Kernel Crash Logging and Core Dump

Cong Wang <amwang@redhat.com>
Software Engineer
Red Hat

Why we need these?

- We want to diagnose kernel failures
- Kernel logging messages may be lost in user-space
- We want to gather as much useful information as we can
- We need to store these information in a more reliable way

Available methods in Linux kernel

- **Early printk, serial console, netconsole**
- **Kmsg dumpers: ramoops, mtdoops**
- **Kdump: core dump of the whole kernel**
- **Pstore: persistent store filesystem**
- **NVRAM: Non-Volatile RAM (in progress)**
- **MCE: hardware errors**

Netconsole

- Easy to setup, arch-independent
- Based on netpoll
- Send kernel messages with UDP via network
- Relies on network stack and network drivers
- Not all network devices support netpoll
- All kernel messages can be stored
- Supports multiple targets

Kernel message dumpers

- Uses platform drivers, arch-dependent
- kmsg_dump_register(), kmsg_dump()
- Ramoops, mtdoops
- Not that easy to setup
- Relies on storage drivers, hardware equipments
- Only selected levels of kernel messages are stored

Pstore and NVRAM

- **Newly developed technologies**
- **Relies on platform drivers**
- **Relies on APEI or UEFI**
- **Pstore provides a generic FS layer for lower persistent storage**
- **NVRAM is still under development**

Kdump

- No dependencies, theoretically ideal, but...
- Based on kexec
- Not all arch support kexec
- Not easy to setup
- Boots a second kernel to retrieve the crash vmcore
- Stores almost all information of the crashed kernel in ELF core
- Needs assistance of other tools for analysis

Why kdump is so special

- A second kernel needs to be started when crashing
- Not all drivers work fine in the second kenrel
- Very limited memory for the second kernel
- We need to construct a new initrd for the second kernel

Kdump problems

- Needs to specify the crashkernel memory manually
- 64-bit kernel needs to go 32-bit first, 4G limited
- Initrd can only be loaded to a limited address, some drivers too
- Virtualization support, especially Xen
- Some kernel parameters are harmful to kdump

Future directions

- Ideally remove crashkernel=X@Y
- MCE friendly
- Load the second kernel into higher memory
- Dump to multiple targets
- Upstream, unify distro initrd construction code

Using kdump on RH Linux

- Install kexec-tools (default on RHEL6)
- Specify crashkernel= kernel parameter (default on RHEL6)
- Configure dump target in /etc/kdump.conf
- Start kdump service: service kdump start
- The second kernel will be automatically started on crash
- Vmcore will be copied to desired place
- Use crash utility to analyze kernel crash

Configure kdump by yourself

- CONFIG_KEXEC=y
- CONFIG_CRASH_DUMP=y
- CONFIG_PROC_VMCORE=y
- CONFIG_RELOCATABLE=y
- CONFIG_PHYSICAL_START=0x1000000

Configure kdump by yourself (cont.)

- Install upstream kexec-tools
- Setup a configure file, /etc/kdup.conf or /etc/sysconfig/kdump
- Make a new initrd for the second kernel
- Make an init script to load/unload the second kernel

Questions and discussion

Thank you!