Record and vPlay
Debugging Container Crashes with “Partial Checkpoints”

Dinesh Subhraveti
CTO / Cofounder, appOrbit
Application Lifecycle Management

Key part of maintenance is rapid bug fixing
Problems with Problem Determination

1) **Report-Problem**
   - Users don’t know what is relevant
   - Overwhelm or mislead developer
   - Persistence data: Privacy concerns
Issue 86: Segment Violation

2 people starred this issue and may be notified of changes.

Owner: adrian.c...@gmail.com
Type: Defect
Priority: Critical
Version: Head

What steps will reproduce the problem?
Unsure

What is the expected output? What do you see instead?
FATAL: Received Segment Violation...dying.
2010/03/01 13:47:42 | storeDirWriteCleanLogs: Starting...
2010/03/01 13:47:42 | WARN: Closing open FD 9
2010/03/01 13:47:42 | commSetEvents: epoll_ctl(EPOLLC_CTL_DEL): failed on fd=9: (1) Operation not permitted
2010/03/01 13:47:42 | WARN: Closing open FD 27
2010/03/01 13:47:42 | commSetEvents: epoll_ctl(EPOLLC_CTL_DEL): failed on fd=27: (1) Operation not permitted
2010/03/01 13:47:42 | WARN: Closing open FD 29
2010/03/01 13:47:42 | commSetEvents: epoll_ctl(EPOLLC_CTL_DEL): failed on fd=29: (1) Operation not permitted
2010/03/01 13:47:42 | Took 0.0 seconds (266024.7 entries/sec).
CPU Usage: 354.986 seconds = 152.658 user + 202.329 sys
Maximum Resident Size: 0 KB
Page faults with physical i/o: 42
Memory usage for squid via mallinfo():

Comment 1 by project member adrian.c...@gmail.com, Oct 19, 2009

Ok. I really, really need to make sure I understand this issue here. If I had to take a stab at it, I'd say it has to do with the way I tidied up the rebuilding code and I've somehow stuffed up the 32/64 (normal, large file) support.

I've got a FreeBSD-7.2 i386 box here. Can you please give me your config file and "uname -a" output?

Thanks!

Status: Accepted
Owner: adrian.chadd
Labels: -Priority-Medium -Version-1.0 Priority-High Version-Head

Comment 2 by project member adrian.c...@gmail.com, Jan 19, 2010

I just can't reproduce this weakness on any of my FreeBSD boxes - and I've tried amd64/i386 for FreeBSD-7.2, I x, 8.0. The rebuild log seems to work. I can't figure out which combination of things doesn't work.

Are you able to still reproduce it? If so, may I please have access to some hardware which exhibits this?

Comment 3 by project member adrian.c...@gmail.com, Mar 25, 2010

Is this still a problem Chudy?
Problems with Problem Determination

1) **Report-Problem**
   - Users don’t know what is relevant
   - Overwhelm or mislead developer
   - Persistence data: Privacy concerns

2) **Reproduce-Problem**
   - Replicating environment: tedious and error-prone
   - Complex config
   - Nondeterminism: Repeated testing

Time spent on exchanges and setting up environment rather than fixing the problem
Problems with Problem Determination

1) Report-Problem
   - Users don’t know what is relevant
   - Overwhelm or mislead developer
   - Persistence data: Privacy concerns

2) Reproduce-Problem
   - Replicating environment: tedious and error-prone
   - Complex config
   - Nondeterminism: Repeated testing

Containers address the problem to some extent

Caveats
   * Container images are quite heavy
   * Not cross-platform
Complete and Concise Recording

Application consumes a variety of data as inputs while executing.

Inputs include:
- Data read from files, network sockets etc.
- Data returned by the OS via system calls
Complete and Concise Recording

Application consumes a variety of data as inputs while executing.

Inputs include:
- Data read from files, network sockets etc.
- Data returned by the OS via system calls
- Bits of application and library code accessed by the application
- Memory pages read by the application
Partial Checkpointing

Execute strcpy function in libc at page 0xb75f3000 gettimeofday

Error Propagation Distance
**Partial Checkpointing**

- **Checkpoint**: Complete intermediate state of a running application at one point of its execution
- **Partial checkpoint**: Partial state of itself that an application accesses in a specified interval
State Composition

- **Processor context:** At the beginning of the recording interval

- **System calls:** Results of system calls made by the application

- **Virtual Memory:** Memory pages accessed by the application

- **Nondeterministic events:** Meta data necessary for deterministic replay: interleaved shared memory accesses, signals

**No Kernel state**
Tracking Memory Pages

- Complications due to threads and changes in memory geometry
  - Processes and threads are created and deleted
  - Memory regions are added and removed: mappings change over time
  - Shared memory regions persist even without being attached to a process
Virtual Replay

- Consists of two steps: **Load phase** and **Replay phase**

- **Load Phase**: performed by a purpose-built binary loader
  1. Sets up the initial sparsely-populated application address space
  2. Recursively creates application threads
  3. Transfers control to the application code as per register context
Replay Phase

- Execute the instructions produced by the application
- Most instructions are executed natively
  - No privileged instructions
- Two types of instructions need emulation
  - Instructions referencing user-defined segment registers
    (fs, gs — modify_ldt())
  - Instructions that invoke a system call
    (int 0x80, sysenter)
Partial Checkpoint in a Debugger

No different than debugging with a standard binary
# Evaluation

<table>
<thead>
<tr>
<th>Application</th>
<th>Workload</th>
<th>Bug</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysql</td>
<td>httpperf (20,000 connection requests)</td>
<td>data race</td>
</tr>
<tr>
<td>apache</td>
<td>httpperf (20,000 connection requests)</td>
<td>library incompatibility</td>
</tr>
<tr>
<td>squid</td>
<td>httpperf (20,000 connection requests)</td>
<td>heap overflow</td>
</tr>
<tr>
<td>bc</td>
<td>Compute pi to 5000 decimal places</td>
<td>heap overflow</td>
</tr>
<tr>
<td>gzip</td>
<td>Compress 200 MB /dev/urandom data</td>
<td>global buffer overflow</td>
</tr>
<tr>
<td>ncomp</td>
<td>Compress 200 MB /dev/urandom data</td>
<td>stack smash</td>
</tr>
<tr>
<td>mplayer</td>
<td>Play 10 MB 1080p video file</td>
<td>device incompatibility</td>
</tr>
</tbody>
</table>

- **Record-Replay on Debian**: IBM HS-22 blade center
- **Replay on Gentoo**: VMware Fusion on MacBook Pro, different Linux kernel
- **Replay on Windows**: Lenovo T61p
Overhead below 3% for all except Squid (9%) and MySQL (17%)

Replay is faster than recording
Largest partial checkpoint was ~5 MB -- fraction of application memory footprint.
Grows with recording interval but not with application run time

Largest log size for a 5 s interval was 59 MB
dinesh@apporbit.com