Debugging using Kdump

Takashi Iwai <tiwai@suse.de>

SUSE Labs
SUSE Linux Products GmbH, Nuremberg, Germany

July 26, 2006

Oh, customer got a problem

- Haughty kernel developer requests a dump
- Dump image is useful for post-crash analysis
  - A snapshot on critical kernel error (panic)
  - You can see the kernel state via crash, gdb, ...
- Different dump methods: kdump, LKCD, ...

Old Dump Methods

- Dedicated dump driver
  - Limited support of hardwares
  - Difficult to cooperate with filesystems
    - Usually dumped to a partition
- LKCD (linux kernel crash dump)
  - Dump mechanism on SLES9 (still valid for SLES10 ia64)
  - Doesn’t work with many devices
  - netdump, diskdump (requires poll mode)
  - Can’t initialize hardware properly for dumping

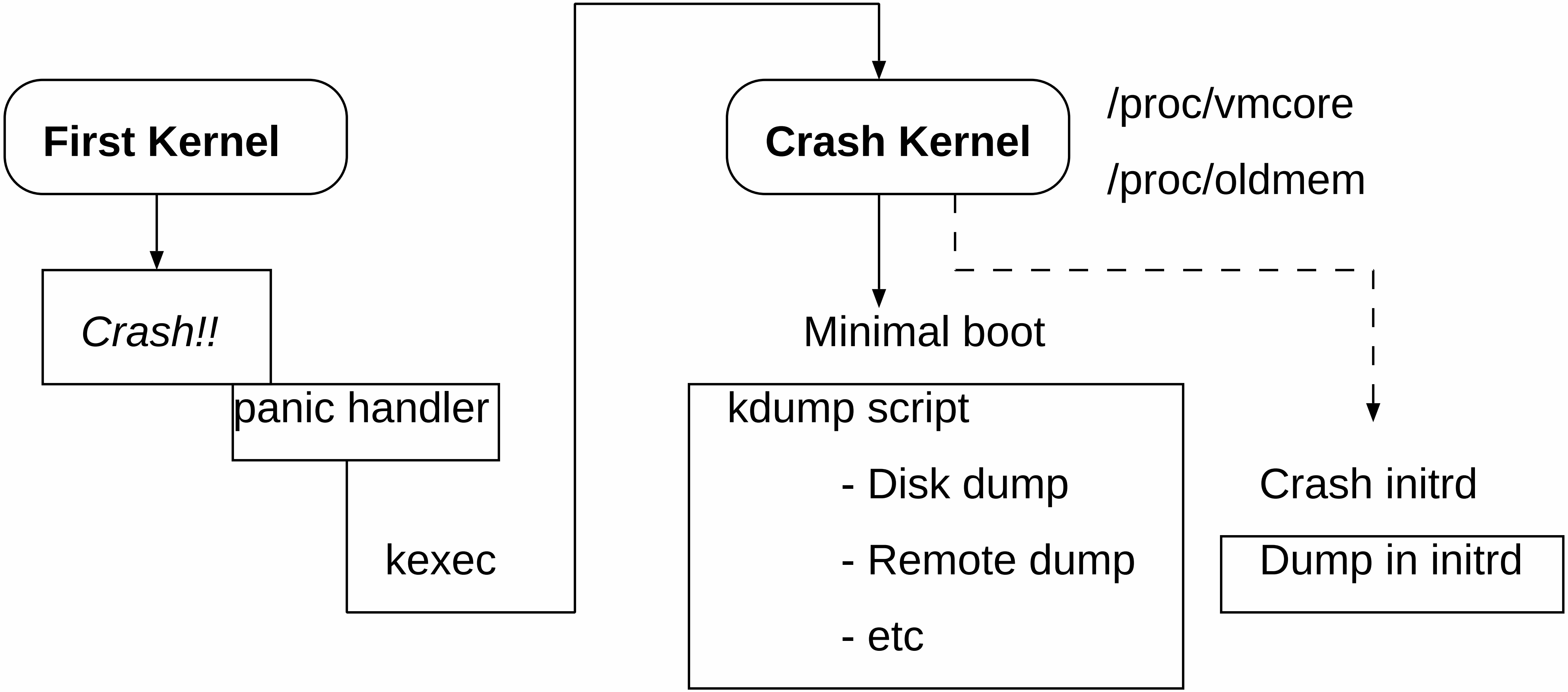
Kdump

- Integrated in mainline kernel
- Standard on SLES10 i386, x86-64 and ppc64
- Reboot-based dump mechanism
  - More robustness and flexibility
- Requires more resources
  - A dedicated dump kernel binary
  - A fixed memory area for 2nd kernel
- Cannot dump non-disruptively

Design Overview

- A secondary (crash) kernel is started after crash
- Kexec is used for kernel-to-kernel switch
- The crash kernel runs in a reserved area
  - The old kernel memory is preserved & untouched
  - ELF image accessible via /proc/vmcore
  - Raw image accessible via /dev/oldmem
- Dump is done on the capture kernel context
  - Devices are re-initialized to sane state
  - You can do almost everything there...

Design Overview (Diagram)



First Kernel

Crash!!

panic handler

kexec

Crash Kernel

/proc/vmcore

/proc/oldmem

Minimal boot

kdump script

- Disk dump
- Remote dump
- etc

Crash initrd

Dump in initrd

Kdump on SLES10

- Minimal boot to runlevel 1 on crash kernel
  - Dump is done on init script: /etc/init.d/kdump
  - Easier setup for complex system (LVM, etc)
  - Netdump possible (not provided by SLES)
- Dump-and-dash tactic
  - Get a dump on /var/log/dump/*
  - Immediately reboot after dump
- Highly configurable via sysconfig

- Reference:
  - /usr/share/doc/packages/kexec-tools/README.SUSE

Setup Kdump on SLES10

- □ Install kexec-tools package
- □ Install kernel-kdump package
- □ Install kernel-*-debuginfo package
- □ Edit /etc/sysconfig/kdump
- □ Enable kdump init service
  - ○ via YaST runlevel manager
  - ○ Alternatively

```
# /sbin/chkconfig kdump on
```

  - ○ "rckdump start" doesn’t suffice!

Setup Kdump on SLES10 (cont’d)

- Add "crashkernel=64M@16M" boot option
  - YaST2 boot loader configuration (or edit GRUB config)
  - 64M = Reserved memory size for capture kernel
  - 16M = Offset of capture kernel (fixed at 16M)
  - For PPC64, 128M@16M is recommended

- Reboot once (what, on linux??)
  - You can use kexec if you’re in hurry

```
# kexec -l /boot/vmlinuz --initrd=/boot/initrd \
  --append=‘cat /proc/cmdline‘" crashkernel=64M@16M"
# kexec -e
```

If You Prefer Manual Operation

□ Loading kdump kernel manually:

```
# kexec -p /boot/vmlinux-kdump \
   --initrd=/boot/initrd-kdump \
   --append="root=/dev/XXX irqpoll ..." \
   --args-linux
```

□ If failed...

- ○ Check /proc/iomem whether your have "Crash" area

Some Internals

- First Kernel
  - CONFIG_KEXEC=y
  - CONFIG_PHYSICAL_START=0x100000 (=1M)
- Capture Kernel
  - CONFIG_CRASH_DUMP=y
  - CONFIG_PHYSICAL_START=0x1000000 (=16M)
  - Stripped configurations
- Additional boot parameters
  - irqpoll, elevator=deadline, sysrq=1 (added automatically)
  - Reduce boot parameters (limited 256 chars)

Editing /etc/sysconfig/kdump

- KDUMP_COMMANDLINE
  - Overrides the default kdump boot parameters
  - You have to set all parameters
- KEXEC_OPTIONS
  - Additional arguments for kexec
  - --args-linux for i386 and x86-64
    - Added automatically at rpm installation
  - --elf32-core-headers is good for gdb on 32bit

More on /etc/sysconfig/kdump

- KDUMP_RUNLEVEL (default: 1)
  - Controls which runlevel to boot kdump kernel
- KDUMP_IMMEDIATE_REBOOT (def: yes)
  - Whether to reboot immediately after kdump script

- KDUMP_TRANSFER
  - The script used as the dumper
  - Empty for the default disk dump
    - Check the available diskspace
    - Create a dump directory from the current time
    - Copy vmcore file
  - You can create your own one here

Let’s Crash

- Do you have a broken driver? Surprise.
- Or, Alt+Sysrq+C triggers crashdump

```
# echo c > /procs/sysrq-trigger
```

- Cross your fingers, sacrifice chickens...

- Screen is kept unchanged during dump
  - Don’t be afraid
  - Serial console is available
    - e.g. boot pameter: console=ttyS0,115200

Post-Crash Analysis

- GDB
  - Can read vmcore (ELF) dump
  - Some helper macros are available
  - gdb-kdump script (in kexec-tools.rpm)
- Crash utility
  - Supports various dump formats
    - LKCD, kdump, xendump, ...
  - Integrated GDB
  - Can examine live system’s kernel internals
  - URL: http://people.redhat.com/~anderson/

Analysis using Crash

- □ Install crash.rpm package
- □ Uncompress /boot/vmlinux-*.gz (if any)
- □ Invokation:

```
# crash /boot/vmlinux-2.6.16-20-smp \
    /var/log/dump/2006-07-24-14:20/vmcore
```

- □ References:
  - ○ "help" command
  - ○ man crash
  - ○ http://people.redhat.com/~anderson/crash_whitepaper/

Analysis using GDB

- Install gdb.rpm package
- Invokation:

```
# gdb-kdump
```

- gdb-kdump helper script
  - Search last vmcore automatically
  - Uncompress vmlinux
  - Add some helper commands
    - bt -- backtrace
    - btpid - pid-specific backtrace
    - dmesg - show kernel message

Remaining Issues

- Kexec doesn’t work on some devices
  - Driver problem -- let’s fix :)
- Can’t kexec from capture kernel
  - Needs either a kernel patch or a hack on kexec-tools
- Requires two different kernels
  - Relocatable kernel?
- Better with initrd?
  - Needs more feedback