. it is not enough to answer true/false, but you have to justify why the answer is true/false)
        iii. *demonstrate* some concept (you may need to draw a diagram or show all the steps/traces of a specific situation which results in the specific situation).
        iv. *Show the full calculation* i.e. if you have to calculate speedup/efficiency and/or using Amdahl's law you will need to show all the steps of your calculation.
    b. Compared to the quizzes the answers will be longer, because you will write the answers in a few short paragraphs instead of *point-and-click* or *drag-and-drop* which you did in the quizzes.

- **Part 2: Practical questions related to *Parallel Programming* (about 25 points)**
    a. You will be given a parallel program and asked to identify and fix errors like:
        i. Forgot to call *join* in main
        ii. Forgot to protect the *critical sections*
        iii. Lock too large of a region, making the program essentially sequential
        iv. Wrong order of *start* and *join* when using the ***divide-and-conquer*** style, making the program essentially sequential
    b. You will be given a parallel program and asked to identify optimizations to the program:
        i. Use a local counter during the looping and assign the result to the global variable (reduces the duration and number of times that the mutex is held).
        ii. Identify where you could put a ***sequential cutoff***

iii. In ***divide-and-conquer*** style program, only create half of the threads by using the "create one and do the other *yourself*" principle.
iv. Reason about how to improve the load balance between the threads or the appropriate number of threads for a specific scenario.

c. You will be given a sequential *algorithm/program* and asked to write a parallel version of the *algorithm/program* using the ***fork/join*** style of parallel programming.
d. You will be given a sequential *algorithm/program* and asked to write a parallel version of the *algorithm/program* using the ***divide-and-conquer*** style of parallel programming.

- **Part 3: Practical questions related to *Concurrent Programming* (about 15 points)**
  a. You will be given a concurrent program and asked to identify and fix errors like:
     i. Data Races
     ii. Bad Interleavings
     iii. When using locks explicitly, forgetting to release the lock due to an exception
  b. You will be given a concurrent program and asked to identify optimizations to the program:
     i. Change the code as to create a shorter critical section (i.e. the thread holds the lock for a shorter period of time)
  c. You will be given a concurrent program which are using explicit locking (i.e. using a mutex or semaphore) and asked to convert it to one that are using the ***synchronized*** statement instead.

The exam will test your knowledge in the above material as well as your ability to apply the concepts to solve new problems.