Continuous Performance Testing

At today’s rapid pace, digital business must stay agile without sacrificing security, availability and performance. Continuous testing is essential for achieving this goal, requiring test tools that are automated and operate at a scale to meet functional, system and performance verification needs.

Performance testing is an essential activity in a software development life cycle. From initial planning to production analysis, application performance drives the development of better software iterations and releases. Application stakeholders, programmers and testers must make performance a primary consideration in all iterations of Agile development process.

Agile Performance Testing

As the term implies, the key to an agile approach is flexibility. QAInfoTech adopts various techniques/methodologies to remain efficient and thorough in an agile environment in all kinds of web applications, mobile applications, web services, APIs and internet based desktop applications.

The objective here is to determine the performance bottlenecks in key business processes as early as possible during the project development cycle. The sooner performance issues are found, the cheaper they are to correct. That’s why we have started to move performance testing earlier – even into development. This transition is known as “shift left” as it shifts testing earlier, in the software development lifecycle.

In QA InfoTech, we perform Agile Performance Testing in a three phases approach:

- **Unit Level Test**: Executing tests to isolate and fix bottlenecks at code level
- **API Level Test**: Executing tests to isolate and fix bottlenecks in application components level
- **Application Level Test**: Testing the critical application flows for user experience under normal and peak loads

Unit Level Test

Here we execute the load test of the Java projects at the unit level by minor modifications in JUnit test cases for JMeter compatible format.

Objective

- Tests are focused on discovering poorly performing methods

Entry Criteria

- Unit performance tuning proceeds when the scalability and performance requirements are specified at method level, along with the development of the application
Activity

- Identifying the methods
- JMeter Scripts are prepared to invoke individual method
- Methods are tested under normal load for response time while measuring CPU and memory utilization
- If method does not show acceptable level of performance, it is fine-tuned and test is re-executed

API Level Test

JMeter is used to quickly create a load test scripts which simulate different types of REST and SOAPUI users as & when APIs are available.

Objective

- Here the tests are targeted towards finding how many concurrent users API can serve before response times exceed acceptable levels
- To ensure that the resources exposed by API are created, updated and deleted appropriately under expected, as well unexpected load
- Validate server scalability and ensure end user performance
- To establish baseline for API performance on which, we validate the impact of infrastructure change over time

Entry Criteria

- The API are being developed and deployed on their respective application server

Activity

- The API can be identified based on following characteristics:
  1. High business criticality and usage.
  2. Can be deployed and invoked independently.
- A proper set of benchmark numbers (SLA) for response time, throughput and CPU Memory usage is being identified before starting the Performance tests of API.
- Tests are executed for each API as listed below:
  1. Step-Up Test: API is tested for simultaneous requests under incremental user load, until failure is identified.
  2. Test for Throughput: API is tested for high number of hits per second with respect to number of users.
- With the help of above results, the API is tuned by optimizing hardware/software configuration and test is re-executed.
Application Load Testing

QA InfoTech experts create the JMeter scripts, configure, run and analyze the load tests results once the application is fully developed. This in turn helps us to find the overall experience of the end user.

Objective

- The major objective is to capture the performance metrics and verify whether the performance objectives are met or not
- Critical application flows are tested for user experience under normal and maximum loads
- Here the tests are targeted towards finding bottlenecks in application tasks and activities
- To evaluate the system behaviour when it is pushed beyond its breaking point

Entry Criteria

- Acceptance Testing is completed and application is stable

Activity

- Load test environment preparation from which we will execute our load tests
- The critical application flows are identified
- JMeter Scripts are prepared for the flows identified
- Load test designing
- Load test monitoring
- Application is tested for normal load and maximum load. Server and client metrics are captured
- As per the results, the application is tuned by optimizing hardware/software configuration and test is re-executed

Continuously test and monitor

Continuous Integration (CI) is considered a ‘Best Practice’ and is a key element of agile development methodologies. Running CI processes without running load tests, is only one part of the equation which would result in an incomplete answer. If the application is going to be deployed to QA or production, it’s important to integrate load testing into the processes.

We at QA InfoTech use JMeter + Jenkin as a solution to complete this equation. As soon as the new build is pushed, Jenkins will trigger the load test according to the performance configuration and validating the performance of the application under load. If result doesn’t pass the thresholds, the build will fail or be declared unstable. After every load test, we can get reports in an easy, comprehensive format – no need to spend hours analyzing data.
Our Performance Testing differentiators

- Support both Open-source as well as commercial tools. Tool of choice—“JMeter”
- Cloud based test labs, for on demand scalability and flexibility. Able to simulate real world traffic from different geographies with minimal cost for and investments by clients
- Performance testing solutions catering to web applications, mobile applications, web services, APIs and internet based desktop applications
- Internal frameworks built on open source tools like JMeter for effective performance test result reporting and system monitoring
  1. Modular realistic scripts, script versioning followed strictly
  2. Simulation of realistic bandwidth, IP spoofing
  3. Right sized load tests
  4. Use of in-house developed plug-ins for creating automated & customized reports

Benefits

- Unit level tuning is done early - so no need for code optimization at later stage on testing lifecycle
- Earlier detection of the performance issues and bugs thus reduces the development effort
- Iterative testing leads to constant awareness of the application performance and therefore more confidence in meeting release dates
- Less number of performance bugs in subsequent phases
- Performance script can be re-used thus saving 60% of the efforts of performance cycle
- Higher quality software in less time

By: Stuti Jaiswal