

SystemTap Testing

William Cohen
Performance Tools Engineer
Red Hat Software, Inc.

Abstract

- For SystemTap to be used by various people on production machine there needs to be assurances that the software is relatively stable and safe to use. The software components and integrated system require testing to verify that things function as expected.

Goals and Motivation

- Ensure that SystemTap does not crash machines
- Show customers that it is tested and safe
- Find and quickly fix faults
- Provide guidance on instrumentation costs and limitations

Testing on Different Levels

- Components:
 - runtime
 - parser/translator
 - kprobes kernel support
- System:
 - Integrated components
 - No unexpected interactions between components

Testing Issues

- User space environment different than kernel
 - Limited kernel stack space
 - Copy from user
 - Some operations not available in user space, e.g. accessing performance monitoring hardware
- Failure could crash machine (or silently corrupt system)
- Need testing on various hardware (e.g. SMP) and various kernel configurations
- Non-determinism, e.g. 10 probe triggers one run and 12 probe triggers on another run (may avoid with careful test selection)



Runtime Test Plan

- Runtime library user space function unit testing:
 - Verify argument checking for each function
 - Verify in user space testing that all paths through code exercised
 - Check that memory leaks do not occur in code (run valgrind)
- Runtime library kernel space testing:
 - Test individual functions
 - Data transport code
 - Test failure modes, e.g. run out of memory

Parser/Translator Component Testing

- Make sure that invalid input identified
- Make sure that parser/translator generate valid constructs

Kprobe and Kernel Instrumentation

Component Testing

- Verify expected functionality of various instrumentation techniques
- Torture tests:
 - Maximum rate calling instrumented function
 - Repeated activation and deactivation of instrumentation
 - Attempts to trigger race conditions
 - Attempts to trigger recursive kprobes
- Identify non-instrumentable locations
- Test performance of instrumentation and provide some estimate of overheads

Red Hat Test Grid

- Internal system to reserve and to manage computers resources
- Collection of machines
- Remote KVM allows bios and boot up configuration (some machines)
- Enables power cycle of machines (same as KVMs)
- CVS repository for tests, e.g. gdb, Oprofile, and kernel torture (cerberus)