Parallel Computer Systems, Fall 2022
Instructions for Databar Exercise 3:
Thinking parallel

September 17, 2022

1 Introduction

In this databar exercise, you will explore how to turn a small sequential program into a parallel program. You will do that at a conceptual level without writing code. This is a normal first step in writing parallel code. We will also profile the code which is essential for making informed decisions on what parts of a program to make parallel.

You will report on this exercise as part of the first report. You will there write about your experiences and findings. Furthermore, you will have to address the questions asked in this exercise. It is expected that you write up to one page for this exercise.

Read through this document in entirely before starting working on the exercises!

2 Learning objectives

During this assignment you will be working towards the following learning objectives:

- Use performance models such as Amdahl's law to predict parallel performance.
- You can write down in text and in your own words definitions or explanations of the following concepts: speedup.
- You can solve ill-structured, ill-formed, open problems related to distributed systems and parallel programming.

3 Report and rules

You will have to hand in written reports during the course. The reports should describe the assignments you have completed. Begin writing on the report as early as possible. You should also maintain a progress diary!

DTU has a zero tolerance policy on cheating and plagiarism. This also extends to the report and indeed all your work. To copy text passages or source code from someone else without clearly and properly citing your source is considered plagiarism. See the study hand book for further detail.

4 Reference material

Before carrying out the exercise study section 5-5.3.5 in Hager&Wellein.

5

Download `calc.c` from DTU Learn.

Study the code carefully.

Your task is to conceptually write a parallel version running on an arbitrary number of processors. You do not write code but rather you informally describe what parts of the code remain serial and what parts can be made parallel. You can, but you are not required to, use pseudo code if you feel that is clearer.

What opportunities for parallelism do you see? What problems or challenges do you see limiting performance? How could the problems be overcome?

Measure the time it takes to execute different parts of the serial program on the HPC machines. If a part of a program has very short execution time, how can you still measure the time relatively accurately?

Use your measurements to estimate performance of your parallel approach. How efficient do you expect your code to be? Why? How many times faster than the sequential version do you expect your code to run on N processors? N being the number of processors available to you. In other words, what speedup do you expect?

Can you enforesee any overheads that can make your estimations inaccurate? What could these overheads be, if any?

6 License