Performance & Capacity 2015 by CMG

Multiple Dimensions of Load Testing

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

November 2, 2015

Agenda

- *Load Testing*

- Five [New] Load Testing Dimensions
  - -Environment
  - -Load Generation
  - -Testing Approach
  - -Life-Cycle Integration
  - -Feedback and Analysis

2

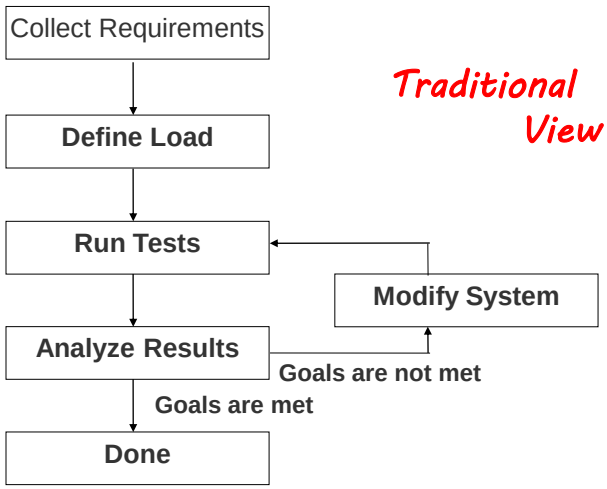
Load Testing

Applying multi-user synthetic load to the system

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

3

Load Testing Process



4

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
  - …

5

Load Testing

- Traditional load testing is not enough anymore
- New industry trends change a lot
  - Cloud
  - Continuous Integration / Delivery / Deployment
  - DevOps
  - Agile
- Some even say that load testing is not needed anymore
  - Due to other ways to mitigate performance risk

6

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

7

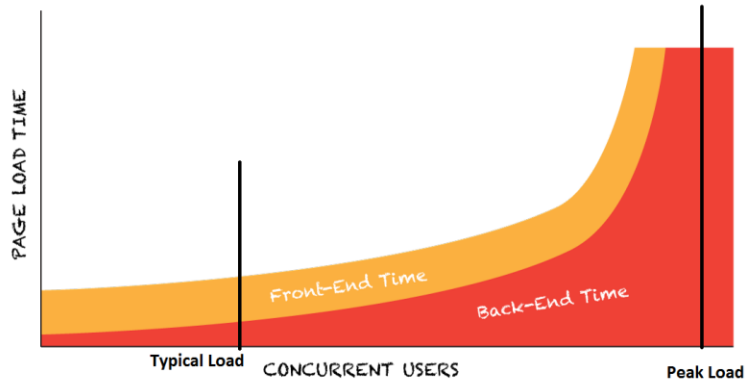
But all of them don't replace load testing:

Load testing complements them in several important ways !

8

Can System Handle Peak Load?

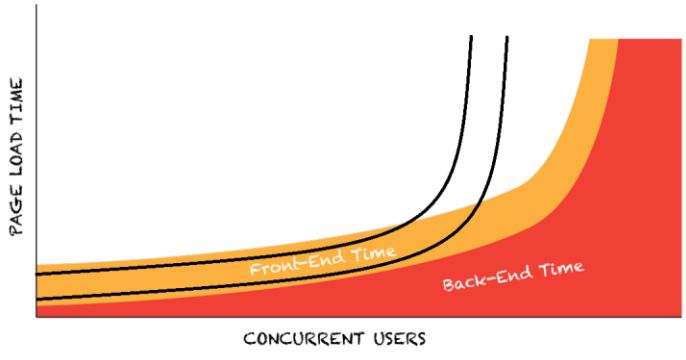
- You can't know without testing:



9

Verify Multi-User Performance

- Single-user improvement may lead to multi-user performance degradation



10

What Else Load Testing Adds

- Performance optimization
  - Apply exactly the same load
  - See if the change makes a difference
- Debugging/verification of multi-user issues
- Testing self-regulation functionality
  - Such as auto-scaling or changing the level of service depending on load

11

So What Is Going On?

- I believe that load testing is here to stay, but should fully embrace the change
  - Not one-time, to become dynamic
- Many things that were practically given became a hard choice of a continuum of options (dimension vs. point)
  - Environment, Load Generation, Testing Approach, Life-Cycle Integration, Feedback and Analysis

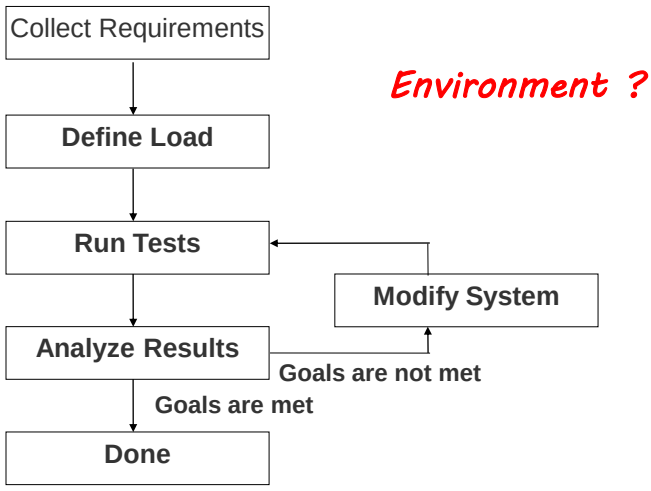
12

Agenda

- Load Testing
- ***Five [New] Load Testing Dimensions***
  - ***Environment***
  - Load Generation
  - Testing Approach
  - Life-Cycle Integration
  - Feedback and Analysis

13

Load Testing Process



Environment ?

14

Deployment

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

15

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

16

Find Your Way

- If performance risk is high it may be a combination of environments, e.g.
  - Outside tests against the production environment to test for max load
  - Lab for performance optimization / troubleshooting
  - Limited performance environments to be used as part of continuous integration

17

Scaling

- Becomes ***critical*** as you get to a large number of virtual users
- The number of supported users per unit of computing power may differ drastically
  - Depending on tool, protocol, scenario, system…
- If you need deploy it on a large number of machines automation would be helpful

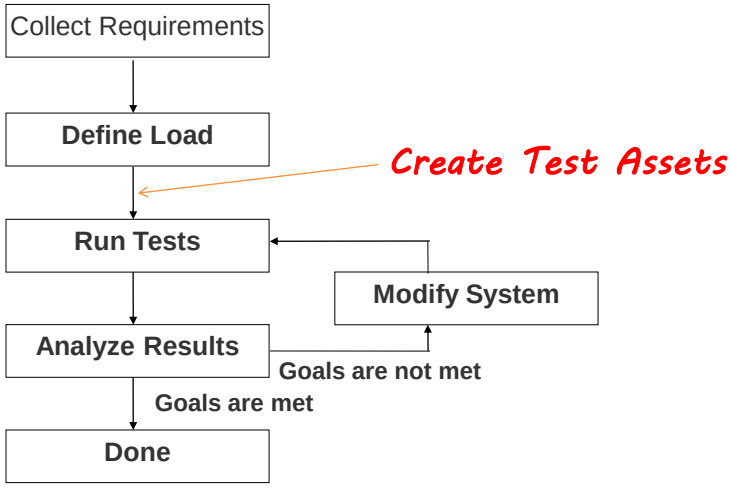
18

Agenda

- Load Testing
- ***Five [New] Load Testing Dimensions***
  - -Environment
  - ***-Load Generation***
  - -Testing Approach
  - -Life-Cycle Integration
  - -Feedback and Analysis

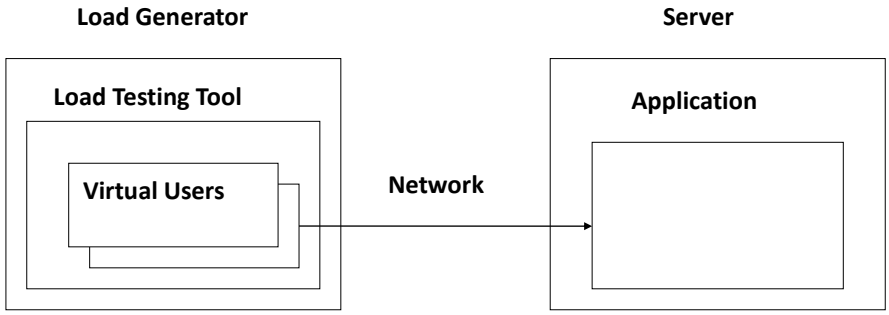
19

Load Testing Process



20

Record and Playback: Protocol Level



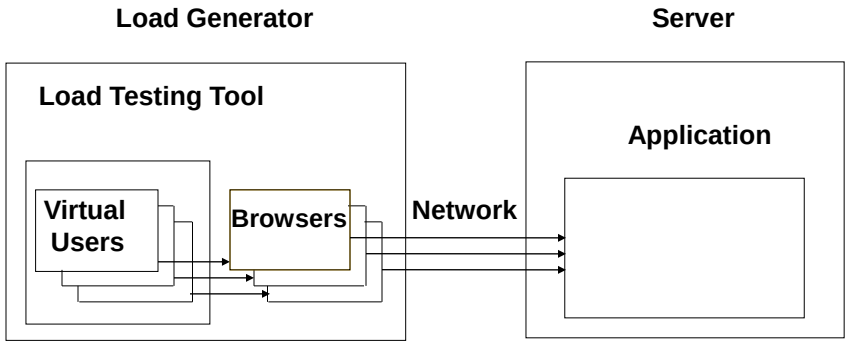
21

Considerations

- 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

22

Record and Playback: UI Level



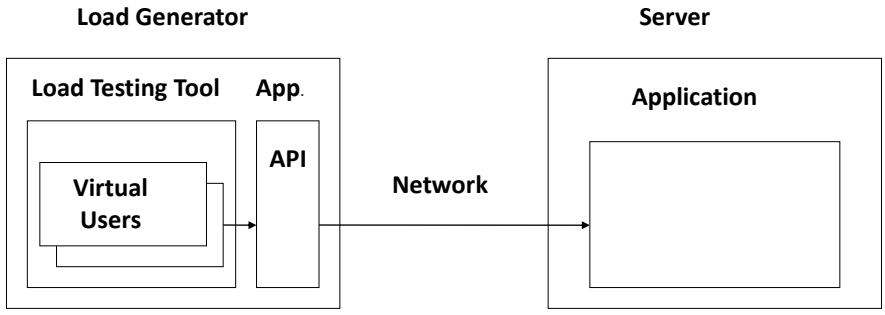
23

Considerations

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

24

Programming



25

Considerations

- Requires programming / access to APIs
- Tool support
  - Extensibility
  - Language support
- May require more resources
- Environment may need to be set

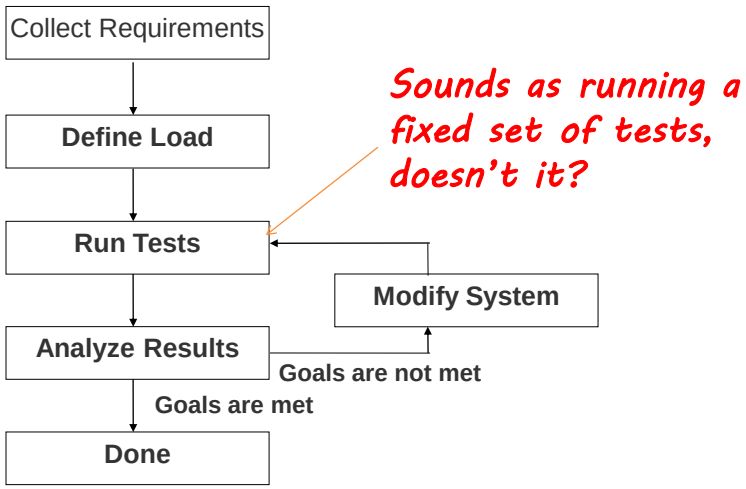
26

Agenda

- Load Testing
- ***Five [New] Load Testing Dimensions***
  - -Environment
  - -Load Generation
  - -***Testing Approach***
  - -Life-Cycle Integration
  - -Feedback and Analysis

27

Load Testing Process



28

Mentality Change

- Making performance everyone's job
- Late record/playback performance testing -> Early Performance Engineering
- System-level requirements -> Component-level requirements
- Record/playback approach -> Programming to generate load/create stubs
- "Black Box" -> "Grey Box"

Performance Testing

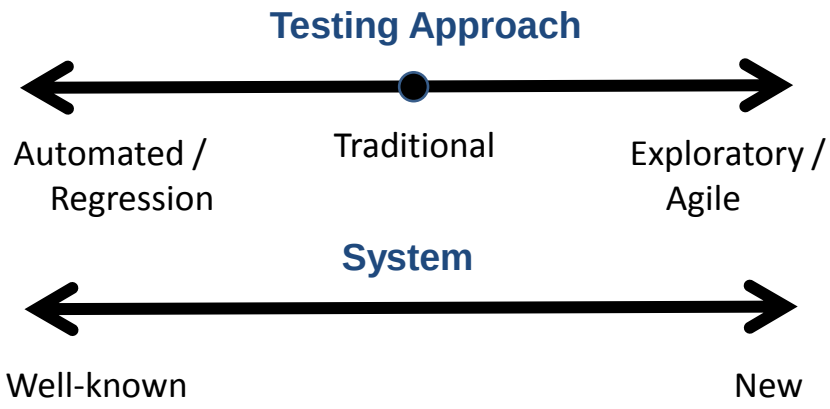
- Usually is not separated from:
  - Tuning
    - System should be properly tuned
  - Troubleshooting / Diagnostics
    - Problems should be diagnosed further to the point when it is clear how to handle them
  - Capacity Planning / Sizing
- "Pure" performance testing is rare
  - Regression testing ?

Exploratory Testing

- Rather alien for performance testing, but probably more relevant than for functional testing
- We learn about system's performance as we start to run test
  - Only guesses for new systems
- Rather a performance engineering process bringing the system to the proper state than just testing

31

Test Approach Dimension



Test Management

- Managing test execution / collecting results
- Virtual users coordination, e.g.:
  - Synchronization points
  - Data exchange
  - Sophisticated scheduling
- Environment simulation, e.g.:
  - Browser Simulation
  - Network simulation (including mobile)
  - IP spoofing

33

Agenda

- Load Testing

- ***Five [New] Load Testing Dimensions***

  -Environment

  -Load Generation

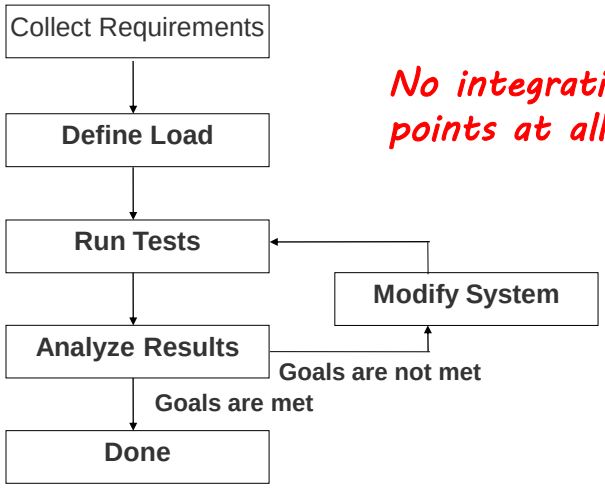
  -Testing Approach

  -Life-Cycle Integration

  -Feedback and Analysis

34

Load Testing Process



No integration points at all !

35

Agile Support

- Agile / CI support becoming the main theme
- Integration with Continuous Integration Servers
  - Jenkins, Hudson, etc.
  - Several tools announced integration recently
  - Making a part of automatic build process
- Automation support
- Easiness to use
- Support of newest technologies

36

Automation: Difficulties

- Complicated setups
- Long list of possible issues
- Complex results (no pass/fail)
- Not easy to compare two result sets
- Changing Interfaces
- Tests may be long

Automation: Considerations

- You need know system well enough to make meaningful automation
- If system is new, overheads are too high
  - So almost no automation in traditional environments
- If the same system is tested again and again
  - It makes sense to invest in setting up automation
- Automated interfaces should be stable enough
  - APIs are usually more stable on early stages

Tool Support

- Not much tool support was until recently
- Some vendors claimed that their load testing tool better fits agile processes
  - Often it meant that the tool is a little easier to use
- Was difficult to find what is available
  - Ability to automate: command line, API, data access
  - Ability to extend scripts
  - Supported technologies

Tool Support: Recent Developments

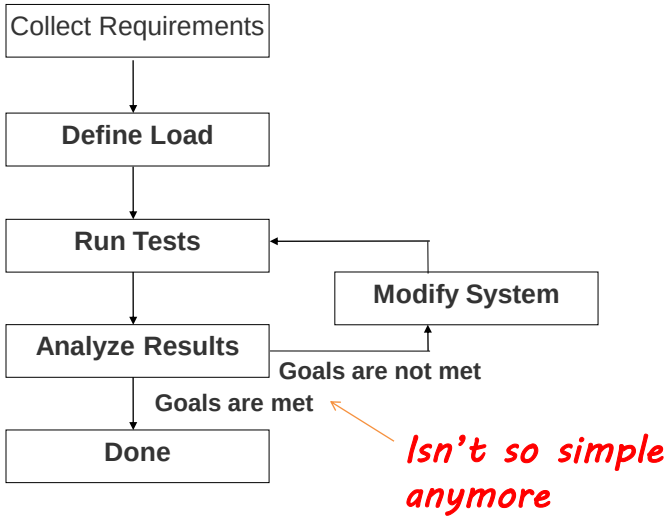
- Recently agile support became the main theme
  - A lot of new developments
- Integration with Continuous Integration Servers
  - Several tools announced integration recently
- Cloud integration
- Support of newest technologies

Agenda

- Load Testing
- ***Five [New] Load Testing Dimensions***
  - -Environment
  - -Load Generation
  - -Testing Approach
  - -Life-Cycle Integration
  - ***-Feedback and Analysis***

41

Load Testing Process



42

Reporting and Analysis

- Good integrated reporting and analysis greatly increases ***efficiency***
  - Getting all data in one place and synchronized
  - Integration of monitoring data is a great help
- Weak spot of many open source tools

43

Monitoring

- System level
- Application level (APM)
  - AppDynamics, New Relic, Dynatrace, etc.
  - Many tools have integration
- Integration allows analyze monitoring data *together* with test results

44

The Main Change in Monitoring

- Configuration becomes dynamic, changing on the fly
  - Auto scaling, auto provisioning, etc.
  - Challenge to monitor all moving parts
  - Challenge to compare results of dynamic configurations
  - Shift to application monitoring

45

The Main Change in Analysis

- Not only comparison with the requirements
- Many different forms of analysis depending on the tests
  - Adjusting to configuration / type of the test
    - Component testing
  - Automatic analysis / alerting
    - Continuous Integration / Delivery / Deployment
  - Input for tuning / optimization / sizing

46

Summary

- The industry is rapidly changing – performance testing should change too
  - Fully embrace agile
- Five [new] dimensions introduced by the changes
  - Environment, Load Generation, Testing Approach, Life-Cycle Integration, Feedback and Analysis
- Good tools help, but there is no best tool – it depends on your needs

47

Questions?

Alexander Podelko

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