Configuring Debugging as Search: Finding the Needle in the Haystack

Andrew Whitaker, Richard S. Cox and Steven D. Gribble.

University of Washington

Divya Muthukumaran

Some slides borrowed from Aditya Y.S.V
Whats the big picture?

- Can we automate some of the diagnostic tasks of the system administrator?
- This paper – *Partial automation of diagnosis!*
Configuration Debugging

• Configuration changes can cause system failure
  – Dynamic library upgrades
  – Installing an incompatible library
  – Windows Registry Modifications
  – Security policy change
• What caused the failure?
Configuration Debugging

• This work addresses the problem of diagnosing configuration errors that cause a system to function incorrectly.
• The basic idea is to search for the time when the system transitioned to a failed state.
• The paper presents a tool CHRONUS which automates this.
Motivation

- System experts are expensive!

Total ownership cost breakdown

- People costs
  - 1970’s
  - 2000’s

- Hardware costs
Existing Approaches

• **Prevention**: Complex systems, Difficult to anticipate side-effects of change
• **Recovery**: Windows XP restore. The problem with this is that it is a transition in itself and so it isn’t always safe.
• **Expert Systems**: “Static Database” of known error configurations. Correction from this can be automated.
  – Complex systems -> complex rule database
The Basic Approach

System failure

Chronus

When?

External analysis tools

Why?
System Overview

- Chronus reveals *when* a system failed

- Chronus pro-actively *logs* system states
## System Overview

<table>
<thead>
<tr>
<th>Design components</th>
<th>Design choices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Travel</td>
<td>Time travel disks, virtual machines</td>
</tr>
<tr>
<td>Testing</td>
<td>Software probes, copy-on-write disks</td>
</tr>
<tr>
<td>Search</td>
<td>Binary search</td>
</tr>
</tbody>
</table>
Time Travel

• Persistent vs. Transient state captures
• Chronus :- Only persistent storage.
  – Lacks Completeness
  – Less Overhead
• Some configuration changes need system restarts.
Virtual Machines

• The various states are checked by doing a virtual reboot of the system.
• **Virtual reboot** is faster than physical reboot
• Good way for terminating failed tests.
Disadvantages of VM

• Performance Overhead
• May not be able to expose the latest devices and device drivers
• Cannot diagnose errors within the virtualization layer itself such as updates to physical device driver.
Testing

- Automated diagnosis uses a user supplied "software probe"
- Written on the fly
- It has a manual method of software probe if all you remember is a series of GUI actions
Search

• Binary search
• Spurious Errors
  – Implicate a past upgrade
• Strategy to overcome spurious errors.
  – Run Chronus several times.
  – Different time ranges for each search
Binary Search

Transition

System was working

System was NOT working
Phase #1: Normal operation

- Child VM runs normal user programs
- Parent VM records disk writes to a time-travel disk
  - Each block write represents an instant in time
Phase #2: Debug Mode

User command: search $T_{\text{begin}}$ $T_{\text{end}}$

Was the system correct?

disk requests

Parent Virtual Machine

Child Virtual Machine

μDenali Virtual Machine Monitor
Testing

• Internal and external probes
• **Pre-processing** - wrap TTDisk with a Copy-on-write disk
• **Execute the probe** on boot
• **Halt** the child VM
• Mount the COW disk and do **post processing**
Implementation

- Command-line interface
- **Search** (TTDisk, Range log indices, probe)
- **Attach** - Mounts child system before and after state change
- **diff** - What precise change caused the failure?
Debugging experience - sshd

• Fault-injection: Random configuration errors
• sshd doesn't respond to remote login requests
• Probe: login via ssh and execute the UNIX date command
Binary search

Time

failure transition

system was working

system was NOT working

>>> search netbsd andrew.time
0000: SSHD UP  5267: SSHD DOWN  2633: SSHD UP
3950: SSHD UP  4608: SSHD UP  4937: SSHD DOWN
4772: SSHD UP  4854: SSHD UP  4895: SSHD UP
4916: SSHD UP  4926: SSHD DOWN  4921: SSHD DOWN
4918: SSHD UP  4919: SSHD UP  4920: SSHD DOWN
Debugging experience: sshd

Time transition

system was working

system was NOT working

>>> search netbsd andrew.time
0000: SSHD UP  5267: SSHD DOWN  2633: SSHD UP
3950: SSHD UP  4608: SSHD UP  4937: SSHD DOWN
4772: SSHD UP  4854: SSHD UP  4895: SSHD UP
4916: SSHD UP  4926: SSHD DOWN  4921: SSHD DOWN
4918: SSHD UP  4919: SSHD UP  4920: SSHD DOWN
Debugging Experience- sshd

- >>> attach andrew.time 4919 4920
- >>> diff -r /child1 /child2
  - Binary file /etc/ssh/ssh_host_key differs
Case Study: Mozilla Web Browser

- Mozilla Web Browser on the NetBSD OS
- Does Chronus apply to all errors?
  - No, 15 out of 24
  - 7-> scripts, 8 -> manual control (GUI)
- Methodology: install several extensions
- Symptom: Mozilla freezes on startup
  - Fails to respond to user input
Debugging the Mozilla Hang

- Step 1: write a probe that tests the behavior:

```bash
#!/bin/sh
mozilla &
sleep 5
mozilla -remote ping()
echo 'SUCCESS' > /TTOOUTPUT
```
Mozilla Hang .......

Step 2: invoke search over a time range:

% search -begin 169354 -end 180025

169354: SUCCESS 180025: FAILURE 174689: FAILURE
172021: SUCCESS 173355: SUCCESS 174022: FAILURE
173688: FAILURE 173521: SUCCESS 173604: FAILURE
173562: FAILURE 173541: SUCCESS 173551: SUCCESS
173556: FAILURE 173553: FAILURE 173552: SUCCESS
Mozilla Hang .......

• Step 3: compute the change:

  % attach time-travel-disk 173552 173553
  % diff -r /before /after

  file /.mozilla/default/zclirw5u.slt/chrome/chrome.rdf differs:

  <RDF:Description about="urn:mozilla:package:stockticker"
  ...
  c:author="Jeremy Gillick"
  c:authorURL="http://jgillick.nettripper.com/"
  c:description="Shows your favorite stocks in a customized ticker."
  c:displayName="StockTicker 0.4.2"
Performance

• Log Inflation:
  – File system meta-data operations
  – Deleting Mozilla directory (rm -rf) generates 1432 MB of log data

• Debug Execution Time:
  – Grows logarithmically
  – 20 seconds to conduct a single probe
Take away

- Can we automate system administrator tasks?
- Partially!
THANK YOU