Assignment 1 (Part 1): Theoretic Exercises

Lars Karlsson, Mikael Rännar, Bo Kågström

2009-04-06

Assignment 1 consists of a number of compulsory theretic exercises gathered from the course literature and elsewhere. Solutions will be discussed during a compulsory oral session scheduled at Monday 6th of April (see the schedule for details). After the oral session you should hand in your solutions in written form (computer printed or nicely formatted hand written).

The exercises are meant to be solved in pairs of two and the main purpose is to give continuity to the theoretic studies and to increase your understanding of scalability and analysis thereof.

1 Part 1

- Exercise 4.6 on page 191.
- Exercise 5.10 on page 231.
- Exercise 5.13 on page 231, also.
- Exercise 8.5 on page 373.
- Describe an algorithm to find the maximum among n numbers on a $\sqrt{n} \times \sqrt{n}$ mesh. Is your algorithm cost optimal?
- The matrix multiplication $C = C + AB$ where A, B, C are $n \times n$ matrices can be performed in parallel on a ring of p processors. Assume that the matrices A and C from the start are distributed using a row block distribution so that processor P_i holds row block i. The matrix B is distributed using a column block distribution so that processor P_i holds column block i.
  1. Construct a parallel algorithm to perform the matrix multiplication on a ring of processors. Use figures to show that you understand the data distribution scheme.
  2. Assume that addition and multiplication together takes one unit of time. Compute the problem size $W = T_s$, the parallel runtime T_p, the speedup S_p, and the efficiency E_p. Assume the communicational cost model $t_s + t_w m$ for an m-word message.
  3. Decide if the algorithm is cost optimal or not. If it is cost optimal, determine the relation (if any) between W and p which makes the system cost optimal. Find the asymptotic iso-efficiency function (and don't forget to consider the maximum degree of concurrency).
  4. Cannon's algorithm for matrix multiplication on a mesh mapped to a hypercube is reported in the course literature to have a parallel runtime of
  $$T_p = \frac{n^3}{p} + 2\sqrt{p}t_s + 2t_w\frac{n^2}{\sqrt{p}}.$$
  The asymptotic iso-efficiency function is reported to be $W = \Theta(p\sqrt{p})$. Try to explain the difference between your ring algorithm and Cannon's algorithm that gives rise to their different scalability.