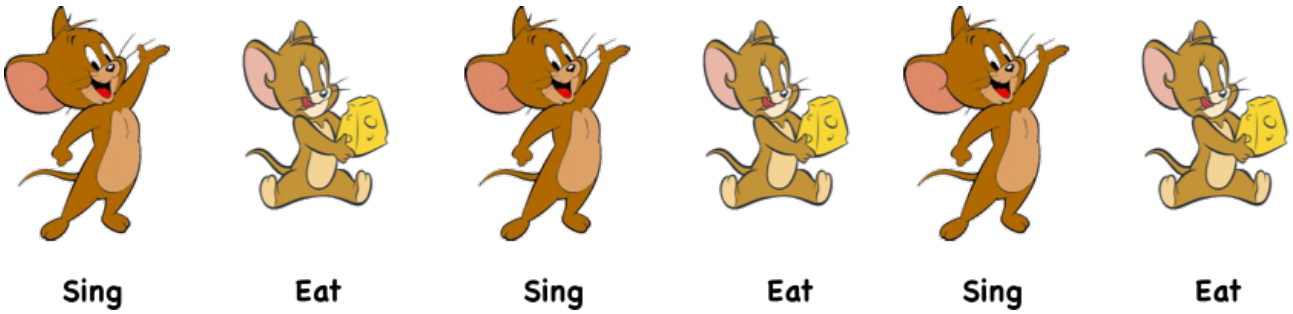
Concurrency vs. Parallel

Concurrency

let's take an example in real life: There's a challenge that requires you to both eat a whole huge cake and sing a whole song. You'll win if you're the fastest who sings the whole song and finishes the cake. So the rule is that you sing and eat simultaneously, How you do that does not belong to the rule. You can eat the whole cake, then sing the whole song, or you can eat half a cake, then sing half a song, then do that again, etc.



Concurrency means executing multiple tasks at the same time but not necessarily simultaneously.

Parallelism

If we keep going with the same example as above, the rule is still singing and eating concurrently**,** but this time, you play in a team of two. You probably will eat and let your friend sing (because she sings better and you eat better). So this time, the two tasks are really executed simultaneously, and it's called *parallel*.

Parallelism requires hardware with multiple processing units, essentially. In single-core CPU, you may get concurrency but NOT parallelism. Parallelism is a specific kind of concurrency where tasks are really executed simultaneously.

Now The Question is..??? What is the difference between Concurrency and Parallel..???

The key concept and difference between these definitions is the phrase **“in progress.”**

This definition says that, in concurrent systems, multiple actions can be *in progress* (may not be executed) at the same time. Meanwhile, multiple actions are simultaneously executed in parallel systems. In fact, concurrency and parallelism are conceptually overlapped to some degree, but “in progress” clearly makes them different.

Concurrency is about **dealing with lots of things** at once. Parallelism is about **doing lots of things at once**.

An application can be concurrent — but not parallel, which means that it processes more than one task at the same time, but no two tasks are executing at the same time instant.

An application can be parallel — but not concurrent, which means that it processes multiple sub-tasks of a task in multi-core CPU at the same time.

An application can be neither parallel — nor concurrent, which means that it processes all tasks one at a time, sequentially.

An application can be both parallel — and concurrent, which means that it processes multiple tasks concurrently in multi-core CPU at the same time.