EPFL

Parallel and High Performance Computing

Dr. Pablo Antolín

Solution Series 0

February 20 2025

Some basic concepts

Exercise 1: *Big O notation*

a) Define in one sentence time and space complexities of an algorithm.

- Time complexity: The impact of more data on the time required to complete an algorithm (total execution time).
- Space complexity: The impact of more data on the memory required to complete an algorithm.

b) Assume you have two algorithms A1 (with time complexity $\mathcal{O}_t(N \log N)$ and space complexity $\mathcal{O}_s(N)$) and A2 (with $\mathcal{O}_t(N^3)$ and $\mathcal{O}_s(N^3)$). Which one is worth to be parallelized?

Algorithm A1 is close to linear in terms of time and linear in memory. The second A2 presents a cubic complexity for both time and memory. Thus the second (A2) is probably a better candidate.

Exercise 2: *Vocabulary*

a) Why a DAG (Direct Acyclic Graph) is useful for the analysis of a parallelization strategy for a particular algorithm?

A DAG contains all the tasks involved in an algorithm. Once all dependencies (relations between tasks) have been defined, it is straightforward to point the independent tasks that can be executed in parallel.

b) What is the critical path in a DAG?

It is the longest execution time path throughout the tasks.

c) What are the three main kinds of parallelism?

- Bit-level parallelism: SIMD in the register (like SSE or AVX)
- Task parallelism or pipelining (Instruction-level parallelism)

- Data parallelism (loop parallelism)

d) What is one of the main solutions for solving the "irregular parallelism" problem?

Introducing load balancing for all the tasks involved in the parallel computation should lead to the same amount of work to perform by each process.

Exercise 3: ***Machine model***

a) What ILP, TLP, and DLP stand for?

- ILP: Instruction-level parallelism (how many instructions can be executed simultaneously on a CPU).
- TLP: Thread-level (or Task-level) parallelism (how many tasks can be executed simultaneously on the same data).
- DLP: Data-level (or vector) parallelism (one instruction is executed on a vector of data in one clock cycle).

b) What is a vector intrinsic?

An instruction of the processor (CPU) performed on a vector of data. For instance, the Intel intrinsic `_mm256_fmadd_pd(a,b,c)` performs a fused multiply-add $a = a + (b \times c)$ on 256 bits long vectors (4 double precision floating points).

c) Sort the following memories according to their speed of access: DRAM, L2, L1, L3, SSD.

L1, L2, L3, DRAM, SSD

d) Which are the 4 types of parallel architectures according to Flynn's taxonomy? Provide an example of each.

	Single Data	Multiple Data
Single intruction	SISD	SIMD
Multiple intruction	MISD	MIMD

- SISD: one intruction in one core of a CPU.
- SIMD: vector intrinsics (like Intel AVX).
- MISD: very rare!!
- MIMD: distributed memory machines, like clusters.

Exercise 4: ***Threads and processes***

a) What are the main differences between thread and process?

Process	Thread
each process has its own memory space	each thread uses the process' memory
heavyweight operations	lightweight operations
context switching is expensive	context switching is cheaper
no memory sharing among processes	each thread uses the process' memory

Exercise 5: ***Performance aspects***

a) What is the definition of speedup?

   If T_1 is the (best) sequential execution time and T_n the execution time on n processes, then the speedup is defined as $S = T_1/T_n$. The closer to n, the better.

b) What is the definition of parallel efficiency?

   The parallel efficiency is defined by $E = \frac{T_1}{nT_n} = S/n$. The closer to 1, the better.

Exercise 6: ***Performance aspects of the 2D Poisson solver***

Introductory remark: The communication between processors (steps 2 and 3) is a mechanism to transfer data from one processor to/from another. We will have much insight into several libraries and tools for performing such operations in the coming weeks.

a) Compute the computational complexity of the algorithm ("big O" notation) in time.

   The complexity in time is $\mathcal{O}(N^2)$. This is determined by the two nested `for` loops in each iteration. If the size of the grid is doubled, the number of computations is 4 times larger.

b) Compute the computational complexity of the algorithm ("big O" notation) in space.

   The complexity in space is $\mathcal{O}(N^2)$, due to the requirement of storing data for $N \times N$ points in the grid. If the size of the grid is doubled, the memory requirement increases a factor 4.

c) Is this problem worth to be parallelized?

   Due to the computational complexities in $\mathcal{O}(N^2)$, the 2D Poisson problem is worth to be parallelized.