Enhacement of JMeter

B. Sushmitha
KIET

Dhruv Joshi
NIT Rourkela

Manisha Choudhury
NIT Rourkela

Naman Choudhary
NIT Jamshedpur

Shekhar Saurav
NIT Jamshedpur

Surabhi Mour
SVNIT Surat

03 July 2013
Our Guides

Principle Investigator
Prof. D. B. Phatak

Project In-Charge
Mr. Nagesh Karmali

Mentors
Miss. Silpa T.
Miss Firuza Aibara
Mr. Sukhdeo Gupta
INTRODUCTION
Why do we use JMeter?

Considering a typical web-application, we could have 1000s of users trying to access it concurrently. We may not choose to employee 1000 testers for every such user for performance evaluation of the application.
More on Testing

Testing Parameters:
- User
- Data
- Time

Performance testing Comprises of:
1. Load Testing
2. Stress Testing
3. Scalability Testing

and calculating Response Time, Latency, Throughput and other such Metrics.
A sample JMeter 'TESTPLAN'
Some JMeter Plugins

- **Thread group Plugins**
  - Stepping Thread Group
  - Ultimate Thread Group

- **Timeline Graph Plugins**
  - Active Threads Over Time
  - Response Times Over Time
  - Response Latency Over Time
  - Transactions per Second
  - Server Hits per Seconds
  - Bytes Throughput Over Time
Dynamic Bandwidth Throttling
for the requests being sent, based on response error percentage
JMeter Enhancements Implemented

- **Dynamic Bandwidth Throttling**
  for the requests being sent, based on response error percentage
- **IP Spoofing**
  distinct IP addresses for each virtual user
JMeter Enhancements Implemented

- **Dynamic Bandwidth Throttling**
  for the requests being sent, based on response error percentage

- **IP Spoofing**
  distinct IP addresses for each virtual user

- **Auto CSV Generation**
  creating a .csv file directly from the database table mentioned
JMeter Enhancements Implemented

- **Dynamic Bandwidth Throttling**
  for the requests being sent, based on response error percentage
- **IP Spoofing**
  distinct IP addresses for each virtual user
- **Auto CSV Generation**
  creating a .csv file directly from the database table mentioned
- **Automating TPC-C Testing**
  Test script for Oracle and MySQL that enables a tester to carry out preliminary TPCC testing
Filtered Results Table
Filters the sampler results, based on user specified parameters
- **Filtered Results Table**
  Filters the sampler results, based on user specified parameters

- **Constant Increasing Timer**
  Stepping Up time interval between Samples requested
- **Filtered Results Table**
  Filters the sampler results, based on user specified parameters

- **Constant Increasing Timer**
  Stepping Up time interval between Samples requested

- **Enhanced Assertion results**
  Details of the Sampler passing or failing the test
- **Filtered Results Table**
  Filters the sampler results, based on user specified parameters

- **Constant Increasing Timer**
  Stepping Up time interval between Samples requested

- **Enhanced Assertion results**
  Details of the Sampler passing or failing the test

- **SMTP Defaults**
  A configuration element for setting data for SMTP Samplers under it
AutoCSV Generation
Application to be tested

Airport Management System

Username: 
Password: 
Login
The Auto CSV Generation GUI

Auto CSV Generation GUI

Object: Auto CSV Generation

Connection Pool Configuration:
- Max Number of Connections: 10
- Pool Timeout: 10000
- Idle Cleanup Interval (ms): 50000

Connection Validation by Pool:
- Keep-Alive: True
- Max Connection age (ms): 5000
- Validation Query: Select 1

Database Connection Configuration:
- Database URL (without Database name): jdbc:mysql://localhost:3306/
- Database Name: jdbc
- Table Name (whose CSV is needed): user
- JDBC Driver class: null
- Username: root
- Password: null
The .csv file generated in /bin folder
The `ams_user.csv` generated

<table>
<thead>
<tr>
<th></th>
<th>username</th>
<th>password</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>j2ee</td>
<td>j2ee</td>
</tr>
<tr>
<td>3</td>
<td>jpet</td>
<td>store</td>
</tr>
<tr>
<td>4</td>
<td>manisha</td>
<td>manisha</td>
</tr>
<tr>
<td>5</td>
<td>shekhar</td>
<td>shekhar1</td>
</tr>
<tr>
<td>6</td>
<td>surabhi</td>
<td>surabhi13</td>
</tr>
<tr>
<td>7</td>
<td>j2ee</td>
<td>j2ee</td>
</tr>
<tr>
<td>8</td>
<td>manc</td>
<td>3456</td>
</tr>
</tbody>
</table>
The CSV data config element is added as child of HTTP Sampler.
The HTTP sampler where the parameters of .csv file are set.
AutoCSV Generation

Output - CSV data verification

![Screenshot of JMeter Test Plan and View Results Tree](image)

- Test Plan:
  - Thread Group
  - HTTP Sample 1
  - HTTP Sample 2
  - CSV Data Set Config
  - View Results Tree
  - Auto CSV Generation

- View Results Tree:
  - Name: View Results Tree
  - Filename: Automatically generated CSV file

- Sampler result:
  - POST http://localhost:8090/TestAMS/validateLogin
  - POST data: username=manisha&pass=manisha
  - Request Headers:
    - Connection: keep-alive
    - Content-Type: application/x-www-form-urlencoded
    - Content-Length: 29
    - Host: localhost:8090
    - User-Agent: Apache-HttpClient/4.2.3 (java 1.5)
Bandwidth Throttling
Bandwidth throttling is the intentional slowing of Internet service.

In real world scenario, people use different web services at different bandwidths. Using Bandwidth Throttling, JMeter can be used to create test plans to simulate slower connections.
Bandwidth Throttling

- Bandwidth throttling is the intentional slowing of Internet service.
- In real world scenario, people use different web services at different bandwidths.
Bandwidth Throttling

- Bandwidth throttling is the intentional slowing of Internet service.
- In real world scenario, people use different web services at different bandwidths.
- Using Bandwidth Throttling, JMeter can be used to create test plans to simulate slower connections.
To use bandwidth throttling in JMeter, a gui component has been added to HTTP Request Defaults Config Element.

- Bandwidth Throttling
  - Use Bandwidth Throttling
  - Bandwidth (character per second (cps)): 1024000
Test Plan

- Thread groups : 2
- Thread count : 5
- Loop count : 5
- HTTP Samplers : 1
# HTTP Default Settings for Samplers

## HTTP Request Defaults

<table>
<thead>
<tr>
<th>Name</th>
<th>HTTP Request Defaults2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comments</td>
<td></td>
</tr>
<tr>
<td>Web Server</td>
<td></td>
</tr>
<tr>
<td>Server Name or IP</td>
<td><a href="http://www.acm.net/jsr.org">www.acm.net/jsr.org</a></td>
</tr>
<tr>
<td>HTTP Request</td>
<td></td>
</tr>
<tr>
<td>Implementation</td>
<td></td>
</tr>
<tr>
<td>Protocol [http]</td>
<td></td>
</tr>
<tr>
<td>Path</td>
<td></td>
</tr>
<tr>
<td>Parameters</td>
<td></td>
</tr>
</tbody>
</table>

**Optional Tasks**

- Retrieve All Embedded Resources from HTML Files
- Use Compression

**Bandwidth Throttling**

- Use Bandwidth Throttling
- Bandwidth (character per second (cps)): 1024000

## HTTP Request Defaults

<table>
<thead>
<tr>
<th>Name</th>
<th>HTTP Request Defaults</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comments</td>
<td></td>
</tr>
<tr>
<td>Web Server</td>
<td></td>
</tr>
<tr>
<td>Server Name or IP</td>
<td><a href="http://www.acm.net/jsr.org">www.acm.net/jsr.org</a></td>
</tr>
<tr>
<td>HTTP Request</td>
<td></td>
</tr>
<tr>
<td>Implementation</td>
<td></td>
</tr>
<tr>
<td>Protocol [http]</td>
<td></td>
</tr>
<tr>
<td>Path</td>
<td></td>
</tr>
<tr>
<td>Parameters</td>
<td></td>
</tr>
</tbody>
</table>

**Optional Tasks**

- Retrieve All Embedded Resources from HTML Files
- Use Compression

**Bandwidth Throttling**

- Use Bandwidth Throttling
- Bandwidth (character per second (cps)): 1024
## Result Table for Thread group 1

<table>
<thead>
<tr>
<th>Sample #</th>
<th>Start Time</th>
<th>Thread Name</th>
<th>Label</th>
<th>Sample Time (ms)</th>
<th>Status</th>
<th>Bytes</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15:03:36.961</td>
<td>Thread Group 1-4</td>
<td>HTTP Request</td>
<td>3588</td>
<td>✔️</td>
<td>14164</td>
<td>3347</td>
</tr>
<tr>
<td>2</td>
<td>15:03:36.961</td>
<td>Thread Group 1-1</td>
<td>HTTP Request</td>
<td>3600</td>
<td>✔️</td>
<td>14164</td>
<td>3359</td>
</tr>
<tr>
<td>3</td>
<td>15:03:36.976</td>
<td>Thread Group 1-5</td>
<td>HTTP Request</td>
<td>3588</td>
<td>✔️</td>
<td>14164</td>
<td>3347</td>
</tr>
<tr>
<td>4</td>
<td>15:03:36.959</td>
<td>Thread Group 1-2</td>
<td>HTTP Request</td>
<td>3614</td>
<td>✔️</td>
<td>14164</td>
<td>3371</td>
</tr>
<tr>
<td>5</td>
<td>15:03:36.972</td>
<td>Thread Group 1-3</td>
<td>HTTP Request</td>
<td>3614</td>
<td>✔️</td>
<td>14164</td>
<td>3347</td>
</tr>
<tr>
<td>6</td>
<td>15:03:40.594</td>
<td>Thread Group 1-2</td>
<td>HTTP Request</td>
<td>1035</td>
<td>✔️</td>
<td>14164</td>
<td>794</td>
</tr>
<tr>
<td>7</td>
<td>15:03:40.571</td>
<td>Thread Group 1-1</td>
<td>HTTP Request</td>
<td>1064</td>
<td>✔️</td>
<td>14164</td>
<td>823</td>
</tr>
<tr>
<td>8</td>
<td>15:03:40.636</td>
<td>Thread Group 1-3</td>
<td>HTTP Request</td>
<td>1008</td>
<td>✔️</td>
<td>14164</td>
<td>767</td>
</tr>
<tr>
<td>9</td>
<td>15:03:40.572</td>
<td>Thread Group 1-5</td>
<td>HTTP Request</td>
<td>1075</td>
<td>✔️</td>
<td>14164</td>
<td>834</td>
</tr>
<tr>
<td>10</td>
<td>15:03:40.553</td>
<td>Thread Group 1-4</td>
<td>HTTP Request</td>
<td>1747</td>
<td>✔️</td>
<td>14164</td>
<td>1506</td>
</tr>
<tr>
<td>11</td>
<td>15:03:41.632</td>
<td>Thread Group 1-2</td>
<td>HTTP Request</td>
<td>751</td>
<td>✔️</td>
<td>14164</td>
<td>510</td>
</tr>
<tr>
<td>12</td>
<td>15:03:41.680</td>
<td>Thread Group 1-3</td>
<td>HTTP Request</td>
<td>1049</td>
<td>✔️</td>
<td>14164</td>
<td>808</td>
</tr>
<tr>
<td>13</td>
<td>15:03:41.679</td>
<td>Thread Group 1-1</td>
<td>HTTP Request</td>
<td>1057</td>
<td>✔️</td>
<td>14164</td>
<td>816</td>
</tr>
<tr>
<td>14</td>
<td>15:03:41.682</td>
<td>Thread Group 1-5</td>
<td>HTTP Request</td>
<td>1060</td>
<td>✔️</td>
<td>14164</td>
<td>819</td>
</tr>
<tr>
<td>15</td>
<td>15:03:42.303</td>
<td>Thread Group 1-4</td>
<td>HTTP Request</td>
<td>747</td>
<td>✔️</td>
<td>14164</td>
<td>506</td>
</tr>
<tr>
<td>16</td>
<td>15:03:42.387</td>
<td>Thread Group 1-2</td>
<td>HTTP Request</td>
<td>745</td>
<td>✔️</td>
<td>14164</td>
<td>504</td>
</tr>
<tr>
<td>17</td>
<td>15:03:42.733</td>
<td>Thread Group 1-3</td>
<td>HTTP Request</td>
<td>881</td>
<td>✔️</td>
<td>14164</td>
<td>525</td>
</tr>
<tr>
<td>18</td>
<td>15:03:42.803</td>
<td>Thread Group 1-1</td>
<td>HTTP Request</td>
<td>862</td>
<td>✔️</td>
<td>14164</td>
<td>541</td>
</tr>
<tr>
<td>19</td>
<td>15:03:42.803</td>
<td>Thread Group 1-5</td>
<td>HTTP Request</td>
<td>904</td>
<td>✔️</td>
<td>14164</td>
<td>662</td>
</tr>
<tr>
<td>20</td>
<td>15:03:43.135</td>
<td>Thread Group 1-2</td>
<td>HTTP Request</td>
<td>845</td>
<td>✔️</td>
<td>14164</td>
<td>604</td>
</tr>
<tr>
<td>21</td>
<td>15:03:43.054</td>
<td>Thread Group 1-4</td>
<td>HTTP Request</td>
<td>946</td>
<td>✔️</td>
<td>14164</td>
<td>705</td>
</tr>
<tr>
<td>22</td>
<td>15:03:43.618</td>
<td>Thread Group 1-3</td>
<td>HTTP Request</td>
<td>875</td>
<td>✔️</td>
<td>14164</td>
<td>633</td>
</tr>
<tr>
<td>23</td>
<td>15:03:43.724</td>
<td>Thread Group 1-5</td>
<td>HTTP Request</td>
<td>780</td>
<td>✔️</td>
<td>14164</td>
<td>539</td>
</tr>
<tr>
<td>24</td>
<td>15:03:43.722</td>
<td>Thread Group 1-1</td>
<td>HTTP Request</td>
<td>810</td>
<td>✔️</td>
<td>14164</td>
<td>569</td>
</tr>
<tr>
<td>25</td>
<td>15:03:44.119</td>
<td>Thread Group 1-4</td>
<td>HTTP Request</td>
<td>1059</td>
<td>✔️</td>
<td>14164</td>
<td>818</td>
</tr>
</tbody>
</table>
## Result Table for Thread group 2

### View Results in Table

<table>
<thead>
<tr>
<th>Sample #</th>
<th>Start Time</th>
<th>Thread Name</th>
<th>Label</th>
<th>Sample Time(msec)</th>
<th>Status</th>
<th>Bytes</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15:03:36.983</td>
<td>Thread Group 2-1</td>
<td>HTTP2 Request</td>
<td>32450</td>
<td></td>
<td>14164</td>
<td>16449</td>
</tr>
<tr>
<td>2</td>
<td>15:03:36.996</td>
<td>Thread Group 2-2</td>
<td>HTTP2 Request</td>
<td>32438</td>
<td></td>
<td>14164</td>
<td>16437</td>
</tr>
<tr>
<td>3</td>
<td>15:03:37.007</td>
<td>Thread Group 2-4</td>
<td>HTTP2 Request</td>
<td>32426</td>
<td></td>
<td>14164</td>
<td>16428</td>
</tr>
<tr>
<td>4</td>
<td>15:03:37.008</td>
<td>Thread Group 2-5</td>
<td>HTTP2 Request</td>
<td>32426</td>
<td></td>
<td>14164</td>
<td>16427</td>
</tr>
<tr>
<td>5</td>
<td>15:03:37.002</td>
<td>Thread Group 2-3</td>
<td>HTTP2 Request</td>
<td>33412</td>
<td></td>
<td>14164</td>
<td>17411</td>
</tr>
<tr>
<td>6</td>
<td>15:04:09.437</td>
<td>Thread Group 2-1</td>
<td>HTTP2 Request</td>
<td>32400</td>
<td></td>
<td>14164</td>
<td>16399</td>
</tr>
<tr>
<td>7</td>
<td>15:04:09.439</td>
<td>Thread Group 2-2</td>
<td>HTTP2 Request</td>
<td>32398</td>
<td></td>
<td>14164</td>
<td>16397</td>
</tr>
<tr>
<td>8</td>
<td>15:04:09.441</td>
<td>Thread Group 2-4</td>
<td>HTTP2 Request</td>
<td>32398</td>
<td></td>
<td>14164</td>
<td>16397</td>
</tr>
<tr>
<td>9</td>
<td>15:04:09.442</td>
<td>Thread Group 2-5</td>
<td>HTTP2 Request</td>
<td>32398</td>
<td></td>
<td>14164</td>
<td>16397</td>
</tr>
<tr>
<td>10</td>
<td>15:04:10.418</td>
<td>Thread Group 2-3</td>
<td>HTTP2 Request</td>
<td>32397</td>
<td></td>
<td>14164</td>
<td>16396</td>
</tr>
<tr>
<td>11</td>
<td>15:04:41.849</td>
<td>Thread Group 2-5</td>
<td>HTTP2 Request</td>
<td>32522</td>
<td></td>
<td>14164</td>
<td>16521</td>
</tr>
<tr>
<td>12</td>
<td>15:04:41.841</td>
<td>Thread Group 2-1</td>
<td>HTTP2 Request</td>
<td>32530</td>
<td></td>
<td>14164</td>
<td>16529</td>
</tr>
<tr>
<td>13</td>
<td>15:04:41.842</td>
<td>Thread Group 2-2</td>
<td>HTTP2 Request</td>
<td>32530</td>
<td></td>
<td>14164</td>
<td>16529</td>
</tr>
<tr>
<td>14</td>
<td>15:04:41.847</td>
<td>Thread Group 2-4</td>
<td>HTTP2 Request</td>
<td>32525</td>
<td></td>
<td>14164</td>
<td>16524</td>
</tr>
<tr>
<td>15</td>
<td>15:04:42.818</td>
<td>Thread Group 2-3</td>
<td>HTTP2 Request</td>
<td>33397</td>
<td></td>
<td>14164</td>
<td>17396</td>
</tr>
<tr>
<td>16</td>
<td>15:05:14.374</td>
<td>Thread Group 2-5</td>
<td>HTTP2 Request</td>
<td>32407</td>
<td></td>
<td>14164</td>
<td>16406</td>
</tr>
<tr>
<td>17</td>
<td>15:05:14.375</td>
<td>Thread Group 2-1</td>
<td>HTTP2 Request</td>
<td>32406</td>
<td></td>
<td>14164</td>
<td>16405</td>
</tr>
<tr>
<td>18</td>
<td>15:05:14.380</td>
<td>Thread Group 2-4</td>
<td>HTTP2 Request</td>
<td>32401</td>
<td></td>
<td>14164</td>
<td>16400</td>
</tr>
<tr>
<td>19</td>
<td>15:05:14.379</td>
<td>Thread Group 2-2</td>
<td>HTTP2 Request</td>
<td>32402</td>
<td></td>
<td>14164</td>
<td>16401</td>
</tr>
<tr>
<td>20</td>
<td>15:05:16.219</td>
<td>Thread Group 2-3</td>
<td>HTTP2 Request</td>
<td>32426</td>
<td></td>
<td>14164</td>
<td>16425</td>
</tr>
<tr>
<td>21</td>
<td>15:05:46.784</td>
<td>Thread Group 2-1</td>
<td>HTTP2 Request</td>
<td>32412</td>
<td></td>
<td>14164</td>
<td>16412</td>
</tr>
<tr>
<td>22</td>
<td>15:05:46.785</td>
<td>Thread Group 2-2</td>
<td>HTTP2 Request</td>
<td>32416</td>
<td></td>
<td>14164</td>
<td>16415</td>
</tr>
<tr>
<td>23</td>
<td>15:05:46.783</td>
<td>Thread Group 2-5</td>
<td>HTTP2 Request</td>
<td>32425</td>
<td></td>
<td>14164</td>
<td>16425</td>
</tr>
<tr>
<td>24</td>
<td>15:05:46.784</td>
<td>Thread Group 2-4</td>
<td>HTTP2 Request</td>
<td>32426</td>
<td></td>
<td>14164</td>
<td>16427</td>
</tr>
<tr>
<td>25</td>
<td>15:05:46.850</td>
<td>Thread Group 2-3</td>
<td>HTTP2 Request</td>
<td>32401</td>
<td></td>
<td>14164</td>
<td>16409</td>
</tr>
</tbody>
</table>
### Results Comparison

#### Bandwidth: 1MBps

<table>
<thead>
<tr>
<th>Status</th>
<th>Bytes</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>14164</td>
<td>3347</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>3359</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>3347</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>3371</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>3340</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>794</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>823</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>767</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>834</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>1506</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>510</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>808</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>816</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>819</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>506</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>504</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>525</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>541</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>662</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>604</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>705</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>633</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>539</td>
</tr>
</tbody>
</table>

#### Bandwidth: 1KBps

<table>
<thead>
<tr>
<th>Status</th>
<th>Bytes</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>14164</td>
<td>16449</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16437</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16428</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16427</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>17411</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16399</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16397</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16397</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16396</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16521</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16529</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16529</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>17396</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16406</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16405</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16400</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16400</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16401</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16425</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16412</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16415</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16425</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16427</td>
</tr>
<tr>
<td></td>
<td>14164</td>
<td>16400</td>
</tr>
</tbody>
</table>

---

JMeter Team

Enhancement of JMeter

03 July 2013
## Aggregate Reports

### Bandwidth: 1MBps

<table>
<thead>
<tr>
<th>Label</th>
<th># Samples</th>
<th>Average</th>
<th>Median</th>
<th>90% Line</th>
<th>Min</th>
<th>Max</th>
<th>Error %</th>
<th>Throughput</th>
<th>KB/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP Request</td>
<td>25</td>
<td>1494</td>
<td>1035</td>
<td>3588</td>
<td>745</td>
<td>3662</td>
<td>0.00%</td>
<td>3.0/sec</td>
<td>42.1</td>
</tr>
<tr>
<td>TOTAL</td>
<td>25</td>
<td>1494</td>
<td>1035</td>
<td>3588</td>
<td>745</td>
<td>3662</td>
<td>0.00%</td>
<td>3.0/sec</td>
<td>42.1</td>
</tr>
</tbody>
</table>

### Bandwidth: 1KBps

<table>
<thead>
<tr>
<th>Label</th>
<th># Samples</th>
<th>Average</th>
<th>Median</th>
<th>90% Line</th>
<th>Min</th>
<th>Max</th>
<th>Error %</th>
<th>Throughput</th>
<th>KB/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP Request</td>
<td>25</td>
<td>32511</td>
<td>32425</td>
<td>32530</td>
<td>32397</td>
<td>33412</td>
<td>0.00%</td>
<td>9.1/min</td>
<td>2.1</td>
</tr>
<tr>
<td>TOTAL</td>
<td>25</td>
<td>32511</td>
<td>32425</td>
<td>32530</td>
<td>32397</td>
<td>33412</td>
<td>0.00%</td>
<td>9.1/min</td>
<td>2.1</td>
</tr>
</tbody>
</table>
Dynamic Bandwidth Throttling
DBT deals with the variation of bandwidth at runtime.
Dynamic Bandwidth Throttling

1. DBT deals with the variation of bandwidth at runtime.
2. DBT can be used to test performance of web services under varying bandwidth (load).
Dynamic Bandwidth Throttling

1. DBT deals with the variation of bandwidth at runtime.
2. DBT can be used to test performance of web services under varying bandwidth (load).
3. DBT can be used to measure and manage errors during the test at runtime.
Dynamic Bandwidth Throttling

1. DBT deals with the variation of bandwidth at runtime.
2. DBT can be used to test performance of web services under varying bandwidth (load).
3. DBT can be used to measure and manage errors during the test at runtime.
4. Based on error rate.
Dynamic Bandwidth Throttling

1. DBT deals with the variation of bandwidth at runtime.
2. DBT can be used to test performance of web services under varying bandwidth (load).
3. DBT can be used to measure and manage errors during the test at runtime.
4. Based on error rate.
5. A distributed testing can be simulated using DBT and IP spoofing.
To use Dynamic Bandwidth Throttling in JMeter, an extended GUI component has been added to Bandwidth Throttling in HTTP Request Defaults.
Test Plan

- Thread Group: 1
- Thread Count: 1000
- Ramp Up period: 0 sec
- Timeout period: 22 sec
- No. of Samplers: 11
- Permissible error: 7%
- Applicable bandwidth: 1MBps
- Minimum applicable bandwidth: 1KBps
# Aggregate Report

## Table of Results

<table>
<thead>
<tr>
<th>Label</th>
<th># Samples</th>
<th>Average</th>
<th>Median</th>
<th>90% Line</th>
<th>Min</th>
<th>Max</th>
<th>Error %</th>
<th>Throughput</th>
<th>KB/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site</td>
<td>1000</td>
<td>3431</td>
<td>3603</td>
<td>6341</td>
<td>8</td>
<td>8315</td>
<td>0.00%</td>
<td>82.9/sec</td>
<td>56.9</td>
</tr>
<tr>
<td>Catalog</td>
<td>1000</td>
<td>11100</td>
<td>10505</td>
<td>19757</td>
<td>44</td>
<td>35987</td>
<td>0.00%</td>
<td>22.5/sec</td>
<td>134.0</td>
</tr>
<tr>
<td>signon</td>
<td>1000</td>
<td>17621</td>
<td>21450</td>
<td>22039</td>
<td>549</td>
<td>33596</td>
<td>20.10%</td>
<td>12.9/sec</td>
<td>44.7</td>
</tr>
<tr>
<td>Login submit</td>
<td>1000</td>
<td>19773</td>
<td>20851</td>
<td>22037</td>
<td>1410</td>
<td>33216</td>
<td>9.00%</td>
<td>9.1/sec</td>
<td>33.0</td>
</tr>
<tr>
<td>Catalog 2</td>
<td>1000</td>
<td>21013</td>
<td>20961</td>
<td>21993</td>
<td>4290</td>
<td>36751</td>
<td>4.60%</td>
<td>6.9/sec</td>
<td>32.5</td>
</tr>
<tr>
<td>reptiles</td>
<td>1000</td>
<td>21312</td>
<td>20746</td>
<td>21841</td>
<td>10094</td>
<td>37571</td>
<td>2.90%</td>
<td>5.7/sec</td>
<td>26.8</td>
</tr>
<tr>
<td>inside reptiles</td>
<td>1000</td>
<td>21670</td>
<td>21341</td>
<td>22020</td>
<td>18911</td>
<td>35739</td>
<td>5.80%</td>
<td>5.1/sec</td>
<td>23.5</td>
</tr>
<tr>
<td>select reptile</td>
<td>1000</td>
<td>21637</td>
<td>22030</td>
<td>22404</td>
<td>16554</td>
<td>27198</td>
<td>64.70%</td>
<td>5.2/sec</td>
<td>14.5</td>
</tr>
<tr>
<td>new order form</td>
<td>1000</td>
<td>21707</td>
<td>22028</td>
<td>22037</td>
<td>16557</td>
<td>23676</td>
<td>82.10%</td>
<td>5.3/sec</td>
<td>13.7</td>
</tr>
<tr>
<td>submit order</td>
<td>1000</td>
<td>21772</td>
<td>21928</td>
<td>22037</td>
<td>16605</td>
<td>23561</td>
<td>79.50%</td>
<td>5.4/sec</td>
<td>14.9</td>
</tr>
<tr>
<td>Back to catalog</td>
<td>1000</td>
<td>18737</td>
<td>19294</td>
<td>22030</td>
<td>11655</td>
<td>23560</td>
<td>28.40%</td>
<td>5.6/sec</td>
<td>22.9</td>
</tr>
<tr>
<td>TOTAL</td>
<td>11000</td>
<td>18161</td>
<td>21116</td>
<td>22037</td>
<td>8</td>
<td>37571</td>
<td>26.92%</td>
<td>39.2/sec</td>
<td>142.5</td>
</tr>
</tbody>
</table>
# Aggregate Report

<table>
<thead>
<tr>
<th>Error %</th>
<th>Throughput</th>
<th>KB/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00%</td>
<td>82.9/sec</td>
<td>56.9</td>
</tr>
<tr>
<td>0.00%</td>
<td>22.5/sec</td>
<td>134.0</td>
</tr>
<tr>
<td>20.10%</td>
<td>12.9/sec</td>
<td>44.7</td>
</tr>
<tr>
<td>9.00%</td>
<td>9.1/sec</td>
<td>33.0</td>
</tr>
<tr>
<td>4.60%</td>
<td>6.9/sec</td>
<td>32.5</td>
</tr>
<tr>
<td>2.90%</td>
<td>5.7/sec</td>
<td>26.8</td>
</tr>
<tr>
<td>5.80%</td>
<td>5.1/sec</td>
<td>23.5</td>
</tr>
<tr>
<td>64.70%</td>
<td>5.2/sec</td>
<td>14.5</td>
</tr>
<tr>
<td>82.10%</td>
<td>5.3/sec</td>
<td>13.7</td>
</tr>
<tr>
<td>78.50%</td>
<td>5.4/sec</td>
<td>14.9</td>
</tr>
<tr>
<td>28.40%</td>
<td>5.6/sec</td>
<td>22.9</td>
</tr>
<tr>
<td>26.92%</td>
<td>39.2/sec</td>
<td>142.6</td>
</tr>
</tbody>
</table>
JMeter Log Report

Dynamic Bandwidth Throttling

![Dynamic Bandwidth Throttling Graph](image)

Enhacement of JMeter

03 July 2013
JMeter Log Report

![Dynamic Bandwidth Throttling Graph](image)

- Y-axis: Bandwidth wrt time
- X-axis: Error wrt time

- The graph shows the relationship between bandwidth and error over time.

Enhancement of JMeter
IP Spoofing
What is IP address Spoofing?

It is the creation of IP Packets with forged source IP address, with the purpose of concealing identity of the sender or for impersonating another computer system.
IP Spoofing in JMeter

1. JMeter is capable of generating thousands of threads that act as virtual users.
2. On the server side, these requests appear from the same IP address on which JMeter resides.
3. On servers which have IP dependent response, the testplan with a thousand virtual fails.
4. To eliminate this drawback, we use IP Spoofing in JMeter.
Without IP spoofing

Load balancing inactive

Figure: Server treating multiple requests without IP spoofing
With IP spoofing

Load balancing active
JMeter Implementation

IP and Subnet Details
We need to provide JMeter with IP address of the machine and the subnet it belongs to, and also specify the number of IP addresses required.

IP allocation
JMeter internally allocates virtual IPs to the same machine, and each virtual user can send request from a distinct IP from newly allocated virtual IPs.
Interface of IP Spoofing Config Element in JMeter

Figure: GUI of IP Spoofing
Virtual IPs allocated to machine

Figure: Server records same IP for each virtual user
Figure: Server records same IP for each virtual user
Server Response With IP Spoofing

Figure: Server records distinct IP for each virtual user
Automating TPC-C Benchmarking
Automating TPC-C Benchmarking

- TPC- Transaction Processing Council It defines transaction processing and database benchmarks and delivers trusted results.
TPC- Transaction Processing Council It defines transaction processing and database benchmarks and delivers trusted results.

Some benchmarks undertaken under TPC:
- TPC-APP
- TPC-H
- TPC-C
WHY TPC-C?

- A number of these benchmarks have been deprecated. TPC-C is currently in use and a rather complex process.
- A number of business houses use this benchmark to showcase their performance for OLTP transactions. It gives the measure of Server speed for online transaction processing.
The actual benchmarking process is a time taking and a costly affair.

A preliminary test would be a highly useful tool to test a server for performance and hence improve it where it lacks.
The Benchmarking Model
The model emulated for TPC-C
Tables in the TPC-C schema

- Item
- Warehouse
- History
- District
- Customer
- New Order
- Orders
- Order Line
- Stock
Transactions

- New-order: enter a new order from a customer
- Payment: update customer balance to reflect a payment
- Delivery: deliver orders
- Order-status: retrieve status of customers most recent order
- Stock-level: monitor warehouse inventory
TPC-C Workflow

1. Select txn from menu:
   1. New-Order 45%
   2. Payment 43%
   3. Order-Status 4%
   4. Delivery 4%
   5. Stock-Level 4%

2. Measure menu Response Time
   - Menu = 0.3

3. Measure txn Response Time
   - Keying time = 9.6
   - Txn RT = 2.1
   - Think time = 11.4

Cycle Time Decomposition
(typical values, in seconds, for weighted average txn)

Average cycle time = 23.4

Go back to 1
Why JMeter to automate??

- JMeter is already capable of spawning a large number of virtual users to simulate the interaction of the real users with the system under test.
Why JMeter to automate??

- JMeter is already capable of spawning a large number of virtual users to simulate the interaction of the real users with the system under test.
- The firing of a request and measurement of response time as well as throughput is embedded in JMeter.
TPC-C Testing in JMeter

TPC Sampler

Database URL: jdbc:mysql://localhost:3306/
Driver Class: com.mysql.jdbc.Driver
Username: root
Password: ********
Database Name: TPCC
Number Of Warehouses: 2
MySQL
Create Database
Start Test
Controllers

Include Controller is a component of JMeter used to run saved scripts in JMeter.

![JMeter Screenshot with Include Controller]

*JMeter Team*
Test with 1 warehouse
Procedures

Automating TPC-C Benchmarking Procedures

JMeter Team ()
Enhacement of JMeter 03 July 2013 58 / 78
Test with 33 Warehouses
JDBC Configuration

JDBC Connection Configuration

Name: DBC Connection Configuration
Comments:
Variable Name Bound to Pool
Variable Name: mypool
Connection Pool Configuration
Max Number of Connections: 0
Pool Timeout: 10000
Idle Cleanup Interval (ms): 60000
Auto Commit: True
Transaction Isolation: DEFAULT

Connection Validation by Pool
Keep-Alive: True
Max Connection age (ms): 5000
Validation Query: Select 1

Database Connection Configuration
Database URL: ${_geturl()}
DBC Driver class: ${_getdriver()}
Username: ${_getusername()}
Password: ***************
Controllers
Timers

Gaussian Random Timer

Name: Think Timer
Comments:

Thread Delay Properties
Deviation (in milliseconds): 5000
Constant Delay Offset (in milliseconds): 5000
Transaction Call

JDBC Request

Name: payment
Comments:
Variable Name Bound to Pool
Variable Name: mypool
SQL Query
Query Type: Callable Statement

Function Helper

Automating TPC-C Benchmarking

Enhancement of JMeter

03 July 2013
Procedure Request

View Results Tree

Name: View Results Tree
Comments:

Write results to file / Read from file
Filename

Sampler result | Request | Response data

Response
<table>
<thead>
<tr>
<th>Label</th>
<th># Samples</th>
<th>Average</th>
<th>Median</th>
<th>90% Line</th>
<th>Min</th>
<th>Max</th>
<th>Error %</th>
<th>Throughput</th>
<th>KB/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>payment</td>
<td>74</td>
<td>30</td>
<td>27</td>
<td>59</td>
<td>8</td>
<td>149</td>
<td>0.00%</td>
<td>14.8/min</td>
<td>.1</td>
</tr>
<tr>
<td>NEW_ORDER</td>
<td>67</td>
<td>85</td>
<td>76</td>
<td>120</td>
<td>8</td>
<td>632</td>
<td>0.00%</td>
<td>13.4/min</td>
<td>.0</td>
</tr>
<tr>
<td>order_status</td>
<td>4</td>
<td>29</td>
<td>30</td>
<td>52</td>
<td>4</td>
<td>52</td>
<td>0.00%</td>
<td>53.3/hour</td>
<td>.0</td>
</tr>
<tr>
<td>delivery</td>
<td>6</td>
<td>146</td>
<td>125</td>
<td>174</td>
<td>104</td>
<td>216</td>
<td>0.00%</td>
<td>2.9/min</td>
<td>.0</td>
</tr>
<tr>
<td>stock_level</td>
<td>6</td>
<td>62</td>
<td>6</td>
<td>12</td>
<td>4</td>
<td>335</td>
<td>0.00%</td>
<td>2.9/min</td>
<td>.0</td>
</tr>
<tr>
<td>TOTAL</td>
<td>157</td>
<td>59</td>
<td>43</td>
<td>120</td>
<td>4</td>
<td>632</td>
<td>0.00%</td>
<td>30.0/min</td>
<td>.2</td>
</tr>
</tbody>
</table>
Automating TPC-C Benchmarking

Aggregate Graph

[Bar chart showing performance metrics such as SQL level, table manager, order status, delivery, and new orders, with data for each category depicted in different colors representing average, median, 50% line, min, and max values.]
Response time over time
Challenges

1. Auto CSV Generation: Generating CSV file for a particular table from a specified Database
Challenges

1. Auto CSV Generation: Generating CSV file for a particular table from a specified Database

2. Dynamic Bandwidth Throttling: Changing bandwidth in runtime and determining a reliable parameter (Percentage Error) to implement throttle bandwidth
Challenges

1. **Auto CSV Generation:**
   Generating CSV file for a particular table from a specified Database

2. **Dynamic Bandwidth Throttling:**
   Changing bandwidth in runtime and determining a reliable parameter (Percentage Error) to implement throttle bandwidth

3. **IP Spoofing:**
   Finding an array of un-used IPs and automating allocation of virtual IPs for a system
Challenges

1. Auto CSV Generation: Generating CSV file for a particular table from a specified Database

2. Dynamic Bandwidth Throttling: Changing bandwidth in runtime and determining a reliable parameter (Percentage Error) to implement throttle bandwidth

3. IP Spoofing: Finding an array of un-used IPs and automating allocation of virtual IPs for a system

4. Automating TPC-C testing: Making tpcc testing capable of being run with different databases incorporating all the standards into JMeter
Future Work

1. Incorporating other Benchmarking support such as TPC-H, TPC-E etc. into JMeter.
Future Work

1. Incorporating other Benchmarking support such as TPC-H, TPC-E etc. into JMeter.
2. Automation of the test scripts, as in user may not have to create the test script, and Jmeter can itself do it for the tester by techniques such as web crawling, etc.
Future Work

1. Incorporating other Benchmarking support such as TPC-H, TPC-E etc. into JMeter.
2. Automation of the test scripts, as in user may not have to create the test script, and Jmeter can itself do it for the tester by techniques such as web crawling, etc.
3. The instability of JMeter on large loads could be worked out with some solution.
Future Work

1. Incorporating other Benchmarking support such as TPC-H, TPC-E etc. into JMeter.
2. Automation of the test scripts, as in user may not have to create the test script, and Jmeter can itself do it for the tester by techniques such as web crawling, etc.
3. The instability of JMeter on large loads could be worked out with some solution.
4. Bringing large download efficiency into JMeter.
Future Work

1. Incorporating other Benchmarking support such as TPC-H, TPC-E etc. into JMeter.

2. Automation of the test scripts, as in user may not have to create the test script, and Jmeter can itself do it for the tester by techniques such as web crawling, etc.

3. The instability of JMeter on large loads could be worked out with some solution.

4. Bringing large download efficiency into JMeter.

5. Better analysis of the results produced by JMeter via some complex graphs and better comparison between different graph results.
Conclusion

- The basic aim of the project was to enhance JMeter with the introduction of some new features and overcome some drawbacks.
The basic aim of the project was to enhance JMeter with the introduction of some new features and overcome some drawbacks.

We have made JMeter capable of performing more practical tests with the introduction of bandwidth throttling and IP Spoofing.
Conclusion

- The basic aim of the project was to enhance JMeter with the introduction of some new features and overcome some drawbacks.
- We have made JMeter capable of performing more practical tests with the introduction of bandwidth throttling and IP Spoofing.
- The user friendliness of JMeter has been improved with the introduction of auto csv generation and results filtering.
Conclusion

The basic aim of the project was to enhance JMeter with the introduction of some new features and overcome some drawbacks.

We have made JMeter capable of performing more practical tests with the introduction of bandwidth throttling and IP Spoofing.

The user friendliness of JMeter has been improved with the introduction of auto csv generation and results filtering.

Finally Jmeter has been extended from just a load testing tool to a Preliminary TPC-C benchmarking tool.
References


Learning Shapes

A simple application to teach young minds the sense of shapes and colors. This app tests the recognition skill shapes and colors among children.

Traingles for High School

An application to aid learning about Triangles, allows construction of Triangles, also drawing Triangles from user Input.
Educational Application

Mathematics Playground

Best friend of any student during examination for quick revision of the formulas. The detailed proofs and animation helps the student in recalling the logic behind the formula.

Base Conversion

This application teaches Base Conversion, it describes the conversion procedure from any positive integral base to any other positive integral base.
Educational Application

Area of Polygons

This application teaches methods to calculate area of any Polygon. Polygon is drawn by user input and area is calculated and displayed with explanation.

Currency Converter

This application provides the methods of currency conversion as well as the details of the country to which the currency belongs.
Enhancement of JMeter - Team Members

B. Sushmitha
Dhruv Joshi
Manisha Choudhury
Naman Choudhary
Shekhar Saurav
Surabhi Mour