

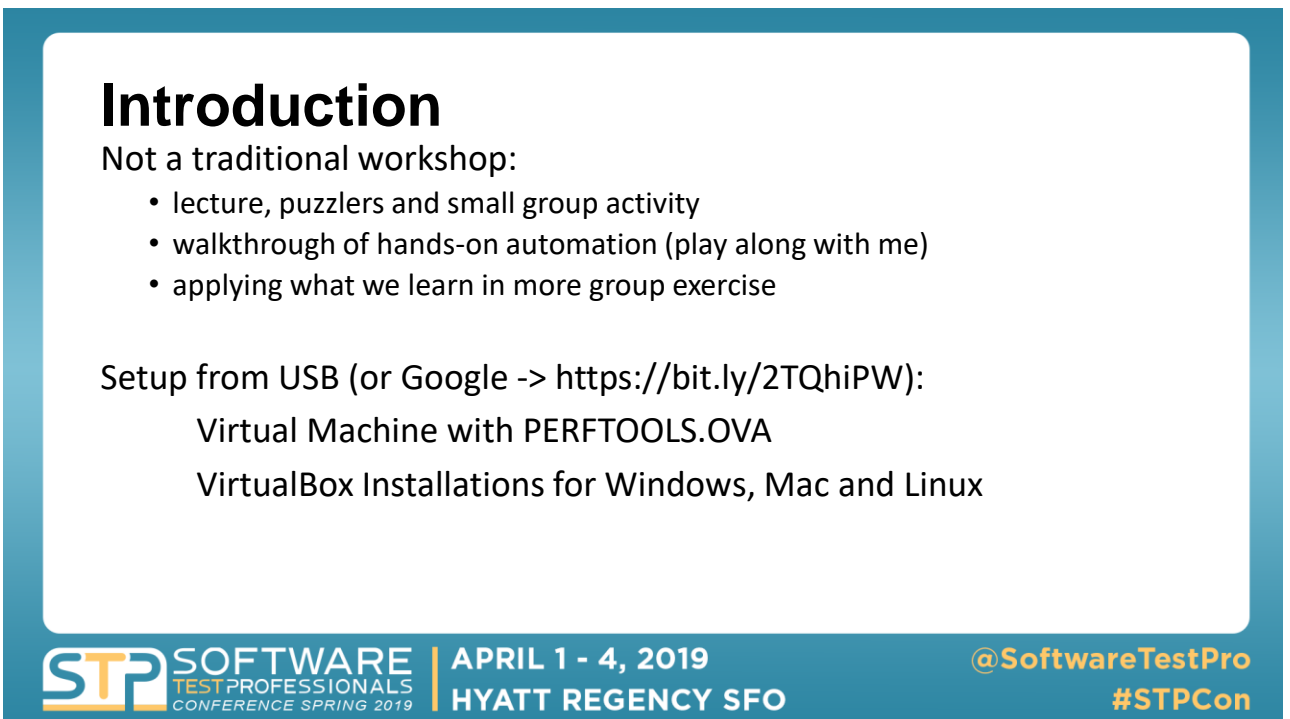


STP SOFTWARE
TEST PROFESSIONALS
CONFERENCE SPRING 2019

Performance Automation for Everyone

Mark Tomlinson, Performance Architect, Freedompay

1



Introduction

Not a traditional workshop:

- lecture, puzzlers and small group activity
- walkthrough of hands-on automation (play along with me)
- applying what we learn in more group exercise

Setup from USB (or Google -> <https://bit.ly/2TQhiPW>):

- Virtual Machine with PERFTOOLS.OVA
- VirtualBox Installations for Windows, Mac and Linux

STP SOFTWARE | APRIL 1 - 4, 2019
TEST PROFESSIONALS | HYATT REGENCY SFO
CONFERENCE SPRING 2019

@SoftwareTestPro
#STPCon

2

Objectives for Today

- Expand our ability to influence everyone and everything...
- Enhance our delivery strategy for results...
- Improve how our tools interoperate...
- Practice our new understanding of skills...
- Elaborate on ways to cooperate...



...ABOUT PERFORMANCE!

3

The Next Three Hours...

Part One:

- Performance Is Everywhere, Performance is for Everyone
- Performance Tool Basics and Architectures

Part Two:

- Expanding Performance Automation
- Challenges of Shifting Left or Right (and up...and down)
- With Hands-on Walkthrough of Automation

Part Three:

- Early Performance in Functional Automation

4



5

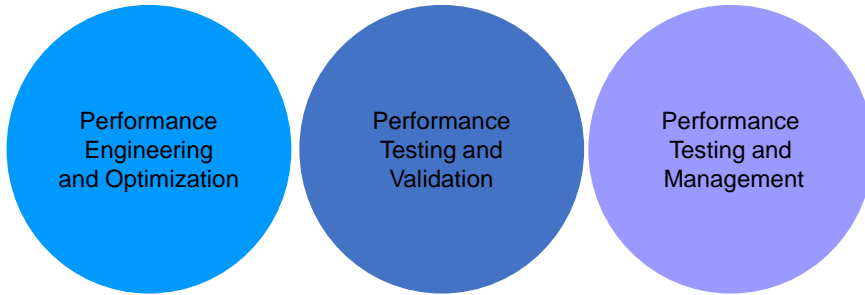
Performance is Everywhere

Sure, lots of people are talking about performance everywhere, but:

- tools seem to be stuck in load + monitoring paradigm
- performance information doesn't travel the pipeline efficiently
- training for "non-testing" disciplines is limited...quite so
- we still just expect that good performance is built-in by devs
-

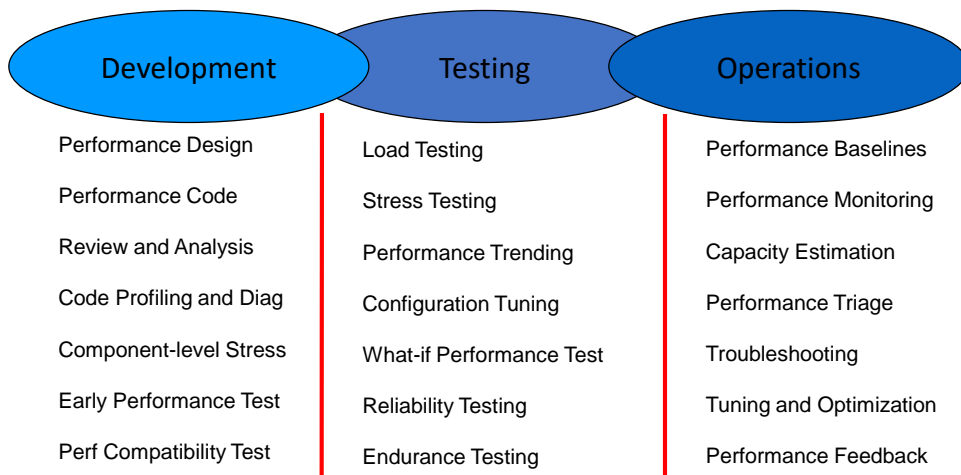
6

Performance is Everywhere



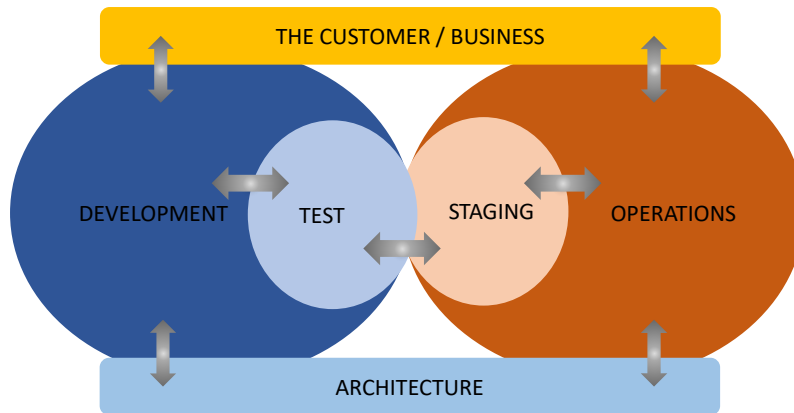
7

Performance is Everywhere



8

Performance is for Everyone



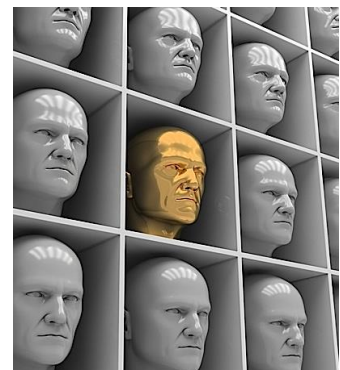
9

Performance Personalities

Different people respond to performance...

- at different speeds
- with different temperament
- with different biases,
- From different contexts
- with different goals/agendas

If we can adjust how and when performance information is shared across the team, we can help people to respond functionally and efficiently.



10

Performance Roles & Personalities

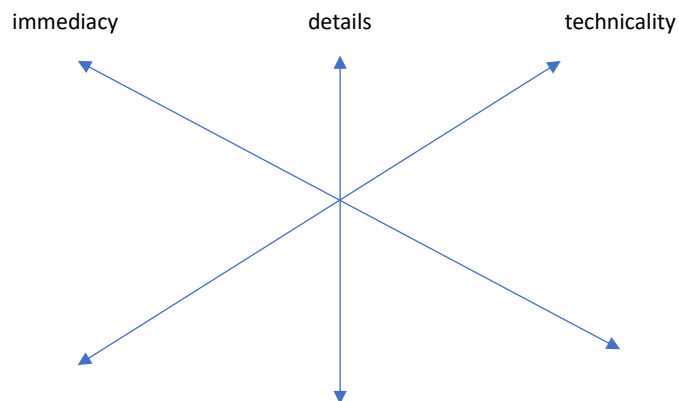
Extroverted:

- product owners
- sales managers
- team leads
- scrum masters
- marketing people
- business analysts
- leadership
- customer support

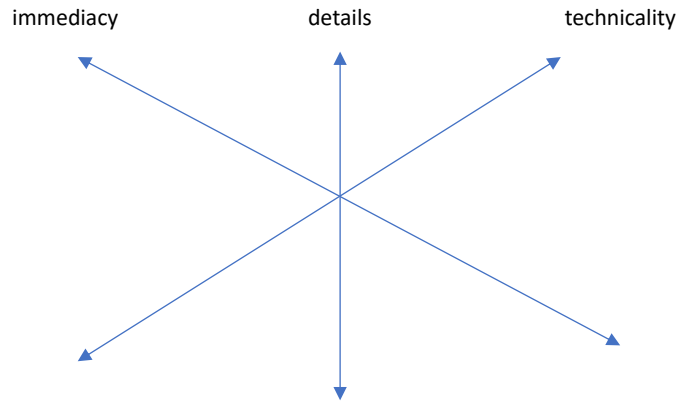
Introverted:

- developers / engineers
- administrators
- data scientists
- testers
- automation engineers
- technical writers

Axes of Performance Acumen



Axes of Performance Acumen



13

Axes of Performance Acumen

immediacy	Fast & Urgent	Slow & Delayed
details	Low & ½ page	High & 10 pages
technicality	Users & Business	Engineers & Geeks

14

Example Exercise

The database administrator was hired recently from another company where he was the manager of the data team. When the database alerts go off, he's the first one on the emails giving direction in a series of one-liner sentences.



Describe how this person likely responds to performance:

- Immediacy?
- Detail?
- Technicality?

Product Owner Performance

Mary is a product owner who loves sprint planning. She's got experience with both UX design and coding as well as data science and big data tooling. When she hears of a performance challenge she asks several people on the team to dig in with her to figure it out.



Describe how this person likely responds to performance:

- Immediacy?
- Detail?
- Technicality?

Production Support Engineer (SRE)

Tim never knows what hit him. In the middle of the night when the report queues back up and he's got to start digging. He never had to do this kind of troubleshooting in his last job. And management is always on his back for production escalation metrics. Tim might have a drinking problem.



Describe how this person likely responds to performance:

- Immediacy?
- Detail?
- Technicality?



Expanding Performance Automation

When shifting left, we have new performance challenges:

- Our automation should support rapid repeated early performance
- The scripts should be reusable and adaptable
- Testing early means integrating with the CI pipeline
- Virtualization of missing or late components is required

When shifting right, we have other performance challenges:

- Test's should be highly accurate – for reproducing bugs
- Scripting should support different scenarios
- Test results should recommend thresholds for prod

Challenge: Change the Test, Run again NOW!

It's so common to hear this with your load test executions:

- "Oh, I think we need to run more load. Can you change the test and run again?"
- "Oh, I think we need to run less load. Can you change the test and run again?"
- "Well, we need to run fewer threads faster. Can you change the test and run again?"
- "Ummm...maybe we need to run more threads slower? Can you change the..."
- "Hmmm...is this different from the previous run? Did you change it...and run it again?"

The tips to help you include:

- make the run-time settings of the load test easy to change and reconfigure
- automatically compare different runs of the test

WALKTHROUGH:

command—line parameters
rundeck job options

login: perftools / P3rfT00l\$

21

Challenge: you need to test sooner and more

Wouldn't it be nice if we could get those developers to do some small load tests?

- Even running 10 threads concurrent can find race conditions and blocking.
- Running 100 iterations a.f.a.p. as a benchmark for current build & trend.
- But developers don't use my test tool, because reasons.
- And my scenarios are built for huge load tests - not ones that devs understand.
- I don't have the time/people to build all this and keep up with my real job.

The tips to help you include:

- make the scripts modular and share with developers
- they have scenarios that are very small - at unit level, leveraging input params
- you have scenarios that are bigger - integrating multiple modules, different params

22

WALKTHROUGH:

script modularization/reuse

same script – different scale

23

Challenge: reduce time during test cycles

Load test script and scenario maintenance is a pain...especially in the midst of execution:

- If script code is not reused efficiently, lots of last-minute code updates.
- If scenario configuration is unshared that's even more time spent updating stuff...
- As we plug our load test tools into CI - the scripts and scenarios should be smarter.
- And if you're getting bored with tedious tweaks of your test tool, you'll go insane!

The tips to help you include:

- parameterize the metadata configuration for the script
- parameterize the metadata configuration for the scenario
- use a pre-test check to determine which calls/scripts/scenarios to exclude

24

WALKTHROUGH:

command line parameters rundeck job options

25

Challenge: test what's working as of now

We are testing every week (or every night) but sometimes dependencies fail!!

- Excuses: "The system is not ready to test." (hint: qualify the word system)
- Consider a distributed services layer sharing only parts of data sources.
- Is the entire test run invalid if you miss just one call failing? One script?
- What if the cause of the failure is known and "approved" for the build?

The tips to help you include:

- use conditional logic in the scripts to exclude failing calls or components
- use conditional logic in the scenario to exclude failing scripts
- don't arbitrarily comment out calls if they're not working - that's not okay!
- use a pre-test and post-test logging of the inventory for the system under test

26

WALKTHROUGH:

samplers and sampler mix

data setup

pre-script / post-script jobs

27

Challenge: make the test more realistic

To produce valuable and meaningful results, make the test more like PROD:

- What are the average transactions per minute in production?
- What's the future estimation for transactions per second?
- Is it realistic to use the same data values over and over again?
- For larger and larger load tests, do you have enough data?

The tips to help you include:

- configure a dynamic command-line parameter for pacing and think time
- randomize your test data - some tools are easier than others
- use unique test data across all scripts or virtual users
- configure more real-world schedules for complex execution

28

WALKTHROUGH:

randomizing data for use

extracting and splitting unique data

data-driven pacing/think-time

29

Challenge: performance thresholds

Sometimes there is a very specific threshold for the max response times:

- The trick is this - is it a transient error under load, or an absolute failure?
- Start by setting an higher-than required SLA/Assertion - learn from your observation.
- If the response time goes over that, then fail the transaction.
- If there are enough violations, then end the script.
- If there are enough scripts failing, then end the whole scenario.

Exercise:

- enable the Duration Assertions in the YouLoveLoadTesting5.jmx - or other script.
- determine what the duration assertion should be
- run the test

30

WALKTHROUGH:

duration assertions

results logging and sharing

notifications

31

Challenge: supporting multiple scenarios

Your work should be to determine what kinds of scenarios are most valuable?

- What kind of application are you testing?
- What's the pattern for application usage in the real world?
- Is there a history of outages? What caused them?
- Do you have marketing promotions that cause huge load?
- Does the traffic come from real users or from external web service calls?

Exercise:

- choose 2 kinds of scenarios that most valuable
- make 2 copies of YouLoveLoadTesting5.jmx and build those scenarios
- experiment with the Jmeter Plugins different Thread Groups and Timers

32

WALKTHROUGH:
multiple thread groups
data offsets
integrated scenarios / jobs

33



34

In Summary

- We learned about performance personalities
- We looked at the 3 axes of performance:
 - Immediacy
 - Detail
 - Technicality
- We looked at the challenges of shifting left/right/up/down
- We walked through performance automation examples
- We walked through more deeply on web performance in testing

In Summary

As your awareness of performance and impact to performance expands across the team, so must your automation ideas expand to enable efficient performance practices and behaviors.

Thank You!

Mark Tomlinson
 Performance Sherpa
www.perfbytes.com
 mark@perfbytes.com
 +1-773-775-8780
 perfsherpa.blogspot.com
 @perfsherpa
 Host of STPRADIO Podcast

Web Performance for the Functional Tester

Beyond a simple loading of data into the browser, consider the following:

- the volume of data moving to/from the client (e.g. page weight)
- the configuration of the content and traffic interactions (e.g. headers)
- the rendering overhead on the client (e.g. CPU and memory, and battery)

And keep in mind two very important limitations:

the QA environment is very rarely configured to the same size as PROD
 the test data in QA is very rarely at the same volume as PROD

REMEMBER: YOU ARE NOT LOAD TESTING!

WHAT'S THIS HAR FILE THING?

HTTP archive, a file format that contains recorded HTTP tracing data.

JSON Format makes it easy to save, store, archive, edit, manipulate.

Contains almost everything you'd need to analyze web performance.

<http://www.maxcdn.com/one/visual-glossary/har-file>

```

• {
•   "log": {
•     "version": "1.1",
•     "creator": {
•       "name": "Firefox",
•       "version": "43.0.4"
•     },
•     "browser": {
•       "name": "Firefox",
•       "version": "43.0.4"
•     },
•     "pages": [
•       {
•         "startedDateTime": "2016-03-01T22:21:03.861-05:00",
•         "id": "page_1",

```

MORE HANDS-ON!

Generate a HAR file: <http://www.webpagetest.org/> or <http://gtmetrix.com>

Download HAR and inspect it manually in notepad/textpad/editor

Copy HAR from FireFox and paste into

<http://www.softwareishard.com/har/viewer/>

Upload to https://toolbox.googleapps.com/apps/har_analyzer/

Integrating Performance into Functional Automation

- your test scripts are already written and running, we don't want to re-write them or refactor anything in a major way to capture performance
- already we could capture the overall elapsed time of the test script execution and trend that in a simple spreadsheet or graphing database
- generating a HAR file is what makes everything work, let's find a way to create it automatically and leverage it into the other tools

Could we automate the calls to these tools with the functional test tool itself?

- webpagetest is a browser-based app
- perfwars.com is a browser-based app

Practice with Browser Tools

Manually capture traffic from Firefox or Chrome

Upload to https://toolbox.googleapps.com/apps/har_analyzer/

Or upload to <http://www.softwareishard.com/har/viewer/>

What other things can you see about performance in the Browser Tools?

WALKTHROUGH:

Waterfall Diagram: <http://www.webpagetest.org/>

Graded Evaluation: <http://www.gtmetrix.com/>

45

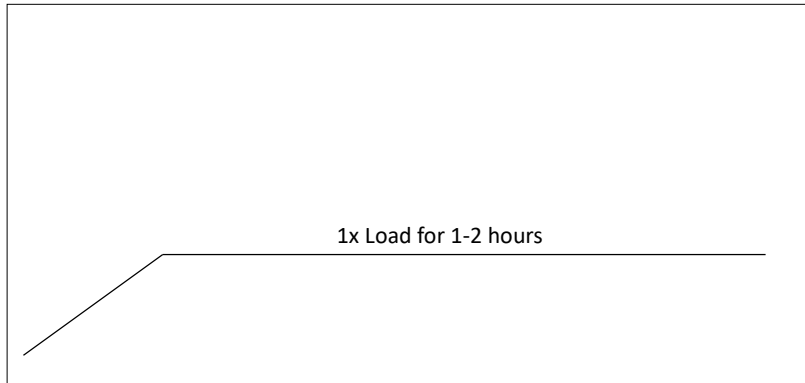
APPENDIX

PERFTOOLS VIRTUAL MACHINE:

- Installed components: InfluxDB, Grafana, Jmeter, SAMBA file sharing
- Terminal login: perftools/ P3rfT00I\$
- Open Rundeck: <http://perftools:4440> with login: perftools / P3rfT00I\$
 - Most jobs are written in BASH scripting
 - the PerfTools projects is stored under /var/rundeck/projects/PerfTools folder
 - There are 3 scenarios scheduled: Baseline Test, Scale Test and Stress Test
- Open Grafana: <http://perftools:3000> with login: admin / admin
 - Demo Dashboard is the primary visualization of running tests
- Jmeter is installed under /opt/jmeter – version 5.11 with all jmeter-plugins
- From terminal session type “influx” command to login to InfluxDB database
- Test assets are under /usr/data/shared folders (scripts, data, results, logs)

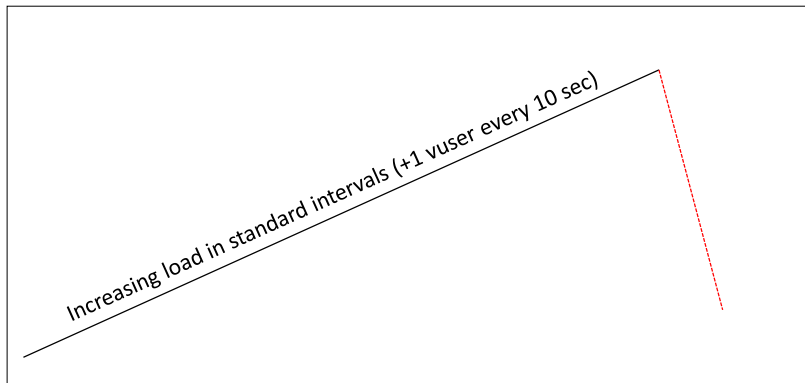
46

Load Test



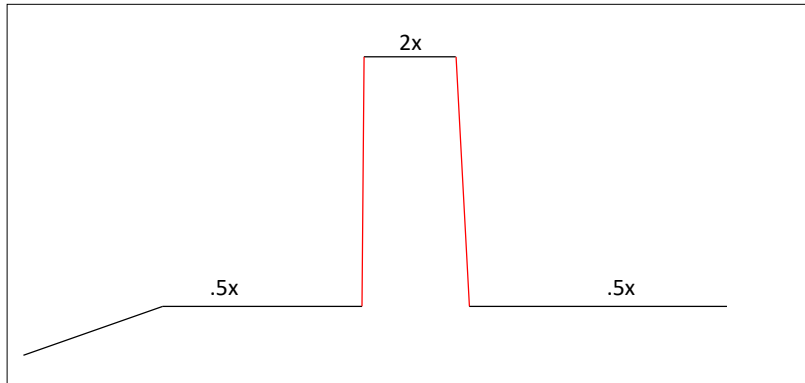
Objective: validate the system's ability to handle a sustained load at a constant specific level (e.g. Run for 2 hours at 1000 trans/sec)

Stress Test



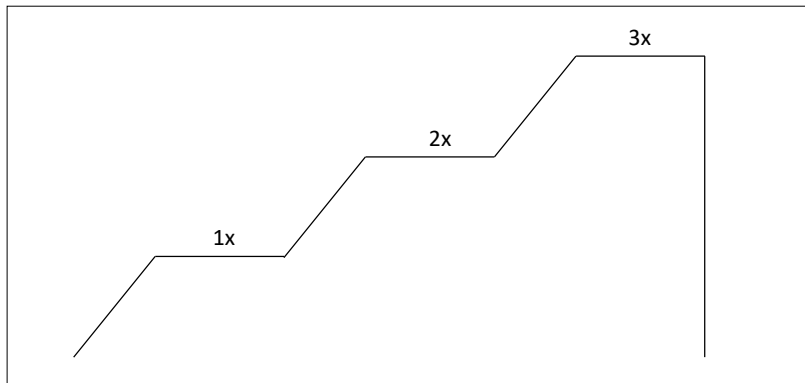
Objective: apply gradually, continuously increasing load to the system until an error state and/or failure occurs (e.g. increase # of virtual users until the system crashes)

Spike Test



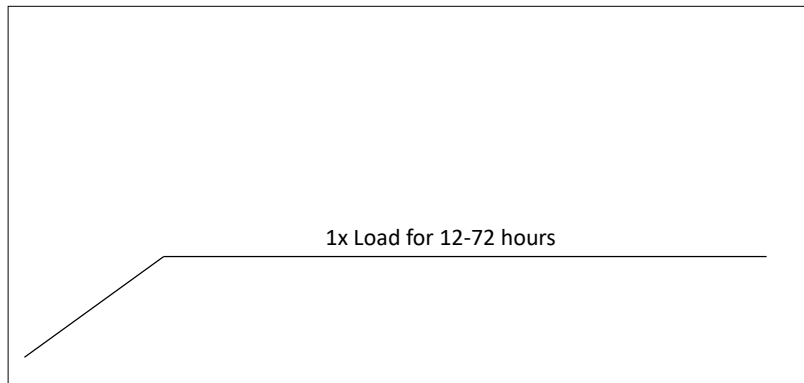
Objective: measure the system's ability to handle extreme demand curve, sustain high demand load for a short period of time and then cleanly recover resources post-demand.

Estimated Scalability Test



Objective: measure system performance at consistent intervals of increased load (1x, 2x and 3x), useful for calculating a trend for performance projections

Soak Test



Objective: running standard load for an extended period of time to validate resource usage can be sustained without exhaustion (e.g. measure disk usage, memory leaks, ramp conditions)

Automatically capturing HAR files

Hey...we're automating this stuff, I'm not going to manually copy and paste the HAR data up to the web, eh? Can't I just automatically generate the HAR files for each page load?

Try the HAR Export Trigger: <http://www.softwareishard.com/blog/har-export-trigger>

- open the about:config page in FireFox to configure the settings
- this will generate an HTTP Archive file for every page you post
- then configure your automated testing tool:
 - clear the /har/logs before each test run
 - after each test run upload each HAR file for visual rendering
 - after each test run compare the current HAR file to a previous HAR file?

Live Proxy / HAR Tools

Here are a few more live website tools to explore (YMMV):

- www.perfwars.com
- <http://requestmap.webperf.tools>
- <http://heatmap.webperf.tools>
- <http://www.httpwatch.com>
- flood.io/har2jmx will convert your copied HAR string into a JMeter
- <https://github.com/andrewf/pcap2har>