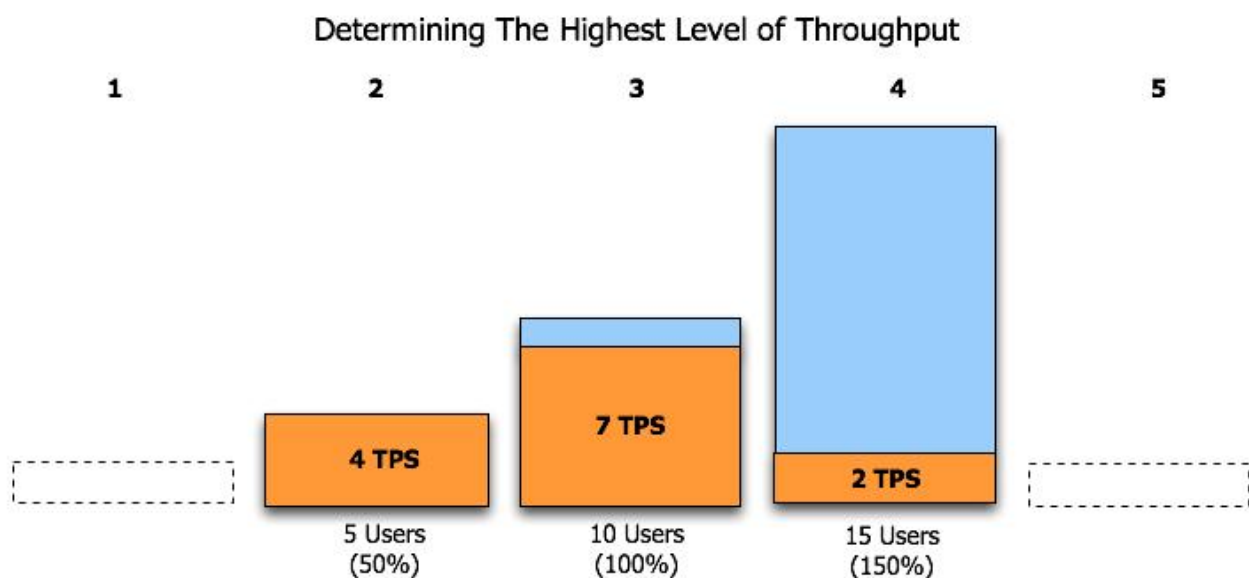


The Methodology

Here are the steps to this methodology:

1. Determine a starting number of concurrently running virtual users. I use the following table from past experiences: Web apps with links and forms, start at 30 virtual users Ajax apps with Javascript, start at 10 virtual users SOAP/REST services, start at 50 virtual users message Queues (TIBCO Business Works, Oracle ESB, IBM Process Server), start at 100 virtual users Java objects method calls, start at 200 virtual users
2. Identify the average time to operate one virtual user session. Multiply this value by 2.5. We will refer to this value as calibration test time (CTT.) For example, if the average time to run a virtual user session is 34 seconds set the CTT value at 85 seconds
3. Operate a performance test for the CTT value at 3 levels of virtual users: 50%, 100%, and 150% of the starting number. For example, run a test of Ajax apps at 5, 10, 15 virtual users

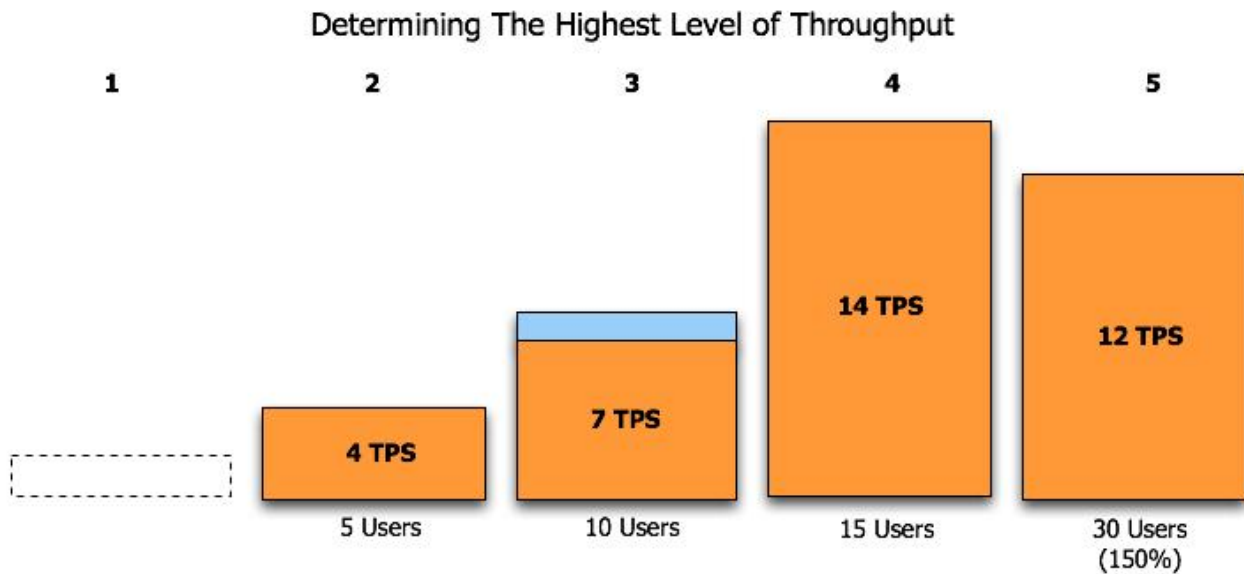


The above results show linear scalability within 10% from test 2 to test 3. The blue area in test 3 shows where the TPS needs to be for linear scalability from test 2. Test 4 shows a performance bottleneck. TPS at linear scalability in test 4 needs to be at the top of the blue area (14 TPS.) Instead, the test server or the app is seriously constrained to perform. TPS is a measurement of the rate of completed transactions. Since the test time is the same in tests 2, 3, and 4 the TPS value is so low because the time to operate each transaction has grown longer at test 4. Appvance Performance Cloud's (AQI's) Transaction Distribution chart shows the transaction times. See the Results Analysis chapter of the AIQ User Guide for details.

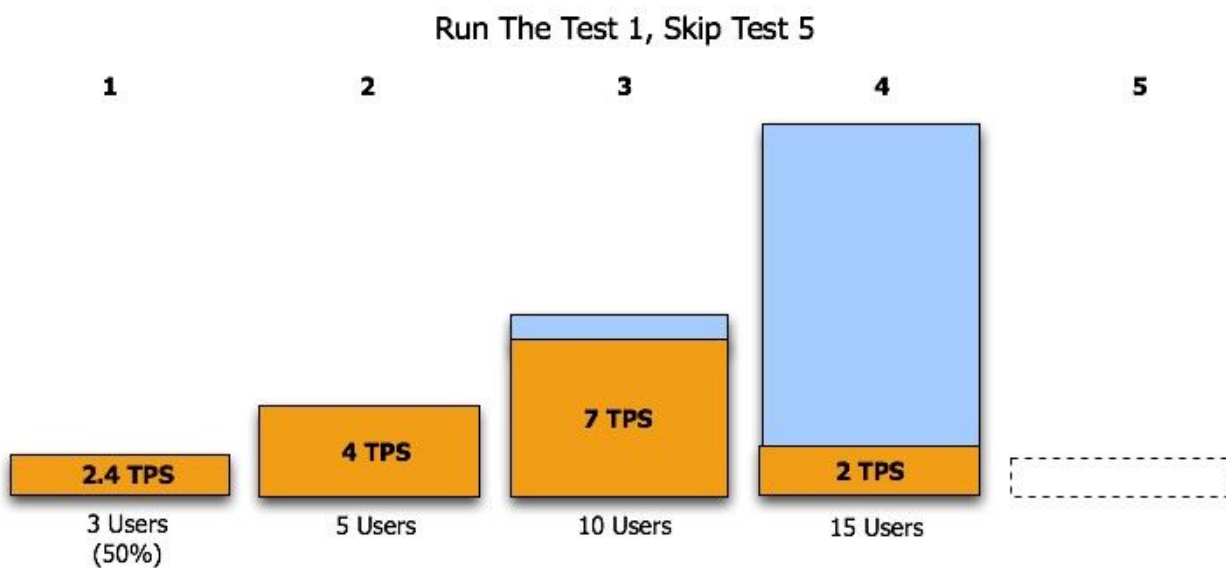
4. Investigate the cause of the performance results in test 4. Look at the test server resources as test 4 operates. Is the CPU running above 90%? Is the network interface above 80%? Are there available network sockets? Is a data source providing the test script

operating slowly? Is there free memory? Is there a logical problem in the test script? If you find any of these causes, resolve them and rerun the test in step 1.

5. If test 4 shows linear scalability from test 3, then operate test 5 at 150% of test 4 virtual user level



If test 4 performs 90% or less of the linear scalability from test 3, then operate test 1 at 50% of the test 2 level.



Continue increasing by 150% or decreasing by 50% the virtual user level until the test reaches 1 virtual user or the test fails to reach 10% of linear scalability from the previous high test measurement

6. Use the required total virtual user count divided by the virtual user count from the test with the highest test measurement on a single test server to determine the number of test servers required to run the performance test. For example, if the total virtual user count is required at 5000, and a single test server produces scalable results at 10 virtual users, then the test requires 500 test servers.

With limited performance testing time available, Appvance recommends using the Calibration Testing Methodology to rapidly identify the optimal test levels for the

application under test and avoid running levels that are not meaningful. For example, we saved 45 minutes in the above test by not running test 1 when test 3 demonstrates linear scalability. Additionally, you now have the flexibility to run the performance test at lower levels of virtual users per machine, a greater number of machines, or any combination of both.

Last updated: June 02, 2023

If you have any feedback on this topic, please contact us at documentation@appvance.ai

Revision #3

Created 2026-07-31 15:09:26 UTC by Admin

Updated 2026-07-31 15:31:24 UTC by Admin