The Comprehensive Guide to Performance Testing Considerations

Everything you need to consider
when deciding **what tests to implement** and
when building a **performance testing framework**.

OptimusQA
Your Trusted QA Partner

Share:

Table of Contents

When possible, all types of performance tests below should be executed. At a minimum single-user and endurance tests should be executed. It is recommended to execute performance tests as early as possible since execution and root cause analysis of performance problems can take much longer than for functional problems.

Single-user

- One user executes a test script a given number of times.
- Pass/Fail criteria
    - Functional success - actual system state is as expected 100% of the time
    - Response times at or below known thresholds. In cases where target response times are unspecified the following rules will apply for non-administrator operations:
        - Under 5 seconds - pass
        - 5 - 10 seconds - warning
        - Over 10 seconds - failure
    - System resources on client and server within acceptable limits. In cases where system resource targets are unspecified the following rules, if not met will result in failure:
        - Client-side memory consumption delta under 50 MB
        - Client-side CPU utilization never stays at 100% for more than 3 seconds
        - Server-side memory consumption increases by less than 10% from idle state during test, returns to idle state after the last user interaction finishes
        - Server-side CPU consumption never exceeds 10%
    - All server-side and client-side applications and services continue running without crashing or generating error messages or log entries

Load

- Prerequisite - Before a proper load test is executed, the saturation point is found by gradually increasing the number of users (or increasing the data volume) until system utilization is between 60 and 80%.
- User load - Multiple users execute a test script a given number of times.
- Data load - One or more users execute a test script which causes large amount of data to be transferred and manipulated, relative to the amount of user activity.
- Pass/Fail criteria (Data load)
    - Response and throughput criteria for data load operations must be specified by the Product Owner before the test
    - Resource utilization is allowed to remain high, so long as system performance for *other* users is fairly unaffected, that is, all User load response time criteria below is met with an allowance of 50%
- Pass/Fail criteria (User load)
    - All Single-user response time criteria as above, with the exception of functional success, which is expected to be 99% or greater
    - System resources on client and server within acceptable limits. In cases where system resource targets are unspecified the following rules, if not met will result in failure:
        - Before the test is complete all system resource consumption must stabilize. That is, resource usage may climb for a period but must eventually reach a steady state while the test continues executing.
        - Any system resource may never exceed 80% consumed for more than 3 seconds

Stress

- This is a load test with the number of users or the data volume exceeding the known saturation point.
- Pass/Fail criteria
    - All Load criteria as above must be met with the follow exceptions
        - Resource consumption may be 100% for any length of time
        - Functional success must be 90% or greater
    - Once the stress is removed, and without restarting any processes or machines
        - All system state and data must be correct
        - All subsequent user interactions should complete as quickly as before the stress was applied

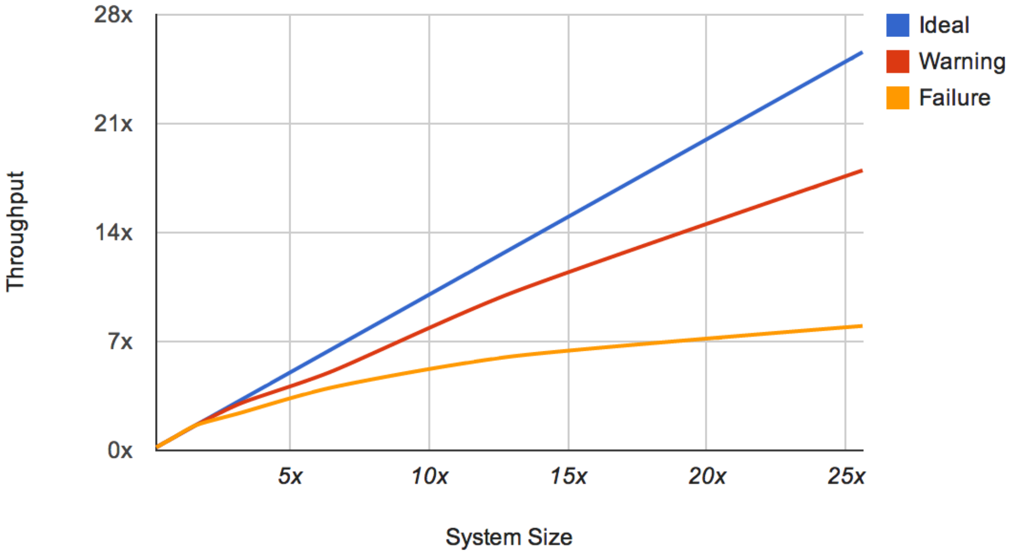
Endurance / Reliability

- This is a load test that is executed for a longer duration. Suggested durations are 12 hours, 24 hours, or 72 hours.
- Pass/Fail criteria
    - All Load criteria as above
    - Throughput per user remains constant (+/- 10%)
    - Response times remain constant (+/- 10%)

Scalability

- This kind of testing typically involves increasing the system load and the system resources by equal amounts. For example, double the number of users and double the hardware. The purpose is to see how throughput and response time change as the load and hardware footprint grows. Ideally, performance per user remains unchanged as you scale.

- Pass/Fail criteria
    - Functional success criteria according to Load as above
    - Response times and throughput within acceptable limits. In cases where performance targets are unspecified, ensure that total system throughput grows with system size according to the following chart



Performance Metrics

All of the metrics below should be recorded for every performance tests. Some are, by default, used directly or in combination for Pass/Fail criteria. Others are used for test analysis purposes to help diagnose and discover the source of problems. Any may be specified by the customer as Pass/Fail criteria, upon request.

- Response time - the amount of time between a user making a request and receiving a response
- Throughput - the number of system transactions per second or the number of user operations performed per second
- Failure rate - the ratio of failed requests to total requests made
- Time to first failure - the amount of time (or the number of operations) before the first failure occurs
- Mean time between failures - the average amount of time in between failures
- Availability - % of time the system was successfully responding to requests

Simulation Strategy

- UI driven - Actual mouse clicks and keyboard events will be sent to the UI
- API driven - Test scripts will make calls on APIs in the same way the UI does
- Protocol driven - Test scripts will exchange data with the server in the same way the client does
- No simulation - People manually interact with the system either at random or according to script
- Hybrid approach - Some combination of above. For example, 98% API-driven and 2% UI-driven

Test Configuration

- Users objects in system - Number of user objects that exist in the system DB / repository
- Active concurrent users - Number of users that will simultaneously generate load
- Type of users - Content creator or consumer, end-user or administrator
- Think time - The (potentially randomized) amount of time between user interaction steps
- Data set size - Number of objects in dataset, number of records in database or size on disk (GB)
- Data uniqueness - Do users or artifacts share data, is much of the data copied, randomization?
- Data cache state - Cold (on disk), hot (in memory)
- Test duration - Number of times each virtual user will execute the test or max duration of execution
- Ramp-up time - Amount of time it will take to transition all virtual users from idle to active
- Synchronization - Will all users execute the scripts independently, will some perform the same actions at the same time?

System-wide and Per-Process Resource Metrics

- Memory consumption - Leaks may be detected or inefficient use of caches
- CPU utilization - Un-optimized algorithms if too high, bottlenecks or contention if too low
- Page faults - Page faults can have a big impact on performance. If there is not enough physical memory available, the number of page faults will be high.
- Disk IO - Could be a sign of unused caches or indicative of a need for a caching strategy
- Network IO - Could indicate excess or suboptimal DB queries, caching weaknesses, lack of compression technology for transfers, or suboptimal network infrastructure

Machine / Configuration Management

The following details should be examined and recorded on all systems on the client and server side before testing.

- Virtualization - Vendor, technology, hardware support
- CPU - Number of Processors, number of cores, speed (clock frequency), L1, L2, L3 cache size
- Memory - Total amount of physical memory, total amount of virtual memory
- Disks - Disk type and technology (Rotational, SSD), host adapter (Fibre Channel, SATA, Infiniband), performance (IOPS, throughput, seek time)
- Network - Throughput, latency, packet loss, LAN or WAN, duplexing, topology, firewalls, routers, switches, hubs, load balancers, traffic shaping, QoS
- Operating System - Major and minor version, service packs, 64 or 32 bits
- 3rd-party configuration - DB indexing, app server memory settings, timeouts, max threads, log levels, anti-virus or malware scanners
- Up-time - When is the last time the system was restarted? When did it last crash?
- System logs - Are there any red flags, for example, a large number of errors logged in the last few days?
- Vendor recommendations and best practices - Has each independent system been configured and sized according vendor specifications or requirements?

Discuss Your Performance Testing Challenges with Optimus

Call us at **1-604-737-4600** to set up a free one hour consultation.

About the Author

Jason DeMelo is currently a Solutions Architect with Optimus Information Inc. His performance testing experience includes one year operating as the owner of performance, reliability and scalability requirements for SAP BusinessObjects BI.