Introduction

The terms *concurrency* and *parallelism* are often used in relation to multithreaded programs. But what exactly does concurrency and parallelism mean, and are they the same terms or what?

The short answer is, **NO**. They are not the same terms, although they appear quite similar on the surface.

Concurrency

Concurrency is when two tasks can start, run, and complete in overlapping time periods. It doesn't necessarily mean they will ever both be running at the same instant.

For example, Multitasking on a single-core machine. If the computer has only one CPU (single-core machine), only one application will make progress at one time (because only one process is executing at a time), but the impression that user will get is that more than one applications (processes) are being processed at a time.

It is because the CPU keeps moving to other applications (in timesharing manner) before completing one application. This shows that one than one applications are executing concurrently.

Parallelism

Parallelism is when more than one tasks literally run at exactly the same time. For example, if my machine is dual-core, than there are two main threads executing two tasks at the level of CPU at any one time. If one application can be split into two smaller tasks that can be processed in parallel (e.g, In quick sort algorithm, partition logic divide the array in two parts, these two parts are then sorted using quick sort. Sorting of these two parts is independent of each other and hence can be done parallel, at exactly the same time on two different CPUs of a dual core machine).

The term Parallelism refers to techniques to make programs faster by performing several computations in parallel. ***This requires hardware with multiple processing units.*** In many cases the sub-computations are of the same structure, but this is not necessary. Graphic computations on a GPU are parallel.

More...

The term Concurrency refers to techniques that make program more usable. Concurrency can be implemented and is used a lot on single processing units, nonetheless it may benefit from multiple processing units with respect to speed.

- ✓ If an operating system is called a multi-tasking operating system, this is a synonym for supporting concurrency.
- ✓ If you can load multiple documents simultaneously in the tabs of your browser and you can still open menus and perform more actions, this is concurrency.
- ✓ If you run distributed net computations in the background, that is concurrency.

Concurrency and parallelism are NOT the same thing. Two tasks T1 and T2 are concurrent if the order in which the two tasks are executed in time is not predetermined,

- ✓ T1 may be executed and finished before T2,
- ✓ T2 may be executed and finished before T1,
- ✓ T1 and T2 may be executed simultaneously at the same instance of time (parallelism),
- ✓ T1 and T2 may be executed alternatively,

If two concurrent threads are scheduled by the OS to run on one single-core non-SMT non-CMP processor, you may get concurrency but not parallelism.

Parallelism is possible on ***multi-core***, ***multi-processor*** or ***distributed systems***. Concurrency is often referred to as a property of a program, and is a concept more general than parallelism.

Concurrent Programming v/s Parallel Programming

Interestingly, we cannot say the same thing for concurrent programming and parallel programming. They are overlapped, but neither is the superset of the other. The difference comes from the sets of topics the two areas cover. For example, concurrent programming includes topic like signal handling, while parallel programming includes topic like memory consistency model. The difference reflects the different original hardware and software background of the two programming practices.