Software Requirements Specification

For

Enhancements to JMeter

Requirements for JMeter version 2.9 and above

Prepared by JMeter Group, Summer Interns

IIT Bombay

18/06/2013

Table of Contents

Contents

1.Introduction....................3
1.1Purpose 3
1.2Document Conventions....................3
1.3Intended Audience and Reading Suggestions....................3
1.4Product Scope....................4
1.5References....................4
2.Overall Description....................4
2.1Product Perspective....................5
2.2Product Functions....................5
2.3User Classes and Characteristics....................5
2.4Operating Environment....................6
2.5Design and Implementation Constraints....................6
2.6User Documentation....................6
2.7Assumptions and Dependencies....................6
3.External Interface Requirements....................6
3.1User Interfaces....................7
3.2Hardware Interfaces....................7
3.3Software Interfaces....................7
3.4Communications Interfaces....................7
1.System Features....................8
3.5Automating CSV File Generation....................8
3.5.1Description and Priority....................8
3.5.2Stimulus/Response Sequences....................8
3.5.3Functional Requirements....................8
3.6SMTP Configuration Element....................8
3.6.1Description and Priority....................8
3.6.2Stimulus/Response Sequences....................9
3.6.3Functional Requirements....................9
3.7Bandwidth Throttling Element....................9
3.7.1Description and Priority....................9
3.7.2Stimulus/Response Sequences....................10
3.7.3Functional Requirements....................10
3.8Dynamic Bandwidth Throttling Element....................10
3.8.1Description and Priority....................10
3.8.2Stimulus/Response Sequences....................10
3.8.3Functional Requirements....................11
4.Other Nonfunctional Requirements....................11
4.1Performance Requirements....................11
4.2Safety Requirements....................11
4.3Security Requirements....................11
4.4Software Quality Attributes....................11
4.5Business Rules....................12
5.Other Requirements....................12

1. Introduction

1.1 Purpose

The purpose of this document is to specify the requirements and preview some elements of the Load Testing tool JMeter.

Apache JMeter desktop application is open source software. The application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions.

Various Experiments are to be conducted using JMeter. Variety of applications can be tested and graded using JMeter, bottlenecks can be identified by varying parameters of each experiment. JMeter can load test various types of servers like:

- Web - HTTP, HTTPS
- SOAP
- Database via JDBC
- LDAP
- JMS
- Mail - SMTP(S), POP3(S) and IMAP(S)
- Native commands or shell scripts

JMeter is highly extensible, many open source plugins are available for JMeter for each of its major component like samplers, listeners, thread group, etc.

1.2 Document Conventions

In general this document prioritizes in writing the functional enhancements of JMeter and performing various kinds of tests using JMeter. This document also analyzes in detail the plugins that are available for variety of components of JMeter. Therefore there are lots of abstractions to represent in a more convenient way the objects and their behavior on the system. Every requirement statement is assumed to have its own priority as to define in most appropriate way the system behavior. In addition there are various figures that represent the described system, where it is needed, and serve only for better understanding of the deployment. Please refer to the official documentation of the application at jmeter.apache.org if you have specific questions based on your system.

1.3 Intended Audience and Reading Suggestions

This document is intended for any individual user, developer, tester, project manager or documentation writer that needs to understand the basic system architecture and its specifications.

Here are the potential uses for each one of the reader types:

- ***Developer:*** *The developer who wants to read, change, modify or add new requirements into the existing program, must firstly consult this document and update the requirements with appropriate manner so as to not destroy the actual meaning of them and pass the information correctly to the next phases of the development process.*

- ***User:*** *The user of this program reviews the diagrams and the specifications presented in this document and determines if the software has all the suitable requirements and if the software developer has implemented all of them.*

- ***Tester:*** *The tester needs this document to validate that the initial requirements of this programs actually corresponds to the executable program correctly.*

Overview

1. ***Introduction:*** *Provide an overview of the application, describe the document structure and point the individual objectives.*

2. ***Overall Description:*** *Provide the specification of the system model, the classes model, the main constraints and the list any assumed factors that used within this document.*

3. ***System Features:*** *Provide the analysis of the requirements by feature.*

4. ***External Interface Requirements:*** *Provide the visualization of the program and the requirements that are related with hardware, software and networking.*

5. ***Other Nonfunctional Requirements:*** *Provide some other constraints that apply to factors such as performance, safety and security.*

1.4 Product Scope

Apache JMeter is a Java desktop application designed to load test client/server software (such as a web application). It may be used to test performance both on static and dynamic resources such as static files, Java Servlets, CGI scripts, Java objects, databases, FTP servers, and more. JMeter can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types.

Additionally, JMeter can help you regression test your application by letting you create test scripts with assertions to validate that your application is returning the results you expect. For maximum flexibility, JMeter lets you create these assertions using regular expressions.

1.5 References

- *http://jmeter.apache.org/*
- *"Apache JMeter: A practical beginner's guide to automated testing and performance measurement for your websites"- Emily H. Hallili*
- *http://jmeter.apache.org/usermanual/component_reference.html#introduction*

- *http://www.code.google.com/p/jmeter-plugins/*
- *http://blackanvil.blogspot.in/2006/06/shootout-load-runner-vs-grinder-vs.html*
- *http://shantonusarker.blogspot.in/2013/05/introdution-to-jmeter-google-plugin.html*
- *http://www.methodsandtools.com/tools/jmeterplugins.php/*

2. Overall Description

2.1 Product Perspective

Since this is an open source program by Apache it is under the Apache License Version 2.0. The source code is free to download. There are various reasons why should anyone use this program. First it's a framework where you can create, edit and store Testplans for Application under test (AUT). Second is an easy and reliable testing program that is very unique in its category where you can use the testplans you have created in a way that represents user sessions. Third is number of free plugins available to increase the functionality of JMeter. And fourth due to its open source nature you can modify it according to your needs.

The major components of the system as noted in the previous sections are the Samplers and Listeners.

- *The Sampler tells JMeter to send requests to a server and wait for a response. They are processed in the order they appear in the tree. Controllers can be used to modify the number of repetitions of a sampler.*

- *The Listeners provide access to the information JMeter gathers about the test cases while JMeter runs. The Graph Results listener plots the response times on a graph. The "View Results Tree" Listener shows details of sampler requests and responses, and can display basic HTML and XML representations of the response. Other listeners provide summary or aggregation information.*

2.2 Product Functions

The major features in this set of enhancements are:

- ***IP Spoofing:** When using multiple threads, using IP spoofing, JMeter uses a unique IP from available IPs on the subnet. Thus a test on server which is IP dependent will not fail for multiple users, as each of them will appear to have a different IP.*

- ***Dynamic Bandwidth Throttling:** Dynamic Bandwidth Throttling deals with the variation of bandwidth at runtime. Through dynamic bandwidth throttling we would be able to vary bandwidth dynamically based on our requirements.*

- ***Auto CSV Generation:** The main aim of the "Auto CSV Generation" is to automate the generation of the .csv file from the mentioned table of the Database, to be used with samplers.*

- ***TPCC plugin:*** *TPC-C is an on-line transaction processing benchmark. The goal of TPC-C benchmarks is to define a set of functional requirements that can be run on any transaction processing system, regardless of hardware or operating system.*

- ***SMTP sampler:*** *Smtp Defaults Config Element is a configuration element that can be used to set default values for Smtp samplers.*

- ***Filtered Results:*** *The Filtered results Listener plugin works as an add-on to "View Results in Table" Listener in JMeter. The plug-in has a wide spread application in performance testing and load testing, also in benchmarking any system.*

2.3 User Classes and Characteristics

Physical actors:

Tester: The tester is the one who designs the test plan to be run on JMeter. JMeter can be used for remote testing, where one Instance acts as Master and other instances act as slaves. In such instances there are Primary and Secondary testers to execute the test.

Analyst: To analyze the results of a test run on JMeter a system analyst is required to review the results, find out whether the system performed as expected, or find the bottlenecks and loop holes in the system.

System actors:

JMeter: This application acts as a master in remote testing mode. In normal mode it is used as an independent application.

JMeter-server: This acts as a slave in JMeter remote testing mode, the interface is non-GUI and only logs are visible on the terminal.

2.4 Operating Environment

This program will operate in the following operating environment for the application and the server mode:

- Apple Mac OS X (Universal)
- Linux/Unix (Source code)
- Microsoft Windows (Installer)

2.5 Design and Implementation Constraints

This program is created using Java programming language and uses the Java Swings and AWT libraries for the main GUI of all modules. So a minimum PC having at least 128mb of RAM and CPU over 800 MHz is required to run the program with good speed. Also the program uses at least 60 megabytes of hard disk space to store the program libraries. A file compressor is used to extract all files in a valid location. For language support apart from the English language pack there is also French, German, Norwegian, Polish, Portuguese (Brazilian), Spanish, Turkish, Japanese, Chine (simplified and traditions) language packs that can be enabled within the program.

For the connection stream Tcp-Ip is used as it's the common gateway for internet applications.

2.6 User Documentation

Here are the official links of the project where you can retrieve more information about it and download the latest version:

Online Documentation of extending JMeter is available at:

http://jmeter.apache.org/extending/jmeter_tutorial.pdf

User manual is available at:

http://jmeter.apache.org/usermanual/

2.7 Assumptions and Dependencies

It is assumed that the user is an experienced tester and knows the usage of appropriate components for various kinds of tests like Performance test, load test, regression test, etc.

3. External Interface Requirements

3.1 User Interfaces

Users of Apache JMeter are testers who perform rigorous tests systems and provide feedback for system enhancement and improvement. Any user can download JMeter from jmeter.apache.org. After downloading user has to extract the files to a directory and open ApacheJmeter.jar. For remote testing the user has to open jmeter-server to act as a slave machine.

The enhancements of JMeter will include addition of new Sampler, Configuration Element, and optional task under various elements and listeners. The GUIs of the intended additions

- The "Auto CSV Configuration" GUI will take the connection details, including the Database name, table name and username password for the connection to be established.

- The "SMTP Defaults" GUI will take as input, server, port, address to the sender and then generates the default configuration for all the SMTP sampler under it.

- The "Bandwidth Throttling" unit of the HTTP Default GUI, will take the value of bandwidth as input.

- The "Dynamic Bandwidth Throttling" unit of the HTTP Default GUI, will take the input of minimum Bandwidth, permissible error percentage and the maximum Bandwidth as bandwidth value from "bandwidth Throttling" unit, by default.

- The “TPC-C Sampler” will take the input of host, port, username, password, name of database, duration of test and number of warehouses and the “create database” button will creates the new TPCC standard Database, while the “start test” button will perform the standard TPCC test.

3.2 Hardware Interfaces

For the communication protocol the program needs these protocols to be installed:

- TCP for the client to connect to the test server in online mode.

- Storing devices for the client to run a test and store results for future references.

Also in case that the JMeter runs behind a firewall the appropriate ports must be port forwarded or port triggered for the clients to connect.

3.3 Software Interfaces

Apache JMeter version 2.9 is a single user desktop application which can generate thousands of virtual users (threads) for testing purpose. Apache JMeter is a cross platform testing tool. It requires only a JVM to run on any operating system. The software implements JAVA Swings and AWT packages for GUI.

3.4 Communications Interfaces

Setting up the server requires that there will be open ports for accepting connections from the clients (here, JMeter). The connection between the client and the server uses Connection-oriented communication, via TCP/IP—Transfer Control Protocol/Internet Protocol, implements reliable delivery of messages. Connection-oriented communication makes programming easier because the protocol includes mechanisms for detecting and handling errors and an acknowledgment mechanism between client and service. Again we assume that the reader is not interested of how that is implemented into the program at this phase so there will be no more details of this now.

1. System Features

The JMeter is a very vast and multi – utility software, designed for variety of testing and analysis of various kinds of software available/developed by the rapidly growing software world.

3.5 Automating CSV File Generation

3.5.1 Description and Priority

The idea is to help facilitate testers with automatically generated .csv file from existing database of the application under test. This element is of medium priority to tester, because it will be used only when an application consists of a form, and requires the users to fill in unique field data.

3.5.2 Stimulus/Response Sequences

The user will have to input the connection details, including username and password of database with table name, then play the test plan having the Auto CSV Config Element only. This would resemble a dry test or a run with non test element, like Proxy Server recording from browser. Thus the new .csv file would be created, preferable in the bin, from where it can be directly used in "CSV Data Set Config" Element of JMeter. Thus a new test plan can be made now. After this, there is no use of the "Auto CSV Generation" Element.

3.5.3 Functional Requirements

- The libs folder of JMeter should contain the .jar file of Driver class being used
- Jmeter should have the predefined element, configuaration element – "CSV Data Set Config".
- The named Database should exist with the mentioned table in it.
- The Database should be opened.

3.6 SMTP Configuration Element

3.6.1 Description and Priority

In cases when the test plan is having more than one sampler of SMTP type, data such as like Server or IP address of the SMTP server, the port address of the smtp server, the mail address where the mail is to be sent, mail address of the person sending the mail, the mail addresses where the copies of the mail is to be sent, the authorization settings, if required, the user-name and the password is to be specified, the message settings, the subject of the mail, headers to be added, message to be sent, any attachment to be sent along with the mail and other optional parameters; have to be specified for all the samplers which is too cumbersome as well as several entries in the samplers are common which make these entries redundant. In this situation, there is a need of a **SMTP config element** where default values can be set which don't have to be specified in the samplers to be used.

Smtp Defaults Config Element is a configuration element that can be used to set default values for Smtp samplers.

3.6.2 Stimulus/Response Sequences

The SMTP Default Config, will be added as child to thread group, and this element will contain as child, the various SMTP Samplers. Thus all the samplers will use the default configuration to build their connections with SMTP Server.

The utility of the SMTP Default Config is the same as any other default configuration element.

3.6.3 Functional Requirements

- The SMTP Sampler should be present. These samplers will be using the Config Element.
- The SMTP protocol implemented in the JMeter.

3.7 Bandwidth Throttling Element

3.7.1 Description and Priority

Bandwidth throttling is the intentional slowing of Internet service. This process is employed in the communication networks to regulate the network traffic and minimize bandwidth to control congestion. Throttling can be used to actively limit a user's upload and download rates on programs.

Jmeter is a testing tool employed to test web services under various loads, configurations, situations, and environments. There is a need to test these services from users using different available bandwidth. Such tests can provide the usability, scalability of the servers under different network connections. Such test can help is the recognizing the scenarios in which the service is unusable or to test the minimum requirement to use the service. Such functionality in Jmeter can increase its usability in testing web services where the tester can specify the available bandwidth which can less than or equal to the available bandwidth to the system and Jmeter will use this bandwidth to send or receive requests to/from the servers. Jmeter can measure the response time, throughput, amount of data exchanged, latency and errors in the transaction using variable bandwidth as required in the scenario.

3.7.2 Stimulus/Response Sequences

This element should take the value of bandwidth, and cause all those samplers under the HTTP Defaults, to take that value of bandwidth as mentioned. The HTTP Samples under the request default will send and receive data with the mentioned bandwidth.

3.7.3 Functional Requirements

- The HTTP Request Default Configuration element of which the “Bandwidth throttling” is an optional task
- Samplers should be present that will be implementing the throttling effect.
- The HTTP protocol should be implemented.

3.8 Dynamic Bandwidth Throttling Element

3.8.1 Description and Priority

Dynamic Bandwidth Throttling deals with the variation of bandwidth at runtime. Through dynamic bandwidth throttling we would be able to vary bandwidth dynamically based on our requirements. As described in the previous section of the report that we can specify bandwidth which JMeter uses to run the samplers in the test plan. The group of samplers under one Http Request Defaults used the bandwidth once specified. In case of dynamic bandwidth throttling, it is required for bandwidth to vary automatically based on the factors like error in the sampler that completed their task or the aggregate throughput of all the samplers or latency. Dynamic Bandwidth Throttling can be used to test the performance of the web services under varying bandwidth. Various types of test plans can be generated to test the performance under varying load, bandwidth, latency etc.

3.8.2 Stimulus/Response Sequences

This element will take as input, the maximum permissible bandwidth, from the “bandwidth throttling element”, the minimum bandwidth from the user and the maximum permissible error, from the user. Now when the test plan is played, and when the percentage of error will start rising, above the permissible limits mentioned, then the band width will automatically start decreasing till it reaches the minimum value of bandwidth mentioned. Following this, if percentage error is removed, that is error is decreased automatically in the test, then the bandwidth will rise automatically and rise upto the mentioned maximum value of bandwidth.

3.8.3 Functional Requirements

- The samplers should be present in the test plan, for which the throttling will be applied.
- The HTTP Request Default Configuration element of which the “Dynamic Bandwidth throttling” is an optional task
- The HTTP protocol should be implemented.
- The Bandwidth throttling value should be mentioned in the optional task, as the maximum bandwidth value will be taken from there.

4. Other Nonfunctional Requirements

Here we specify some nonfunctional constraints that the program satisfies in order to be more concrete and stable.

4.1 Performance Requirements

Performance: checking the fact that the system must perform as what every user expects .So in every action-response of the system, there are no immediate delays. In case of opening windows forms, of popping error messages and saving the settings or sessions there is delay much below 2 seconds, In case of computing there are no delays and the operation is performed in less than 2 seconds for opening, computing > 95% of the files. Also when connecting to the server the delay is based on the distance of the 2 systems and the configuration between them so there is high probability.

4.2 Safety Requirements

Consistency: checking the fact that all threads must be addressed to one server, so there would be appropriate control of the test statistics and information. Also in case of a potential loss of connection between the client and the server the clients test progress so far is maintained. When the client finishes its test then its progress is written to a file if specified. In case of a potential server breakdown only the so far finished tests are saved to the log file.

4.3 Security Requirements

This program uses object oriented mechanisms to protect its data passed using methods. Also there is no currently a security schema of this program. Thus the log files that are being created are readable using a simple text reader.

4.4 Software Quality Attributes

Availability: Checking that the system always has something to function and always pop up error messages in case of component failure. In that case the error messages appear when something goes wrong so to curtail availability problems.

Usability: Checking that the system is easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its states.

Functionality: Checking that the system provides the right tools for editing testsplans, creating tests and analyzing the test results. In that case the tools that the JMeter Listener provides is the one that provides that attribute.

4.5 Business Rules

Only a certified tester must have the access to perform test on live servers. Regular back-up of the system under test must be taken to restore the system in case of a test failure or unforeseen mishap. While testing, the system under test must be isolated from outside world for accurate results and not hamper its performance in that duration in which the test runs.

5. Other Requirements

Apart from technical requirements, a tester needs to have quality experience in testing background, an amateur cannot be expected to use software like JMeter and generate excellent testplans for any system under test.

The analyst also has to be experienced and must know every detail of the system well to analyze the results generated after a test and take necessary actions, modify and perform re-test on the system if required.

Appendix A: Glossary

SRS: Software Requirement Specification
SUT: System Under Test
AUT: Application Under Test
JVM: Java Virtual Machine
OS: Operating System
MHz: Mega Hertz, unit to measure clock frequency
Client: User machine or machine sending request
Server: System Machine or machine which responds
Master: Main controller for remote testing
Slave: Passive controller for remote testing