

Load Testing with JMeter

Presented by Matthew Stout - mat@ucsc.edu



UCSC ITS - APM - Infrastructure & Operations

JMeter Overview

- Java application for load testing and measuring performance
- Originally for web applications but has grown to support lots of other functions, SQL, Java, JUnit, SMTP, SOAP, LDAP, etc.

JMeter Overview continued

- Can run on just about any workstation or environment that can run java
- Has a GUI and non-GUI mode
 - GUI for developing and watching tests
 - non-GUI for running more load or on numerous hosts/load engines only

JMeter Overview continued

- Server mode for controlled distributed tests
- Can generate lots of load or be used with just a single user/thread if you are just interested in testing for specific responses or content.

JMeter Terms

Test Plan	The container for all test objects
Thread Group and Threads	Threads = Users
Controllers	Containers for grouping Samplers, other elements, add logic (if, random, once only)
Samplers	Makes Requests
Assertions	Validations on responses, step results
Listeners	Reports results/monitors performance
Workbench	Temporary working area to add some items of temp use -- Proxy recorder, etc

JMeter Terms



See JMeter Docs for more on components, functions, etc. This workshop is intended to show some real uses and start interest in ITS, so I am not going to go into all the features--Docs are out there though...

- http://jmeter.apache.org/usermanual/component_reference.html
- <http://jmeter.apache.org/usermanual/best-practices.html>
- <http://wiki.apache.org/jmeter/>

Install JMeter

- Use latest version and latest jre at least 1.6
- Can bundle and set your own location for JRE easily on Mac/Unix
- Very easy see Apache docs or my full presentation on the UCCSC site

Install JMeter

- Typical Load by type of Machine
 - Typical laptop can do 150-500 users depending on cpu/ram and test elements.
 - Typical Oracle T4-1 can do 500-2000 users depending on test
 - You WILL need to increase the heap settings for larger tests or complex tests with lots of logic or listeners storing results
 - I then run some client/server coordinated tests, or command line non-gui--the later has lowest resource needs
 - Running more instances of smaller tests of 300-1000 users is better as overloading a JMeter install can make results look slow when it is only JMeter

Installing JMeter

Show an install ...



Ways to Create a Test

- Hand enter samples/requests
- Read in URLs from file--log replay
- Record a test
- Third party tools

Ways to Create a Test - continued

- However you start you'll need to...
 - Identify scenario or steps & create test
 - Debug, refine, parameterize
 - Add validations/Assertions
 - Define load needed and run test
 - Monitor (logs, Listeners, other)
 - Tune/Change App or Systems, retest

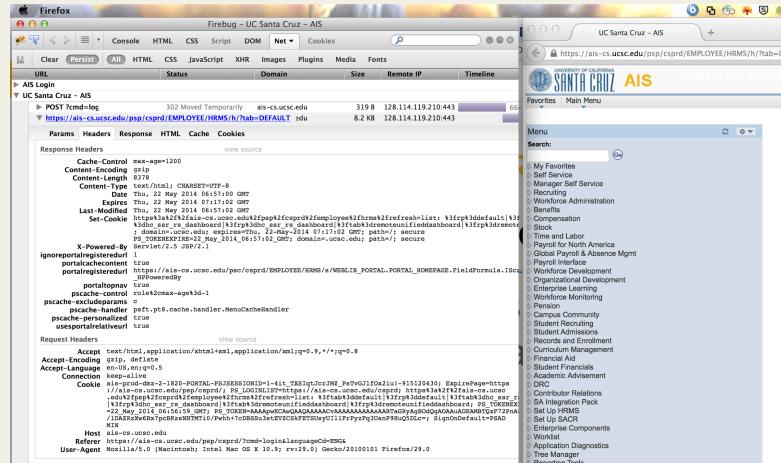
Ways to Create a Test - continued

- I often record or start by hand and then use Firefox and Firebug (lots of other good browser debug tools too)



Ways to Create a Test - continued

- Firebug or similar tool helps see the activity and adjust a test faster



Hand Entering a Test

- Simple for basic tests
- Stress tests of lots of GETs or simple POSTs
- Not likely used for creating real user transactions that test the real user load-- but could be better than no tests

Hand Entering a Test

Queue up the Live Demo...



Read URLs from a File

- Useful for general content sites with no authentication or simple authentication
- Quick way to replay the actual requests your app gets
- Can generate lots of random load
- Not all apps generate URLs in logs that can replay or that are useful for this

Read URLs from a File

Queue up the Live Demo...



Recording a Test

- This is usually where most demos start
- I wanted to show the previous ways as options to get you doing some load testing with less initial effort
- All methods can generate load
- This allows you to record a specific user transaction or experience

Recording a Test - continued

- Specific user transactions more useful for
 - Testing total time and user experience on multiple step processes
 - Many sites require certain steps to set up sessions/authentication
 - More useful if using as a more general testing/release validation tool

Recording a Test - continued

- To play back such a test may take some debugging
- Most sites require parsing responses for values to send in subsequent requests
- Use browser or http tracing tools to watch what happens in a real session to debug and add to what happens in JMeter

Recording a Test

Queue up the Live Demo...



Summary so far...

- Lots of ways to start a test
- Keep evolving the test
- Remove unneeded cruft and add extractors and assertions to remove errors and check for success
- Determine load and how you'll monitor
- Run tests

Summary so far...

- Then once you have a valid test
 - Watch results for sudden spikes,
isolate cause, tune one thing, retest
 - Repeat same exact test to measure
improvement

Assertions

- Add at least basic assertions
 - Check for strings in response
 - Response code if appropriate
 - Without these it is easy to get a success or OK return from JMeter when really the app may be giving an error (Err Msg, but 200)

Ways to Generate Load

- Single Test Machine
 - Easy to run from laptop or workstation for tests under 250 users or so
- Use several servers and non-GUI call to standalone tests
 - Drawback of no single monitoring point

Ways to Generate Load

- Distributed Test
 - Start client instances and invoke tests from control machine
 - Gives you data collection in a single machine, but that machine is eventually still a bottleneck and needs low latency to remote machines

Ways to Generate Load

- Ideal (or things to consider)
 - Depends on your needs, but would
 - Have clients spread out more like your real users so campus network or ISPs are in the mix if you are trying to simulate what they will see

Ways to Generate Load

- Ideal (or things to consider) continued
 - It is possible to see higher load and slower site with slow users than a load test since they keep processes locked up longer on connections than a fast connection

Ways to Generate Load

- Ideal (or things to consider) continued
 - We are exploring VMs and Cloud instances for simulating more unique sources that are more distributed from our site

Ways to Monitor Performance

- JMeter
 - Aggregate Report
 - Summary min, max, average of all samples
 - Aggregate Graph -- graph of same data

Ways to Monitor Performance

- JMeter
 - Plugins from <http://jmeter-plugins.org/> have some improved graphs
- System Logs
 - web server access logs -- add time-taken and compare responses from the real peak times of year/events

Ways to Monitor Performance

- System Monitoring
 - Use system monitoring tools and scripts -- Shinken/Nagios/Cacti, others, or use scripts to capture system metrics (cpu, load, memory), connections, etc. every minute or so

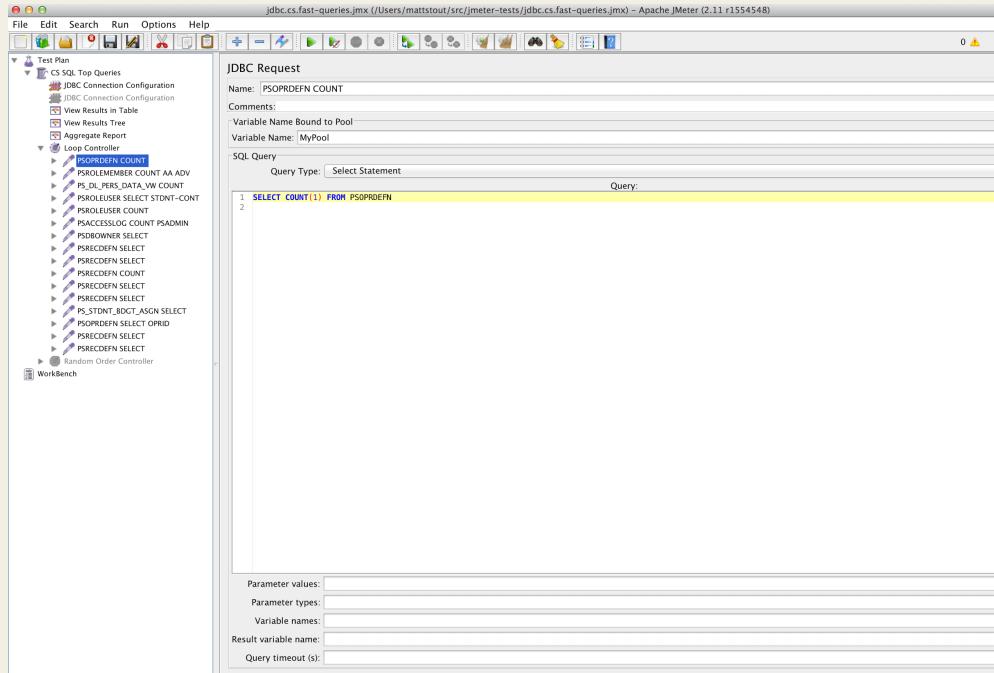
Some more Fun Examples

- I wanted this workshop to show the possibilities and inspire others to use the tool so I have some more uses...
 - SQL, SMTP, Shibboleth Login, Distributed tests, real tests we use, set off some pagers...

Some more Fun Examples

- SQL/JDBC
 - If your platform has JDBC driver likely can run this kind of test
 - Run SQL statements
 - Has some issues with large responses when your load generator is on slow network

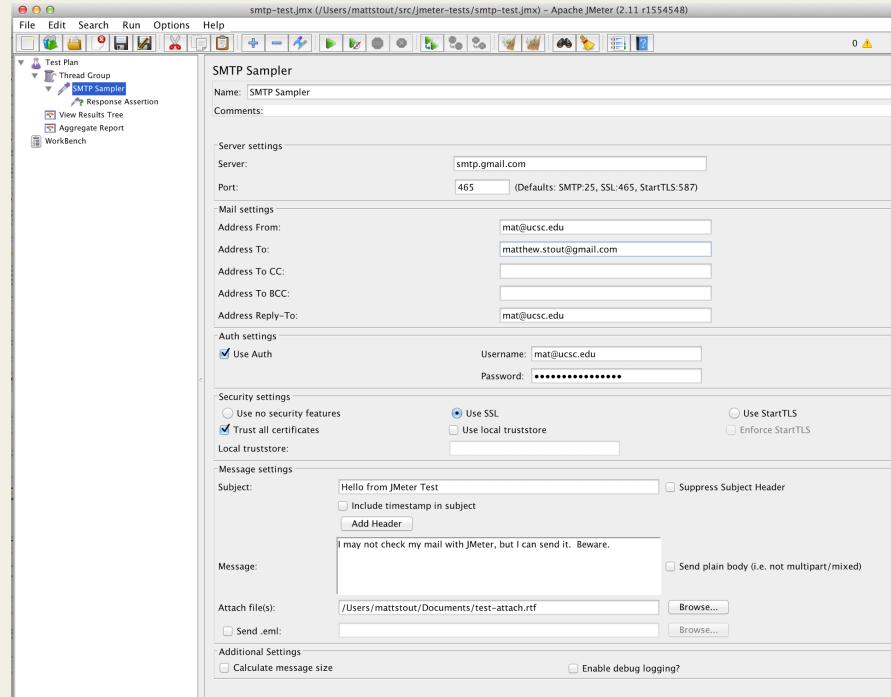
Some more Fun Examples - JDBC



Some more Fun Examples

- SMTP Sampler
 - Had to include this after it was advertised that I use JMeter with my email (be careful when challenging Sys/App Admins)

Some more Fun Examples - SMTP



Some more Fun Examples

- Distributed Test - More Loaded
 - Remote server mode
 - More control/visibility
 - Simple non-GUI across many machines
 - Lowest resources

Some more Fun Examples

- Distributed Test - More Loaded
 - See Apache for Setup:
 - http://jmeter.apache.org/usermanual/jmeter_distributed_testing_step_by_step.pdf



Fun Examples - Distributed Test

Summary Report

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	kB/sec	Avg. Bytes
https://ais-ucsc.edu	4	142	140	144	1.46	0.00%	7.6/sec	2.69	
https://apply.graad.	4	360	252	303	12.95	0.00%	7.6/sec	57.25	7
https://ais-reports.uic.edu	4	269	267	273	2.38	0.00%	9.2/sec	22.20	
https://ais-reports.uic.uchicago.edu	4	2	2	2	0.00	0.00%	23.4/sec	46.21	2
https://ais-ucb.edu	4	127	118	135	0.14	0.00%	13.2/sec	3.98	
https://ais-reports.uchicago.edu	4	147	143	152	4.03	0.00%	12.2/sec	63.24	5
https://ais-reports.uchicago.uchicago.edu	4	228	87	432	147.63	0.00%	6.5/sec	19.42	3
https://ais-reports.uchicago.uchicago.uchicago.edu	4	24	22	26	1.48	0.00%	7.2/sec	21.61	3
https://ais-reports.uchicago.uchicago.uchicago.uchicago.edu	4	129	122	138	5.67	0.00%	6.0/sec	3.17	
https://ais-uic.edu	4	121	103	165	2.07	0.00%	4.9/sec	3.97	
https://ais-uic.uchicago.edu	4	174	169	178	3.25	0.00%	4.7/sec	3.69	
https://slugilla.uchicago.edu	4	2615	2597	2629	11.80	0.00%	1.2/sec	5.60	4
TOTAL	52	367	2	2629	663.13	0.00%	9.6/sec	24.40	2

Some more Fun Examples

- Shib Authentication
 - Yes we finally have working examples
 - One for a simpleSAML.php and the package install--slightly different
 - Demo!

Some more Fun Examples

- Nagios Plugin?
- Maybe...
 - In development I have a simple plugin that invokes a test, loads the results, checks for errors, total time and returns codes for warn, critical, ok, etc.

Resources

- The Encyclopedia, source, mothership:
 - <https://jmeter.apache.org/>
 - <http://wiki.apache.org/jmeter/>
- Blazemeter - browser recorder, blog, a couple good training videos
 - <http://blazemeter.com/>

Resources

- Blazemeter Youtube
 - <https://www.youtube.com/user/BlazeMeterSupport>
- Blazemeter Blog (some are for their products, but also some JMeter)
 - <http://blazemeter.com/blog>

Resources

- Add to JMeter
 - <http://jmeter-plugins.org/>

Q & A

