What's the Fuss About Fastboot and New Kernel Crash Dumping Mechanism

Vivek Goyal
Senior Software Engineer
RedHat
Agenda

- Kernel Crash dumping (RHEL4 and RHEL5)
- What changed and why change
- Fastboot/Kexec
- Kdump
- Relocatable kernel
- How to configure and use kdump
- Dump filtering
- Driver test matrix
Kernel crash dumping in RHEL4
What changed in RHEL5

- Reliability
  - Don't trust a crashed kernel
- Flexibility
- Upstream solution
- Supported arch
  - x86, x86_64, ppc64, IA64
Kernel crash dumping in RHEL5
Kexec design

First Kernel

Load Second Kernel

Kexec -l

Second kernel pages

Kexec -e

First Kernel

Execute Second Kernel

Second Kernel

initrd

Setup page
How fast is kexec?

- Test Hardware: x86_64, 64 processor, 128 GB RAM
- Measured reboot time
- Reboot time reduced by 70% on test system

<table>
<thead>
<tr>
<th></th>
<th>Normal Boot</th>
<th>Kexec Boot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>7.5 minutes</td>
<td>2.2 minutes</td>
</tr>
</tbody>
</table>

Test Hardware: x86_64, 64 processor, 128 GB RAM
How to use Kexec

- `yum install kexec-tools`
- Load Kernel
  - `/sbin/kexec -l <kernel-to-load> --initrd=<initrd-to-load> --command-line=<command-line>`
- `reboot`
Kdump design

Reserved Memory for Capture Kernel
Load Capture Kernel
kexec -p
Regular Kernel
Elf Core Headers
initrd
Capture Kernel
Setup code
Crash
Boot into capture kernel
Capture Kernel
Setup code
Regular Kernel
Currently Running Kernel
Elf format dump file

• Kernel core exported through /proc/vmcore
  • Standard format
  • gdb can open the dump
• All memory chunks represented by PT_LOAD type headers
• All cpu states are captured by NT_PRSTATUS type Elf notes
• Standard tool can operate on /proc/vmcore to save it
  • cp, scp, dd etc.
Relocatable kernel

- Same kernel binary can run from different physical addresses
- Allows one to use regular kernel as capture kernel
- Currently i386, x86_64 and IA64 kernels are relocatable
- ppc64 uses a separate kernel binary as capture kernel
- x86
  - Retains relocation information
  - Performs relocation at run time
  - Kernel compile and run time virtual addresses are different
- x86_64
  - Kernel text region mappings are updated early
  - Kernel compile and run time virtual addresses are same
Enable kdump at installation

- Select “system-config-kdump” and “kexec-tools” packages
  - Optional package in “Base System”
- Enable kdump at first boot time
  - Specify amount of memory reserved for capture kernel
- Enable kdump service
  - `chkconfig kdump on`
  - Or use `system-config-kdump`
How to enable kdump later

- Install relevant packages
  - `yum install kexec-tools`
  - `yum install system-config-kdump`
- Reserve memory for capture kernel
  - Use `system-config-kdump`
- Reboot machine
- Enable kdump service
  - `chkconfig kdump on`
  - Or use `system-config-kdump`
Configuration: GUI interface

- **Enable kdump**: Checked
- **Total System Memory (MB)**: 1517
- **kdump Memory (MB)**: 128
- **Usable Memory (MB)**: 1389
- **Location**: `ext3:///dev/mapper/VolGroup00-LogVol00:/var/crash`
- **Default Action**: `mount rootfs and run /sbin/init`
- **Core Collector**: `makedumpfile -c`
- **Path**: `/var/crash`
What is configurable

• Amount of memory to reserve for crash kernel
• Dump Destination
  • Local file-system
  • NFS
  • SCP
  • Raw partition dump
• Default Action
  • Reboot; halt; shell; mount root and run init
• Dump filtering Options
  • makedumpfile
Behind the scenes

- /boot/grub/menu.lst
  - Modified for `crashkernel=X@Y` parameter
- /etc/kdump.conf
  - Modified for rest of the options
Advance configuration

- More configuration options in /etc/kdump.conf
  - extra_bins
    - Load extra bin/scripts into initrd
  - kdump_post
    - Specify if some binary/scripts need to be run after saving dump. Handle success/failure.
  - extra_modules
- /etc/sysconfig/kdump
  - Various command line, kernel version related option
  - No need to touch it normally
How much memory to reserve?

- Primarily depends on architecture
  - 128 MB for x86 and x86_64
  - 256 MB for ppc64
  - 256 MB (small servers) or 512MB (big servers) for IA64
How fast is dumping?

- Data for a test system
  - RHEL5.2, x86_64, 64 processor, 128 GB RAM

![Graph showing minutes vs. RAM size (128MB, 256MB, 512MB)]
Dump filtering

- makedumpfile is the dump filtering tool
- All filtering takes place in user space
- Output Format
  - ELF format
  - Kdump compressed format
- Allows compression of output pages
- Multiple dump filtering levels
Filtering levels

<table>
<thead>
<tr>
<th>Dump Level</th>
<th>Zero Page</th>
<th>Cache Page</th>
<th>Cache Private</th>
<th>User Data</th>
<th>Free Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>31</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
Filtering design

Flags:
- PG_swapcache
- PG_lru

Mapping:
- PG_MAPPING_ANON

- If set:
  - Swap Cache
  - Page Cache

- If not set:
  - Is set?
    - Yes: User Page
    - No: Scan pages for zeros
      - Zero Page
    - Scan free_list in zone
      - Free Page
How effective is filtering?

- Freshly booted system; mostly free pages
- 128 MB reserved for second kernel; Filtering level highest

<table>
<thead>
<tr>
<th></th>
<th>Unfiltered</th>
<th>Filtered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dump Size</td>
<td>Unfiltered</td>
<td>Filtered</td>
</tr>
<tr>
<td></td>
<td>128GB</td>
<td>234MB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Unfiltered</th>
<th>Filtered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time taken to save dump</td>
<td>Unfiltered</td>
<td>Filtered</td>
</tr>
<tr>
<td></td>
<td>39 Minutes</td>
<td>4 Minutes</td>
</tr>
</tbody>
</table>
How effective is filtering? Contd.

- Wrote a huge file with random numbers to fill page cache
- 128 MB reserved for second kernel; Filtering level highest

<table>
<thead>
<tr>
<th>Unfiltered</th>
<th>128GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filtered</td>
<td>1.08 GB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unfiltered</th>
<th>39 Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filtered</td>
<td>5 Minutes</td>
</tr>
</tbody>
</table>
Is this the perfect world

- Best effort is made to capture the dump
- Device driver initialization issues
  - Software reset capability
  - Reset device at initialization if in capture kernel
## Driver test matrix (storage)

<table>
<thead>
<tr>
<th>Driver/Controller</th>
<th>x86</th>
<th>X86_64</th>
<th>ppc64</th>
<th>IA64</th>
</tr>
</thead>
<tbody>
<tr>
<td>megaraid_sas</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>megaraid_mbox</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mptfusion</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mptspi</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mptsas</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sym53c8xx</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lpfc</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cciss</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>serveraid</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ipr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>adpxxxxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>aic79xx</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>aacraid</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>aic94xx</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stex</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qla1280</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Driver test matrix (networking)

<table>
<thead>
<tr>
<th>Driver/Controller</th>
<th>x86</th>
<th>X86_64</th>
<th>ppc64</th>
<th>IA64</th>
</tr>
</thead>
<tbody>
<tr>
<td>e100</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
</tr>
<tr>
<td>e1000</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
</tr>
<tr>
<td>e1000e</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
</tr>
<tr>
<td>tg3</td>
<td>🟢</td>
<td>🟢</td>
<td>🟢</td>
<td></td>
</tr>
<tr>
<td>q802.1/bonding</td>
<td>🟢</td>
<td>🟢</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bnx2</td>
<td>🟢</td>
<td>🟢</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Mailing lists

- Kexec, Kdump or makedumpfile issues
  - kexec@lists.infradead.org

- “Crash” Issues
  - crash-utility@redhat.com
Questions?