Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Parallel Programming
Assignment 5: Basic Parallel Programming Concepts
Spring Semester 2020

Assigned on: **18.03.2020**

Due by: **(Wednesday Exercise) 23.03.2020**
(Friday Exercise) 25.03.2020

Task 2 – Amdahl's and Gustafson's Law

Assuming a program consists of 50% non-parallelizable code.

a) Compute the speed-up when using 2 and 4 processors according to Amdahl's law.

Answer: Amdahl's law says: $S_p \leq \frac{W_{ser} + W_{par}}{W_{ser} + \frac{W_{par}}{p}} = \frac{1}{f + \frac{1-f}{p}}$

Therefore we have

$S_2 \leq \frac{1}{\frac{1}{2}+\frac{1}{4}} = \frac{4}{3} \approx 1.33$

and

$S_4 \leq \frac{1}{\frac{1}{2}+\frac{1}{8}} = \frac{8}{5} = \approx 1.6$

b) Now assume that the parallel work per processor is fixed. Compute the speed-up when using 2 and 4 processors according to Gustafson's law.

Answer: Gustafson's Law says: $S_p = p - f(p-1)$

So we have

$S_2 = 2 - \frac{1}{2}(2-1) = \frac{3}{2} \approx 1.5$

and

$S_4 = 4 - \frac{1}{2}(4-1) = \frac{5}{2} \approx 2.5$

c) Explain why both speed-up results are different.

Answer: Amdahl's law sees the percentage of non-parallelizable code as a fixed limit for the speedup. So even if we had an infinite amount of processors, according to Amdahl's law, the speedup would never be greater than 2.
On the other hand Gustafson's law assumes that the parallel part of the program increases with the problem size and the sequential part stays fixed.

A more formal explanation:[*]

The fundamental assumption of Amdahl's law is that the sequential part of the executable work W is given by a fixed ratio f of the **overall** work W provided: $W_s = W \cdot f$, $W_p = W \cdot (1 - f)$.

Assuming it would take time T to process W on a single core, we compute the time required to process work W_p with p processors as $T_p = T \cdot f + T \cdot (1 - f)/p$. Thus $T_1/T_p = \frac{1}{f+(1-f)/p}$

The fundamental assumption of Gustafson's law is that the sequential part of the executable work W is given by a fixed ratio f of the work W **before the work size is increased** to $W_p = W \cdot p$ when p processors are available: $W_s = W \cdot f$, $W_p = W \cdot p \cdot (1 - f)$.

Assuming it would take time T to process W on a single core, we compute the time required to process work W_p with p processors as $T_p = T \cdot f + T \cdot p \cdot (1 - f)/p = T$ while with only one processor we get $T_1 = T \cdot f + T \cdot (1 - f) \cdot p$, thus $T_1/T_p = f + (1 - f) \cdot p$.

[*]For the following arguments, we assume that, homogeneously over all processors, the time T [e.g. in seconds] to process some work W [e.g. in instructions] is proportional to W. Moreover, we assume parallelism without overheads.

Task 3 – Amdahl's and Gustafson's Law II

a) The analysis of a program has shown a speedup of 3 when running on 4 cores. What is the serial fraction according to Gustafsons law?

Answer: Gustafson's Law: $S_p = p - f(p-1)$

$$\begin{aligned}
S_p &= p - f(p-1) \\
3 &= 4 - 3f \\
3f &= 4 - 3 \\
f &= \frac{1}{3}
\end{aligned}$$

b) The analysis of a program has shown a speedup of 3 when running on 4 cores. What is the serial fraction according to Amdahls law (assuming best possible speedup)?

Answer: Amdahls Law: $S_p \leq \frac{1}{f + \frac{1-f}{p}}$

$$\begin{aligned}
S_p &\leq \frac{1}{f + \frac{1-f}{p}} \\
3 &= \frac{1}{f + \frac{1-f}{4}} \\
3 &= \frac{1}{\frac{3f+1}{4}} \\
3 &= \frac{1}{\frac{3f+1}{4}} \\
3 &= \frac{4}{3f+1} \\
9f + 3 &= 4 \\
9f &= 1 \\
f &= \frac{1}{9}
\end{aligned}$$

Task 4 – Task graph

Assuming you want add eight numbers, then two options to do this are

$$\underbrace{\underbrace{\underbrace{\underbrace{\underbrace{\underbrace{\underbrace{1+2}_{+}+3}_{+}+4}_{+}+5}_{+}+6}_{+}+7}_{+}+8}_{+}$$

and

$$\underbrace{\underbrace{\underbrace{1+2}_{+}+\underbrace{3+4}_{+}}_{+}+\underbrace{\underbrace{5+6}_{+}+\underbrace{7+8}_{+}}_{+}}_{+}$$

a) Given those two variants, determine the length of the critical path for both computations.

Answer: In the first version 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.

b) For a sequence of length n, determine the length of the critical path using the approaches from above.

Answer: In the first version we add all the numbers in a serial fashion - therefore the length of the critical path is equal to the length of the set of numbers minus one.

$$n-1$$

In the second version we do as many independent additions in parallel as possible resulting in a tree structure for the task graph. The critical path is equal to the height of this tree which is

$$\lceil \log_2(n) \rceil$$