Department of Mathematics and Computer Science
University of Southern Denmark
Daniel Merkle

Due on: Wednesday 1. October, 12:00 p.m. (Department secretaries office (Lone Seidler Petterson) or my office).

Exercises or parts of exercises marked with * are voluntary exercises.

For preparing solutions for the following exercises you need to read the complete Chapter 3 and Sections 4.1 and 4.2 from the course book.

Exercise 1

Task Dependency Graph (5+5=10 points)

$$A = \begin{array}{|c|c|c|c|c|c|c|c|}\hline \bullet & \bullet & & & & & & \\ \hline \bullet & & & & & \bullet & & \\ \hline & & & \bullet & & & & \\ \hline & & & & \bullet & & & \\ \hline & & & \bullet & & & & \\ \hline & & & & \bullet & \bullet & & \\ \hline & & & & & & & \bullet \\ \hline & & & & & \bullet & \bullet & \\ \hline \end{array} \qquad B = \begin{array}{|c|c|c|c|c|c|c|c|}\hline & & \bullet & & & & & \\ \hline \bullet & & & & & & & \\ \hline & & \bullet & \bullet & & & & \\ \hline & \bullet & & & \bullet & & & \\ \hline & & & & & & & \\ \hline & & & & \bullet & & & \\ \hline & & & & & & \bullet & \\ \hline & & & & & \bullet & & \\ \hline \end{array}$$

Given are the two sparse matrices A and B. Consider the problem of sparse matrix-matrix multiplication. A dot corresponds to a non-zero entry. The computation is decomposed into 8 tasks. Let task i the owner of row $A[i, *]$ and of row $B[i, *]$. Task i has to compute row i of the result $C = A \cdot B$.

a) Draw the task interaction graph using directed edges. Draw an edge from task T_i to task T_j, if T_i requires data from T_j.

b) Suppose that task i owns column i of matrix B instead of row i for the computation. Draw the task-interaction graph for this case.

c) * Which decomposition should (usually) to be preferred? Explain why.

Exercise 2

LU factorization (5+3+3+3+3+2+2+4=25 points)

$$\begin{pmatrix} A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,1} & A_{3,2} & A_{3,3} \end{pmatrix} \rightarrow \begin{pmatrix} L_{1,1} & 0 & 0 \\ L_{2,1} & L_{2,2} & 0 \\ L_{3,1} & L_{3,2} & L_{3,3} \end{pmatrix} \cdot \begin{pmatrix} U_{1,1} & U_{1,2} & U_{1,3} \\ 0 & U_{2,2} & U_{2,3} \\ 0 & 0 & U_{3,3} \end{pmatrix}$$

1: $A_{1,1} \rightarrow L_{1,1}U_{1,1}$	6: $A_{2,2} = A_{2,2} - L_{2,1}U_{1,2}$	11: $L_{3,2} = A_{3,2}U_{2,2}^{-1}$
2: $L_{2,1} = A_{2,1}U_{1,1}^{-1}$	7: $A_{3,2} = A_{3,2} - L_{3,1}U_{1,2}$	12: $U_{2,3} = L_{2,2}^{-1}A_{2,3}$
3: $L_{3,1} = A_{3,1}U_{1,1}^{-1}$	8: $A_{2,3} = A_{2,3} - L_{2,1}U_{1,3}$	13: $A_{3,3} = A_{3,3} - L_{3,2}U_{2,3}$
4: $U_{1,2} = L_{1,1}^{-1}A_{1,2}$	9: $A_{3,3} = A_{3,3} - L_{3,1}U_{1,3}$	14: $A_{3,3} \rightarrow L_{3,3}U_{3,3}$
5: $U_{1,3} = L_{1,1}^{-1}A_{1,3}$	10: $A_{2,2} \rightarrow L_{2,2}U_{2,2}$	

Given is the decomposition of the LU factorization into 14 tasks. (We assume that each of the 14 tasks requires the same unit amount of work).

a) Draw the task dependency graph.

b) Determine all critical paths.

c) Determine the average and the maximal degree of concurrency.

d) Describe/draw an efficient mapping of the task-dependency graph of the decomposition onto three processes.

e) Describe/draw an efficient mapping of the task-dependency graph of the decomposition onto four processes.

f) Which of the both mappings solves the problem faster?

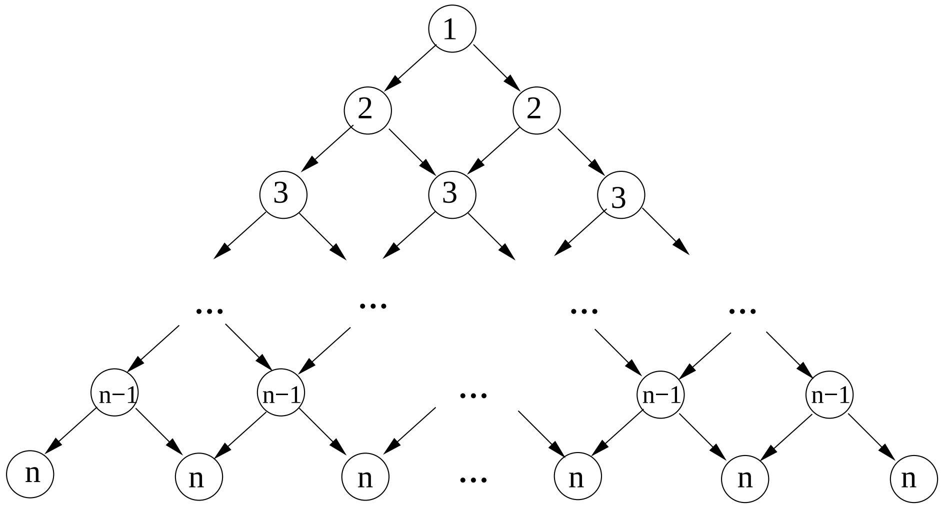
g) What is the maximal speedup that can be achieved and how many processes are necessary for that speedup?

h) What is the maximal efficiency, that can be achieved, if $p > 1$ processes are used? Describe/draw the mapping that you used.

Exercise 3

Task Dependency Graph (2+2+3+3=10 points)

Given is the following task dependency graph:



a) Determine the maximal degree of concurrency.

b) What is the length of the critical path?

c) Determine the average degree of concurrency.

d) What is the maximal speedup that can be achieved, and what is the corresponding efficiency when this speedup is realized?

Exercise 4

All-to-All Broadcast (5+5+5=15 points)

On a ring, all-to-all broadcast can be implemented in two different ways: (i) the standard ring algorithm as shown in Figure 4.9 in the course book, and (ii) the hypercube algorithm as shown in Figure 4.11. in the course book.

a) What is the run time for case (i)?

b) What is the run time for case (ii)?

If k messages have to traverse the same link at the same time, then assume that the effective per-word-transfer time for these messages is $k \cdot t_w$. Also assume that $t_s = 100 \cdot t_w$.

c) Which of the two methods, (i) or (ii), is better if the message size m is very large? Which method is better if m is very small (may be one word)? Explain.

Exercise 5*

All-to-All Broadcast on a Tree (voluntary, but not very complicated)

Given a balanced binary tree as shown in Figure 4.7 from the course book, describe a procedure to perform all-to-all broadcast that takes time $(t_s + t_w \cdot m \cdot p/2) \log p$ for m-word messages on p nodes. Assume that only the leaves of the tree contain nodes, and that an exchange of two m-word messages between any two nodes connected by bidirectional channels takes time $t_s + t_w \cdot m \cdot k$ if the communication channel (or a part of it) is shared by k simultaneous messages.