



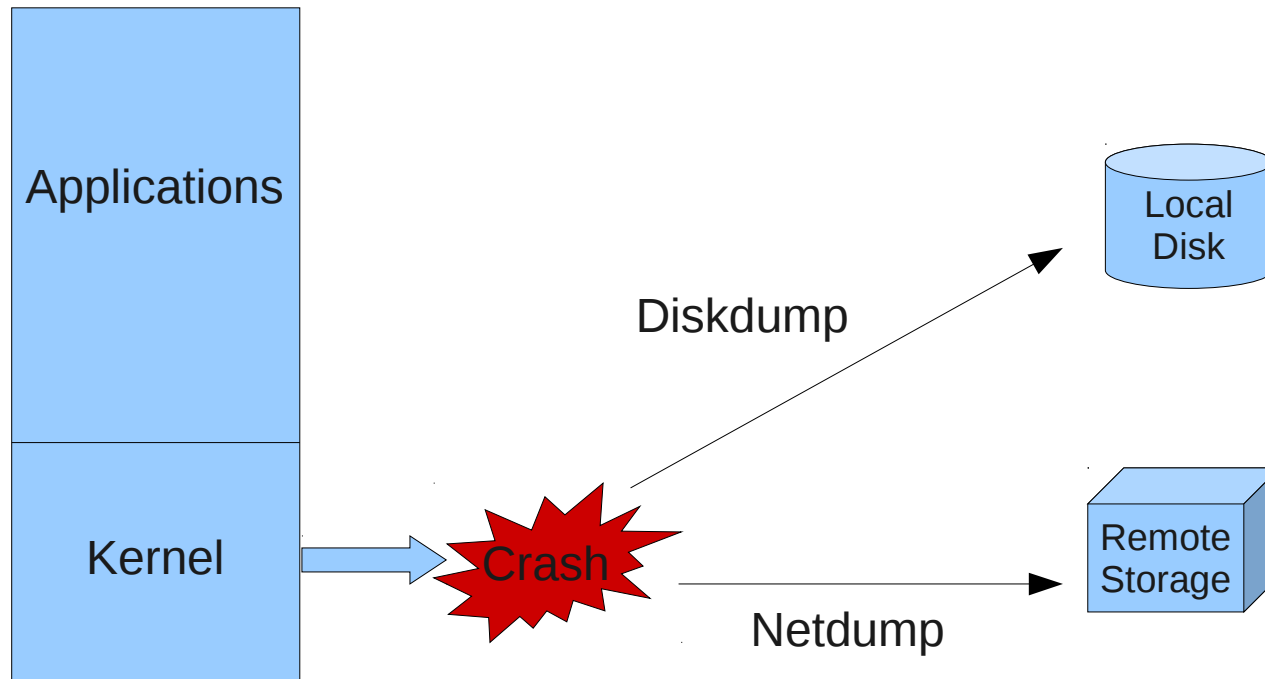
What's the Fuss About Fastboot and New Kernel Crash Dumping Mechanism

Vivek Goyal
Senior Software Engineer
RedHat

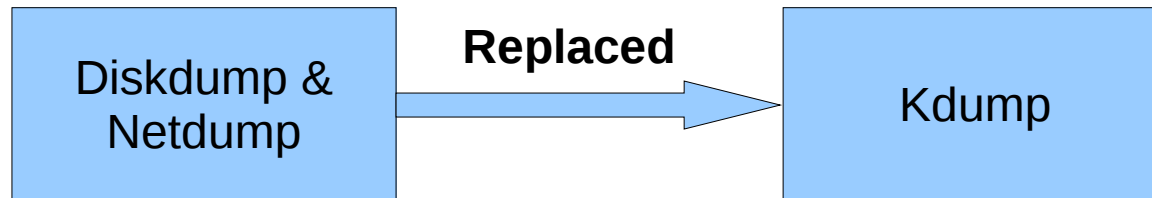
Agenda

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

Kernel crash dumping in RHEL4

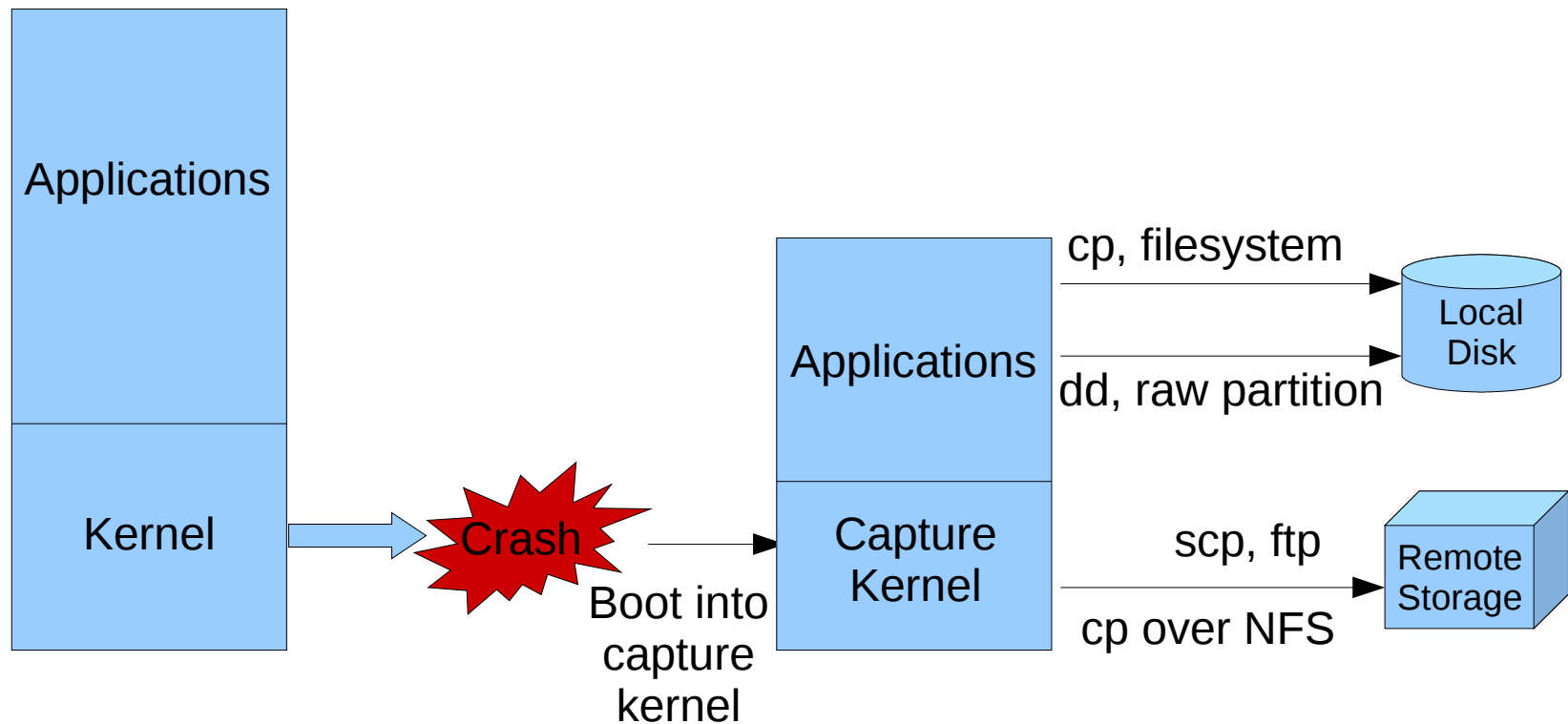


What changed in RHEL5



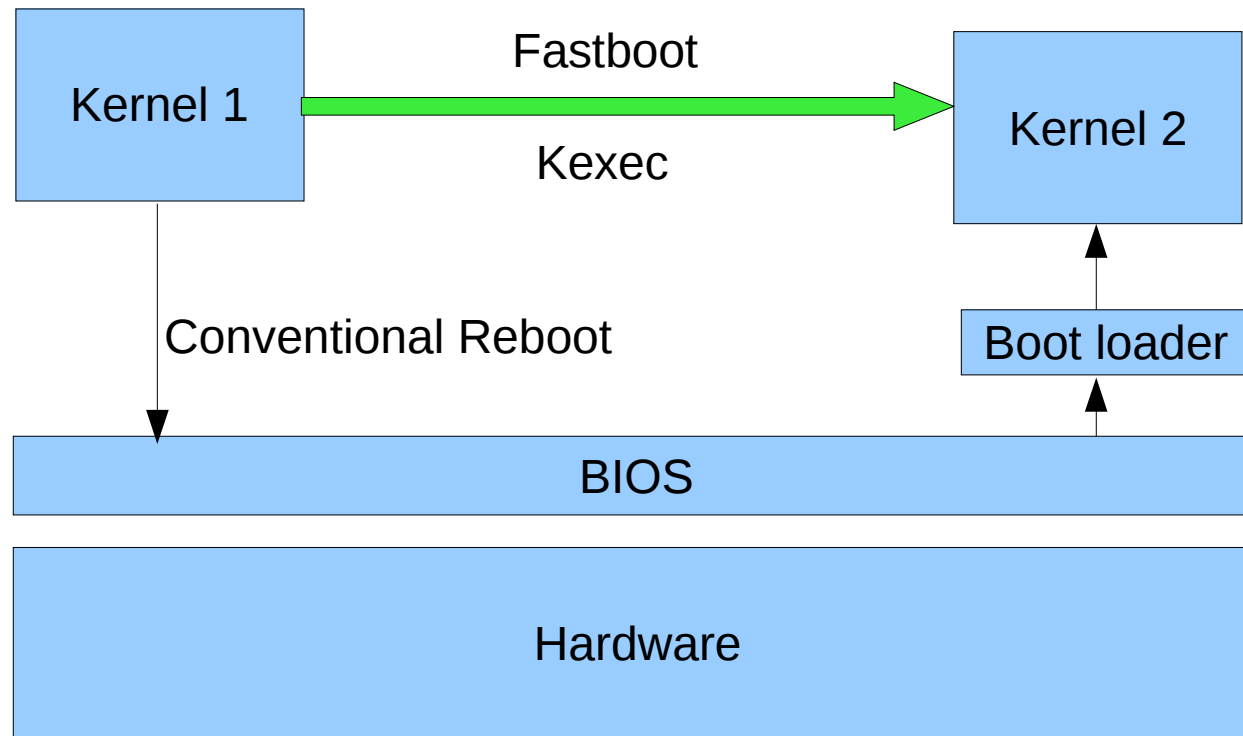
- Reliability
 - Don't trust a crashed kernel
- Upstream Solution
- Flexibility
 - Diskdump and netdump supported limited drivers

Kernel crash dumping in RHEL5

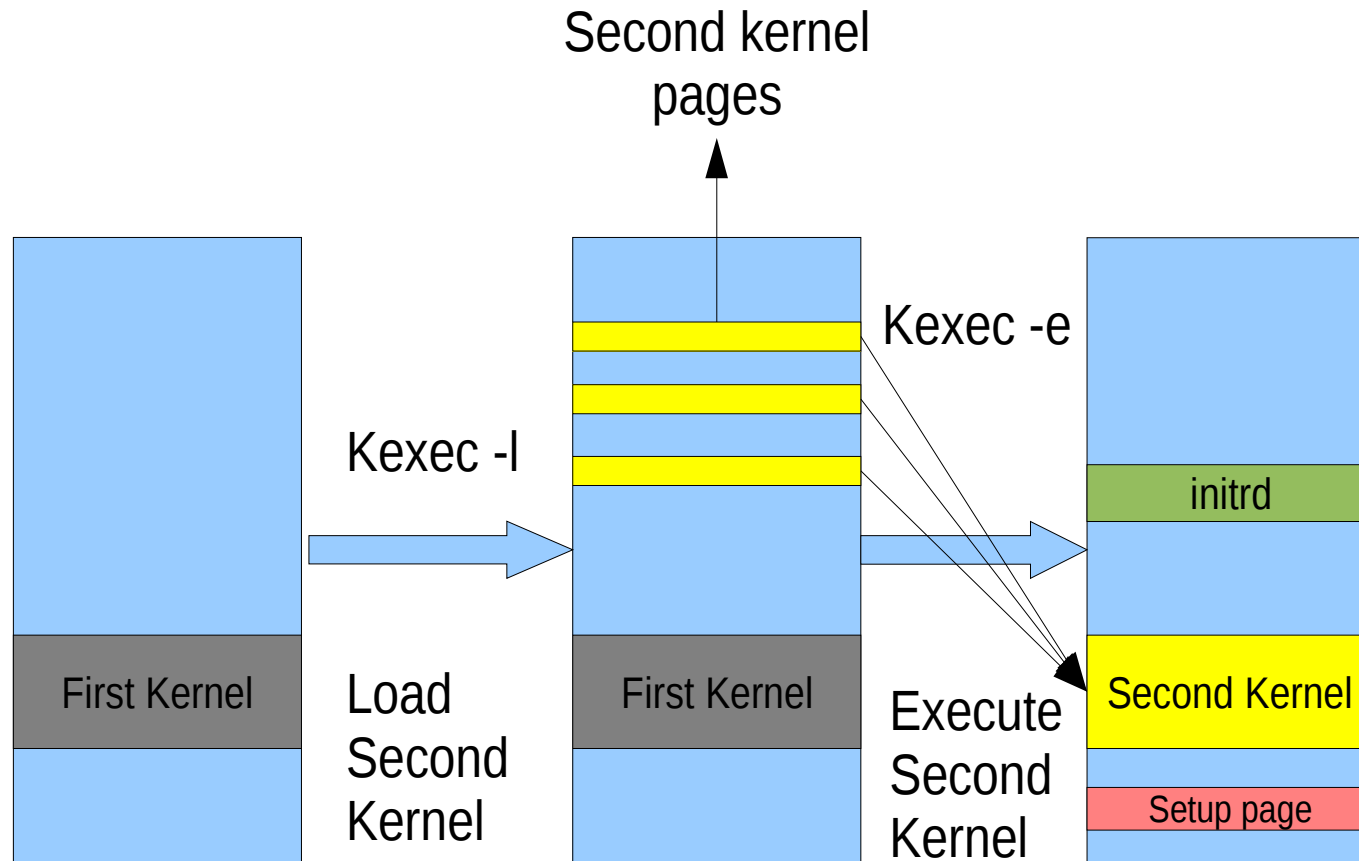


- Supported arch
 - x86, x86_64, ppc64, IA64

Fastboot/Kexec

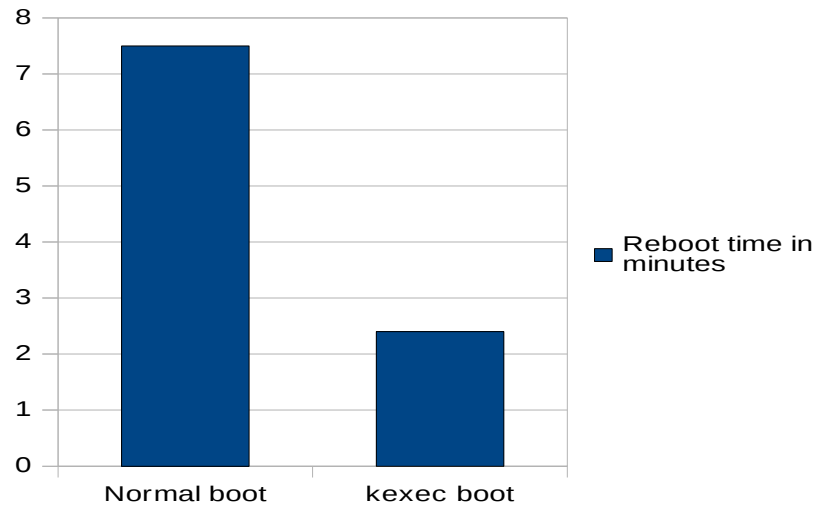


Kexec design



How fast is kexec?

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

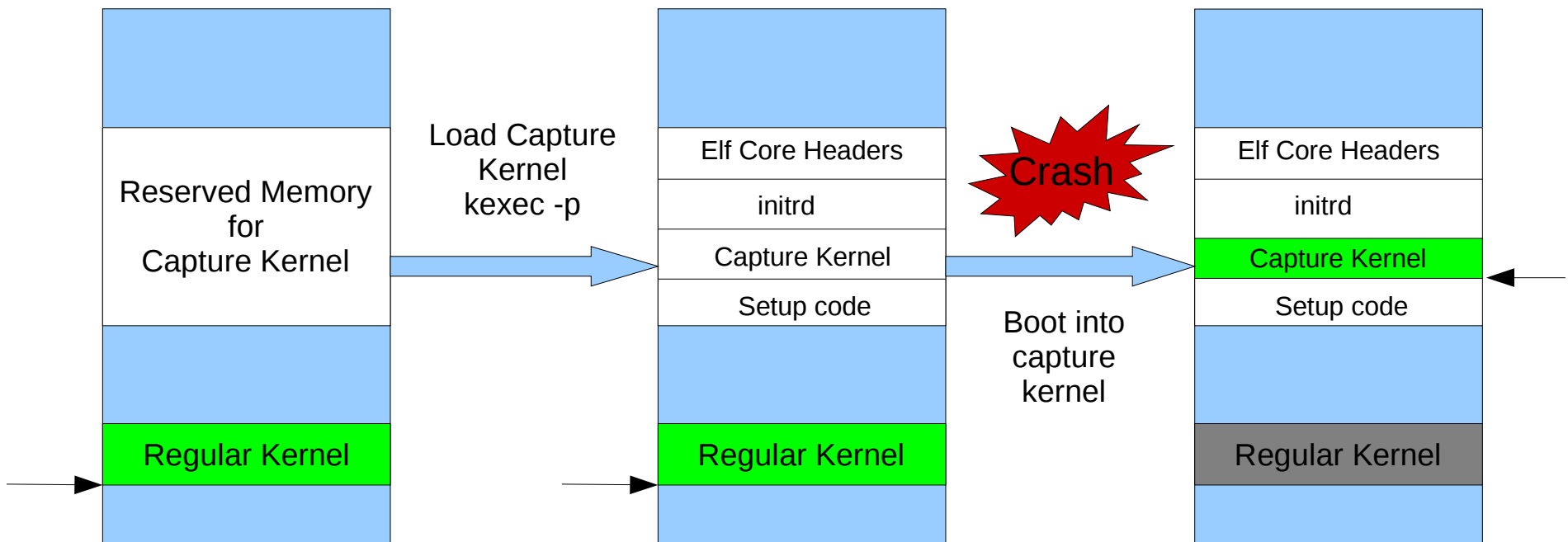


Normal Boot	7.5 minutes
Kexec Boot	2.2 minutes

How to use Kexec

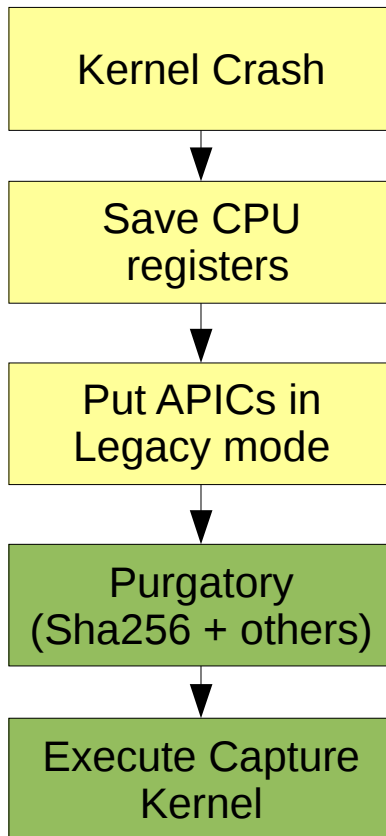
- *yum install kexec-tools*
- Load Kernel
 - */sbin/kexec -l <kernel-to-load> --initrd=<initrd-to-load> --command-line=<command-line>*
- *reboot*
 - Shuts down applications and calls *kexec -e*

Kdump design



- Use `crashkernel=X@Y` to reserve memory for capture kernel
- Capture kernel runs from reserved area unlike kexec
- Protection from ongoing DMA

Control flow after kernel crash



- Minimal dependency on crashed kernel
- Purgatory code ensures pre-loaded capture kernel is not corrupted
- Purgatory code is part of kexec-tools user space package and runs between two kernels

Elf format dump file

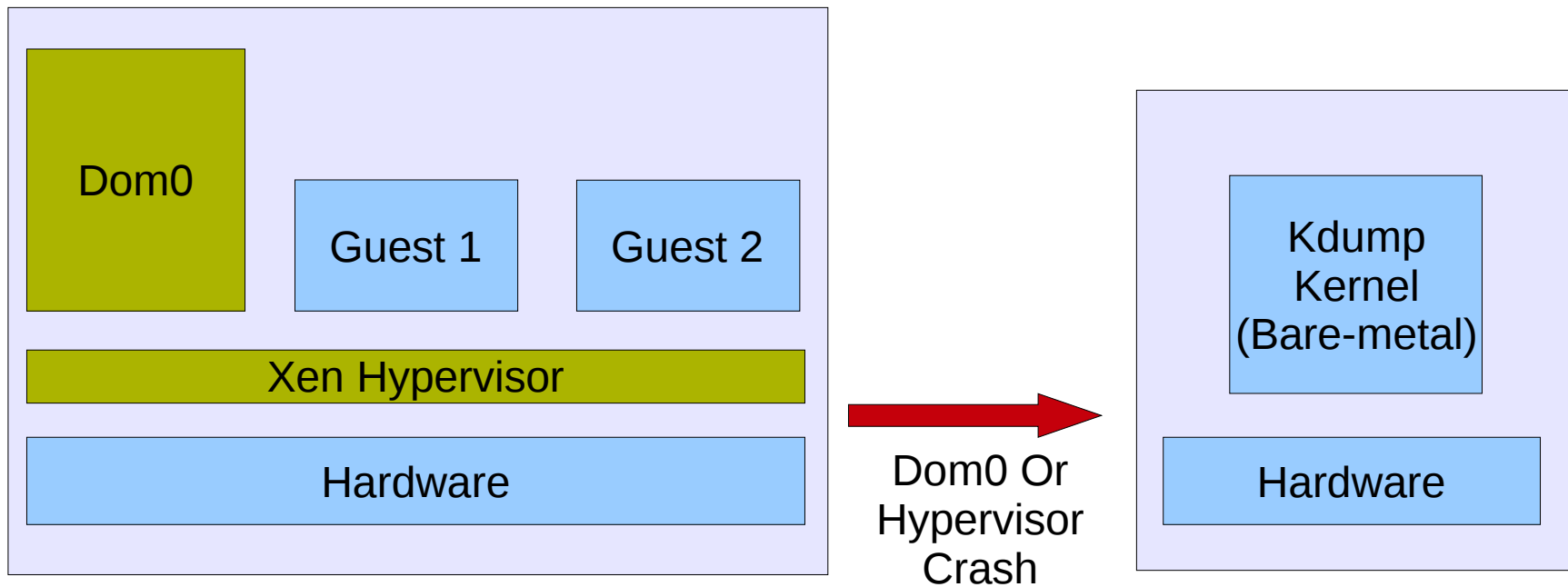
ELF Header	Program Header PT_NOTE	Program Header PT_LOAD	Program Header PT_LOAD	-----	NT_PRSTATUS type Elf Notes	Dump Image
------------	---------------------------	---------------------------	---------------------------	-------	----------------------------------	------------

- 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

Kdump in Xen Environment



- Kdump is used for Dom0 and Hypervisor crashes
- Xendump can be used to capture guest crash dumps

Enabling Kdump

- Enable kdump during installation
 - Firstboot menu gives options to enable kdump
 - Specify amount of memory reserved for capture kernel
- Enable kdump at some point later

Enable kdump at firstboot



The screenshot shows the Kdump configuration screen in a Red Hat installation environment. On the left is a red sidebar with a list of installation steps: Welcome, License Agreement, Firewall, SELinux, Kdump (highlighted with a right-pointing arrow), Date and Time, Set Up Software Updates, Create User, Sound Card, and Additional CDs. At the bottom of the sidebar is the Red Hat logo. The main content area has a light gray background and is titled 'Kdump' with a computer icon. Below the title is a paragraph explaining that kdump is a kernel crash dumping mechanism and that it requires reserving system memory. There is a checkbox labeled 'Enable kdump?' which is currently unchecked. Below this are three memory-related fields: 'Total System Memory (MB): 498', 'Kdump Memory (MB): 128' (with up and down arrows), and 'Usable System Memory (MB): 370'. At the bottom right of the main area are two buttons: 'Back' and 'Forward'.

Welcome
License Agreement
Firewall
SELinux
▶ **Kdump**
Date and Time
Set Up Software Updates
Create User
Sound Card
Additional CDs

Kdump

Kdump is a kernel crash dumping mechanism. In the event of a system crash, kdump will capture information from your system that can be invaluable in determining the cause of the crash. Note that kdump does require reserving a portion of system memory that will be unavailable for other uses.

Enable kdump?

Total System Memory (MB): 498

Kdump Memory (MB): 128

Usable System Memory (MB): 370

[Back](#) [Forward](#)

Enable kdump at firstboot contd.

Welcome
License Agreement
Firewall
SELinux
▶ **Kdump**
Date and Time
Set Up Software Updates
Create User
Sound Card
Additional CDs

Kdump

Kdump is a kernel crash dumping mechanism. In the event of a system crash, kdump will capture information from your system that can be invaluable in determining the cause of the crash. Note that kdump does require reserving a portion of system memory that will be unavailable for other uses.

Enable kdump?

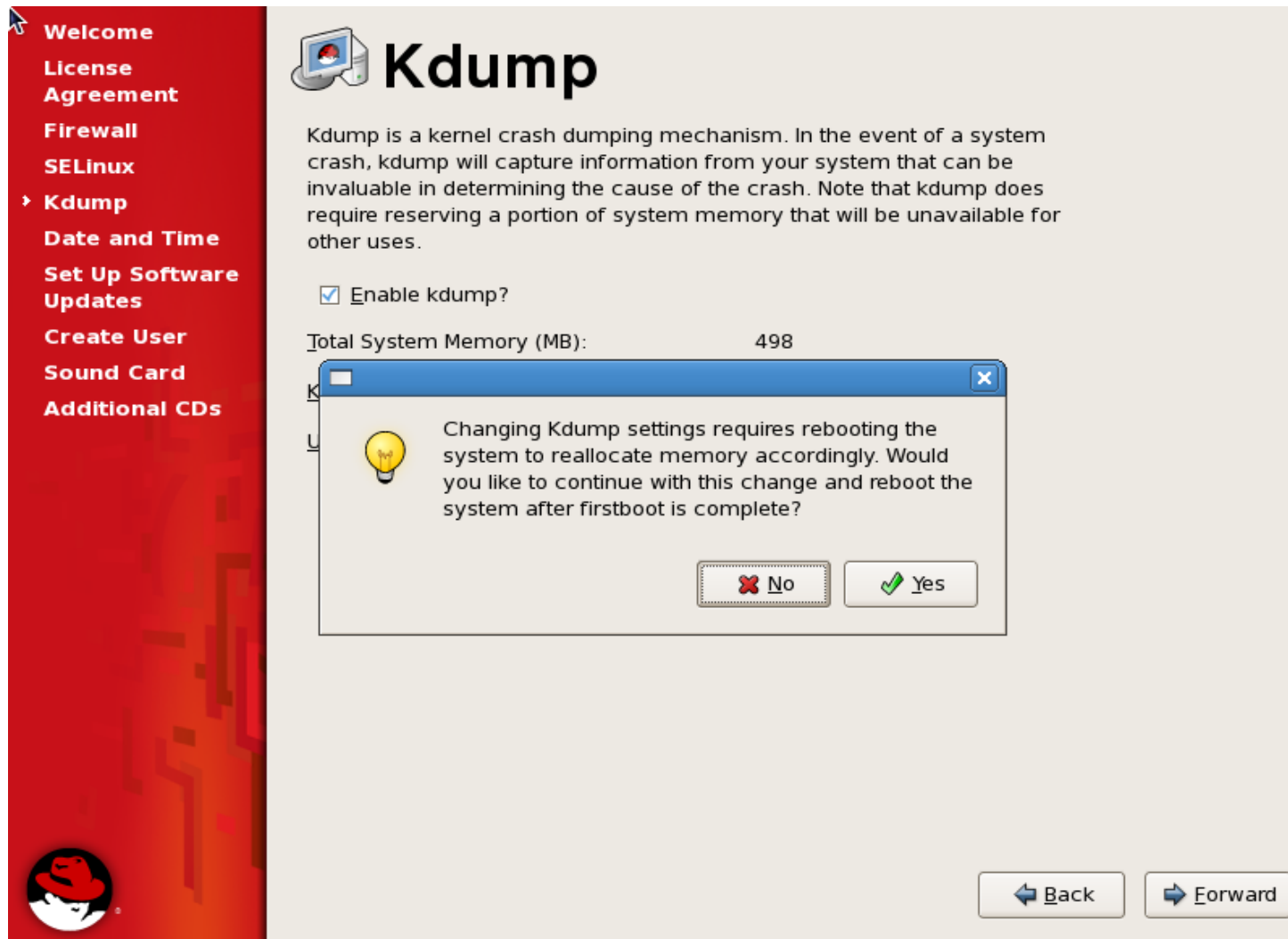
Total System Memory (MB): 498

Kdump Memory (MB): 128

Usable System Memory (MB): 370

[← Back](#) [→ Forward](#)

Enable kdump at firstboot contd.

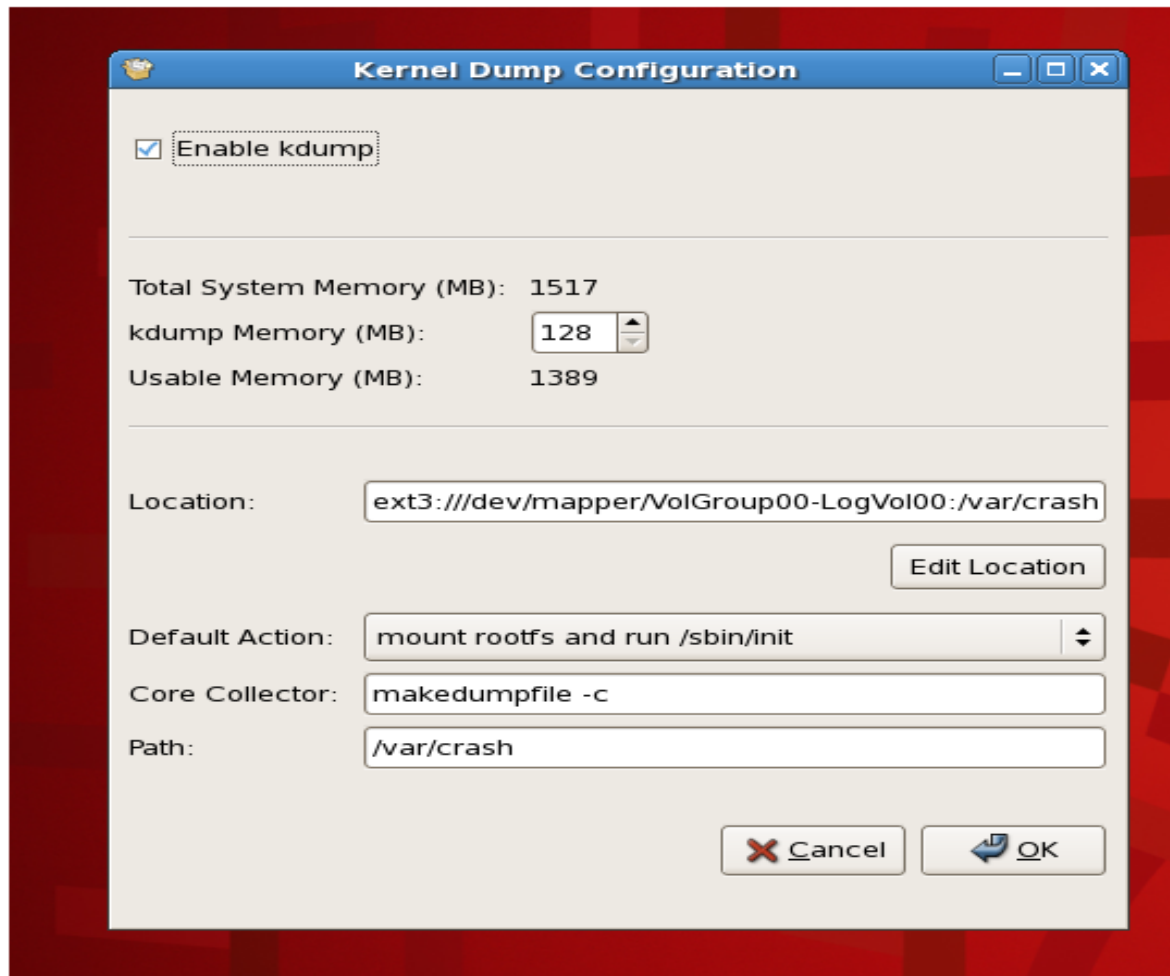


The screenshot shows the 'Kdump' configuration screen in the Red Hat System Configurator. On the left is a red sidebar with navigation options: Welcome, License Agreement, Firewall, SELinux, Kdump (selected), Date and Time, Set Up Software Updates, Create User, Sound Card, and Additional CDs. The main area is titled 'Kdump' and contains the following text: 'Kdump is a kernel crash dumping mechanism. In the event of a system crash, kdump will capture information from your system that can be invaluable in determining the cause of the crash. Note that kdump does require reserving a portion of system memory that will be unavailable for other uses.' Below this text is a checked checkbox labeled 'Enable kdump?'. Underneath, it says 'Total System Memory (MB): 498'. A dialog box is open in the foreground with a lightbulb icon and the text: 'Changing Kdump settings requires rebooting the system to reallocate memory accordingly. Would you like to continue with this change and reboot the system after firstboot is complete?'. The dialog box has two buttons: 'No' (with a red X icon) and 'Yes' (with a green checkmark icon). At the bottom right of the main window are 'Back' and 'Forward' navigation buttons.

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: system-config-kdump



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
- Kdump initrd is rebuilt based and kdump kernel is reloaded

Advance configuration

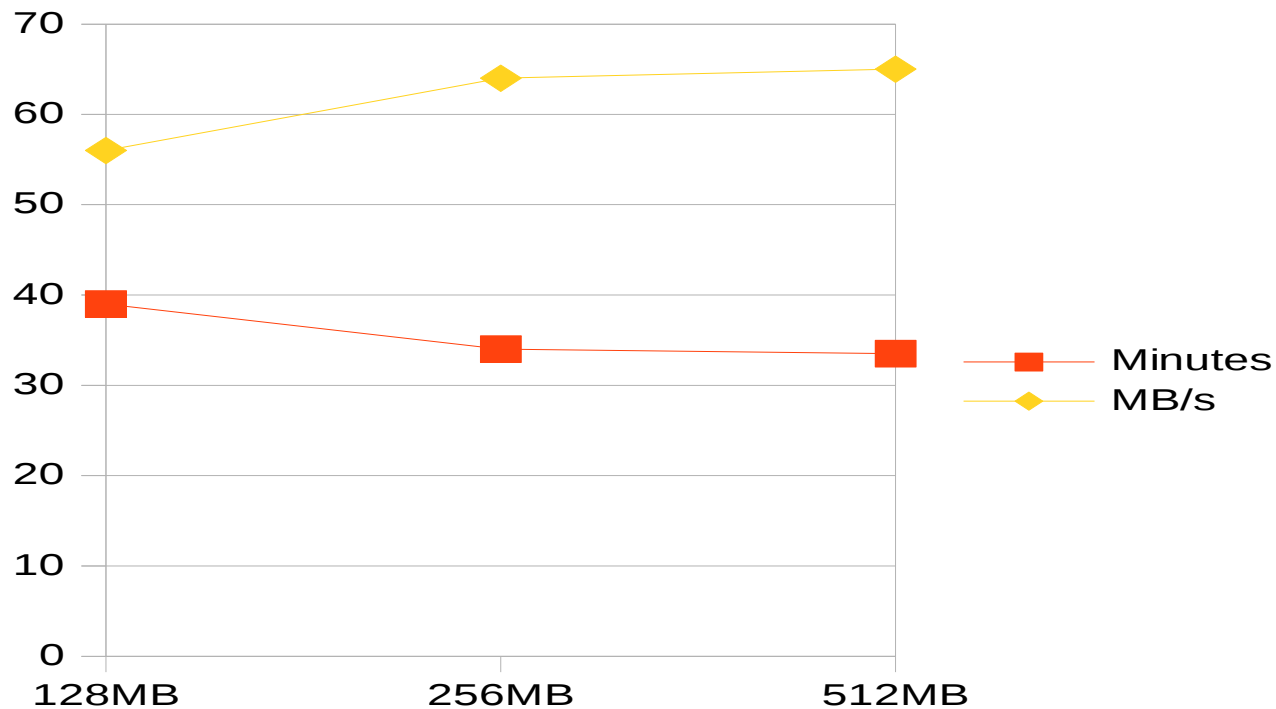
- More configuration options in `/etc/kdump.conf`
 - `extra_bins`
 - Load extra bin/scripts into `initrd`
 - `kdump_post`
 - Specify if some binary/scripts need to be run after saving dump. Handle success/failure.
 - `extra_modules`
- `/etc/sysconfig/kdump`
 - Various command line, kernel version related 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?

- RHEL5.2, x86_64, 64 processor, 128 GB RAM, MPT fusion SAS storage controller
- Took 39 minutes to copy 128 GB file with 128 MB memory



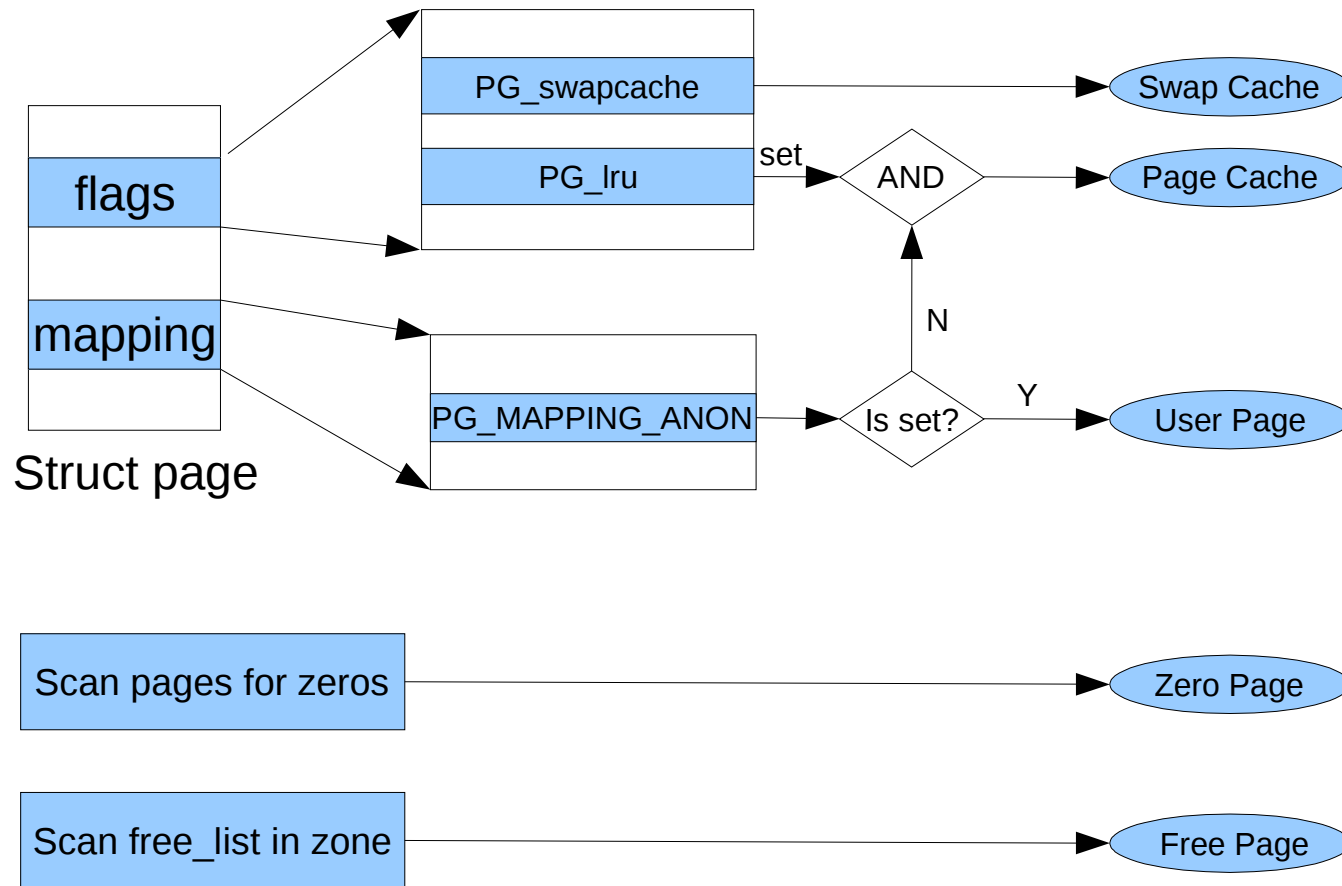
Dump filtering

- 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

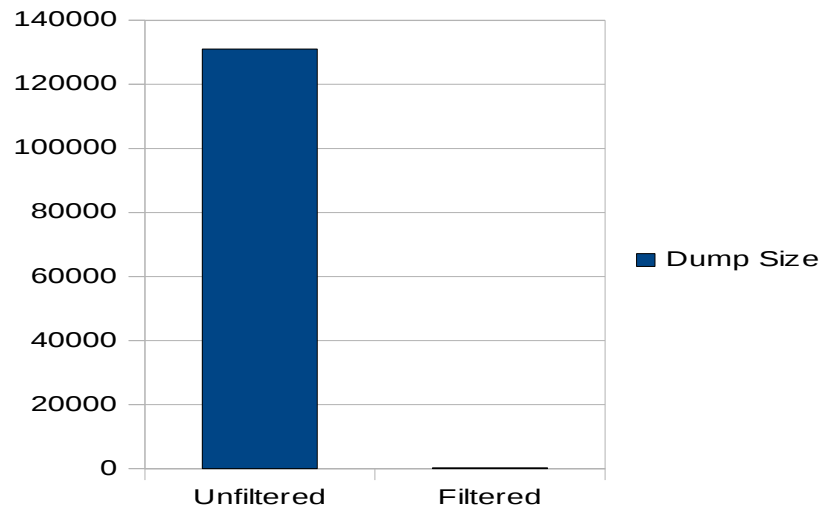
Dump Level	Zero Page	Cache Page	Cache Private	User Data	Free Page
0					
1	x				
2		x			
4		x	x		
8				x	
16					x
31	x	x	x	x	x

Filtering design

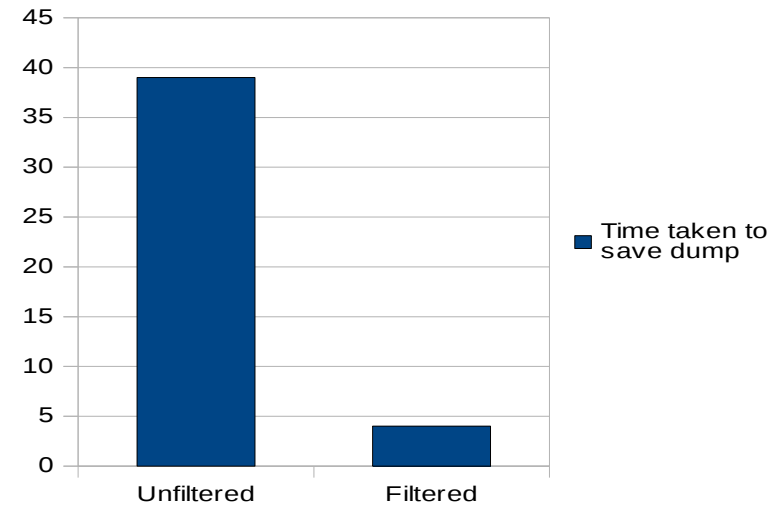


How effective is filtering?

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



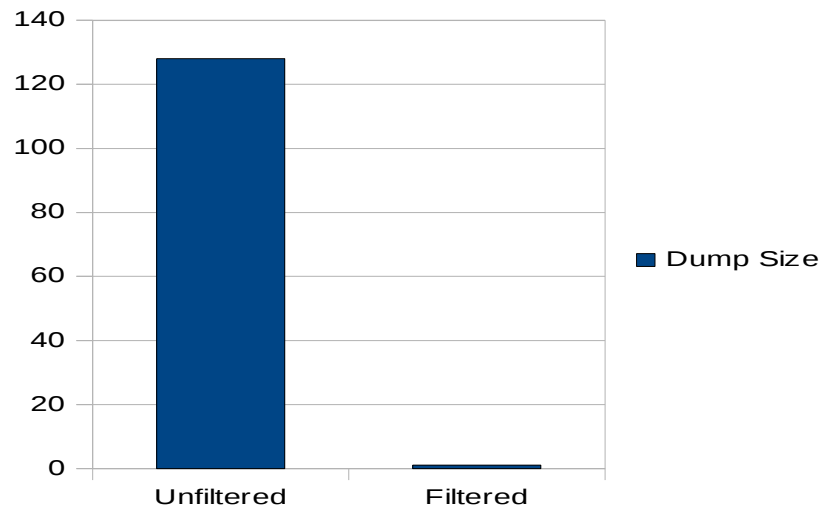
Unfiltered	128GB
Filtered	234MB



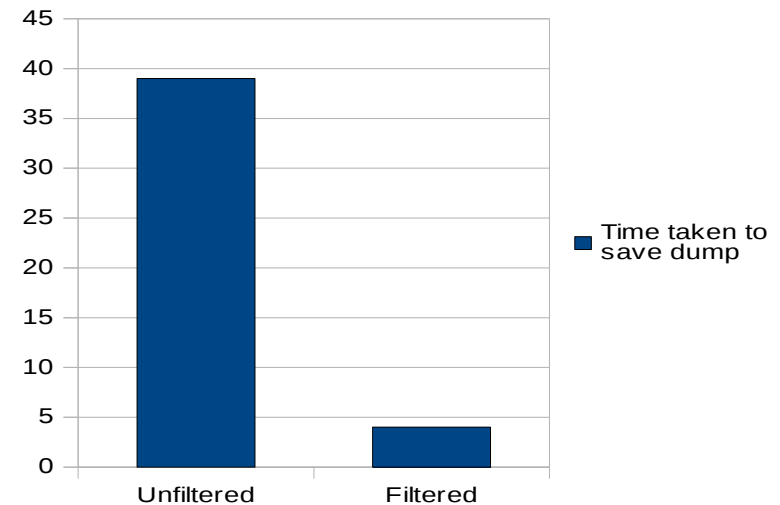
Unfiltered	39 Minutes
Filtered	4 Minutes

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



Unfiltered	128GB
Filtered	1.08 GB



Unfiltered	39 Minutes
Filtered	5 Minutes

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)

Driver/Controller	x86	X86_64	ppc64	IA64
megaraid_sas				
megaraid_mbox				
mptfusion				
mptspi				
mptsas				
sym53c8xx				
lpfc				
cciss				
serveraid				
ipr				
adpxxxx				
aic79xx				
aacraid				
aic94xx				
stex				
qla1280				

Driver test matrix (networking)

Driver/Controller	x86	X86_64	ppc64	IA64
e100				
e1000				
e1000e				
tg3				
q802.1/bonding				
bnx2				

Mailing lists/Documentation/Links

- Kexec, Kdump or makedumpfile issues
 - kexec@lists.infradead.org
- “Crash” Issues
 - crash-utility@redhat.com
- `/usr/share/doc/kexec-tools-1.101/kexec-kdump-howto.txt`
- Kexec man page
- Knowledge base entries
 - http://kbase.redhat.com/faq/FAQ_105_9036.shtm

Questions?

Thank You