Faculty EWI
Department ST

Exam in4049TU, January 25, 2013
9:00 – 12:00 hours

Question 1.

(a) Give four (4) examples of models of parallel computation, explaining the main features for each one of them.

(b) Consider three arrays, A, B, and C, of length N. Explain how the element-wise addition of these two arrays can be parallelized using each of the chosen 4 models of parallel computation.

(c) Give two (2) examples of basic machine architectures, explaining the main features for each of them.

(d) Can each of the four models map on both architectures? Explain why or why not.

Question 2.

John and Mary have a shared bank account, currently having 500 Euro. While John shops at Albert Heijn, Mary is getting money from an ATM. These two transactions happen in parallel.

Transaction John:

```
READ ACCOUNT
ACCOUNT = ACCOUNT - 50
WRITE ACCOUNT
```

Transaction Mary:

```
READ ACCOUNT
ACCOUNT = ACCOUNT - 120
WRITE ACCOUNT
```

(a) What can go wrong because the transactions are in parallel? Give an example of a wrong execution and explain how it could happen.

(b) Can such an incorrect execution be avoided? If yes, please explain how.

(c) How would you implement the solution from b. in a shared memory programming model? How about a message passing one?

(d) Is there a different if the bank account is shared not only between two, but between 100 people?

Question 3.

ACME is a company that has just bought a new multi-core processor with 8 cores. Each core is capable of 4-way SIMD operations, and operates at 1GHz. The company has a weather simulation sequential application, Cloudy.

Cloudy uses 1 core and runs in 10s. Bill is a MSc student and does an internship at ACME. He has to parallelize Cloudy, but he needs some help predictind the performance he might achieve.

Here are some of Bill's questions:

(a) What is the peak performance of the processor ?

(b) If I parallelize 50% of the application (Cloudy) can be parallelized, what is the best speed-up I can achieve using all 8 cores, but no SIMD ?

(c) If I parallelize 80% of Cloudy (no SIMD), how many cores do I have to use for (Cloudy) to run in at most 4s?

(d) Assuming I can use all 8 cores, 50% SIMD, and can parallelize 100% of the application, how many processors should I use to get the execution time below 0.2s ?

Question 4.

(a) When parallelization is done using the domain decomposition approach the so-called interface points between neighbour-subdomains must be exchanged. This can be correctly implemented by first communicating the value of 1 interface point whenever this value is required for the update of a grid point on a neighbour-subdomain. Does this lead to optimal parallel efficiency? If not, how can it be improved?

(b) Describe the main differences between the parallelization of the Jacobi algorithm and the Gauss-Seidel algorithm.

(c) Consider a 2-dimensional grid of n by n points. Suppose that we want to parallelize the Jacobi method using the 1-D (strip-partitioning) and 2-D (block-partitioning) domain decomposition for a parallel computer with P processors. Assume that the update of a grid point during one iteration takes 5 time units, and the communication of m numbers takes $a+m*b$ time units. If you can choose between 1-D and 2-D partitioning, in which cases will you choose a 2-D partitioning scheme? (PS. if you assume a certain configuration of these P processors in your answer please specify it).

(d) Assume the communication time of m numbers is equal to $T_{comm}(m)=0.0005+m*10^{-7}$ (s) and 1 floating-point operation takes 10^{-9} s. If the block partitioning scheme has been applied for the parallelization of the Jacobi method and the multi-grid method, will these two parallel iterative methods have the same data locality ratio? (i.e., the ratio between computation time to communication time).

Question 5.

(a) What are the most important differences between the parallel FFT algorithm with a block distribution and the parallel transposition FFT algorithm? Describe the communication phase(s) of the parallel transposition algorithm for 1-D FFT.

(b) Describe the main steps of a multi-level graph partitioning algorithm. What is the main reason of using a multi-level algorithm instead of directly partition a graph into a number of subgraphs?

(c) One of the parallelization approaches for the Barnes-Hut algorithm uses the Orthogonal Recursive Bisection (ORB) for the spatial partitioning. What is the purpose of using ORB? Describe (or illustrate) the ORB scheme.

(d) If a 9-point finite difference scheme instead of a 5-point difference scheme is used for the discretization of a 2-D domain, will the application of Red-Black ordering result in an efficient parallel Gauss-Seidel iteration? Please explain. (PS. In a 9-point scheme, a grid point has 8-neighbours, the 4 diagonal grid points are also neighbours in this case).

-end of exam-