



ALMFORUMSEATTLE

April 1-3, 2014 • Washington State Convention Center

Sponsored by



Load Testing: See a Bigger Picture

Alexander Podelko
alex.podelko@oracle.com
www.alexanderpodelko.com
@apodelko

April 1, 2014



Speaker Bio

- Have specialized in performance for the last 17 years
- Currently performance testing and optimization of Hyperion products at Oracle
- Board director at CMG (<http://cmg.org>), organization of performance and capacity professionals
 - Next conference November 3-6, 2014 in Atlanta, GA

Disclaimer: The views expressed here are my personal views only and do not necessarily represent those of my current or previous employers. All brands and trademarks mentioned are the property of their owners.

In This Session, You'll Learn...

- *Load testing and its role in performance risk mitigation*
- Different approaches: load generation
- Different approaches: environments
- Load testing tools

Load Testing: Terminology

Applying multi-user synthetic load to the system

- Load testing
- Performance testing
- Stress testing
- Scalability testing
- Volume testing
- Reliability testing
- Concurrency testing
- Endurance testing
- Longevity testing
- Soak testing
- Stability testing

The Stereotype

- Load / Performance Testing is:
 - Last moment before deployment
 - Last step in the waterfall process
 - Protocol level record-and-playback
 - Large corporations
 - Expensive tools requiring special skills
 - Lab environment
 - Scale-down environment
 - ...

Load Testing

- It is only one very specific kind of load testing
 - Most popular due to easier integration in corporate SDLC
- But load testing in no way limited to this one

Technology evaluation

What/if scenarios

Infrastructure evaluation

Performance troubleshooting

Prototypes / POC

Performance optimization

Component / unit

Benchmarking

Performance Risk Mitigation

- Single-user performance engineering
 - Profiling, WPO, single-user performance
- Software Performance Engineering
 - Modeling, Performance Patterns
- Instrumentation / APM / Monitoring
 - Production system insights
- Capacity Planning/Management
 - Resources Allocation
- Continuous Integration / Deployment
 - Ability to deploy and remove changes quickly

But all of them
don't replace
load testing

Load testing
complements them in
several important ways

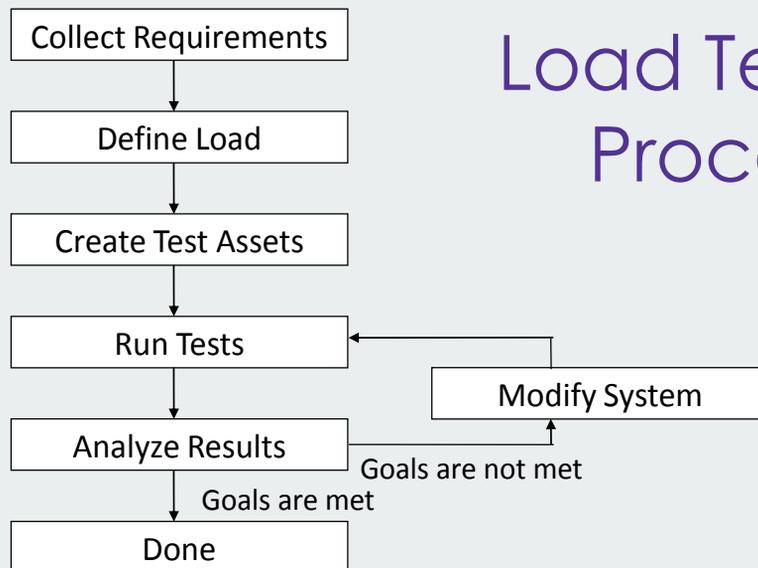
What Load Testing Adds

- Verification that the system handles the load
- Verification of multi-user performance
- Performance optimization
 - Exactly the same load
- Debugging/verification of multi-user issues
- Testing self-regulation functionality
 - Such as auto-scaling or changing the level of service depending on load

Agenda

- Load testing and its role in performance risk mitigation
- *Different approaches: load generation*
- Different approaches: environments
- Load testing tools

Load Testing Process



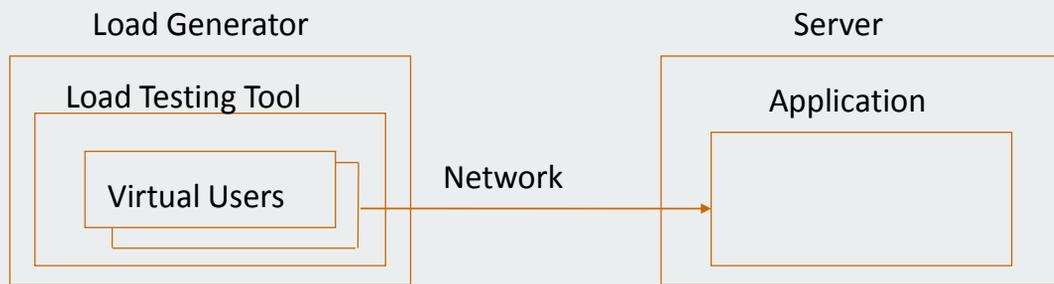
Load Generation

- Create test assets – run test
- A must step in load testing
- 'Test assets' - usually scripts or programs in load testing
- Time constraints can make it very challenging
 - Different for each product / interface

Record and Playback: Protocol Level

- Virtual users: record communication between two tiers and then playback an automatically created script
- Usually after proper correlation / parameterization

Record and Playback: Protocol Level



Issues

- Usually doesn't work for testing components
- Each tool support a limited number of technologies (protocols)
- Some technologies are very time-consuming
- Workload validity in case of sophisticated logic on the client side is not guaranteed

Recalling Challenges

- 199x – SMB (Server Message Block), DCOM, Java RMI
- 200x – Applets and ActiveX controls
- 201x – Rich Internet Applications

It was just a short period of time when Web sites were simple – it was rather an exception

Not so Simple: Example

- Back-end calculation (financial consolidation)
 - Long time, shows progress bar
 - Polling back-end
 - Explicit loop is needed to work properly

Recorded Script

```
web_custom_request("XMLDataGrid.asp_7","URL={URL}/Data/XML
DataGrid.asp?Action=EXECUTE&TaskID=1024&RowStart=1&ColS
tart=2&RowEnd=1&ColEnd=2&SelType=0&Format=JavaScript",
LAST);
web_custom_request("XMLDataGrid.asp_8","URL={URL}/Data/XML
DataGrid.asp?Action=GETCONSOLSTATUS", LAST);
web_custom_request("XMLDataGrid.asp_9","URL={URL}/Data/XML
DataGrid.asp?Action=GETCONSOLSTATUS", LAST);
web_custom_request("XMLDataGrid.asp_9","URL={URL}/Data/XML
DataGrid.asp?Action=GETCONSOLSTATUS", LAST);
```

Working Script

```
web_custom_request("XMLDataGrid.asp_7","URL={URL}/Data/XML
DataGrid.asp?Action=EXECUTE&TaskID=1024&RowStart=1&Col
Start=2&RowEnd=1&ColEnd=2&SelType=0&Format=JavaScript",
LAST);
do {
    sleep(3000);
    web_reg_find("Text=1","SaveCount=abc_count",LAST);
    web_custom_request("XMLDataGrid.asp_8","URL={URL}/Data/X
MLDataGrid.asp?Action=GETCONSOLSTATUS", LAST);
} while (strcmp(lr_eval_string("{abc_count}"),"1")==0);
```

Alternatives

- Manual
- Real Users
- Record and Playback, UI Level
- Programming
- Mixed

Manual

- Not an option for a large number of users
- Always variation in human input times
- Can be a good option to simulate quickly a few users
- Can be used with other methods to verify correctness

Real Users

- Testing in production
 - Full load
 - Partial load (A/B testing, canary testing)
- You trade in the need to generate and validate workload for a possibility of performance issues and load variability

Considerations

- May make sense for the following conditions
 - Potential issues have minimal impact on user satisfaction and company image
 - Easy rollback of the changes
 - Homogenous workload and a way to control it
 - Fully parallel and scalable architecture

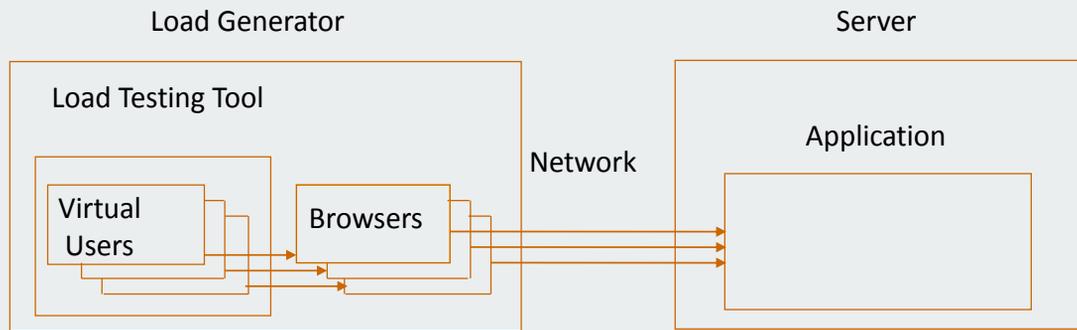
Record and Playback: UI Level

- Functional / regression testing tools
- Record and playback communication between user and client GUI
- Don't care about communication protocols / internals
- Accurate data (real client, end-to-end)

History of the Approach

- Traditional tools, fat clients
 - Require a separate machine (or a terminal session) per user
- Low-level graphical protocols
 - Citrix, Remote Desktop
- Web tools, browser
 - Require a separate browser instance
- Web tools, light-weight browser
 - Require a separate light-weight browser instance
 - For example, HtmlUnit or PhantomJS

Record and Playback: UI Level



Challenges

- Scalability
 - Still require more resources
- Supported technologies
- Timing accuracy
- Playback accuracy
 - For example, for HtmlUnit

Programming: Custom Test Harness

- Special program to generate workload
- Requires access to the API or source code
- Requires programming
- Could be cost effective solution in some simple cases

Advantages

- Doesn't require any special tool
- Starting version could be quickly created by a programmer familiar with API
- Should work if API works
- You don't care what protocol is used for communication

Disadvantages

- Efforts to update and maintain harness increase drastically as you need to add features
- When you have numerous products you really need to create something like a commercial load testing tool

Mixed Approach

- Programming using load testing tools
 - Program a script instead of recording
 - Run it using a load testing tool
- Implementation depends on the particular tool
 - Scripting support
 - Languages support
 - API calls (may require external dll / client stubs)

Advantages

- Eliminates dependency on supporting specific protocols
- Leverages all the features of the load testing tool and allows using it as a test harness
- Sometimes simplifies work with difficult to parameterize protocols

Considerations

- Requires programming / access to APIs
- May require more resources on agents
- Environment may need to be set on all agents
- Results should be cautiously interpreted
- May require a load test tool license

Mixed Approach: Example

- Middleware, no GUI interface
- Functional test scripts in Java
- Solution: creation of load testing scripts from the functional test scripts

Example Script

```
import lrapi.lr;
...
public int action() {
    lr.start_transaction("o1_Create_API_instance");
    ess = IEssbase.Home.create(IEssbase.JAPI_VERSION);
    lr.end_transaction("o1_Create_API_instance", lr.AUTO);
    lr.start_transaction("o2_SignOn");
    IEssDomain dom = ess.signOn(s_userName, s_password,
    s_domainName, s_prefEesSvrName, s_orbType, s_port);
    lr.end_transaction("o2_SignOn", lr.AUTO);
    ...
}
```

Agenda

- Load testing and its role in performance risk mitigation
- Different approaches: load generation
- *Different approaches: environments*
- Load testing tools

Environments

- Lab vs. Service (SaaS) vs. Cloud (IaaS)
 - For both the SUT and load testing tool
- Test vs. Production
- No best solution, depends on your goals / system

Scenarios

- System validation for high load
 - Outside load (service or cloud), production system
 - Wider scope, lower repeatability
- Performance optimization / troubleshooting
 - Isolated lab environment
 - Limited scope, high repeatability
- Testing in Cloud
 - Lowering costs (in case of periodic tests)
 - Limited scope, low repeatability

Agenda

- Load testing and its role in performance risk mitigation
- Different approaches: load generation
- Different approaches: environments
- *Load testing tools*

Load Testing Tools

- Differ drastically
 - Supported approaches / protocols
 - Scripting / extendibility
 - Supported environments
 - Scalability
 - Integration
 - Result analysis
 - Environment monitoring
 - Cost/Licensing
 - Available support and skills

Load Testing Tools

- There is no best tool – it depends on your needs
- Almost every tool works for a simple web site
- If using more sophisticated technologies, always check if the tool / approach supports it

Summary

- Load testing is an important way of performance risk mitigation
 - Other ways don't substitute load testing
- There are many ways to do load testing.
- There is no best approach or tool – it depends on your needs.

Any questions?

Alexander Podelko
alex.podelko@oracle.com
www.alexanderpodelko.com
@apodelko





ALMFORUMSEATTLE

April 1-3, 2014 • Washington State Convention Center

Thank you for joining us!