Performance Metrics

Amdahl's law. Assume an application where the execution of floating-point instructions on a certain processor P consumes 60% of the total runtime. Moreover, let's assume that 25% of the floating-point time is spent in square root calculations.

- Based on some initial research, the design team of the next-generation processor $P2$ believes that they could either improve the performance of all floating point instructions by a factor of 1.5 or alternatively speed up the square root operation by a factor of 8. From which design alternative would the aforementioned application benefit the most?
- Instead of waiting for the next processor generation the developers of the application decide to parallelize the code. What speedup can be achieved on a 16-CPU system, if 90% of the entire program can be perfectly parallelized? What fraction of the code has to be parallelized to get a speedup of 10?

Efficiency. Consider a computer that has a peak performance of 8 GFlops/s. An application running on this computer executes 15 TFlops, and takes 1 hour to complete.

- How many GFlops/s did the application attain?
- Which efficiency did it achieve?

Parallel efficiency Given the data in Tab. 1, use your favorite plotting tool to plot

a) The scalability of the program (speedup vs number of processors)

b) The parallel efficiency attained (parallel efficiency vs number of processors)

In both cases plot also the ideal case, that is, scalability equal to the number of processors and parallel efficiency equal to 1, respectively.

# Processors	Best seq.(1)	2	4	8	16
# GFlops/s	4.0	7.6	14.9	23.1	35.6

Table 1: Performance attained vs number of processors.

Weak scalability. Algorithm $\mathcal{A}$ takes an integer k as input, uses $O(k^2)$ workspace, and has cubic complexity in k. Let $T_p(k)$ be the execution time of $\mathcal{A}$ to solve a problem of size k with p processes. It is known that $T_1(8000) = 24$ minutes. Assuming perfect weak scalability, what is the expected execution time for $k = 128000$?

Metrics.

1. A given program was run on a node consisting of 4 multi-core processors, each of which comprises 10 cores. The peak performance of the node is 320 GFlops/s. To study the scalability of the program, it was run using 1, 2, 4, 8, 16, 24, 32 and 40 of the cores, for fixed problem size. For this problem size, the program executes 22×10^{12} flops. The execution time for each number of cores is collected in Tab. 2. Complete the table with the performance, speedup, efficiency, and parallel efficiency for each case.

# Cores	Time (s)	Perf. (GF/s)	Eff.	Speedup	Par. Eff.
1	34566.227				
2	17133.639				
4	8535.915				
8	4326.953				
16	2206.018				
24	1531.882				
32	1133.534				
40	943.301				

Table 2: Study of performance, speedup and efficiency.

- What is your opinion of the attained efficiency?
- What is your opinion of the scalability (speedup)?