**JMeter Load Testing: Advanced Step-By-Step Guide**

OPEN SOURCE AUTOMATION

PERFORMANCE TESTING

The Stepping Thread Group and the Concurrency Thread Group are convenient and stable ways to find out how much user capacity your system can handle during Apache [JMeter load testing](https://www.blazemeter.com/blog/jmeter-plugins-manager), which is a convenient load testing solution.

By adding users in steps, we know the exact number of virtual users tested when a bottleneck is discovered. If we test in a linear fashion, it is difficult to determine the precise number of users that causes issues in the system.

This blog will delve into how you can use the Stepping Thread Group and Concurrency Thread Group as part of your advanced Apache [JMeter load testing](https://www.blazemeter.com/blog/jmeter-variables) efforts.

**What is JMeter Load Testing?**

JMeter load testing is a testing process that determines whether or not web applications under test can satisfy high load requirements. Techniques used within JMeter to determine issues in the system include the Stepping and Concurrency Thread Groups.

**How to Do Load Testing in JMeter**

Let us take a step-by-step look at how to do load testing in JMeter using the Stepping Thread Group.

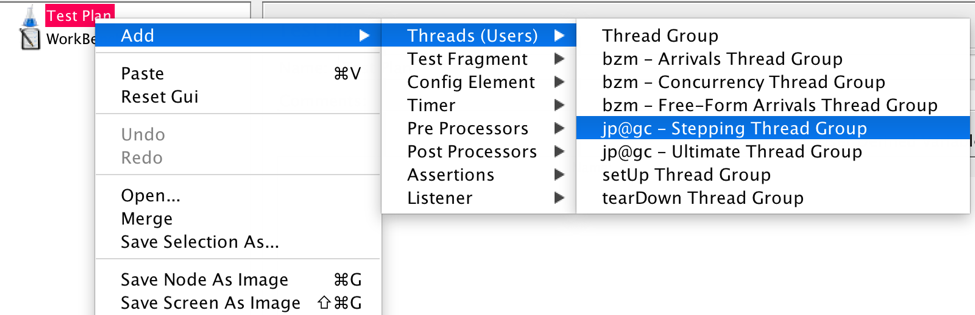
While the standard Thread Group lets us control the:

* Number of threads
* Linear ramp-up period
* Duration

The Stepping Thread Group lets us:

* Determine the same parameters as the standard Thread Group.
* Increase thread load by portions.
* Configure hold target load time.
* Decrease load by portions.

1. Install Customer Thread Groups from the [JMeter Plugins Manager.](https://www.blazemeter.com/blog/jmeter-plugins-manager)
2. Add the Stepping Thread Group from the Test Plan.



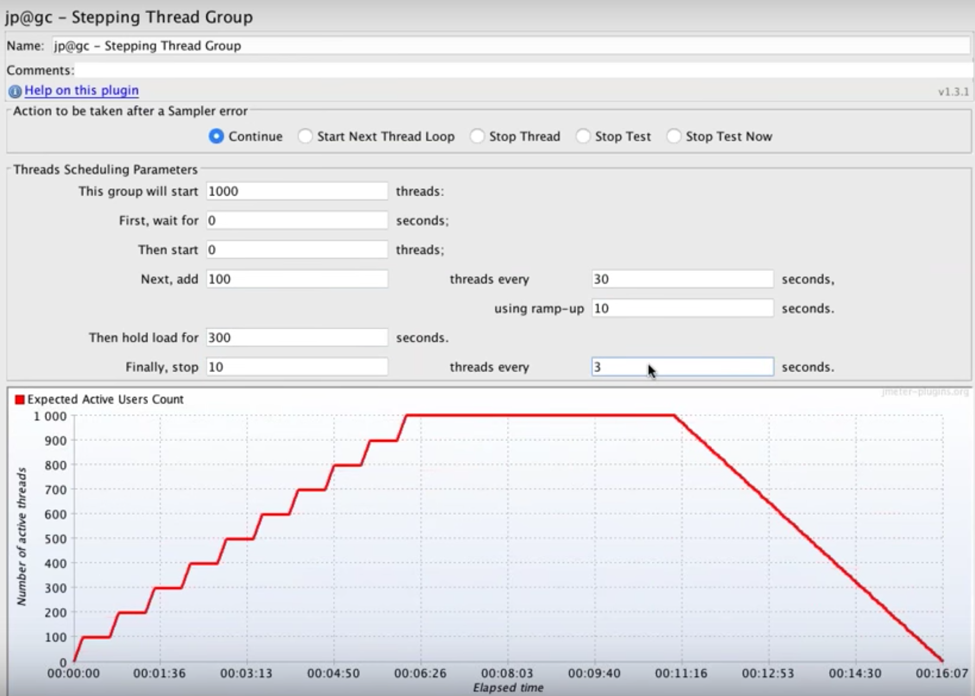
Let’s look at the following scenario:

* 1,000 threads as target load
* 0 seconds waiting after the test starts
* 0 threads run at the immediate beginning of the test
* 100 threads are added every 30 seconds with a ramp-up (or step transition time) of 10 seconds
* The target load is held for 300 seconds (5 minutes)
* Finally, 10 threads are stopped every 3 seconds.

This means that:

* The test begins immediately when JMeter starts, since there is 0 seconds waiting.
* Every 30 seconds 100 users will be added, until we reach 1,000 users. The first step is 1-100, the second 101-200, etc., because we defined 0 threads to run at the beginning. If we defined 50 threads to run the first step would be 51 - 150, the second 151 - 250, etc. and the test would be completed faster.
* It will take each of the steps (with 100 users each) 10 seconds to complete. After that JMeter waits 30 seconds before starting the next step.
* After reaching 1,000 threads all of them will continue running and hitting the server together for 5 minutes.
* At the end, 10 threads will stop every 3 seconds.

The Stepping Thread Group shows us the JMeter load test in a real-time preview graph.



By changing the values, you can ramp up the load to monitor your server metrics.

Now that test creation is complete, you're ready to launch the test. Consider adding the [Active Threads Over Time graph](https://jmeter-plugins.org/wiki/ActiveThreadsOverTime/) for tracking your load schedule’s performance and making sure it’s running as expected.

Using the Stepping Thread Group on BlazeMeter requires slight configuration.

**How to Add Concurrency Thread Group in JMeter**

You can also use the [Concurrency Thread Group](https://jmeter-plugins.org/wiki/ConcurrencyThreadGroup/) during JMeter load testing, which is a more modern alternative to the Stepping Thread group.

The Concurrency Thread Group provides a better simulation of user behaviour because it lets you control the length of your test more easily, and it creates replacement threads in case a thread finishes in the middle of the process. In addition, the Concurrency Thread Group doesn’t create all threads upfront, thus saving on memory.

The parameters that you can define are:

* Target Concurrency (Number of Threads).
* Ramp Up Time: for the whole test.
* Ramp-Up Steps Count.
* Hold Target Rate Time.
* Time Unit: minutes or seconds.
* Thread Iterations Limit (Number of Loops).
* Log Threads Status into File: saving thread start and thread stop events as a log file.

The Concurrency Thread Group doesn’t provide the ability to define initial delay, step transition and ramp-down — which the Stepping Thread Group does.

Add the Concurrency Thread Group from the test plan.