The concurrency challenges

At the beginning the computers used just one core to run their code, by the time the programs became more though to process, an idea to use two cores appeared as a revolution.

Use two or more cores represented a powerful idea to execute a program. The ideas came at the same time as the challenges. Questions as How do I split a program to be run in both cores? Can I run two or more processes by the same time in one core? Appeared as a result of this innovation. Questions that gave birth new paradigms as parallelism and concurrency.

Programmers affronted the challenge with a “black line” discovering and dominating the technique. On these steps, debugging was harder, but the paradigm became more known and the elements to construct programs with parallelism and concurrency became easier by the time.

The parallelism raised to solve a problem of efficiency in the execution time of a process. But move a process from CPU to another sometimes became less efficient that keep it in the current CPU. These problems not only were in the software but in the hardware, the physical components needed to be more powerful to do what the software wanted to do what the programmer wanted. Unfortunately, by that time, companies inverted more money to develop software than hardware (almost 10 times). This was a barrier that didn’t let the movement flows fast.

Nowadays, this challenge of use parallelism is already well studied and used in multiple problems as face recognition, MapReduce paradigm, rendering, GPU core systems, etc.

Is recommended to use libraries already implemented and proved for parallel-programming in your code always taking care of the architecture of your system to achieve a high performance in your instances.