Embedded COMPUTING DESIGN

6th July 2017

JMeter – A preferred tool for performance testing

Maulik Patel, Volansys Technologies

What is Performance Testing?

Performance testing is a non-functional testing technique to determine the system parameters in terms of responsiveness and stability under various workload. Performance testing measures the quality attributes of the system, such as scalability, reliability and resource usage. The performance testing helps product owner to identify benchmarks of the product solution. The goal of performance testing is not only the finding bugs in the code but also to find out the bottlenecks. In performance testing, QE simulates real user scenarios using any of the tools available for performance testing.

Performance Testing Tools:

Various tools are available in the market to do performance testing of an application:

- JMeter - http://jmeter.apache.org
- Open STA - http://opensta.org
- Loadrunner - http://www.hp.com
- Web Load - http://www.radview.com

From above mentioned tools, JMeter is a preferred tool for performance testing.

JMeter:

How Apache JMeter simulates multiple users?
Apache JMeter is an open source software, a Java desktop application. It is designed to perform load test as well as check functional behavior of web applications. QE can use JMeter to analyze and measure the response time of web application or variety of services (including Rest APIs). It 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. JMeter is compatible with JDK 5 or the higher versions of JDK.

Additionally, JMeter helps with 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.
Note- JMeter is not a browser, it works at protocol level.

Protocols supported by JMeter:

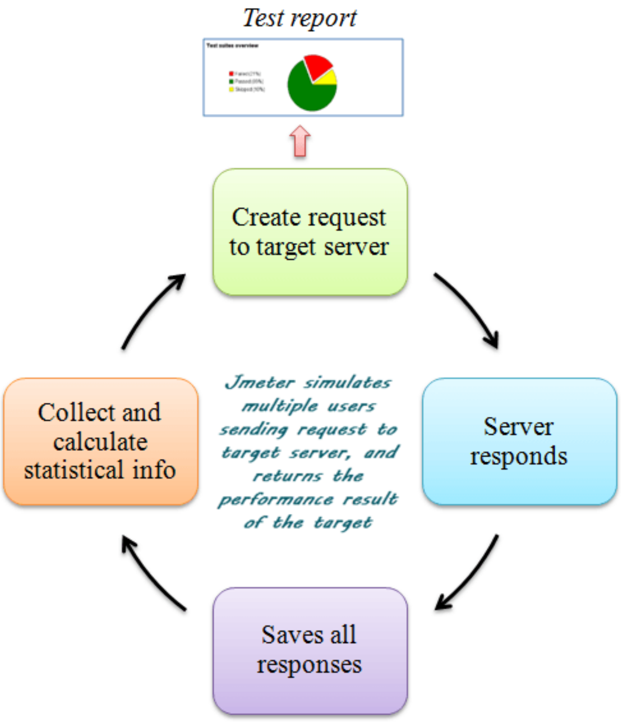
- HTTP, HTTPS sites
- Web Services
- Database via JDBC drivers
- Directory – LDAP
- JMS
- SMTP
- FTP Services

JMeter Features:

- Simple and user-friendly GUI, even a layman with little information about JMeter can use it.
- JMeter is a platform independent application. On Windows, testing on JMeter can be initiated using the JMeter.bat file, and on Unix/Linux, by clicking on JMeter Shell Scripts.
- JMeter creates and stores its test plan in XML format. That indicates that we can generate or create a test plan using a text editor.
- JMeter is a multi-threading framework which is used for concurrent sampling using multiple threads.
- It is also helpful in automated functional testing.

How it works?

JMeter simulates a group of users sending requests to a target server, and returns statistics that show the performance/functionality of the target server/application via tables, graphs, etc.

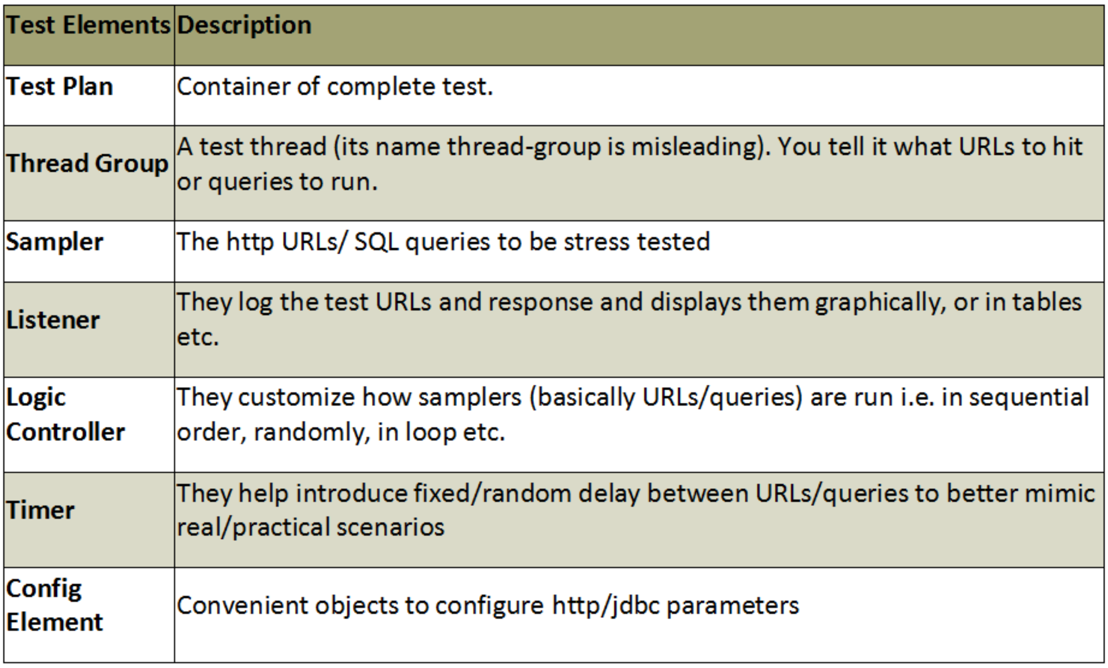

Test report
Test suites overview
Failed
Passed
Skipped
Create request to target server
Server responds
Saves all responses
Collect and calculate statistical info
Jmeter simulates multiple users sending request to target server, and returns the performance result of the target

To use JMeter, one need to define the test plan. Which basically means, one must specify

- http or other URL to hit or SQL queries to run
- how to store/display the results
- how many threads to run to create desired load/stress

In JMeter terms, these are called Threads, Samplers, Listeners etc. So, a complete test plan contains all of these. To create it, you right click on "Test Plan" on the left-hand pane of JMeter GUI and add required elements.

JMeter's Test Elements:



Test Elements	Description
Test Plan	Container of complete test.
Thread Group	A test thread (its name thread-group is misleading). You tell it what URLs to hit or queries to run.
Sampler	The http URLs/ SQL queries to be stress tested
Listener	They log the test URLs and response and displays them graphically, or in tables etc.
Logic Controller	They customize how samplers (basically URLs/queries) are run i.e. in sequential order, randomly, in loop etc.
Timer	They help introduce fixed/random delay between URLs/queries to better mimic real/practical scenarios
Config Element	Convenient objects to configure http/jdbc parameters

How Apache JMeter simulates multiple users?

Let us take a view at how a request for web page is sent to the web server from a user's browser, and how a web page is finally displayed. Understanding this work flow will give us opportunity to imagine how JMeter fits in this work flow and how it can do same thing as a browser does.

Step 1: User takes actions in browser

1.Types site address or
2.Click a link or
3.Fills form and press submit
JMeter: This is not required.

Step 2: Browser

Browser creates request in HTTP format and sends to server.
JMeter: This becomes HTTP request sampler. HTTP request sampler sends request in HTTP format.

Step 3: Web Server

Server receives HTTP request, creates HTML response and sends back HTML to browser.
JMeter: For server, it is same request.

Step 4: Browser receives HTML.

JMeter: HTTP request sampler receives HTML response and records it for analysis.

Step 5: Browser renders the HTML on the screen so user can see text and colors (GUI).

JMeter: JMeter does not show like browser.

So, what makes one user transaction in terms of JMeter?

HTTP request sampler sends request in HTTP format. --> Server receives HTTP request, creates HTML response and sends back HTML to HTTP request sampler. --> HTTP request sampler receives HTML response and records it for analysis.

When we add a HTTP Request Sampler in JMeter Test plan, it stores one webpage calling and receiving sequence. We can call this as a single user transaction. As soon as we tell this to JMeter it can do the same thing n - number of times. It also does this multiple times simultaneously.

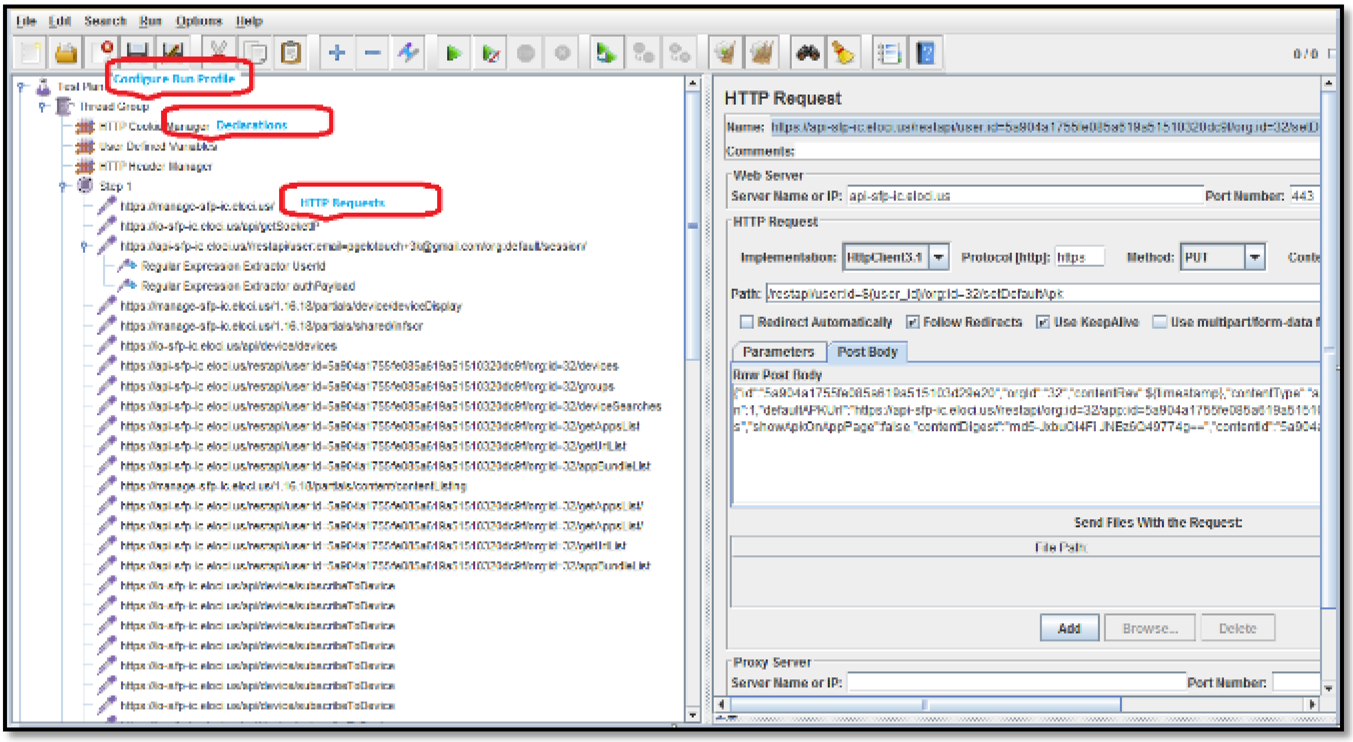
Finally, JMeter simulates multiple users by making copies of samplers and running them in separate threads.

As JMeter is not exactly a browser, the HTTP Request Sampler makes all the received data available for further analysis. View Tree Listener are used for visualizing the received response. View Tree Listener is a good way to know what request was sent, what was the response and the result of the

Why JMeter?

- JMeter is the open source tool and have numerous virtual users support
- Scripting isn't essential in JMeter, however tools like Loadrunner requires scripting in C
- Easy to do parameterization and correlation compared to other tools
- No additional configurations required for virtual user's setup because it auto generates v-users on JVM
- JMeter is cross-platform

JMeter interface:



Test Plan
Thread Group
HTTP Cookie Manager
User Defined Variables
HTTP Header Manager
Step 1
https://manage-sfp-ic.elocl.us/
https://io-sfp-ic.elocl.us/api/getSocketIP
https://api-sfp-ic.elocl.us/restapi/user.email=pgeklouch+3@gmail.com/org.default/session/
Regular Expression Extractor UserId
Regular Expression Extractor authPayload
https://manage-sfp-ic.elocl.us/1.16.18/partials/device/deviceDisplay
https://manage-sfp-ic.elocl.us/1.16.18/partials/shared/infsor
https://io-sfp-ic.elocl.us/api/device/devices
https://api-sfp-ic.elocl.us/restapi/user.id=5a904a1755fe085a618a51510320dc9f/org.id=32/devices
https://api-sfp-ic.elocl.us/restapi/user.id=5a904a1755fe085a618a51510320dc9f/org.id=32/groups
https://api-sfp-ic.elocl.us/restapi/user.id=5a904a1755fe085a618a51510320dc9f/org.id=32/deviceSearches
https://api-sfp-ic.elocl.us/restapi/user.id=5a904a1755fe085a618a51510320dc9f/org.id=32/getAppsList
https://api-sfp-ic.elocl.us/restapi/user.id=5a904a1755fe085a618a51510320dc9f/org.id=32/getUrlList
https://api-sfp-ic.elocl.us/restapi/user.id=5a904a1755fe085a618a51510320dc9f/org.id=32/appBundleList
https://manage-sfp-ic.elocl.us/1.16.18/partials/content/content_listing
https://api-sfp-ic.elocl.us/restapi/user.id=5a904a1755fe085a618a51510320dc9f/org.id=32/getAppsList/
https://api-sfp-ic.elocl.us/restapi/user.id=5a904a1755fe085a618a51510320dc9f/org.id=32/getAppsList/
https://api-sfp-ic.elocl.us/restapi/user.id=5a904a1755fe085a618a51510320dc9f/org.id=32/getUrlList
https://api-sfp-ic.elocl.us/restapi/user.id=5a904a1755fe085a618a51510320dc9f/org.id=32/appBundleList
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice

HTTP Request

Name: https://api-sfp-ic.elocl.us/restapi/user.id=5a904a1755fe085a618a51510320dc9f/org.id=32/setD

Comments:

Web Server

Server Name or IP: api-sfp-ic.elocl.us Port Number: 443

HTTP Request

Implementation: HttpClient3.1 Protocol [http]: https Method: PUT Conte

Path: /restapi/userId=${user_Id}/orgId=32/setDefaultApk

□ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive □ Use multipart/form-data f

Parameters | Post Body

Raw Post Body

```
{"id":"5a904a1755fe085a619a515103d29a20","orgId":"32","contentRev":${timestamp},"contentType":"a
n":1,"defaultAPKUrl":"https://api-sfp-ic.elocl.us/restapi/org.id=32/app.id=5a904a1755fe085a619a5151
s","showApkOnAppPage":false,"contentDigest":"md5-JxbuOI4FI.JNEz6Q49774g==","contentId":"5a904a
```

Send Files With the Request:

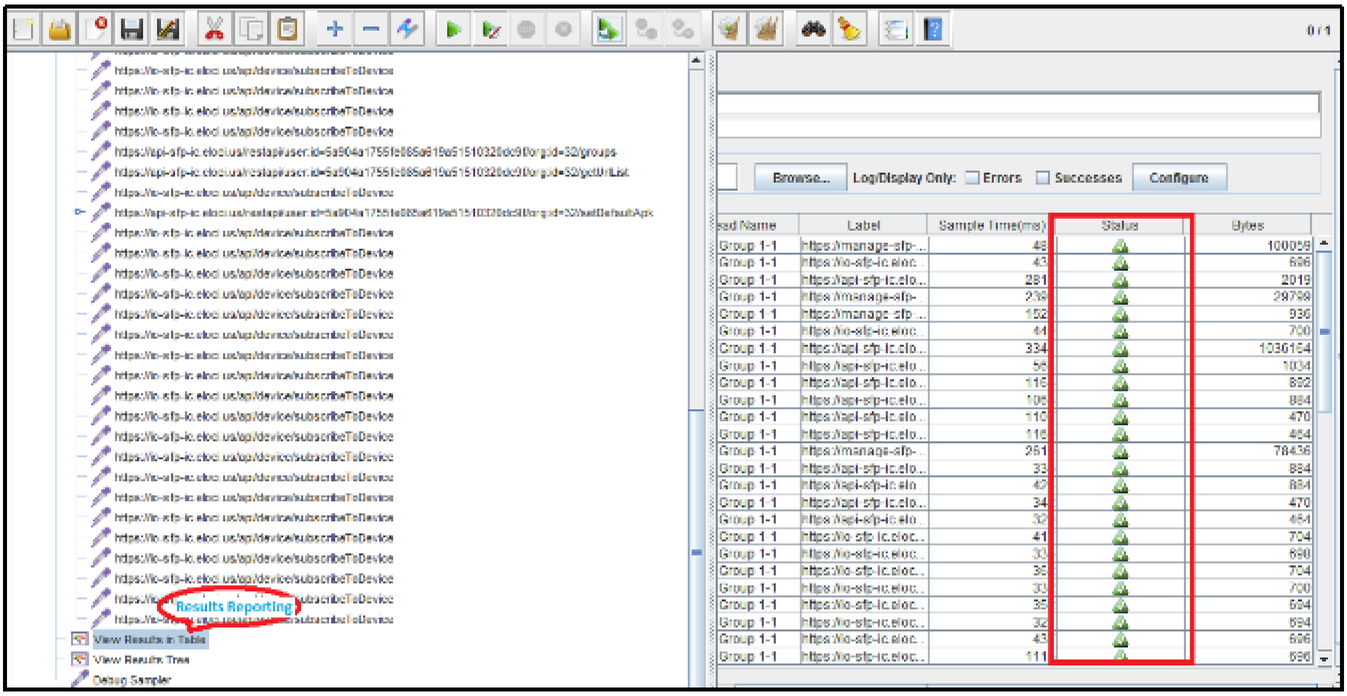
File Path:

Add | Browse... | Delete

Proxy Server

Server Name or IP: Port Number:

Report in Tabular Format:

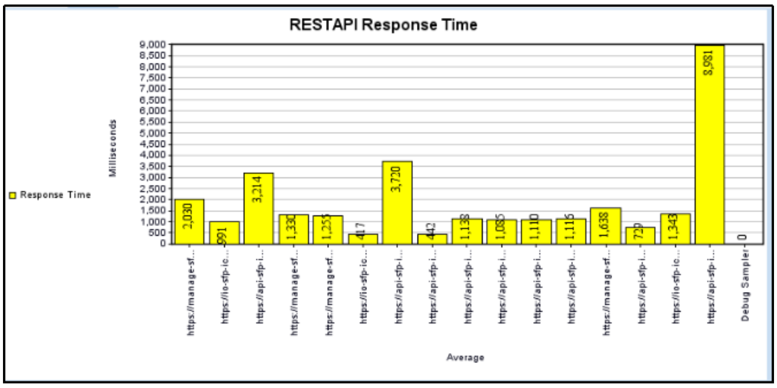


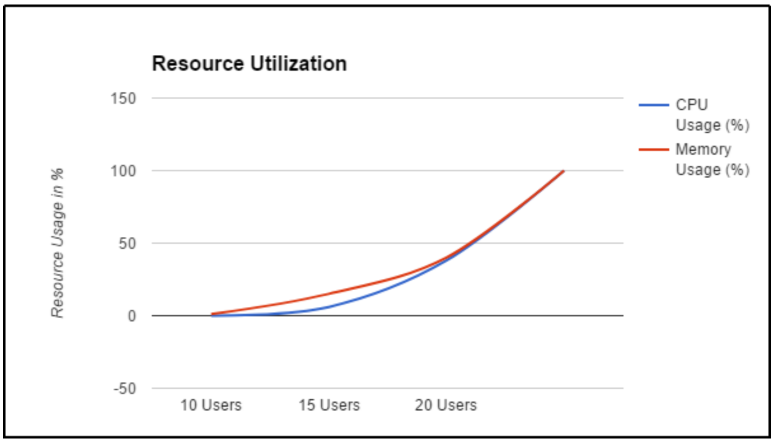
```
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://api-sfp-ic.elocl.us/restapi/user:id=5a904a17551e085a619a51510328dc9l/orgid=32/groups
https://api-sfp-ic.elocl.us/restapi/user:id=5a904a17551e085a619a51510328dc9l/orgid=32/getUrlList
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://api-sfp-ic.elocl.us/restapi/user:id=5a904a17551e085a619a51510328dc9l/orgid=32/setDefaultApk
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://io-sfp-ic.elocl.us/api/device/subscribeToDevice
https://i          ubscribeToDevice
https://io-        ubscribeToDevice
View Results in Table
View Results Tree
Debug Sampler
```

□ Errors □ Successes

ead Name	Label	Sample Time(ms)	Status	Bytes
Group 1-1	https://manage-sfp-...	48	▲	100058
Group 1-1	https://io-sfp-ic.eloc...	43	▲	696
Group 1-1	https://api-sfp-ic.elo...	281	▲	2019
Group 1-1	https://manage-sfp-...	239	▲	29799
Group 1-1	https://manage-sfp-...	152	▲	936
Group 1-1	https://io-sfp-ic.eloc...	44	▲	700
Group 1-1	https://api-sfp-ic.elo...	334	▲	1036164
Group 1-1	https://api-sfp-ic.elo...	56	▲	1034
Group 1-1	https://api-sfp-ic.elo...	116	▲	892
Group 1-1	https://api-sfp-ic.elo...	106	▲	884
Group 1-1	https://api-sfp-ic.elo...	110	▲	470
Group 1-1	https://api-sfp-ic.elo...	116	▲	464
Group 1-1	https://manage-sfp-...	261	▲	78436
Group 1-1	https://api-sfp-ic.elo...	33	▲	884
Group 1-1	https://api-sfp-ic.elo...	47	▲	884
Group 1-1	https://api-sfp-ic.elo...	34	▲	470
Group 1-1	https://api-sfp-ic.elo...	32	▲	464
Group 1-1	https://io-sfp-ic.eloc...	41	▲	704
Group 1-1	https://io-sfp-ic.eloc...	33	▲	690
Group 1-1	https://io-sfp-ic.eloc...	36	▲	704
Group 1-1	https://io-sfp-ic.eloc...	33	▲	700
Group 1-1	https://io-sfp-ic.eloc...	35	▲	694
Group 1-1	https://io-sfp-ic.eloc...	32	▲	694
Group 1-1	https://io-sfp-ic.eloc...	43	▲	696
Group 1-1	https://io-sfp-ic.eloc...	111	▲	696

Report in Graphical Format:





CONCLUSION

JMeter tool is easy to use for injecting load on web application servers. Dev/QE engineers can easily simulate a heavy load on a server using JMeter tool without doing extra setup for virtual users, extra efforts for scripting and can measure accurate response time in form of charts and table summary. JMeter results charts are useful locate bottlenecks and improve performance of the web server application. QE engineers can identify the culprit RestAPI by analyzing JMeter results charts. JMeter tool is also used to perform server benchmark tests and scalabiity tests. Recently Volansys helped with performance testing with JMeter to one of its reputed client. To know more about such solutions, visit https://volansys.com/case-studies or email us at business@volansys.com .

Maulik Patel
Sr. QA Engineer
Volansys Technologies.