Recent Issues in Software Testing:
Part B

W. Eric Wong
Department of Computer Science
The University of Texas at Dallas
ewong@utdallas.edu
http://www.utdallas.edu/~ewong
Speaker Biographical Sketch

• Professor & Director of International Outreach  
  Department of Computer Science  
  University of Texas at Dallas

• Vice President, IEEE Reliability Society

• Secretary, ACM SIGAPP (Special Interest Group on Applied Computing)

• Principal Investigator, NSF TUES (Transforming Undergraduate  
  Education in Science, Technology, Engineering and Mathematics) Project:  
  *Incorporating Software Testing into Multiple Computer Science and  
  Software Engineering Undergraduate Courses*

• Founder & Steering Committee co-Chair for the SERE conference  
  (*IEEE International Conference on Software Security and Reliability*)  
  (http://paris.utdallas.edu/sere12)
Outline

- Testing Web Applications
- Security Testing
- Cloud Testing
Testing Web Applications
Common Mistakes for Web Applications

Data Tier
- Configuration errors
- Application design issues
- Code defects
- Insufficient infrastructure

Data Access Tier
- Inconsistent geo performance
- Bad performance under load
- Blocking content delivery

Business Tier
- Incorrect or Insufficient functions
- Unsatisfied requirement

Presentation Tier
- Poorly performing JavaScript
- Browser/device incompatibility
- Page size too big
- Too many objects
- Low cache hit rate
**Objective**

- The objective is to detect and fix bugs in web applications before they are published

- Concerns
  - Security
  - Functionality
  - Accessibility
  - High traffic handling
  - etc.
Five Steps for Web Application Testing

- Prioritization
- Process and reporting
- Defect analysis and tracking
- Testing environment set-up
- Test execution
Checklist for Web Application Testing (1)

• Functionality Testing
  – Test *links* for correctness and make sure none of them are broken
  – Test *forms* against expected behavior
  – Test *cookies* against expected behavior
  – Test *HTML* and *CSS* (cascading style sheets) to ensure accessibility
  – Test *business workflows*

• Usability Testing
  – Test site *navigation*
  – Test site *content*
Checklist for Web Application Testing (2)

• Interface Testing
  – Application
  – Web server
  – Database server

• Database Testing
  – Correctness and response time of query execution
  – Data integrity during database update

• Compatibility Testing
  – Display across different browsers
  – Correct rendering of web elements
Checklist for Web Application Testing (3)

• Performance Testing
  – Behavior and response time under different loads
  – Break point beyond normal load
  – Recovery from a crash due to peak load
  – Data compression

• Security testing
  – Unauthorized access
  – File protection
  – Inactivity timeout

• Crowd Testing
  – Many users with different background
Load Runner for Load & Performance Testing (1)

- A tool from HP (formerly Mercury) for performance measurement and bottleneck detection
Load Runner for Load & Performance Testing (2)
Load Runner for Load & Performance Testing (3)
**IBM Rational Performance Tester**

- A tool from IBM/Rational for testing performance of web applications under different loads
**Link Tiger for Detection of Broken Links**

- Link Tiger helps detect broken links in web applications.
- Results can be reported through email alerts, dashboard display, and customized formats.
Browsera for Cross-Browser Testing

- Testing the display of web applications across operating systems in different browsers

![Same page in different browsers](image-url)
Security Testing
Six Fundamentals (1)

- Confidentiality
  - Managing information accessibility
  - Preventing information theft

- Integrity
  - Determining correctness of information received

- Authentication
  - Confirming user identity
  - Tracing origins of artifacts
  - Ensuring product consistency with packaging and labeling
  - Assuring trustworthiness of computer programs
Six Fundamentals (2)

• Authorization
  – Validating requesters’ permission to receive a service or perform an operation

• Availability
  – Assuring readiness of information and communication services
  – Maintaining information on-demand for authorized users

• Non-repudiation
  – Preventing subsequent denial of past actions
    □ Example: The sender of a message cannot deny having sent the message and the recipient cannot deny having received the message.
Importance of Security Testing

- Information and access security
  - Discover loopholes for possible information loss or intrusion into the systems

- System stability
  - Prevent system disruptions

- System integrity
  - Eliminate possible flaws in system design and implementation

- Economical efficiency
  - Cheaper to prevent possible problems rather than reactively resolving them and repairing their consequences
Techniques for Security Testing (1)

• Review
  – Examine systems, applications, networks, policies, and procedures for vulnerabilities
  – Generally conducted manually
  – Variations
    - Documentation review
    - Log review
    - Ruleset review
    - System configuration review
    - Network sniffing
    - File integrity checking
Target Identification and Analysis

- Identify potential vulnerabilities of systems, ports, services, etc.
- Generally performed using automated tools
- Variations
  - Network discovery
  - Network port and service identification
  - Vulnerability scanning
  - Wireless scanning
Techniques for Security Testing (3)

• Target Vulnerability Demonstration
  – Corroborate the existence of vulnerabilities
  – Performed manually or by using automatic tools
  – Variations
    - Password cracking
    - Penetration testing
Challenges for Security Testing

- Significantly larger test space
  - Security testing needs to consider an exponentially larger test space than functional testing

- Only able to partially automate the process

- Rarity of skilled testers with right competencies
  - Need to understand implementation details and also be capable of looking under the hood
Tools for Security Testing

- BackTrack provides a collection of more than 300 security-related tools for network discovery, scanning and sniffing, password cracking, remote access testing, penetration testing, etc.
  - http://livecdlist.com/backtrack

- Knoppix STD (Security Tools Distribution) includes tools for authentication, password cracking, encryption, intrusion detection, penetration, packet sniffers, vulnerability assessment, etc.
  - http://livecdlist.com/knoppix-std

Cloud Testing
**Cloud Computing (1)**

- Cloud computing is internet-based computing
- Shared resources and software provided on-demand with reduced management effort
- Example

![Apple iCloud Diagram](image)

- Take a picture
- Picture is saved (synchronization service)
- Same picture is shared by other devices
Cloud Computing (2)

SaaS (Software as a Service)
- Software delivered as a service over the Internet
- No application installed on the customer's side
- Simplified maintenance and support

PaaS (Platform as a Service)
- A computing platform or a solution stack as a service
- Consuming cloud infrastructure and sustaining cloud application
- Providers: Google, Microsoft, Apple, etc.

IaaS (Infrastructure as a Service)
- Computing infrastructure along with data storage and networking as a service
- Providers: IBM, Amazon, etc.
Cloud Testing & Its Benefits

• Cloud testing uses cloud infrastructure for software testing

• Benefits
  – Reduction in capital expenditure and tool license costs
    □ Pay-as-you-use basis
    □ No need to set up and maintain in-house infrastructures for testing
    □ No need to purchase expensive tools not fully used
  – Highly flexible, scalable and wide range
    □ Allow different platforms, scenarios, and geographic locations
    □ Choose the right tools
  – Minimum overhead on paperwork
    □ Contact selected cloud vendors in a timely manner to set up testing environments
Types of Testing Done in Cloud

- Stress
- Compatibility
- Load & Performance
- Browser Performance
- Functional
- Latency

Cloud Testing
Traditional Load Testing vs Cloud Load Testing

### Traditional Load Testing
- Not scalable
- Not flexible
- No way to reach production-like load maintenance
- Hardware is never fully utilized
- Unfit for distributed team
- You never have enough hardware

1) Decide load level
2) Choose an appropriate cloud provider based on the number of virtual clients, load levels, platforms, etc.
3) Test for the selected load level, platform, etc.
4) Change parameter values and repeat testing for different load levels, platforms, etc.

### Cloud Load Testing
- Infinite load
- From private clouds
- Scalable
- No hardware
- Easy maintenance
- Affordable
- Geo enabled
- Accessible

1) Decide load level
2) Purchase appropriate tools if necessary
3) Set up customized testing environments for each client
4) Test under pre-selected load level
5) Repeat 1) to 4) to establish different testing environments for different load levels and clients
Example of Testing in Cloud using SOASTA (1)

• Recording sequences of actions (e.g., clicking on a link followed by filling out a text field, etc.) for testing a web application
Example of Testing in Cloud using SOASTA (2)

- Creating customized testing environments
Example of Testing in Cloud using SOASTA (3)

- Creating customized testing environments (cont’d)
Example of Testing in Cloud using SOASTA (4)

- Creating customized testing environments (cont’d)
  - Selecting multiple *virtual monitors* each of a different set-up
Example of Testing in Cloud using SOASTA (5)

- Analysis and report generation (with respect to pre-recorded sequences of actions and customized testing environments)
**Challenges for Testing in Cloud**

- Not all applications are suitable for testing in cloud

- Certain characteristics must be present
  - Test cases independent of each other
    - Performance improved through concurrent test execution
  - A self-contained and easily identifiable operational environment
  - A programatically accessible interface for test automation

- Testing results may not be accurate due to varying performance of service providers’ network and Internet connection
  - Bandwidth limitation

- Not suitable for critical applications (e.g., complex software systems in defense, medicine, nuclear power generation, etc.)

- Difficult to manage large amount of complicated data sets for testing
Cloud Testing Provider - SOASTA

- The SOASTA CloudTest Platform offers a capability to execute functional and performance tests from in-house to production environments.
Cloud Testing Provider - PushToTest (1)

- Combine grid technology and cloud computing to run tests across multiple cloud-based testing environments
Cloud Testing Provider - PushToTest (2)

Report generation

Creation of different testing scenarios
**Cloud Testing Provider - CloudSleuth (1)**

- Comparison of performance and availability of different PaaS and IaaS providers

<table>
<thead>
<tr>
<th>Provider</th>
<th>Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google App Engine</td>
<td>7.67 sec</td>
</tr>
<tr>
<td>Windows Azure (US Central - Chicago)</td>
<td>8.20 sec</td>
</tr>
<tr>
<td>Cloud (US East - New York)</td>
<td>8.54 sec</td>
</tr>
<tr>
<td>Joyent (US West - San Francisco)</td>
<td>9.37 sec</td>
</tr>
<tr>
<td>BlueLock (US Central - Indiana)</td>
<td>9.60 sec</td>
</tr>
<tr>
<td>TekLinks (US South - Alabama)</td>
<td>9.61 sec</td>
</tr>
<tr>
<td>Codero (US South - Arizona)</td>
<td>9.71 sec</td>
</tr>
<tr>
<td>GoGrid (US East - Virginia)</td>
<td>9.78 sec</td>
</tr>
<tr>
<td>SoftLayer (US East - Virginia)</td>
<td>9.84 sec</td>
</tr>
<tr>
<td>GoGrid (US West - California)</td>
<td>9.91 sec</td>
</tr>
<tr>
<td>Iland Cloud (US South - Texas)</td>
<td>9.97 sec</td>
</tr>
<tr>
<td>eApps (US East - Georgia)</td>
<td>9.99 sec</td>
</tr>
<tr>
<td>SoftLayer (US South - Dallas)</td>
<td>10.80 sec</td>
</tr>
<tr>
<td>CloudSigma (US West - Las Vegas)</td>
<td>10.14 sec</td>
</tr>
<tr>
<td>Green House Data (US West - Wyoming)</td>
<td>10.15 sec</td>
</tr>
<tr>
<td>Tier3 (US West - Seattle)</td>
<td>10.20 sec</td>
</tr>
</tbody>
</table>

Image of a map and a table showing performance metrics for various cloud testing providers.
Cloud Testing Provider – CloudSleuth (2)

- Reporting performance of web applications based on user locations