**Difference between Concurrency and Parallelism**

Last Updated : 25 Nov, 2020

**Concurrency:**   
Concurrency relates to an application that is processing more than one task at the same time. Concurrency is an approach that is used for decreasing the response time of the system by using the single processing unit. Concurrency is creates the illusion of parallelism, however actually the chunks of a task aren’t parallelly processed, but inside the application, there are more than one task is being processed at a time. It doesn’t fully end one task before it begins ensuing.   
Concurrency is achieved through the interleaving operation of processes on the central processing unit(CPU) or in other words by the context switching. that’s rationale it’s like parallel processing. It increases the amount of work finished at a time.

A group of rectangular boxes with green text

Description automatically generated

In the above figure, we can see that **there is multiple tasks making progress at the same time.** This figure shows the concurrency because concurrency is the technique that deals with the lot of things at a time.

**Parallelism:**   
Parallelism is related to an application where  tasks are divided into smaller sub-tasks that are processed seemingly simultaneously or parallel. It is used to increase the throughput and computational speed of the system by using multiple processors. It enables single sequential CPUs to do lot of things “seemingly” simultaneously.

Parallelism leads to overlapping of central processing units and input-output tasks in one process with the central processing unit and input-output tasks of another process. Whereas in concurrency the speed is increased by overlapping the input-output activities of one process with CPU process of another process.

A diagram of a task

Description automatically generated

In the above figure, we can see that **the tasks are divided into smaller sub-tasks that are processing simultaneously or parallel.** This figure shows the parallelism, the technique that runs threads simultaneously.

**Difference between Concurrency and Parallelism:-**

| **S.NO** | **Concurrency** | **Parallelism** |
| --- | --- | --- |
| 1. | Concurrency is the task of running and managing the multiple computations at the same time. | While parallelism is the task of running multiple computations simultaneously. |
| 2. | Concurrency is achieved through the interleaving operation of processes on the central processing unit(CPU) or in other words by the context switching. | While it is achieved by through multiple central processing units(CPUs). |
| 3. | Concurrency can be done by using a single processing unit. | While this can’t be done by using a single processing unit. it needs multiple processing units. |
| 4. | Concurrency increases the amount of work finished at a time. | While it improves the throughput and computational speed of the system. |
| 5. | Concurrency deals lot of things simultaneously. | While it do lot of things simultaneously. |
| 6. | Concurrency is the non-deterministic control flow approach. | While it is deterministic control flow approach. |
| 7. | In concurrency debugging is very hard. | While in this debugging is also hard but simple than concurrency. |

***Concurrency***

**Concurrency** means working or processing on things time by time like some time for this process(work) some time for this process(work) in simple words Concurrency means dealing with multiple task time by time peridically.

Example: Imagine you are cooking and also watching TV. You cook for a few minutes, then watch TV for a few minutes, and keep switching between the two tasks. Both tasks are making progress, but not necessarily at the same time.

***Parallelism***

**Parallelism** means working on multiple things at same time. both process is work on different cpu.

Example: Imagine two people in a kitchen. One person is cooking while the other is watching TV. Both activities are happening simultaneously, without any switching between the two.

may be this image clear your remaining doubts.

