# Part 1:

There are several ways to solve the problem

1. Using for - loop from 1 to 1000000
2. Recursively add number from 1000000 down to 0 (stop condition)
3. Mathematical formula n\*(n+a)/d  
   In this case n = 1000000, a = 1, d = 2

For the sake of simplicity, I would like to pick option #3 as it is simple and does not require expensive computer resources.

Pseudo code should look like this (In Groovy script)

long end = 1000000, start = 1, divide = 2, sum = 0;

sum = end\*(end+start)/divide;

# Part 3:

1. Load testing

Using SoapUI, create a Load testing project, SoapUI will create Load Test for every testcase.

Create test scenarios then configure data for the scenario in the right panel.

Set test Parameters:

+ Duration – 10 minutes.

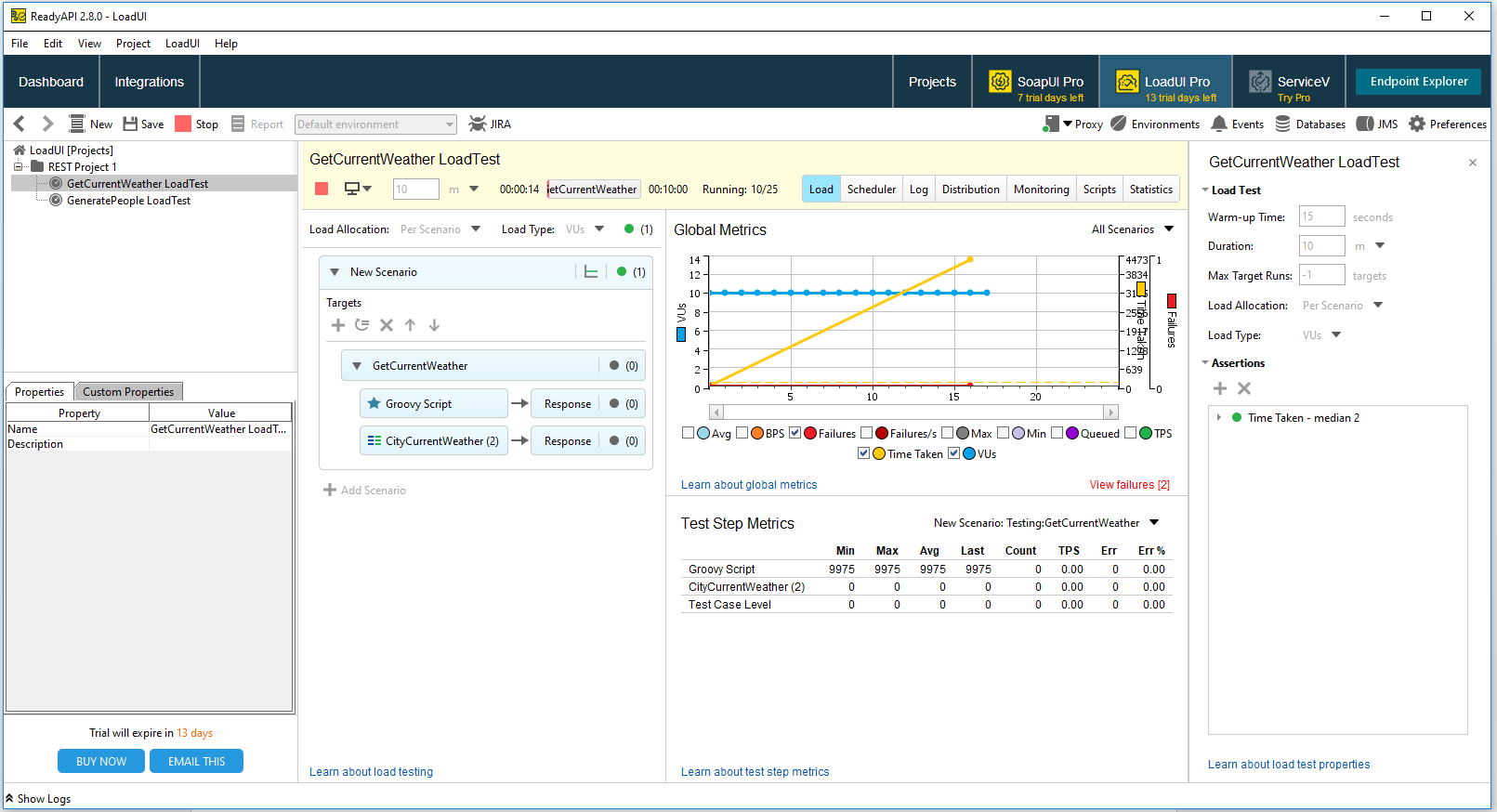
+ Load type – VUs.

+ Simulated load – 10 virtual users.

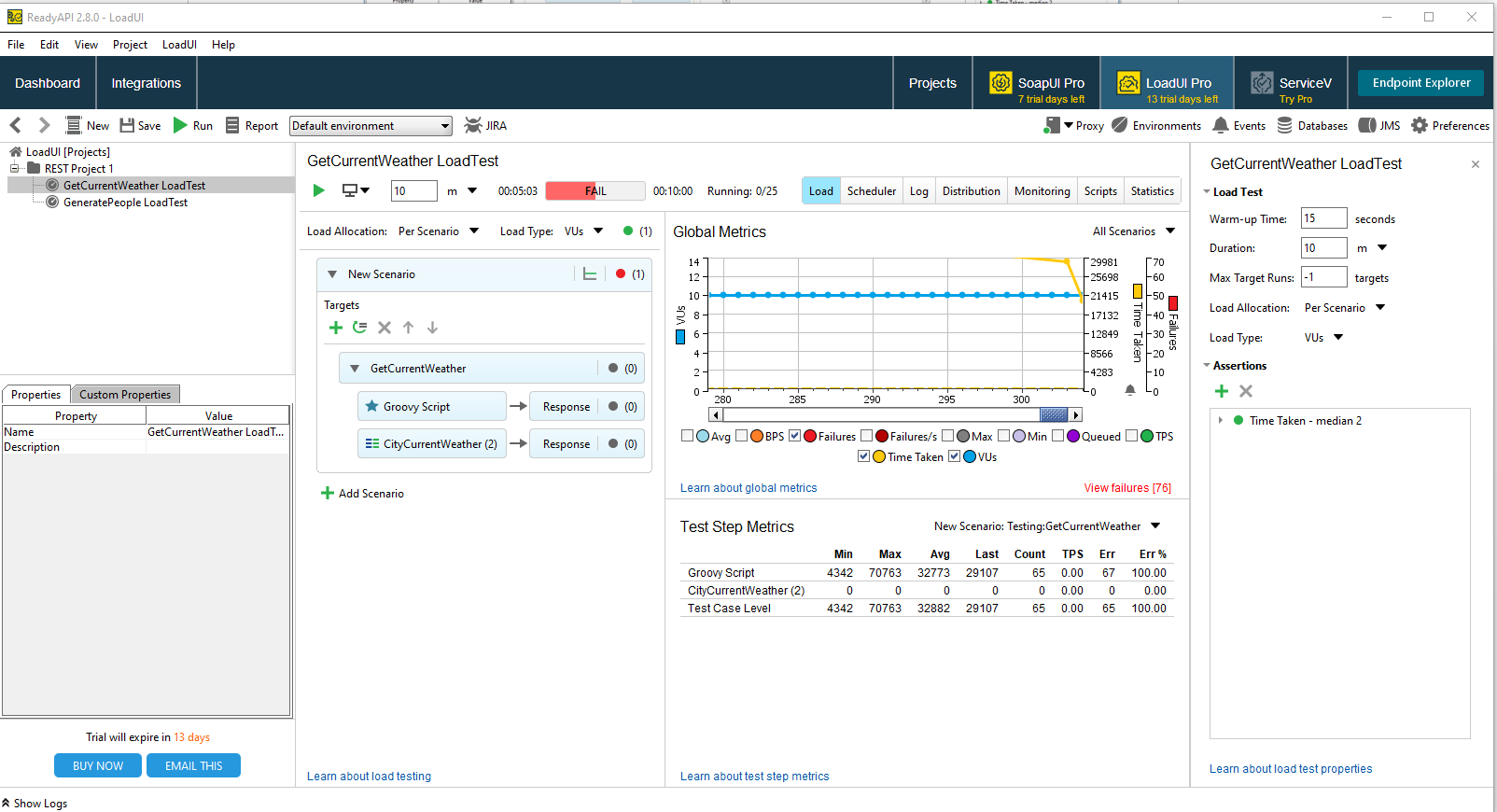
+ Wait time – One second.

+ Warmup time – 15 seconds.

+ Assertions – Time taken – Median.



Run the Load test, then result will be showed:



Generate to a report and save it for the comparison.

Customizing the test for serve purposes:

Change the Load Profile to better simulate user behavior.

+ Switch to Rate Load Type to simulate users coming to service without waiting until other users finish their work.

+ Add Assertions to check service performance.

+ Add additional scenarios to simulate more complex user behavior, or to simulate the actions of several users working with the service.

Save the test scenarios for the next load test to compare.

1. Stress Test:

To create a stress test, use the Ramp Up load profile with a large number of users.

Parameters:

+ Duration: 120 minutes

+ Load type: Rate

+ Simulated load: Ramp Up, from 0 to maximum number of arriving users per second supported by license

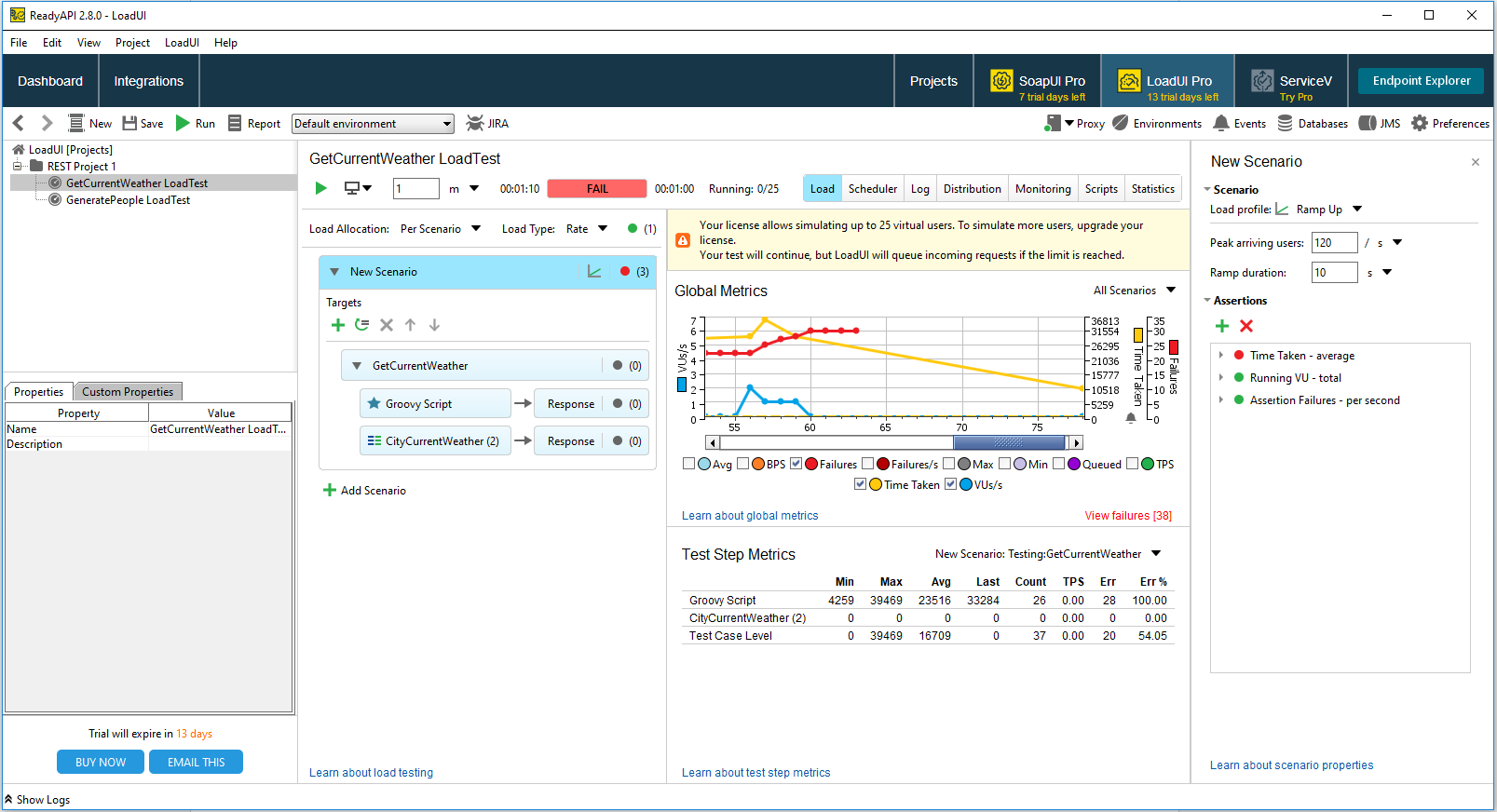
+ Think time: 1 second (10 ms for trial users)

+ Warmup time: 0 seconds

+ Assertions: Time taken – Average;

Running VU;

Assertion failures – Per Second;

Result:

Generate report or add a statistic graph for tracking.

1. Endurance Testing – Soak testing:

Use soak testing to find issues that come up during extended work with the server, like server memory leaks.

Ideally, at the end of a test run, the server performance should be the same as it was at the beginning of the test. The decrease in performance can indicate that the server code has some issues.

Load Profile: Fixed

Parameters:

+ Duration: 24 hours

+ Load type: VUs

+ Simulated load: Fixed, 50 virtual users (25 for trial users)

+ Wait time: One second (100 ms for trial users)

+ Warmup time: 15 seconds

+ Assertions: Time taken – Average;

Time taken – Max;

Assertion failures – Per Second;

To estimate the server performance, we can track various server-side metrics or use assertions.

Run the test and generate report/ add statistic graph.

1. Spike Test:

Spike testing helps to check how the tested server responds to a sudden spike in the number of users. The server should be able to stabilize and return to normal functioning after the spike passes.

Load Profile: Burst

Use Burst + Rate load type -> create a large number of virtual users coming to the server. This will simulate the sudden spike that can easily overwhelm the server.

Burst +VUs -> specify the exact top number of users that will work with the server.

Configure the test:

+ Duration: 20 minutes, burst load spikes every three minutes for one minute

+ Load type: VUs

+ Simulated load: 10 virtual users base, 100 virtual users during burst

+ Wait time: One second

+ Warmup time: 10 seconds

+ Assertions Failures – Per second;

+ Time taken – Average;

+ Time taken – Max;

+ Created monitors Default metrics for the created monitor

+ Created statistics Failures – Per second;

Time taken – Average;

Time taken – Max;

Running VUs;

Run the test.

Generate report and/ or add a statistic graphic to compare.

1. Volume test is a stress test.

Use Load Profile: Ramp up

When the responses take longer than the SLA(service-level agreement) specifies, your server runs out of the processing power, or errors start happening – you have reached the maximum capacity of your server.

1. Scalability testing – Peak testing:

Use peak testing to check how your server works during the busiest periods. Peak testing is similar to soak testing, but with a much heavier load and shorter duration.

Parameters:

+ Duration: 60 minutes

+ Load type: VUs

+ Simulated load: Ramp Sequence, from 0 to the number of VUs defined by the user, 5 minute ramp up, 50 minute peak time, 5 minute ramp down

+ Wait time: One second (500 ms for trial users)

+ Warmup time: 10 seconds

+ Assertions: Time taken – Average;

Time taken – Max;

Assertion failures – Per Second;

Run the test.

Generate report and/ or add a statistic graphic to compare.