Capturing System Crash Information for Postmortem Analysis

BBLISA

July 13th, 2005

Presenter: Jeff Moyer
<jmoyer@redhat.com>

Overview

- Motivation
- Current solutions
- Netdump Details
- Crash Demo
- Future Directions
- Q & A

Motivation

- Minimize downtime
  - collect as much information as possible
- Provide administrators with enough information to file useful bug reports
- Provide developers with the exact state of the machine at the time of a crash

Motivation

- Minimize downtime
  - collect as much information as possible
- Provide administrators with enough information to file useful bug reports
- Provide developers with the exact state of the machine at the time of a crash

- Problem: Software bugs cause system down time.

Current Solutions

- kernel panic and oops messages
- kernel crash dumps
  - lkcd
  - diskdump
  - netdump
- kprobes and jprobes

What's in a panic message?

```
01 kernel BUG at fs/aio.c:1248!
02 invalid operand: 0000 [#1]
03 SMP
04 Modules linked in: nfs lockd scsi_dump diskdump md5 ipv6 parport_pc
05 lpparport autofs4 sunrpc dm_mod button battery ac uhci_hcd ehci_hcd
06 soundcore tg3 floppy ext3 jbd mptscsih mptbase sd_mod scsi_mod
07 CPU:    0
08 EIP:    0060:[<c017513a>]    Not tainted VLI
09 EFLAGS: 00010286   (2.6.9-1.849_EL.rootsmp)
10 EIP is at io_destroy+0xa6/0xb2
11 eax: ffffffff   ebx: ce52de80   ecx: 00000000   edx: ce52dea4
12 esi: d10c0980   edi: 00000000   ebp: ce52de80   esp: f1a26fa0
13 ds: 007b   es: 007b   ss: 0068
14 Process randasys (pid: 3098, threadinfo=f1a26000 task=f68005a0)
15 Stack: 15685d28 fffffff2 fffffff2 c01751cf 0000a4d5 25648bb6 5dc5839e
f1a26000
16        c02c5a07 0000a4d5 15685d28 70744532 25648bb6 5dc5839e 22079297
000000f5
17        c02c007b 0000007b 000000f5 0063c7a2 00000073 00000282 bfeca8d8
0000007b
```

What's in a panic message?

```
01 kernel BUG at fs/aio.c:1248!
02 invalid operand: 0000 [#1]
03 SMP
04 Modules linked in: nfs lockd scsi_dump diskdump md5 ipv6 parport_pc
05 lpparport autofs4 sunrpc dm_mod button battery ac uhci_hcd ehci_hcd
06 soundcore tg3 floppy ext3 jbd mptscsih mptbase sd_mod scsi_mod
07 CPU:    0
08 EIP:    0060:[<c017513a>]    Not tainted VLI
09 EFLAGS: 00010286   (2.6.9-1.849_EL.rootsmp)
10 EIP is at io_destroy+0xa6/0xb2
11 eax: ffffffff   ebx: ce52de80   ecx: 00000000   edx: ce52dea4
12 esi: d10c0980   edi: 00000000   ebp: ce52de80   esp: f1a26fa0
13 ds: 007b   es: 007b   ss: 0068
14 Process randasys (pid: 3098, threadinfo=f1a26000 task=f68005a0)
15 Stack: 15685d28 fffffff2 fffffff2 c01751cf 0000a4d5 25648bb6 5dc5839e
f1a26000
16        c02c5a07 0000a4d5 15685d28 70744532 25648bb6 5dc5839e 22079297
000000f5
17        c02c007b 0000007b 000000f5 0063c7a2 00000073 00000282 bfeca8d8
0000007b
```

What's in a panic message?

```
01 kernel BUG at fs/aio.c:1248!
02 invalid operand: 0000 [#1]
03 SMP
04 Modules linked in: nfs lockd scsi_dump diskdump md5 ipv6 parport_pc
05 lpparport autofs4 sunrpc dm_mod button battery ac uhci_hcd ehci_hcd
06 soundcore tg3 floppy ext3 jbd mptscsih mptbase sd_mod scsi_mod
07 CPU:    0
08 EIP:    0060:[<c017513a>]    Not tainted VLI
09 EFLAGS: 00010286   (2.6.9-1.849_EL.rootsmp)
10 EIP is at io_destroy+0xa6/0xb2
11 eax: ffffffff   ebx: ce52de80   ecx: 00000000   edx: ce52dea4
12 esi: d10c0980   edi: 00000000   ebp: ce52de80   esp: f1a26fa0
13 ds: 007b   es: 007b   ss: 0068
14 Process randasys (pid: 3098, threadinfo=f1a26000 task=f68005a0)
15 Stack: 15685d28 fffffff2 fffffff2 c01751cf 0000a4d5 25648bb6 5dc5839e
f1a26000
16        c02c5a07 0000a4d5 15685d28 70744532 25648bb6 5dc5839e 22079297
000000f5
17        c02c007b 0000007b 000000f5 0063c7a2 00000073 00000282 bfeca8d8
0000007b
```

What's in a panic message?

```
01 kernel BUG at fs/aio.c:1248!
02 invalid operand: 0000 [#1]
03 SMP
04 Modules linked in: nfs lockd scsi_dump diskdump md5 ipv6 parport_pc
05 lpparport autofs4 sunrpc dm_mod button battery ac uhci_hcd ehci_hcd
06 soundcore tg3 floppy ext3 jbd mptscsih mptbase sd_mod scsi_mod
07 CPU:    0
08 EIP:    0060:[<c017513a>]    Not tainted VLI
09 EFLAGS: 00010286   (2.6.9-1.849_EL.rootsmp)
10 EIP is at io_destroy+0xa6/0xb2
11 eax: ffffffff   ebx: ce52de80   ecx: 00000000   edx: ce52dea4
12 esi: d10c0980   edi: 00000000   ebp: ce52de80   esp: f1a26fa0
13 ds: 007b   es: 007b   ss: 0068
14 Process randasys (pid: 3098, threadinfo=f1a26000 task=f68005a0)
15 Stack: 15685d28 fffffff2 fffffff2 c01751cf 0000a4d5 25648bb6 5dc5839e
f1a26000
16        c02c5a07 0000a4d5 15685d28 70744532 25648bb6 5dc5839e 22079297
000000f5
17        c02c007b 0000007b 000000f5 0063c7a2 00000073 00000282 bfeca8d8
0000007b
```

What's in a panic message?

```
01 kernel BUG at fs/aio.c:1248!
02 invalid operand: 0000 [#1]
03 SMP
04 Modules linked in: nfs lockd scsi_dump diskdump md5 ipv6 parport_pc
05 lpparport autofs4 sunrpc dm_mod button battery ac uhci_hcd ehci_hcd
06 soundcore tg3 floppy ext3 jbd mptscsih mptbase sd_mod scsi_mod
07 CPU:    0
08 EIP:    0060:[<c017513a>]    Not tainted VLI
09 EFLAGS: 00010286   (2.6.9-1.849_EL.rootsmp)
10 EIP is at io_destroy+0xa6/0xb2
11 eax: ffffffff   ebx: ce52de80   ecx: 00000000   edx: ce52dea4
12 esi: d10c0980   edi: 00000000   ebp: ce52de80   esp: f1a26fa0
13 ds: 007b   es: 007b   ss: 0068
14 Process randasys (pid: 3098, threadinfo=f1a26000 task=f68005a0)
15 Stack: 15685d28 fffffff2 fffffff2 c01751cf 0000a4d5 25648bb6 5dc5839e
f1a26000
16        c02c5a07 0000a4d5 15685d28 70744532 25648bb6 5dc5839e 22079297
000000f5
17        c02c007b 0000007b 000000f5 0063c7a2 00000073 00000282 bfeca8d8
0000007b
```

What's in a panic message?

```
01 kernel BUG at fs/aio.c:1248!
02 invalid operand: 0000 [#1]
03 SMP
04 Modules linked in: nfs lockd scsi_dump diskdump md5 ipv6 parport_pc
05 lpparport autofs4 sunrpc dm_mod button battery ac uhci_hcd ehci_hcd
06 soundcore tg3 floppy ext3 jbd mptscsih mptbase sd_mod scsi_mod
07 CPU:    0
08 EIP:    0060:[<c017513a>]    Not tainted VLI
09 EFLAGS: 00010286   (2.6.9-1.849_EL.rootsmp)
10 EIP is at io_destroy+0xa6/0xb2
11 eax: ffffffff   ebx: ce52de80   ecx: 00000000   edx: ce52dea4
12 esi: d10c0980   edi: 00000000   ebp: ce52de80   esp: f1a26fa0
13 ds: 007b   es: 007b   ss: 0068
14 Process randasys (pid: 3098, threadinfo=f1a26000 task=f68005a0)
15 Stack: 15685d28 fffffff2 fffffff2 c01751cf 0000a4d5 25648bb6 5dc5839e
f1a26000
16        c02c5a07 0000a4d5 15685d28 70744532 25648bb6 5dc5839e 22079297
000000f5
17        c02c007b 0000007b 000000f5 0063c7a2 00000073 00000282 bfeca8d8
0000007b
```

What's in a panic message?

```
01 kernel BUG at fs/aio.c:1248!
02 invalid operand: 0000 [#1]
03 SMP
04 Modules linked in: nfs lockd scsi_dump diskdump md5 ipv6 parport_pc
05 lpparport autofs4 sunrpc dm_mod button battery ac uhci_hcd ehci_hcd
06 soundcore tg3 floppy ext3 jbd mptscsih mptbase sd_mod scsi_mod
07 CPU:    0
08 EIP:    0060:[<c017513a>]    Not tainted VLI
09 EFLAGS: 00010286   (2.6.9-1.849_EL.rootsmp)
10 EIP is at io_destroy+0xa6/0xb2
11 eax: ffffffff   ebx: ce52de
12 esi: d10c0980   edi: 000000
13 ds: 007b   es: 007b   ss: 0
14 Process randasys (pid: 3098
15 Stack: 15685d28 fffffff2 ff
f1a26000
16        c02c5a07 0000a4d5 15
000000f5
17        c02c007b 0000007b 00
0000007b
```

P – Proprietary module loaded

F – Module forcibly loaded

S – SMP with CPUs not designed for it

R – Forced module unload

M – MCE

B – System hit a bad page

What's in a panic message?

```
01 kernel BUG at fs/aio.c:1248!
02 invalid operand: 0000 [#1]
03 SMP
04 Modules linked in: nfs lockd scsi_dump diskdump md5 ipv6 parport_pc
05 lpparport autofs4 sunrpc dm_mod button battery ac uhci_hcd ehci_hcd
06 soundcore tg3 floppy ext3 jbd mptscsih mptbase sd_mod scsi_mod
07 CPU:    0
08 EIP:    0060:[<c017513a>]    Not tainted VLI
09 EFLAGS: 00010286   (2.6.9-1.849_EL.rootsmp)
10 EIP is at io_destroy+0xa6/0xb2
11 eax: ffffffff   ebx: ce52de80   ecx: 00000000   edx: ce52dea4
12 esi: d10c0980   edi: 00000000   ebp: ce52de80   esp: f1a26fa0
13 ds: 007b   es: 007b   ss: 0068
14 Process randasys (pid: 3098, threadinfo=f1a26000 task=f68005a0)
15 Stack: 15685d28 fffffff2 fffffff2 c01751cf 0000a4d5 25648bb6 5dc5839e
f1a26000
16        c02c5a07 0000a4d5 15685d28 70744532 25648bb6 5dc5839e 22079297
000000f5
17        c02c007b 0000007b 000000f5 0063c7a2 00000073 00000282 bfeca8d8
0000007b
```

What's in a panic message?

```
01 kernel BUG at fs/aio.c:1248!
02 invalid operand: 0000 [#1]
03 SMP
04 Modules linked in: nfs lockd scsi_dump diskdump md5 ipv6 parport_pc
05 lpparport autofs4 sunrpc dm_mod button battery ac uhci_hcd ehci_hcd
06 soundcore tg3 floppy ext3 jbd mptscsih mptbase sd_mod scsi_mod
07 CPU:    0
08 EIP:    0060:[<c017513a>]    Not tainted VLI
09 EFLAGS: 00010286   (2.6.9-1.849_EL.rootsmp)
10 EIP is at io_destroy+0xa6/0xb2
11 eax: ffffffff   ebx: ce52de80   ecx: 00000000   edx: ce52dea4
12 esi: d10c0980   edi: 00000000   ebp: ce52de80   esp: f1a26fa0
13 ds: 007b   es: 007b   ss: 0068
14 Process randasys (pid: 3098, threadinfo=f1a26000 task=f68005a0)
15 Stack: 15685d28 fffffff2 fffffff2 c01751cf 0000a4d5 25648bb6 5dc5839e
f1a26000
16        c02c5a07 0000a4d5 15685d28 70744532 25648bb6 5dc5839e 22079297
000000f5
17        c02c007b 0000007b 000000f5 0063c7a2 00000073 00000282 bfeca8d8
0000007b
```

What's in a panic message?

```
01 kernel BUG at fs/aio.c:1248!
02 invalid operand: 0000 [#1]
03 SMP
04 Modules linked in: nfs lockd scsi_dump diskdump md5 ipv6 parport_pc
05 lpparport autofs4 sunrpc dm_mod button battery ac uhci_hcd ehci_hcd
06 soundcore tg3 floppy ext3 jbd mptscsih mptbase sd_mod scsi_mod
07 CPU:    0
08 EIP:    0060:[<c017513a>]    Not tainted VLI
09 EFLAGS: 00010286   (2.6.9-1.849_EL.rootsmp)
10 EIP is at io_destroy+0xa6/0xb2
11 eax: ffffffff   ebx: ce52de80   ecx: 00000000   edx: ce52dea4
12 esi: d10c0980   edi: 00000000   ebp: ce52de80   esp: f1a26fa0
13 ds: 007b   es: 007b   ss: 0068
14 Process randasys (pid: 3098, threadinfo=f1a26000 task=f68005a0)
15 Stack: 15685d28 fffffff2 fffffff2 c01751cf 0000a4d5 25648bb6 5dc5839e
f1a26000
16        c02c5a07 0000a4d5 15685d28 70744532 25648bb6 5dc5839e 22079297
000000f5
17        c02c007b 0000007b 000000f5 0063c7a2 00000073 00000282 bfeca8d8
0000007b
```

What's in a panic message?

```
01 kernel BUG at fs/aio.c:1248!
02 invalid operand: 0000 [#1]
03 SMP
04 Modules linked in: nfs lockd scsi_dump diskdump md5 ipv6 parport_pc
05 lpparport autofs4 sunrpc dm_mod button battery ac uhci_hcd ehci_hcd
06 soundcore tg3 floppy ext3 jbd mptscsih mptbase sd_mod scsi_mod
07 CPU:    0
08 EIP:    0060:[<c017513a>]    Not tainted VLI
09 EFLAGS: 00010286   (2.6.9-1.849_EL.rootsmp)
10 EIP is at io_destroy+0xa6/0xb2
11 eax: ffffffff   ebx: ce52de80   ecx: 00000000   edx: ce52dea4
12 esi: d10c0980   edi: 00000000   ebp: ce52de80   esp: f1a26fa0
13 ds: 007b   es: 007b   ss: 0068
14 Process randasys (pid: 3098, threadinfo=f1a26000 task=f68005a0)
15 Stack: 15685d28 fffffff2 fffffff2 c01751cf 0000a4d5 25648bb6 5dc5839e
f1a26000
16        c02c5a07 0000a4d5 15685d28 70744532 25648bb6 5dc5839e 22079297
000000f5
17        c02c007b 0000007b 000000f5 0063c7a2 00000073 00000282 bfeca8d8
0000007b
```

What's in a panic message?

```
01 kernel BUG at fs/aio.c:1248!
02 invalid operand: 0000 [#1]
03 SMP
04 Modules linked in: nfs lockd scsi_dump diskdump md5 ipv6 parport_pc
05 lpparport autofs4 sunrpc dm_mod button battery ac uhci_hcd ehci_hcd
06 soundcore tg3 floppy ext3 jbd mptscsih mptbase sd_mod scsi_mod
07 CPU:    0
08 EIP:    0060:[<c017513a>]    Not tainted VLI
09 EFLAGS: 00010286   (2.6.9-1.849_EL.rootsmp)
10 EIP is at io_destroy+0xa6/0xb2
11 eax: ffffffff   ebx: ce52de80   ecx: 00000000   edx: ce52dea4
12 esi: d10c0980   edi: 00000000   ebp: ce52de80   esp: f1a26fa0
13 ds: 007b   es: 007b   ss: 0068
14 Process randasys (pid: 3098, threadinfo=f1a26000 task=f68005a0)
15 Stack: 15685d28 fffffff2 fffffff2 c01751cf 0000a4d5 25648bb6 5dc5839e
f1a26000
16        c02c5a07 0000a4d5 15685d28 70744532 25648bb6 5dc5839e 22079297
000000f5
17        c02c007b 0000007b 000000f5 0063c7a2 00000073 00000282 bfeca8d8
0000007b
```

What's in a panic message (cont'd)

```
18 Call Trace:
19  [<c01751cf>] sys_io_setup+0x89/0x90
20  [<c02c5a07>] syscall_call+0x7/0xb
21  [<c02c007b>] unix_dgram_recvmsg+0x21/0x1da
22 Code: a7 2d c0 89 d8 e8 98 ef ff ff 89 d8 e8 fe ef ff ff f0 ff 0b
...
```

What's in a panic message (cont'd)

```
18 Call Trace:
19  [<c01751cf>] sys_io_setup+0x89/0x90
20  [<c02c5a07>] syscall_call+0x7/0xb
21  [<c02c007b>] unix_dgram_recvmsg+0x21/0x1da
22 Code: a7 2d c0 89 d8 e8 98 ef ff ff 89 d8 e8 fe ef ff ff f0 ff 0b
...
```

Sample disassembly output:

```
c0175132 <sys_io_setup>:
c0175132:       55                      push   %ebp
c0175133:       57                      push   %edi
c0175134:       56                      push   %esi
c0175135:       53                      push   %ebx
c0175136:       8b 5c 24 18             mov    0x18(%esp),%ebx
c017513a:       8b 4c 24 14             mov    0x14(%esp),%ecx
c017513e:       89 d8                   mov    %ebx,%eax
c0175140:       e8 a7 20 04 00          call   c01b71ec <__get_user_4>
```

Panic/oops summary

- Provides details for the faulting process
- Enough information to debug the “easy” problems
  - recursive locking
  - single bit memory errors
  - etc.
- Deployment tips:
  - serial console
  - netconsole
  - frame buffer console for greater resolution

Crashdump Basics

- Copy contents of memory to permanent storage
  - local disk
  - over the network
  - Full dump / partial dump
- Hooks in the kernel
  - panic
  - oops
  - page flags (optional)
- Analysis tools

LKCD

- Linux Kernel Crash Dump
  - Started by SGI in or around 1999
  - Now supports a modified netdump protocol
  - Supports non-disruptive dumps
  - Supports compressed dump files
  - relies on the block io subsystem of the failed kernel

LKCD (cont'd)

- Crash analysis tools for LKCD
  - lcrash
    - requires System.map, Kern-types file, and a dump file
  - crash – will load LKCD dump files
    - requires a vmlinux with debug symbols, System.map, and a dump file
- Ships with Suse distributions
- Supported architectures:
  - x86, x86_64, ia64, ppc64, s390

Disk Dump

- Dump to either a dedicated dump partition or a swap partition (new).
- Requires hooks in every disk driver
- Currently supports:
  - mpt fusion, cciss, sym53c8xx, ipr (IBM Power series raid adapters), aic7xxx, SATA
- Ships in Red Hat enterprise distributions
- Dump analysis via crash (same dump format as netdump)
- Robust

Netdump

- **Net**work Crash**dump**
- Implemented using the netpoll infrastructure (2.6)
- Requires dedicated netdump server
- Loadable module
  - 2.4 has netconsole.o
  - 2.6 has netconsole.o and netdump.o

Netdump (cont'd)

- 3 bits of functionality
  - network crash dump
  - remote logging to a netdump server (netlog)
  - remote syslog
- 2.4
  - netdump and netlog cannot be configured independently
- 2.6
  - netdump, netlog, and syslog can be configured separately

Deployment Considerations

- Disk based dumps:
  - LKCD
    - Supports compression
    - Supports page selection
    - Uses existing bio subsystem
      - Supports all existing block devices
  - Diskdump
    - No compression; page selection forthcoming
    - Works with a limited subset of storage controllers
    - Uses a separate subsystem to perform disk I/O

Deployment Considerations (cont'd)

- Netdump
  - no compression or page selection
  - requires a single server with a large amount of storage
  - memory contents transmitted over the wire unencrypted
  - Supports most network adapters
    - requires a simple hook in each driver to implement polling
- Reliability?

kprobes and jprobes

- kprobes:
  - Hooks which can be placed at any *instruction* in the kernel
  - support pre, post, and exception handlers
- jprobes:
  - same as kprobes, but can be placed on function entrypoints and examine stack variables easily

kprobes & jprobes (cont'd)

- Useful for
  - Getting more debug output without rebuilding / rebooting
  - understanding the code
  - Patching the kernel on the fly!

Practicum

Netdump client and server setup

Netdump Setup (server)

- Server
  - rpm -i netdump-server-0.7.4-2.i386.rpm
  - /etc/netdump.conf
    - secure=[01]
  - Set the passwd for the netdump user
  - Optionally, copy scripts from

    `/usr/share/doc/netdump-n-v-r/example_scripts`

    to

    `/var/crash/scripts`
  - service netdump-server start

Netdump Setup (client)

- Client
  - rpm -i netdump-0.7.4-2.i386.rpm
  - modify /etc/sysconfig/netdump
  - service netdump propagate
  - service netdump start

/etc/sysconfig/netdump

```
#LOCALPORT=6666
#DEV=
#NETDUMPADDR=<Required>
#NETDUMPPORT=
#NETDUMPMACADDR=
#IDLETIMEOUT=

#SYSLOGADDR=
#SYSLOGPORT=
#SYSLOGMACADDR=

#NETLOGADDR=
#NETLOGPORT=
#NETLOGMACADDR=
```

Netdump (in)security

- ssh key shared between client and server
  - used for the distribution of a shared secret, generated upon netdump startup
  - Secret verification only happens one-way.
- UDP unicast used
  - for switched networks, this is generally O.K.

Netdump: supported platforms

- pre RHEL-3 U5
  - x86
- RHEL 3 U5 and beyond (including RHEL 4)
  - x86
  - x86_64
  - ia64
  - ppc64
- netdump server is platform independent.

Netdump: How it works

- Client/Server
  - Panic()ing system initiates the dump
    - handshake process
  - netdump server then requests pages from the panic()ed system
  - client breaks pages up into 1k chunks, due to the default Ethernet MTU of 1500 bytes.
  - At the end of the dump, a show_state is performed

Dump file format

- ELF core header
  - Can be read by gdb
- ELF header has a NT_TASKSTRUCT note
  - use to squirrel away a pointer to the panic()ing task
- After ELF header, raw dump of memory.

Testing your netdump setup

- You will want to enable the magic sysrq key:
  - `# sysctl -w kernel/sysrq=1`
- And panic_on_oops
  - `# sysctl -w kernel/panic_on_oops=1`
- `Check that netlog is working`
  - `# echo h > /proc/sysrq-trigger`
- On the server, you should see a new directory created:
  - `/var/crash/<IPAddr>`
- In that directory will be a file named 'log'
- You can crash the system with:
  - `# echo c > /proc/sysrq-trigger`
- Or by typing alt-sysrq-c

Crash

- Kernel-specific “debugger”
- Can be used on live systems and dump files
- Requires a vmlinux file with debugging symbols
  - Red Hat builds a -debuginfo package with this (though it isn't distributed)
- Knows about kernel specific data structures
  - custom commands
  - can pretty print these structures

Crash (cont'd)

- Supported file formats
  - Any netdump vmcore
  - LKCD up to dump header version 8 (latest)
  - /dev/mem (2.4 kernels and upstream 2.6)
  - /dev/crash (Red Hat 2.6 kernels)

Preparing the kernel

- FC-3
  - download the SRPM
    - kernel-2.6.9-1.724_FC3.src.rpm
  - install it
    - rpm -i kernel-2.6.9-1.724_FC3.src.rpm
  - This places the kernel tarball and patches in /usr/src/redhat by default
  - Build the kernel
    - rpmbuild -bb /usr/src/redhat/SPECS/kernel-2.6.spec

Preparing the kernel (cont'd)

- Install the -debuginfo kernel
  - rpm -i /usr/src/redhat/RPMS/kernel-debuginfo-2.6.9-1.724_FC3.rpm
- And now you're ready to run crash
  - crash /usr/lib/debug/lib/modules/2.6.9-1.724_FC3/vmlinux
- Crash takes arguments for:
  - mapfile (System.map)
  - namelist (vmlinux)
  - dump file (vmcore or /dev/crash)

crash demo

Future Directions

- Kexec
  - Boot a new kernel without going through the BIOS
    - Big win on larger systems, such as PPC pseries
    - Allows us to preserve memory on reboot
  - Load a new kernel using system call
- Kdump
  - Depends on Kexec
  - Break up the loading of the new kernel and the execution of it

Kdump (cont'd)

- A separate kernel image is provided as the dump kernel.
  - minimal set of drivers
  - provides a proc that is a map of memory
- cp, scp, ftp, or other common utilities can then be used to copy the memory to stable storage
- After this, the system is rebooted
- This approach is attractive since the panic is captured using a stable kernel image.

Summary

- More information is better.
  - Plenty of tools exist to aid in this; use them!
- Deploy what works for you.
- The future hopefully holds a single solution.

Questions?

References

- Crash
  - Where to get it:
    - http://people.redhat.com/anderson
    - RHEL or Fedora repositories
  - Documentation
    - http://people.redhat.com/anderson/crash_whitepaper
- Netdump
  - Kernel patches
    - Available as part of the Red Hat kernel SRPMs
  - Documentation
    - http://www.redhat.com/support/wpapers/redhat/netdump/