The Case For System Testing with Swift Hierarchical VM Fork

Junji Zhi, Sahil Suneja and Eyal de Lara
Department of Computer Science
University of Toronto
Introduction

- Testing large systems is often difficult
- System configuration takes effort
- Executing tests consumes resources and time
System Testing

Test #1

Step #1
Step #2
Step #3

==

??

Test #1

Specification
System Testing

Test #2

Step #1

Step #2

Clean up

Step #3

Specification

==

?
Multiple test cases

Sequential, One machine

Test #1 → Test #2 → Test #3 → Test #4 → ....

Simple parallel, multiple VMs

VM #1 → VM #2 → VM #3 → VM #4 → ....
Observations

- Many commonalities or overlapping steps exist among tests
- Test cases share the same code base.

Simple parallel, multiple VMs

Common Steps are executed in each VM!
VM Fork

![Diagram showing VM Fork process]
Test Execution with VM Fork

- Reuse common steps
- Share memory and disk state
Case Study: Testing MySQL

- MySQL v5.5 “large_tests” suite

<table>
<thead>
<tr>
<th>TC#</th>
<th>Step 1</th>
<th>Step 2</th>
<th>Step 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Construct*</td>
<td>Delete</td>
<td>Insert</td>
</tr>
<tr>
<td>2</td>
<td>Construct</td>
<td>Delete</td>
<td>Add Column</td>
</tr>
<tr>
<td>3</td>
<td>Construct</td>
<td>Select</td>
<td>/</td>
</tr>
<tr>
<td>4</td>
<td>Construct</td>
<td>Update</td>
<td>/</td>
</tr>
</tbody>
</table>

*Construct: construct a new table and populate rows
MySQL Testing with VM Fork

Tester

MySQL

Config

VM

Fork

10

Fork

TC#1 done

TC#2 done

TC#3 done

TC#4 done

Time
Experiment Setup

● All experiments on server x86 12GB RAM
  ○ 64-bit Ubuntu 12.04 (Kernel v3.2.0)
  ○ Mysql v5.5
  ○ KVM v3.8
  ○ QEMU v1.0

● No current hierarchical fork implementation
  ○ Emulated VM fork with QEMU snapshot
  ○ Issues: slow
Results: Execution Time

Sequential & Simple Parallel
Total CPU cycles: 971s

With Emulated VM Fork
Total CPU cycles: 690s
Results: Space Usage

- 1 Snapshot < 5% Base VM
- 70% space saving
Challenges

- Fast hierarchical VM fork implementation
- When to fork
- Where to fork
- Transparency
Summary

- Leverage hierarchical VM fork to optimize system testing
- Advantages
  - Reuse common steps among TCs
  - Saving memory and disk space
- 30% reduction in VM runtime
- 70+% Space Savings
Thank you!
&
Questions?
Software Testing on the Cloud

- Cloud creates new opportunities

- Each VM encapsulates an entire test environment

- VM can run in parallel, isolated from each other
Case Study: Testing MySQL

- Structured automated test scripts

Listing 1: Sample SQL code for two MySQL tests
## Testing on the cloud

<table>
<thead>
<tr>
<th>Problem Domain</th>
<th>Test Level</th>
<th>Test Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud App.s</td>
<td>X</td>
<td>[10], [11], [12], [13], [14], [15], [16]</td>
</tr>
<tr>
<td>Web Services &amp; App.s</td>
<td>X</td>
<td>[19], [20]</td>
</tr>
<tr>
<td>Distributed &amp; Parallel App.s</td>
<td>X</td>
<td>[21], [22], [23], [24]</td>
</tr>
<tr>
<td>Cloud Service Dev. &amp; Deployment</td>
<td>X</td>
<td>[25], [26], [27]</td>
</tr>
<tr>
<td>Migration to Cloud</td>
<td>X</td>
<td>[38], [39], [40], [41], [6]</td>
</tr>
<tr>
<td>Cloud Infrastructure &amp; Storage</td>
<td>X</td>
<td>[42], [43], [44], [45]</td>
</tr>
<tr>
<td>Real-Time Systems</td>
<td>X</td>
<td>[46], [47], [48]</td>
</tr>
<tr>
<td>Network Config.</td>
<td>X</td>
<td>[51]</td>
</tr>
<tr>
<td>Test Task Mang.</td>
<td>[52]</td>
<td>[53], [7], [52]</td>
</tr>
</tbody>
</table>

Incki et al. “A Survey of Software Testing in the Cloud”, ICSSRC’12