University of Lleida

Master's Degree in Informatics Engineering

Higher Polythecnic School

Problem of Metrics

High Performance Computing

Francesc Contreras

Albert Pérez

March 14, 2021

Table of contents

1 Problem 1

A 25% of a program can not be parallelized and the rest can be distributed between different number of processors.

Which is the maximum possible gain with the parallelization?

In general terms, ***Amdahl's Law*** states that in parallelization, if P is the proportion of a system or program that can be made parallel, and *(1-P)* is the proportion that remains serial, then the maximum speedup *S(N)* that can be achieved using N processors is:

$$S(N) = 1/((1 - P) + (P/N))$$

So, when *(P/N)* is approching zero

$$S_M ax = 1/(1 - P) = 1/(1 - 0.75) = 1/0.25 = 4$$

Which is the minimum number of processors needed to achieve a gain over 2?

In this case, there is an inequality that is expressed as follows:

$$1/[(1 - P) + (P/N)] >= 2$$

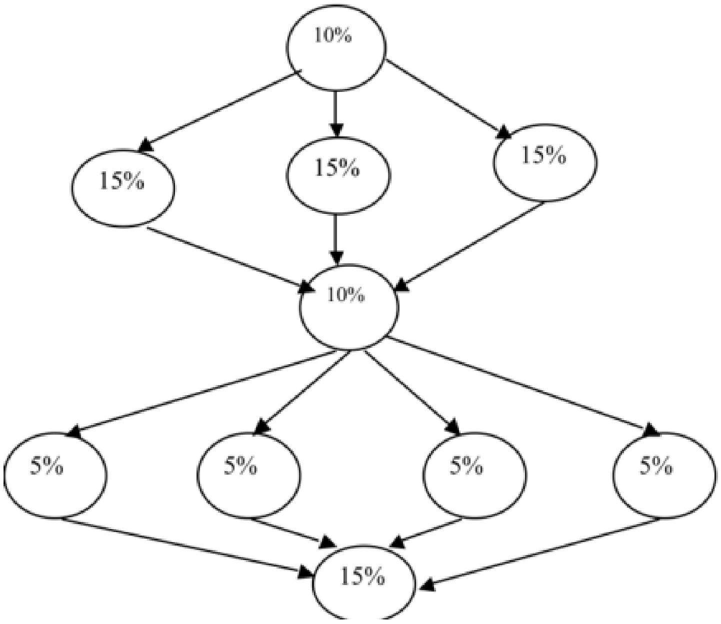
So,

$$1/[(1 - 0.75) + (0.75/N)] >= 2$$
$$1/2 <= 0.25 + (0.75/N)$$
$$N >= 0.75/0.25 >= 3$$

As it is observed, a minimum of 3 processors are required to obtain a gain of 2. If the number of processors rises, the gain will be better but taking into account that its maximum is 4 as its calculated in the previous question.

2 Problem 2

The following graph represents a task dependence graph of an application. The graph shows the percentage of scalar time spent by each task.



]

Figure 1: Task dependency graph

Assume a scalar time of 60s and a negligible communication time.

Obtain the parallel execution time and the speedup for a computer with 4 and 2 processors.

First we will calculate according to the graph and considering the scalar time, the total execution time depending on the number of processors, which implies the tasks that can be parallelized at each time unit.

Finally, the speedup will be calculated by comparing the sequential and parallel execution according to each case.

2.1 4 processors

For each unit time, the execution one it's calulated as:

$$0.1 * 60 = 6s$$

$$0.15 * 60 = 9s$$

$$0.1 * 60 = 6s$$

$$0.05 * 60 = 3s$$

$$0.15 * 60 = 9s$$

Therefore, the total amount of time is,

$$6 + 9 + 6 + 3 + 9 = 33s$$

So,

$$S = 60/33 = 20/11 = 1.81$$

2.2 2 processors

The execution time is calculated for each unit time but now with 2 processors as:

$$0.1 * 60 = 6s$$

$$0.15 * 60 * 2 = 18s$$

$$0.1 * 60 = 6s$$

$$0.05 * 60 * 2 = 6s$$

$$0.15 * 60 = 9s$$

Then, the total is,

$$6 + 18 + 6 + 6 + 9 = 45s$$

So,

$$S = 60/45 = 4/3 = 1.33$$