Stress Your Web App Before It Stresses You: Tools and Techniques for Extreme Web Testing

Vinicius Senger

Architect and Instructor
Globalcode
http://www.globalcode.com.br

Felipe Leme

Architect
Matera Systems
http://matera.com

TS-9235

Quote

*"The designer is concerned with what happens when **1 user** presses a button and the architect is concerned with what happens when **10,000 users** press a button."*

Sun Certified Enterprise Architect for J2EE Technology Study Guide. Page 6.
Mark Cade, Simon Roberts.

java.sun.com/javaone

Reality Check

- Architect designs for 10,000 users
- Developer programs for 1 user
- Murphy crashes it on 100 users

- Stress testing to the rescue!

Goal of This Talk

Understand different methods, techniques and tools to plan, model, and execute stress tests.

Agenda

Introduction

Fundamentals of Stress Testing

Tools

Tips

Conclusion

Q&A

Agenda

Introduction
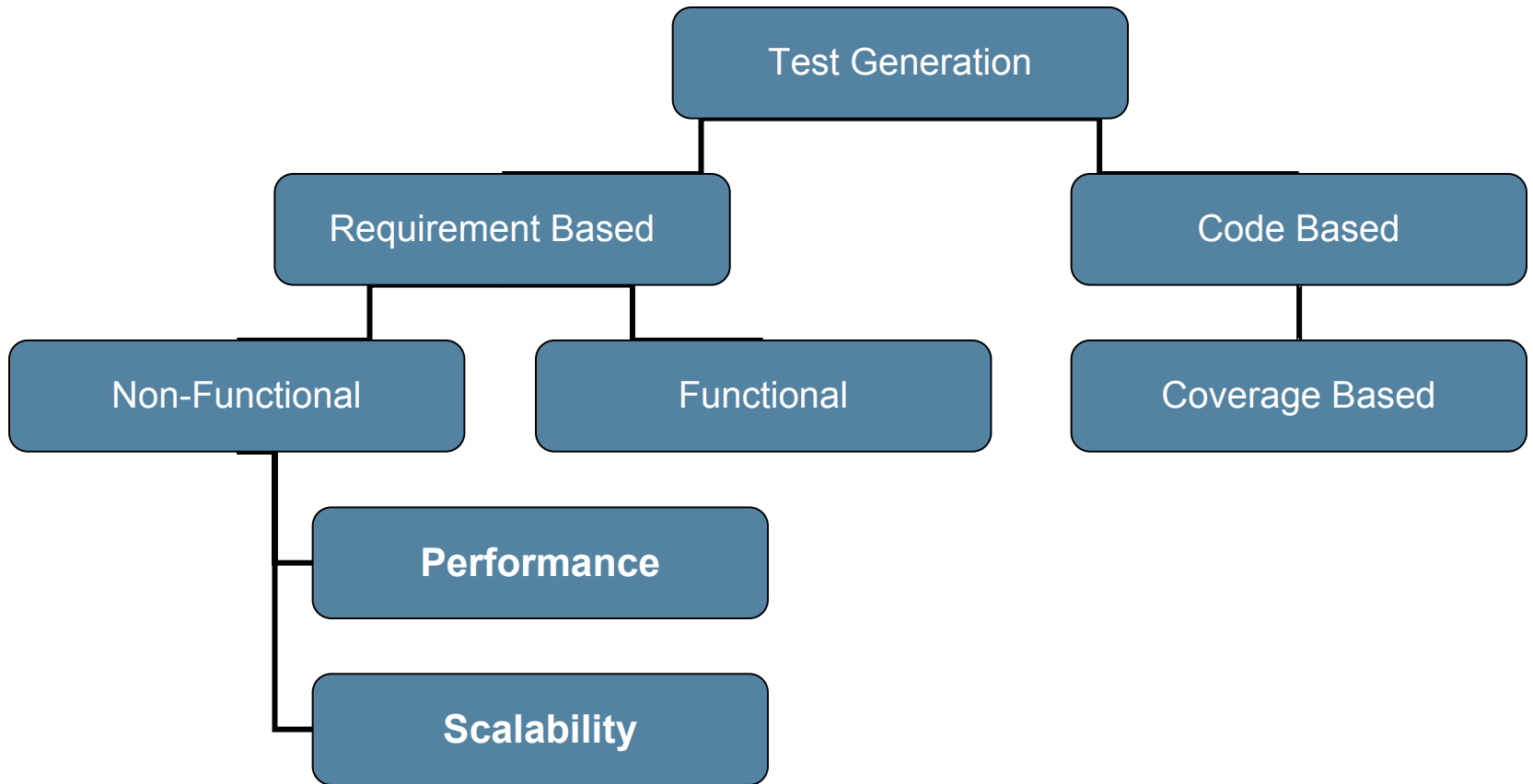Fundamentals of Stress Testing
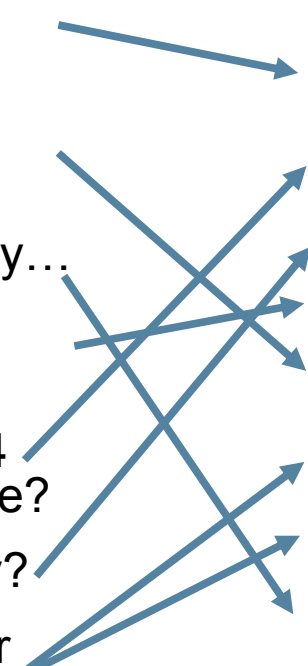Tools
Tips
Conclusion
Q&A

Testing Ecosystem
Different tests, different objectives

```
                          ┌─────────────────┐
                          │ Test Generation │
                          └─────────────────┘
                     ┌───────────┴───────────┐
          ┌──────────────────┐        ┌──────────────┐
          │ Requirement Based│        │  Code Based  │
          └──────────────────┘        └──────────────┘
          ┌────────┴────────┐                 │
  ┌──────────────┐   ┌──────────────┐  ┌──────────────┐
  │ Non-Functional│   │  Functional  │  │Coverage Based│
  └──────────────┘   └──────────────┘  └──────────────┘
        │
        ├──┌──────────────┐
        │  │ Performance  │
        │  └──────────────┘
        └──┌──────────────┐
           │  Scalability │
           └──────────────┘
```

java.sun.com/javaone

Non-Functional Requirements

Different needs, different tests…

- Can my application handle 5K simultaneous users?

- Which architecture should we use?

- My server crashes everyday…

- Will our server survive the next 2 years?

- Will application work after 4 months of continuous usage?

- What server should we buy?

- We need to afford 20K user

- Simultaneous users

- Continuous usage

- Hardware sizing

- Capacity planning

- Architecture decisions

- Stress-test driven development

- Continuous performance

- Find a bottleneck/diagnose performance problem

What Is the X of the Question?

Demand Simulation
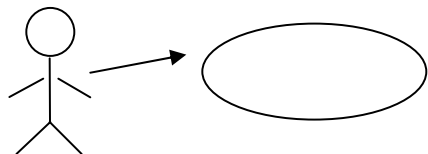
X is the number of users…

- Can my application handle 5K simultaneous users?

- With a given test configuration (data center environment) and Java™ Platform, Enterprise Edition (Java™ EE platform) Web App, how many users can it afford?
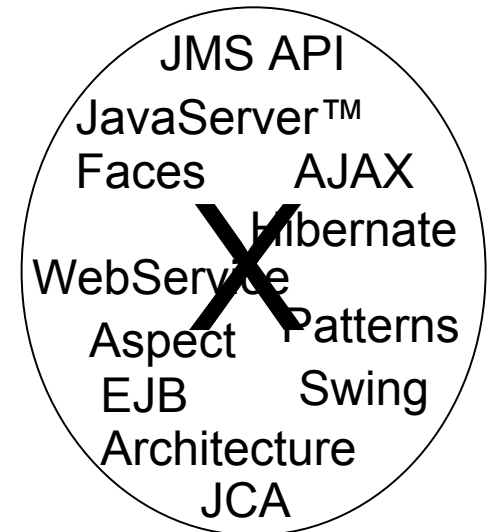
$$app.war \quad + \quad \text{[server]} \quad = \quad \text{X [users]}$$

Architecture Decision

X is the technical architecture…

- Given functional and non-functional requirements, which architecture should we use?

- Should we cluster Web + Enterprise JavaBeans™ (EJB™) technology container together?

+ 5K users =

JMS API
JavaServer™
Faces   AJAX
                    Hibernate
WebService
Aspect   Patterns
EJB   Swing
Architecture
        JCA

JMS = Java Message Service
JCA = J2EE™ Connector Architecture

Hardware Sizing and Capacity Planning

X is the hardware (and network)…

- Given the application and non-functional requirements, how much hardware do we need?

app.war + 5K users = X
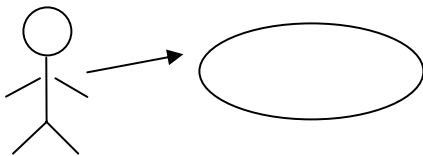
Diagnose Performance Problems

X is the problem

- When a big problem happens and you see someone smiling, probably they already found a someone to blame!

$$5K \text{ users } + \text{ app.war } =$$

Java™ DataBase Connectivity (JDBC™)

EJB Specification Pool?

JDBC software Pool?

Dead-lock?   Memory?

O.S.?   AJAX?

SQL code? Framework? Network?

Murphy's Law?   Garbage collector?

Agenda

Introduction

Fundamentals of Stress Testing

Tools

Tips

Conclusion

Q&A

java.sun.com/javaone

Definition

Functional vs. non-functional requirement testing

- Functional testing is concerned, most of the time, on single-thread and single-user testing

- Stress-testing is concerned on multi-thread /multi-user testing

- Repetition, concurrency, magnitude, and random variation are parts of a stress-test

Stress-Testing Phases

Main activities

- Planning

- Modelling and designing

- Construction and instrumentation

- Execution

- Data analyses

java.sun.com/javaone

Planning a Stress-Test

- Stress-test can have different scenarios:
  - Credit card companies have different demands during the year
  - E-commerce demand changes during promotions
  - Financial market is susceptible to external factors (wars, elections, rumors)

- Different scenarios, different load factor

- The more you plan, the more you cover!

Modelling a Stress-Test
Defining operational profiles

Operation	User profile/role			
	Standard (20%)	Investor (20%)	Manager (10%)	Corporate (50%)
Withdraw	20%	5%		25%
Deposit	15%	15%		25%
Transfer	15%	15%		30%
Account report	50%	20%	20%	20%
Create an account			20%	
Inactive an account			30%	
Make investments		45%	30%	

java.sun.com/javaone

Modelling a Stress-Test

Refining input space

- How do we better represent all possible user inputs?

- Login can be a string:
  - Minimum size is 4
  - Maximum size is 10
  - Only numbers and letters allowed

- Test input space for username:
  - Ana, A1
  - Ana1, James, JonhLennon, JonhLennon12

Construction

Creating test scripts based on operational profiles

Operation	User profile / role			
	Standard (20%)	Investor (20%)	Manager (10%)	Corporate (50%)
Withdraw	20%	5%		25%
Deposit	15%	15%		
Transfer	15%	15%		
Account report	50%	20%	20%	
Create an account			20%	
Inactive an account			3	
Make investments		45%	3	

Para capture.jmx (C:\Familia Senger\Globalcode\JavaOne2007\Para c

File   Edit   Run   Options   Help

- Test Plan
  - Standard
    - Withdraw
    - Deposit
    - Report
    - Transfer
  - Investor
  - Corporate
  - Manager
- WorkBench

Throughput Controller

Name: Withdraw

Percent Executions ▼

Throughput 20.0

☑ Per User

Agenda

Introduction

Fundamentals of Stress Testing

Tools

Tips

Conclusion

Q&A

Apache JMeter

- "*De facto*" tool for stress testing
    - Wide adoption
    - Open source (ASL 2.0)
    - 100% Java technology Desktop Application
    - Very mature (current release: 2.2)

- Originally intended for testing web (HTTP requests)

- Flexible design: Supports JMS technology, POP3, TCP, JUnit, SOAP, FTP, LDAP, JDBC software, and many other requests

java.sun.com/javaone

Essential Elements

- Test plan: JMeter "project"
- Workbench: temporary items
- Thread group: represents demand
- Samplers: protocol-specific requests
- Listeners: handle results
- Minimum usage:
  - Thread Group (X simultaneous users)
  - A request
  - One or more listener

DEMO

Stressed Hello World

Other Elements

- Configuration elements: global configurations
- Processors: transform request/response data
- Assertions: test validity of responses
- Timers: modify the tempo between requests
- Logic controllers: groups elements according to logical conditions
- HTTP Proxy Server: add elements by mimicking real usage

java.sun.com/javaone

DEMO

JavaServer Faces Study Case

Complementary Tools

When JMeter is not enough…

- ## OS monitoring tools
  - vmstat, iostat, dtrace

- ## Java Virtual Machine (JVM™) monitoring
  - Java Management Extensions (JMX™) technology console

- ## Profiler Integration

- ## Build integration
  - Apache Ant tasks
  - JChav

The terms "Java Virtual Machine" and "JVM" mean a Virtual Machine for the Java™ platform.

java.sun.com/javaone

Agenda

Introduction

Fundamentals of Stress Testing

Tools

Tips

Conclusion

Q&A

Planning Tips

- Avoid stress testing only most used scenario

- Special attention to user definition, when talking about simultaneous users stress-testing

- Usage of real input data space instead of synthetic data (whenever possible)

java.sun.com/javaone

Planning Tips
Stress-test environment

- Documentation is paramount:
  - OS version and patches
  - JVM software
  - Database/connection pool
  - Full hardware description
  - Techniques for monitoring
  - Input space
  - Definition of user profiles

java.sun.com/javaone

JMeter Tips

- Automatize whatever you can:
  - Variables
  - Regular expressions
  - Processors
  - HTTP Request Default

- Remember: HTTP is stateless
  - Cookies and Session ID
  - JavaServer Faces state
  - Solution: HTTP Cookie Manager

java.sun.com/javaone

JMeter Tips

- **Beware the interference rule:**
  - Instrumentation consumes resources
  - Test configuration must mirror real world
    - Servers, network, OS, load

- **JMeter proxy is your friend**
  - Powerful and handy tool
  - Ajax requests
  - Session-based workflow
  - JavaServer Faces requests

java.sun.com/javaone

Agenda

Introduction

Fundamentals of Stress Testing

Tools

Tips

Conclusion

Q&A

Summary

- **Stress testing is a must**
  - …Not a might or a should
- **It is not hard to implement**
  - …Plenty of free tools available
- **Plan earlier, test always**
  - …Do not wait until it is too late!

For More Information

- JChav: http://jchav.blogspot.com/

- 2005 BOF: http://globalcode.com.br/j1-2005.pdf

- TS-9646: Performance-Tune Your Ajax Application

Agenda

Introduction

Fundamentals of Stress Testing

Tools

Tips

Conclusion

Q&A

java.sun.com/javaone

Q&A

Vinicius Senger—vinicius@globalcode.com.br
Felipe Leme—felipeal@gmail.com

java.sun.com/javaone/sf

Stress Your Web App Before It Stresses You: Tools and Techniques for Extreme Web Testing

Vinicius Senger

Felipe Leme

Architect and Instructor
Globalcode
http://www.globalcode.com.br

Architect
Matera Systems
http://matera.com

TS-9235