



# *Kernel Crash Logging and Core Dump*

**Cong Wang <[amwang@redhat.com](mailto:amwang@redhat.com)>**  
**Software Engineer**  
**Red Hat**

# Why

- **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**
- **Kdump: core dump of the whole kernel**
- **Kmsg dumpers: ramoops, mtdoops**
- **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**
- **Supports multiple targets**
- **Relies on network stack and network drivers**
- **Not all network devices support netpoll**
- **All kernel messages can be stored**

# Kernel message dumpers

- Relies on platform drivers, arch-dependent
- `kmsg_dump_register()`, `kmsg_dump()`
- Ramoops, mtdoops
- Not that easy to setup
- Only selected kernel messages are stored
- Could be an alternative to `kdump`

# 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**

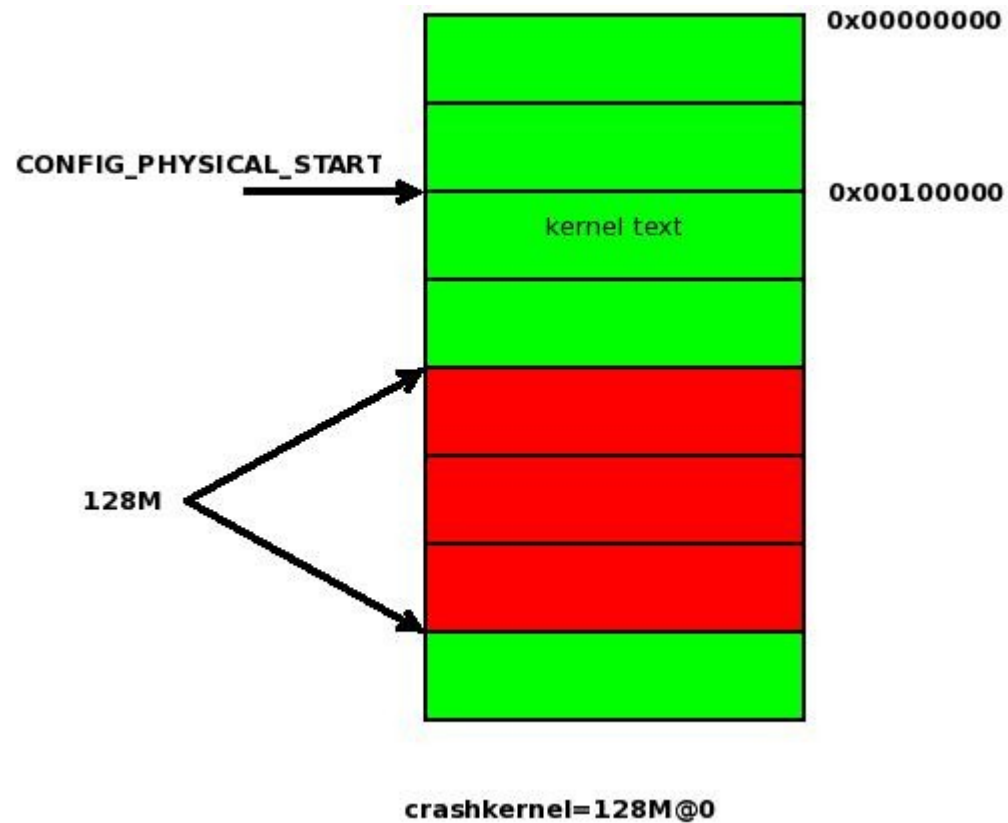
# Kdump

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

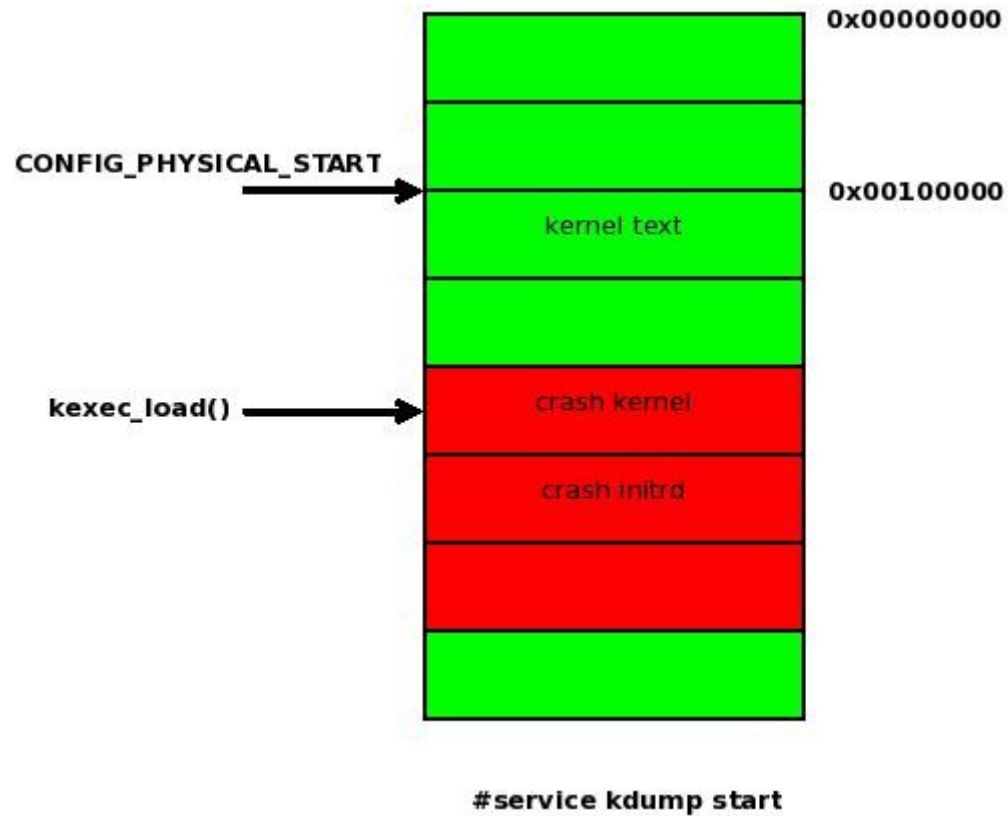
# Kdump advantages

- **Dump both locally and remotely**
- **Fully control of the second kernel**
- **Full information of the crashed kernel**
- **makedumpfile and crash utility**

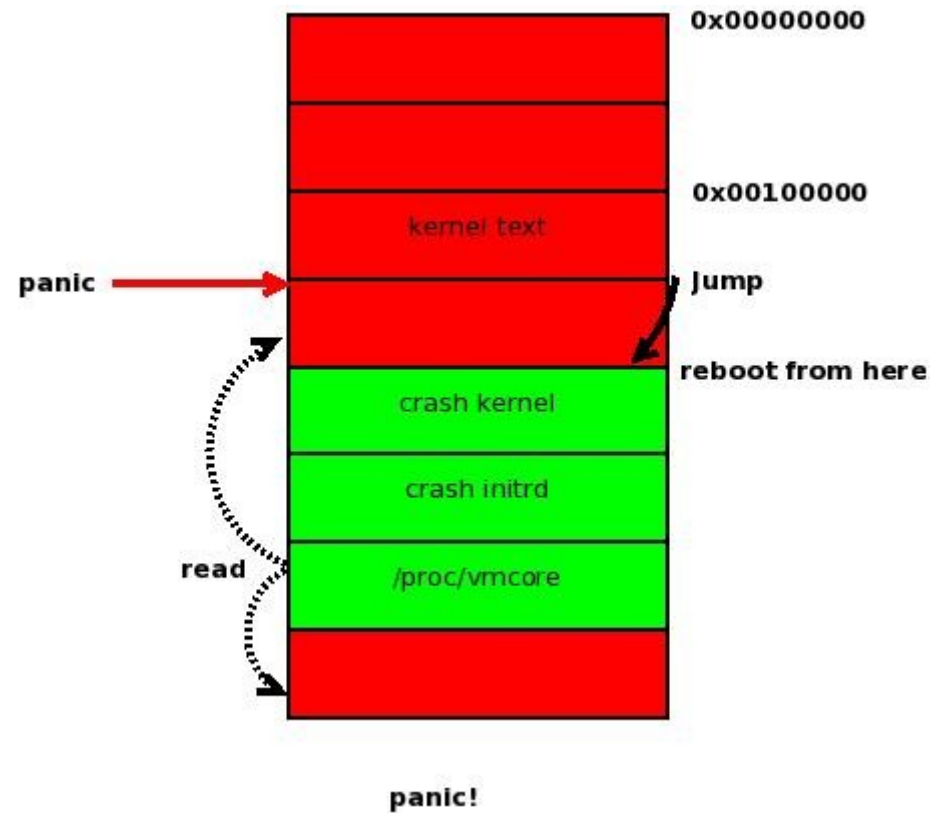
# How it works



# How it works



# How it works



## So kdump is special...

- **A second kernel needs to be started when crashing**
- **Not all drivers work fine in the second kernel**
- **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

# Hack kdump by yourself

- **CONFIG\_KEXEC=y**
- **CONFIG\_CRASH\_DUMP=y**
- **CONFIG\_PROC\_VMCORE=y**
- **CONFIG\_RELOCATABLE=y**
- **CONFIG\_PHYSICAL\_START=0x1000000**

## Hack kdump by yourself (cont.)

- **Install upstream kexec-tools**
- **Setup a configure file, `/etc/kdump.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!**