WebPerfDays

Load Testing: Use a Right Approach to Mitigate Performance Risks

Alexander Podelko
alex.podelko@oracle.com
alexanderpodelko.com/blog
@apodelko

June 28, 2012

About Me

- Specialize in performance last 15 years
- Currently performance testing and optimization of Hyperion products at Oracle
- Blog at http://alexanderpodelko.com/blog, have a collection of performance-related links at http://alexanderpodelko.com, on Twitter as @apodelko
- Board director at CMG http://cmg.org, a volunteer organization of performance and capacity planning professionals
  - Conference December 3-7, 2012 in Las Vegas

Agenda

- ***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
  - …

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
- Infrastructure evaluation
- Prototypes / POC
- Component / unit
- What/if
- Performance troubleshooting
- Performance optimization
- 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
- [Auto] Scalable Architecture
- Continuous Integration / Deployment
  - Ability to deploy and remove changes quickly

But all of them don't replace load testing

Load testing complements them an 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

Velocity

- Web Performance [Front End?]

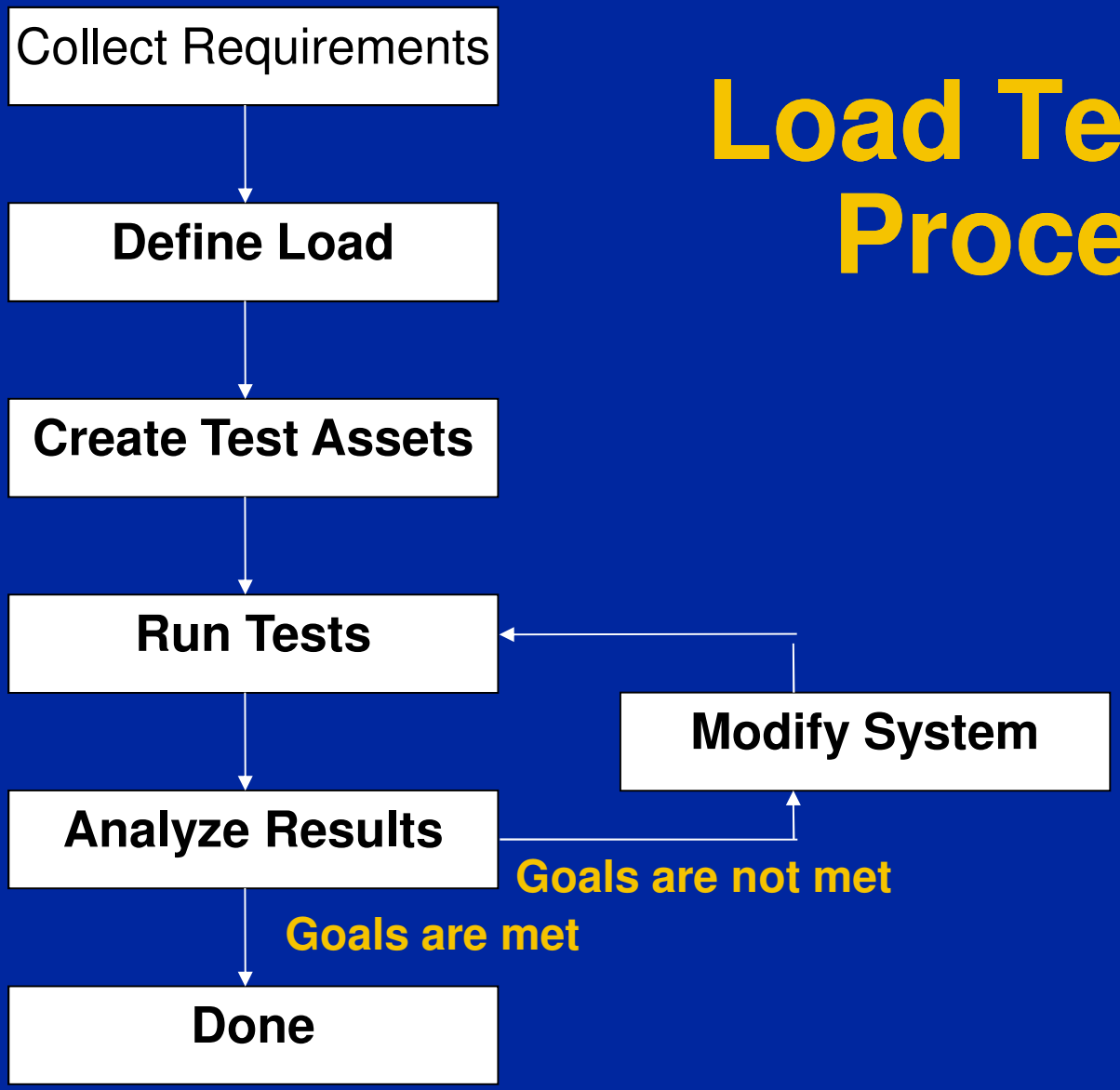
  Where we put, for example, scalable application design and load testing ?

- Operations

Agenda

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

Load Testing Process



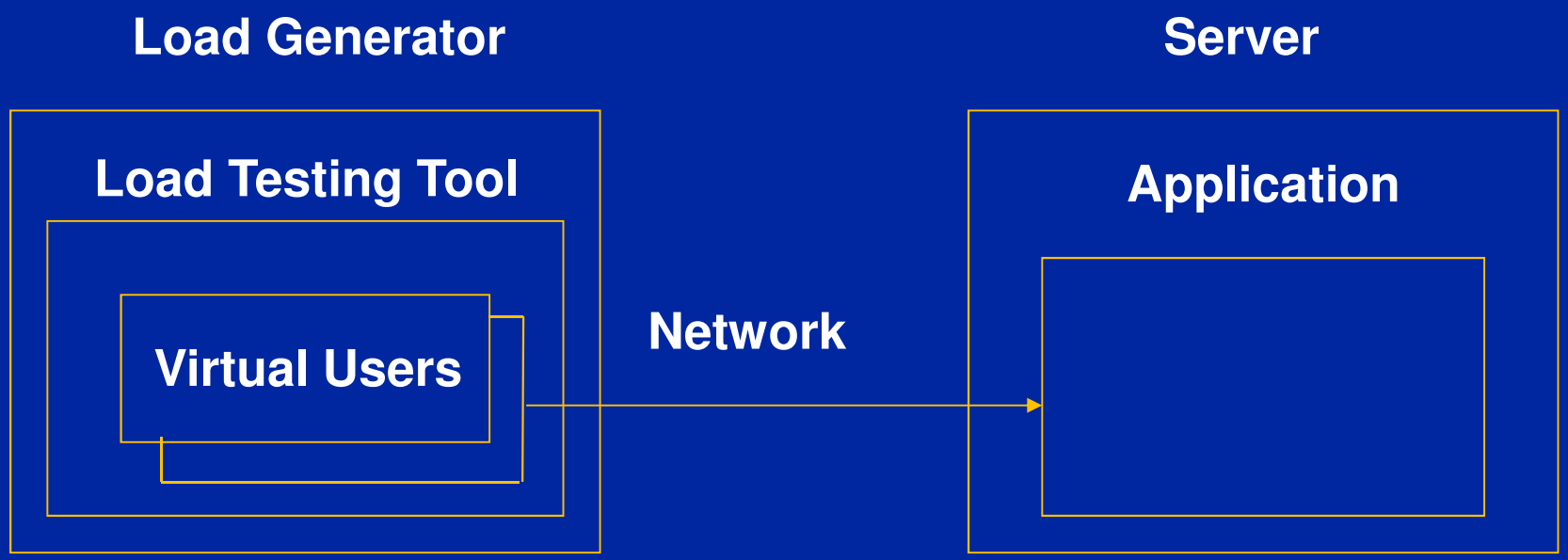
Load Generation

- Create tests assets – run test
- A must step in load testing
- ‘Tests 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 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/XMLDataGrid.asp?Action=EXECUTE&TaskID=1024
   &RowStart=1&ColStart=2&RowEnd=1&ColEnd=2&SelTy
   pe=0&Format=JavaScript", LAST);
web_custom_request("XMLDataGrid.asp_8","URL={URL}/
   Data/XMLDataGrid.asp?Action=GETCONSOLSTATUS",
   LAST);
web_custom_request("XMLDataGrid.asp_9","URL={URL}/
   Data/XMLDataGrid.asp?Action=GETCONSOLSTATUS",
   LAST);
web_custom_request("XMLDataGrid.asp_9","URL={URL}/
   Data/XMLDataGrid.asp?Action=GETCONSOLSTATUS",
   LAST);
```

Working Script

```
web_custom_request("XMLDataGrid.asp_7","URL={URL}/
   Data/XMLDataGrid.asp?Action=EXECUTE&TaskID=1024
   &RowStart=1&ColStart=2&RowEnd=1&ColEnd=2&SelTy
   pe=0&Format=JavaScript", LAST);
do {
   sleep(3000);
   web_reg_find("Text=1","SaveCount=abc_count",LAST);
   web_custom_request("XMLDataGrid.asp_8","URL={UR
   L}/Data/XMLDataGrid.asp?Action=GETCONSOLSTATU
   S",  LAST);
   } while (strcmp(lr_eval_string("{abc_count}"),"1")==0);
```

Alternatives

- Manual
- Record and Playback, GUI Users
- Programming
- Mixed (programming load testing tool)

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

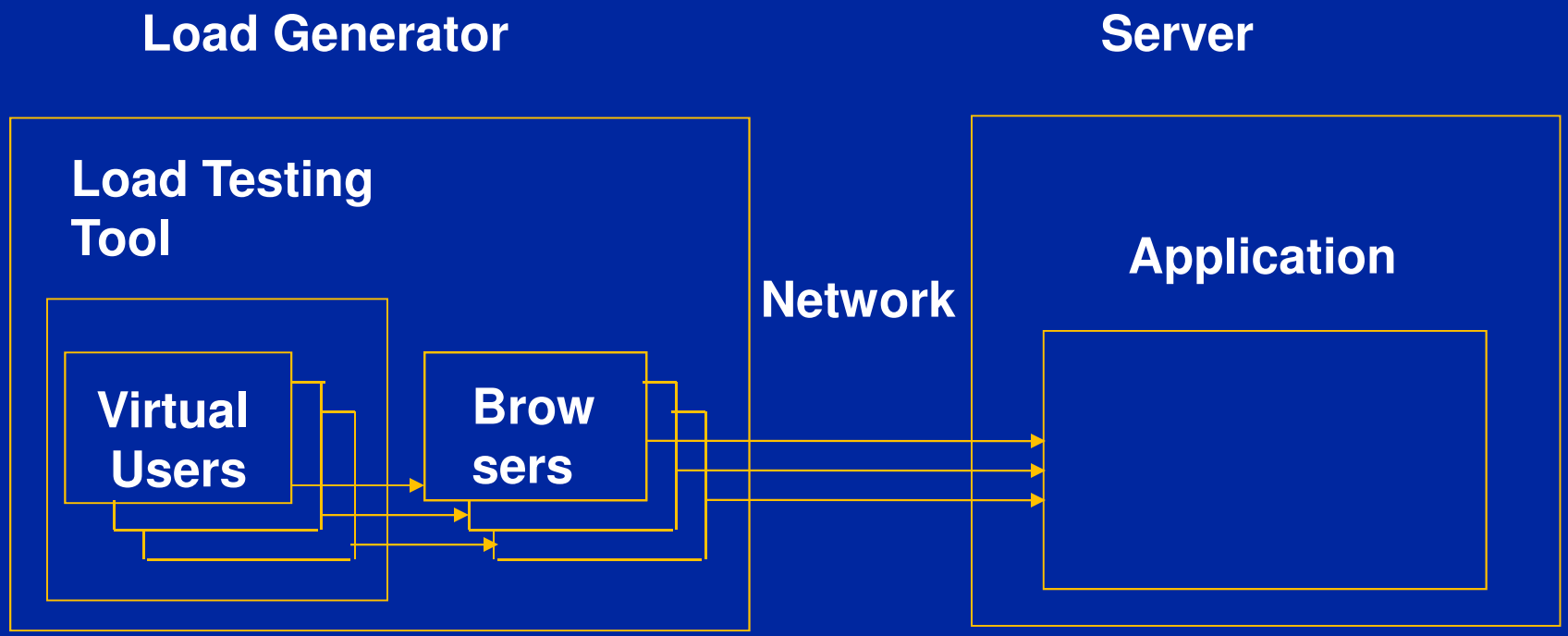
Record and Playback: UI Level

- 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 at least terminal session per user)
- Low-level graphical protocols
  - Citrix, Remote Desktop
- Web tools, browser
  - Require a separate browser instance
- Web tools, light-weight browser (HtmlUnit)
  - Require a separate light-weight browser instance

Record and Playback: UI Level



Challenges

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

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 can increase drastically
- When you have numerous products you really need to create something like a commercial load testing tool

Mixed Approach

- Programming load testing tool
  - Just write a script instead of recording
- Implementation depends on the particular tool
  - Scripting support
  - Supporting languages
  - 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 access to API or source code
- Requires programming
- Minimal transaction that could be measured is a request / API call
- Requires understanding of internals

More Considerations

- Requires a load test tool license for the necessary number of users
- Environment may need to be set on all agents
- May require more resources on agents
- Results should be cautiously interpreted

Agenda

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

Load Testing: Environment

- Lab vs. Service (SaaS) vs. Cloud (IaaS)
  - For both SUT and load testing tool
- Test vs. Production
- Think what you are doing performance testing for

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
- No best tool – depends on your needs

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.

Questions?

Alexander Podelko

alex.podelko@oracle.com
alexanderpodelko.com/blog
@apodelko