ECE445: PARALLEL AND DISTRIBUTED COMPUTING
Fall 2021-2022

Homework 1:

Design of Parallel Algorithms – Implicit Parallelism

(*Deadline: Tuesday 30.11.2021, 23:59)*

General Instructions:

Follow the template text (template.docx) since this is part of your grade.

Write clearly your names and your IDs on the first page.

In the theoretical exercises, write in detail your answers and the sources from which you drew them.

In the programming ones, present and explain the algorithm you programmed and present in detail the experimental results of your programs.

Describe your implementation and the machine you worked on (hardware features, OS, compiler, etc.).

Program in C and Linux, use Makefile to compile / link / execute your programs.

Report your observations and present the results with tables and graphs.

Graphs and figures must be software-generated and not handmade and scanned.

In general, your text should be well written and legible, while you should FULLY justify the steps you followed, and comment on the results of each exercise. (20 credits)

Submit your work via eclass to a zip file (hw1_id1_id2_id3.zip), which will contain the text with your solutions / results and comments, your code (.c, .h files and your Makefile).

It is noted that the teams do not change during the semester.

Question 1. (5)

Describe the basic steps in designing a parallel algorithm. What should we look out for in each of these stages?

Question 2. (5)

Which are the basic properties of mapping two graphs? Describe what each of these reveals with respect the workload and the communication needs, considering that the first graph represents the individual processes of a parallel algorithm and the second one represents a system of computing nodes with their wiring while the embedding shows the assignment of processes to computing nodes.

Question 3. (10)

Assume the symmetric matrix A from Figure 1 below and the problem of matrix-vector multiplication Ab = c, where b and c are vectors of 8 elements. Assume that we divide the problem by block of lines of matrix A and elements of b and c, into 2 processors in the obvious way of Fig. 1. Create the dependence graph of the rows/columns of matrix A. How will the specific multiplication operation be performed in the 2 processors and what are the communication requirements?

Study the dependence graph of the rows/columns in matrix A that you made before and present another (non-obvious) partition of the problem with less communication. Which rows/columns will be assigned to each processor? What are the communication requirements?

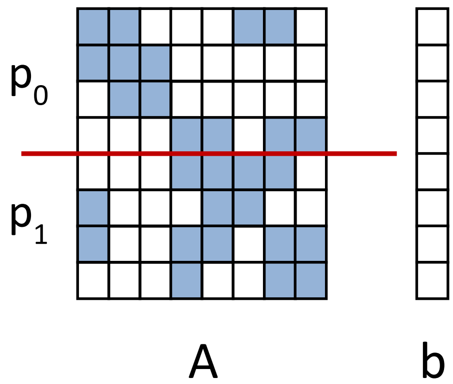
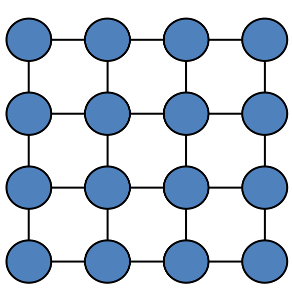


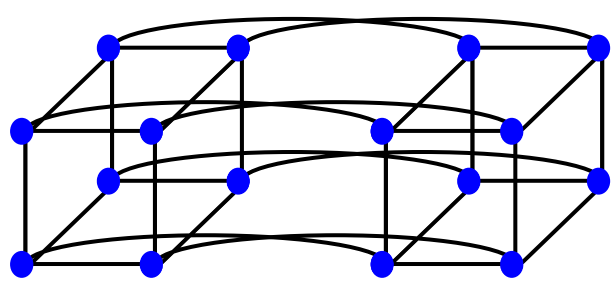
Figure 1. Sparse symmetric matrix A.

Question 4. (15)
Graph the solution to the following problem:
If (xi, yi), i = 1,2,…, n points from the graph of a function, calculate its integral using the trapezoidal rule.
Explain in details what task each node of the graph is working on and use the directional arrows to indicate process dependency and data exchange.
Calculate maximum and average degree of parallelism for general n, while creating your graphs for n = 4.
Try writing a parallel algorithm for this problem running on p processors. What is the relation between p and n so there is uniform work load? Assume this relation for your study.

Question 5. (25)
Assume 2 graphs G (V, E) and G '(V', E '). G is a 2dimensional grid of 16 vertices as in Fig. 2, and G' is a 4th dimension hypercube. Calculate the mapping between the nodes of the 2 graphs so that all three properties of mapping are equal to 1. For the 2D grid, express the nodes' id of the Cartesian coordinate type, while for the hypercube, consider to express the ids of the nodes in binary format, as shown in slides.
a) (10) Write the formula that maps the ids.
b) (15) Write a general mapping that works for general cases, where the hypercube is of k dimensions and the 2D grid has 2^k nodes while k is an even number.



2D πλέγμα



4D-κύβος

Figure 2. 2D grid, G(V,E), hypercube 4th dimension, G' (V', E).

Exercise 6. (60)

Consider the problem of computing the norm 1 of an n by n matrix A, i.e.,

$$\|A\|_1 = \max_{1 \le j \le n} \sum_{i=1}^{n} |a_{ij}|$$

The 1-norm can be calculated with 2 loops. To study the effect of the loop step, we will perform the loop of the sums with different steps (k = 1, 2, 4, 8, 16) with k separate commands.

a. (20) Write corresponding functions that calculate the sums of the elements of each column with a different step k and measure the execution time.

b. (10) Graph the execution time of norm calculation with respect to k.

c. (10) What is the number of operations (addition or multiplication) with floating point numbers in your program?

d. (5) What is the maximum speed of your program in FLOPS (Floating Point Operations Per Second)?

e. (5) Which k or ks are considered critical for efficiency improvement?

f. (10) What is the reason of expecting an improvement in efficiency?

g. Make sure your operations and results are correct. Answer each sub-question separately and clearly justify your remarks.

Hint 1: Write code to create large and dense matrices (e.g., try n = 10^3, 10^5, 10^8, etc. until your computer can not respond) dense with unit units, i.e., $\alpha_{ij} = 1$, $1 \le i, j \le n$, so that $\|A\|_1 = n$. Use this code in your program to work with large matrices and make sure you make the right calculations with execution time measurements as accurately as possible.

Hint 2: Visit netlib.org's LAPACK / BLAS sdot function to figure out how to hcreate for loops with different steps k.
http://www.netlib.org/lapack/explore-html/d0/d16/sdot_8f_source.html

Exercise 7. (60)

Consider the problem of computing the max norm of an n by n matrix A, i.e.,

$$\|A\|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}|$$

The max-norm can be calculated with 2 loops. To study the effect of the loop step, we will perform the loop of the sums with different steps (k = 1, 2, 4, 8, 16) with k separate commands.

a. (20) Write corresponding functions that calculate the sums of the elements of each line with a different step k and measure the execution time.

b. (10) Graph the execution time of norm caqlculation with respect to k.

c. (10) What is the number of operations (addition or multiplication) with floating point numbers in your program?

d. (5) What is the maximum speed of your program in FLOPS (Floating Point Operations Per Second)?

e. (5) Which k or ks are considered critical for efficiency improvement?

f. (10) What is the reason of expecting an improvement in efficiency?

g. Make sure your operations and results are correct. Answer each sub-question separately and clearly justify your remarks.

Hint 1 : As in Hint 1 of exercise 6.

Question 8. (10)

Is there any difference in execution time between the 2 programs of exercises 6 and 7? Justify your answer.