Christian Fischer Pedersen, cfp@ece.au.dk

1 Exercise: Characterization of distributed systems

1. Discuss the pros and cons of distributed versus centralized systems.
2. Discuss the relationship between distributed systems, parallel systems, cluster computing systems, grid computing systems, and cloud computing systems.
3. Discuss the relationship between distributed, pervasive, and ubiquitous systems.
4. Discuss the relationship between parallel and concurrent systems
5. Discuss pros and cons of different forms of transparency in distributed systems.
6. Discuss the fallacies of distributed systems.
7. Discuss current examples of noteworthy distributed systems.
8. Discuss current examples of noteworthy pervasive systems.
9. Discuss Flynn's taxonomy in general and specifically in relation to distributed systems.
10. Make a small processing-step example for each of Flynn's four computer architectural classes SISD, SIMD, MISD and MIMD by taking the concepts "instruction stream", "data stream" and "processing unit" into account.
11. Discuss the consequences of Amdahl's law for distributed systems.
12. Discuss the relationship between Amdahl's and Gustafson's laws and the consequences of Gustafson's law for distributed systems.
13. Discuss the consequences of Moore's, Koomey's and Kryder's laws for distributed systems.
14. Discuss three types of hardware resources, three types of data resources, and three types of software resources that can usefully be shared. Give examples of their sharing as it occurs in practice in distributed systems.
15. Consider the WWW as an example to illustrate the concepts of clients, servers, and resource sharing. Discuss the pros and cons of the technologies HTML, URL and HTTP for information browsing? Are any of these technologies suitable as a basis for client-server computing in general?

2 Exercise: Amdahl's law

1. If there is a tight sequential coupling between steps in a computational problem it is called *inherently serial*. What is the serial fraction, f, in such a problem?
2. If there is no sequential coupling between steps in a computational problem it is called *perfectly parallel* (also called *embarrassingly parallel*). What is the serial fraction, f, in such a problem?
3. What is the theoretical speedup in execution latency of a program if 25% of the original execution time is made twice as fast?
4. The analysis of a program has shown a speedup of 3 when running on 4 cores. What is the serial fraction (best case) according to Amdahls law?

5. Assume 10%, 20%, and 50%, respectively, of a program's run-time is non-parallelizable and the program is supposed to run on a super-computer with 100,000 cores. Also, assume that the program runs at the same speed on all of the cores, and there are no additional overheads. Plot a graph that illustrates the theoretical parallel speedup as a function of number of cores.

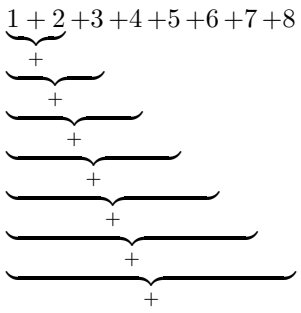
6. Assume 0.1% of a program's run-time is non-parallelizable, that the program is supposed to run on a super-computer with 100,000 cores, and that the program runs at the same speed on all of the cores. Assume also that the program invokes a broadcast operation which adds overhead as a function of the number of cores involved. There are two broadcast implementations available. One adds a parallel overhead of $BC1_{OH}(P) = 0.0001P$, and the other adds $BC2_{OH}(P) = 0.0005\log(P)$. Find, for each of the broadcast implementations, the number of cores that maximizes the speed-up.

3 Exercise: Gustafson's law

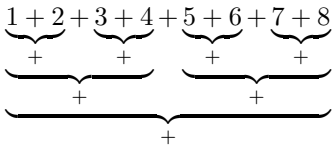
1. Assume a program with a serial fraction of 50%
   (a) Compute the speed-up when using 2 and 4 processors according to Amdahl's law.
   (b) Compute the speed-up when using 2 and 4 processors according to Gustafson's law under the assumption that the parallel work per processor is fixed.
   (c) Why are both the speed-up results different?
2. The analysis of a program has shown a speedup of 3 when running on 4 cores. What is the serial fraction according to Gustafson's law?

4 Exercise: Critical path

1. Consider an integer sequence $s = (i)_{i=1}^{8}$ of length $|s| = 8$. Two alternatives for adding the elements of s are:



and



   (a) Determine the length of the critical path for both computational alternatives.
   (b) Determine the length of the critical path for both computational alternatives for a sequence of length n.

   This exercise is inspired by the parallel programming course at ETH, 2020.

5 Solutions to exercises in section 2

Problem 1: The serial fraction is $f = 1$.

Problem 2: The serial fraction is $f = 0$.

Problem 3: If the parallelizable fraction, $1 - f = 0.25$, is to run twice as fast, $P = 2$, then theoretical net speedup is 1.14.

Problem 4: The serial fraction is $f = \frac{1}{9}$

Problem 6: The serial fraction is 0.1% and the number of cores is P. For each of the broadcast implementations we get a speed-up function:

- $\text{Speedup}(f,P)\big|_{BC1(P)} = \dfrac{1}{f + \frac{1-f}{P} + 0.0001P}$
- $\text{Speedup}(f,P)\big|_{BC2(P)} = \dfrac{1}{f + \frac{1-f}{P} + 0.0005\log(P)}$

To find the number of cores that maximizes the speed-up for each of the broadcast implementations we can minimize the denominators:

$$Re.BC1(P) : \frac{\partial}{\partial n}\left(0.001 + 0.999\frac{1}{P} + 0.0001P\right) = 0 \Leftrightarrow 0.0001 - 0.999\frac{1}{P^2} = 0 \Leftrightarrow P \approx 100$$
$$Re.BC2(P) : \frac{\partial}{\partial n}\left(0.001 + 0.999\frac{1}{P} + 0.0005\log(P)\right) = 0 \Leftrightarrow \frac{0.005P - 0.999}{P^2} = 0 \Leftrightarrow P \approx 1998$$

6 Solutions to exercises in section 3

Problem 1a: $\text{Speedup}(f,P)\big|_{f=0.5,P=2} = \frac{4}{3}$. $\text{Speedup}(f,P)\big|_{f=0.5,P=4} = \frac{8}{5}$

Problem 1b: $\text{Speedup}(f,P)\big|_{f=0.5,P=2} = \frac{3}{2}$. $\text{Speedup}(f,P)\big|_{f=0.5,P=4} = \frac{5}{2}$

Problem 1c: Amdahl's law sees the percentage of non-parallelizable code as a fixed limit for the speedup (cf. the Amdahl corollary). So even if we had an infinite amount of processors, according to Amdahl's law, the speedup would never be greater than $\text{Speedup}(f,P) \leq \frac{1}{0.5} = 2$. On the other hand Gustafson's law assumes that the parallel part of the program increases with the problem size and the serial part stays fixed. The fundamental assumption of Amdahl's law is that the serial part of the executable work is given by a fixed ratio of the **overall** work. The fundamental assumption of Gustafson's law is that the serial part of the executable work is given by a fixed ratio of the work **before the work size is increased**.

Problem 2: $\text{Speedup}(f,P) = f + P(1-f) = P - f(P-1) \Rightarrow 3 = 4 - 3f \Leftrightarrow f = \frac{1}{3}$

7 Solutions to exercises in section 4

Problem 1a: In the first compute-alternative we always have to wait for the previous addition to finish, before we can start the following one. This causes us to have seven additions that can not be executed in parallel, which form the critical path. In the second version, we can do all additions on the same level at the same time. We finish the whole task after 3 levels which is the length of the critical path.

Problem 1b: In the first compute-alternative we add all the numbers in a serial fashion; therefore the length of the critical path is equal to the length, n, of the sequence minus one:

$$\text{Critical path length} = n - 1$$

In the second compute-alternative we do as many independent additions in parallel as possible resulting in a binary task tree structure. The critical path is equal to the height of this tree which is:

$$\text{Critical path length} = \lceil \log_2(n) \rceil$$