**TestNG allows the tests to run in parallel or concurrent mode.**

This means that based on the test suite configuration, different threads are started simultaneously, and the test methods are executed in them. This gives a user a lot of advantages over normal execution, mainly reduction in execution time and the ability to verify a multi-threaded code.

**Advantages of Parallel Tests Execution**

1. **Reduces execution time -** As tests are executed in parallel, multiple tests get executed simultaneously, hence reducing the overall time taken to execute the tests.
2. **Allows multi-threaded tests** – Using this feature, we can write tests to verify certain multi-threaded code in the applications.

The ***parallel***attribute on the **<suite>** tag can take on of the following **values**:

<**suite** name="My suite" parallel="methods" thread-count="5">

* **parallel="methods"**: TestNG will run all your test methods in separate threads. Dependent methods will also run in separate threads but they will respect the order that you specified.

<**suite** name="My suite" parallel="tests" thread-count="5">

* **parallel="tests"**: TestNG will run all the methods in the same <test> tag in the same thread, but each <test> tag will be in a separate thread. This allows you to group all your classes that are not thread safe in the same <test> and guarantee they will all run in the same thread while taking advantage of TestNG using as many threads as possible to run your tests.

<**suite** name="My suite" parallel="classes" thread-count="5">

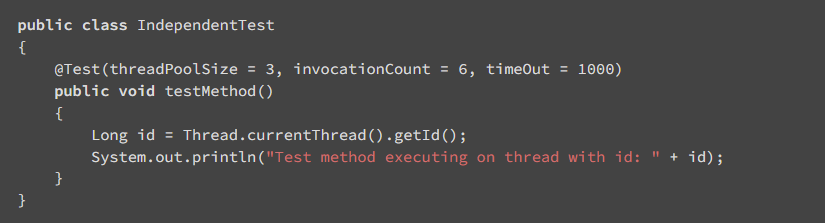
* **parallel="classes"**: TestNG will run all the methods in the same class in the same thread, but each class will be run in a separate thread.

<**suite** name="My suite" parallel="instances" thread-count="5">

* **parallel="instances"**: TestNG will run all the methods in the same instance in the same thread, but two methods on two different instances will be running in different threads.

Additionally, the attribute *thread-count* allows you to specify how many threads should be allocated for this execution.

**A single test can be executed in multiple threads**, and this is achieved by configuring it while using the **@Test annotation on a method.**



* **threadPoolSize** is set to 3: this configures the test method to be run in three different threads.
* **iInvocationCount** and **timeOut** configure the test to be invoked a multiple number of times and fail if the execution takes more time.