

# Kexec: Soft-Reboot and Crash-Dump Analysis for Linux and Xen

Linux.Conf.Au, Sydney, Australia

<http://www.vergenet.net/linux/kexec/>

Simon Horman (Horms) <[simon@valinux.co.jp](mailto:simon@valinux.co.jp)>

Magnus Damm <[magnus@valinux.co.jp](mailto:magnus@valinux.co.jp)>

VA Linux Systems Japan K.K.

<http://www.valinux.co.jp/en/>

18th January 2007

# A Tale in Three Parts

- Part I: Crash-Dump Analysis

# A Tale in Three Parts

- Part I: Crash-Dump Analysis
- Part II: Kexec

# A Tale in Three Parts

- Part I: Crash-Dump Analysis
- Part II: Kexec
- Part III: Port of Kexec to Xen

# Part I

## Crash-Dump Analysis

# Crash-Dump Analysis

- User-Space Core-Dump
  - When an application crashes it is useful to analyse the state of the programme at the time of the crash
  - Operating systems provide a facility to dump the memory core of applications
  - The core can be analysed using tools such as gdb

# Crash-Dump Analysis

- User-Space Core-Dump
  - When an application crashes it is useful to analyse the state of the programme at the time of the crash
  - Operating systems provide a facility to dump the memory core of applications
  - The core can be analysed using tools such as gdb
- Operating System Crash-Dump
  - When an operating system crashes it is also useful to analyse the state of the system at the time of the crash
  - The difficulty arises because it is the operating system that would naturally collect the memory dump, however it has crashed

# Who is Interested In It?

- Enterprise Users
  - Want deep analysis of crashes
  - Are willing to go to third parties for analysis
  - But they want to use vendor-supplied kernels
  - So a standardised solution is needed



# Who is Interested In It?

- Enterprise Users
  - Want deep analysis of crashes
  - Are willing to go to third parties for analysis
  - But they want to use vendor-supplied kernels
  - So a standardised solution is needed
- Kernel Developers
  - If end-users could collect crash dumps, it could be useful for kernel developers to isolate bugs.
  - The facility would need to be provided by distributions.
  - Crash-Dump size may be a problem when transferring between end-users and kernel-developers

# Some Crash-Dump Solutions for Linux

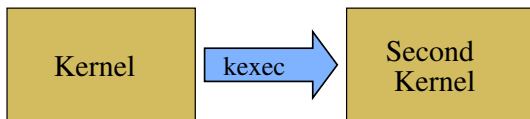
- LKCD — Linux Kernel Crash Dump  
<http://lkcd.sourceforge.net/>
- netdump  
<http://www.redhat.com/support/wpapers/redhat/netdump/>
- diskdump  
<http://sourceforge.net/projects/lkdump/>
- Mini Kernel Dump  
<http://mkdump.sourceforge.net/>
- Kexec/Kdump

Full Disclosure: Mini Kernel Dump was developed in part by my employer, VA Linux Systems Japan K.K.

## Part II

### Kexec

- Kexec is a feature of the Linux kernel that allows a new kernel to run in place of the running kernel



- It is named after `exec(2)`, the system call which allows a process to replace a running process

# Motivation for Kexec

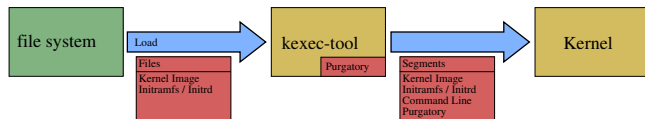
- Allows rebooting to skip the BIOS and hardware initialisation
  - Some hardware has buggy BIOSes that do not allow reboot to occur reliably
  - Some hardware takes a really long time to reboot
  - Rebooting using kexec can be really fast
- Can be very useful for bootstrapping and rebooting in development environments.

# Performing Kexec

- The new kernel is loaded into the running kernel from user-space using kexec-tool  
# kexec -l /boot/vmlinuz
- The new kernel can then be kexeced using the same tool  
# kexec -e
- Kexec is a method of warm or soft-booting machines

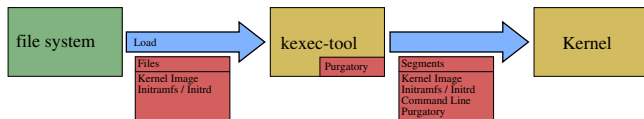
# Kexec Flow

## 1. Kexec Load

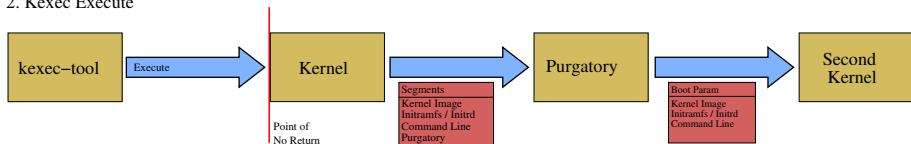


# Kexec Flow

## 1. Kexec Load



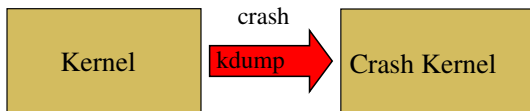
## 2. Kexec Execute





# Kdump

- Kdump is an extension of kexec which allows a crash kernel to be executed in place of a running kernel if the running kernel panics



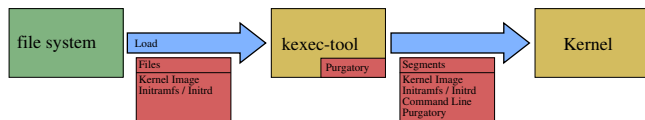
- Kdump and kexec use separate kernel images, which may be loaded independently of each other

# Kexec vs Kdump

- Kexec
  - Uses (all) system memory as any booting kernel would
  - The second kernel runs when triggered by kexec-tool
- Kdump
  - Runs the crash kernel in a reserved area of memory, so as to leave the panicked kernel's memory untouched for forensic analysis
  - The crash kernel runs when the running kernel panics

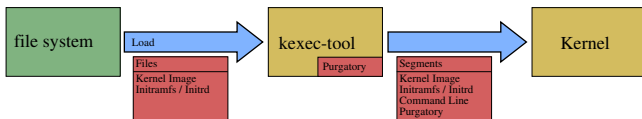
# Kdump Flow

## 1. Kdump Load

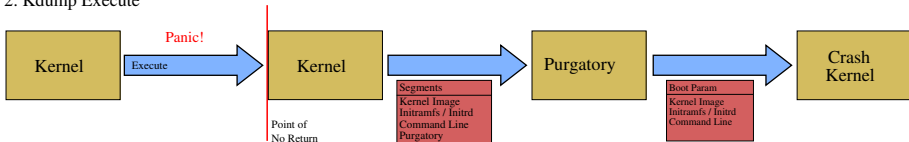


# Kdump Flow

## 1. Kdump Load



## 2. Kdump Execute

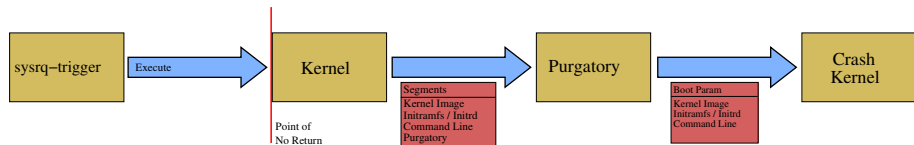


# Triggering Kdump

- Usually triggered by the kernel on panic
- But may be triggered from user-space (for testing purposes? :-)  
`echo c > /proc/sysrq-trigger`

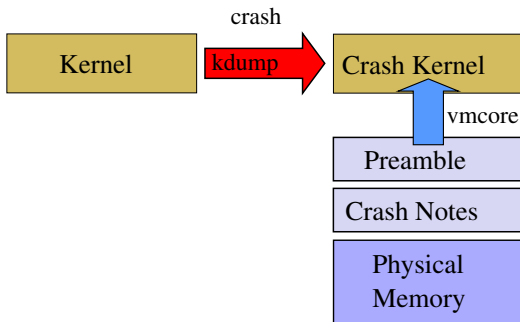
# Triggering Kdump

- Usually triggered by the kernel on panic
- But may be triggered from user-space (for testing purposes? :-)  
`echo c > /proc/sysrq-trigger`



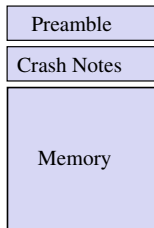
# VMCore

- Visible as `/proc/vmcore` from within a crash kernel run by `kdump`
- Includes the entire physical memory, as visible to the panicked kernel



# Core Files

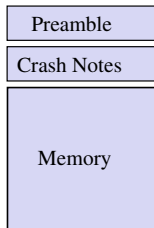
- Linux core files are generally ELF format





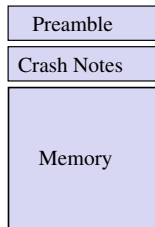
# Core Files

- Linux core files are generally ELF format
- They start with an ELF preamble



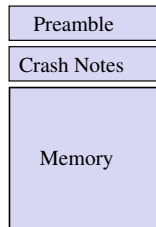
# Core Files

- Linux core files are generally ELF format
- They start with an ELF preamble
- Then there is a crash notes section which includes the pid of the crashing process, cpu registers, and the like



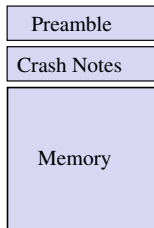
# Core Files

- Linux core files are generally ELF format
- They start with an ELF preamble
- Then there is a crash notes section which includes the pid of the crashing process, cpu registers, and the like
- And then there is the memory core of the crashed process

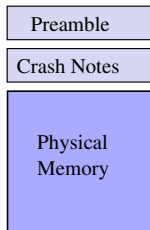


# Core Files

- Linux core files are generally ELF format
- They start with an ELF preamble
- Then there is a crash notes section which includes the pid of the crashing process, cpu registers, and the like
- And then there is the memory core of the crashed process
- Core files may be analysed using tools such as gdb

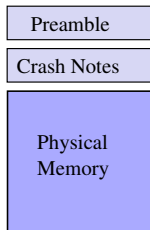


- Kexec's vmcore is also ELF format



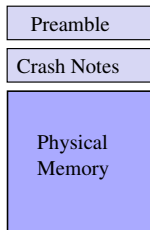
# VMCore Files

- Kexec's vmcore is also ELF format
- Preamble



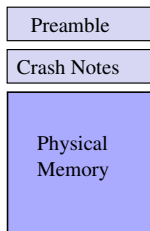
# VMCore Files

- Kexec's vmcore is also ELF format
- Preamble
- Then per-CPU crash notes



# VMCore Files

- Kexec's vmcore is also ELF format
- Preamble
- Then per-CPU crash notes
- And then the entire physical memory of the crashed machine





# Using Kexec for Crash-Dump Analysis

- Kdump can be used to capture the state of a system when a panic occurs

# Using Kexec for Crash-Dump Analysis

- Kdump can be used to capture the state of a system when a panic occurs
- VMcore can be used to save this state to disk or a remote host

# Using Kexec for Crash-Dump Analysis

- Kdump can be used to capture the state of a system when a panic occurs
- VMcore can be used to save this state to disk or a remote host
- The resulting image than then be analysed off-line using tools like crash

[http://people.redhat.com/anderson/crash\\_whitepaper/](http://people.redhat.com/anderson/crash_whitepaper/)

## Part III

# Port of Kexec to Xen

# Motivation for Porting Kexec to Xen

- Provide a crash-dump facility for both domain 0 *and* the hypervisor
  - kexec for domain U is being developed separately by Gerd Hoffmann

# Motivation for Porting Kexec to Xen

- Provide a crash-dump facility for both domain 0 *and* the hypervisor
  - kexec for domain U is being developed separately by Gerd Hoffmann
- Kexec is a strong solution for hypervisor crash-dump

# Motivation for Porting Kexec to Xen

- Provide a crash-dump facility for both domain 0 *and* the hypervisor
  - kexec for domain U is being developed separately by Gerd Hoffmann
- Kexec is a strong solution for hypervisor crash-dump
- And for free it can also provide crash-dump for domain 0

# Motivation for Porting Kexec to Xen

- Provide a crash-dump facility for both domain 0 *and* the hypervisor
  - kexec for domain U is being developed separately by Gerd Hoffmann
- Kexec is a strong solution for hypervisor crash-dump
- And for free it can also provide crash-dump for domain 0
- It has an active developer community (for Linux)



# Motivation for Porting Kexec to Xen

- Provide a crash-dump facility for both domain 0 *and* the hypervisor
  - kexec for domain U is being developed separately by Gerd Hoffmann
- Kexec is a strong solution for hypervisor crash-dump
- And for free it can also provide crash-dump for domain 0
- It has an active developer community (for Linux)
- And as a further bonus, provides a method of soft-booting

# Implementation in Brief

- Linux's kexec code has a generic core and architectures implement several hooks

# Implementation in Brief

- Linux's kexec code has a generic core and architectures implement several hooks
- The implementation of these hooks have been moved into the hypervisor
  - Accessed via a hypercall
  - Only the hypervisor has access to the machine's physical memory which is a requirement for implementing kexec
  - Allows the hypervisor to trigger kdump if it panics

# Implementation in Brief

- Linux's kexec code has a generic core and architectures implement several hooks
- The implementation of these hooks have been moved into the hypervisor
  - Accessed via a hypercall
  - Only the hypervisor has access to the machine's physical memory which is a requirement for implementing kexec
  - Allows the hypervisor to trigger kdump if it panics
- Additional hooks were added to load and unload kernels into the hypervisor

# Xen Kexec Flow

## 1. Kexec Load



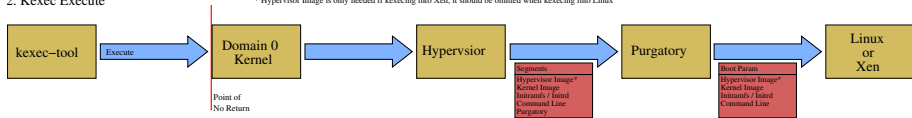
\* Hypervisor Image is only needed if kexecing into Xen, it should be omitted when kexecing into Linux

# Kexec Flow

## 1. Kexec Load

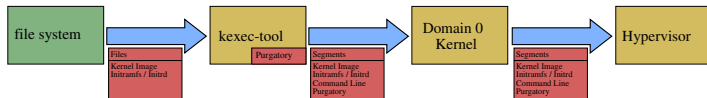


## 2. Kexec Execute



# Xen Kdump Flow

## 1. Kdump Load

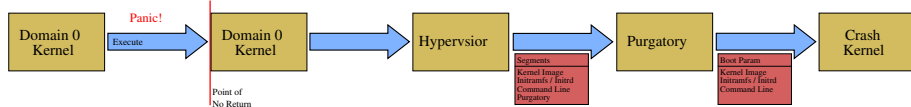


# Xen Kdump Flow

## 1. Kdump Load



## 2a. Kdump Execute - Domain 0 Panics



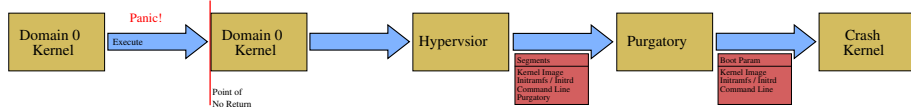


# Xen Kdump Flow

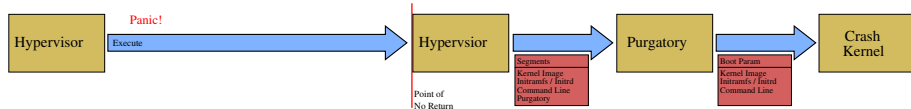
## 1. Kdump Load



## 2a. Kdump Execute - Domain 0 Panics

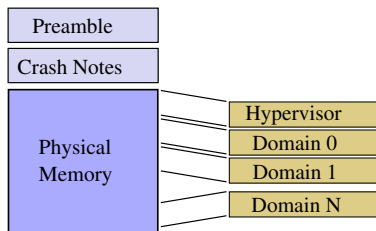


## 2b. Kdump Execute - Hypervisor Panics



# VMCore Files for Xen

- The same as Linux's (for now)
- memory includes the hypervisor, domain 0 and domain U
- Kazuo Moriwaka is working on dumpread, which allows the memory of the hypervisor, or any of the domains to be extracted from the vmcore
- Itsuro Oda is working on xencrash, a fork of crash for handling xen crash-dumps.



Note that domain memory is not linear

\* Full Disclosure: Both Kazuo Moriwaka and Itsuro Oda are colleagues of mine

# Implementation Problems: Constants

- Some constants differ between Linux and Xen  
e.g. PAGE\_OFFSET (ia64)

# Implementation Problems: Constants

- Some constants differ between Linux and Xen  
e.g. PAGE\_OFFSET (ia64)
- Compile time constants in kexec-tool: purgatory

# Implementation Problems: Constants

- Some constants differ between Linux and Xen  
e.g. PAGE\_OFFSET (ia64)
- Compile time constants in kexec-tool: purgatory
- Compile time constants in the Kernel: `relocate_kernel()`

# Implementation Problems: Constants

- Some constants differ between Linux and Xen  
e.g. PAGE\_OFFSET (ia64)
- Compile time constants in kexec-tool: purgatory
- Compile time constants in the Kernel: `relocate_kernel()`
- Globals the Kernel: `relocate_kernel()`

# Implementation Problems: Constants

- Some constants differ between Linux and Xen  
e.g. `PAGE_OFFSET` (ia64)
- Compile time constants in kexec-tool: `purgatory`
- Compile time constants in the Kernel: `relocate_kernel()`
- Globals the Kernel: `relocate_kernel()`
- In practice this has not been a problem for the i386 and x86\_64 implementations, but it does manifest in ia64

# Implementation Problems: Crash Notes

- Problem
  - Per-cpu memory reserved at kernel boot time



# Implementation Problems: Crash Notes

- Problem
  - Per-cpu memory reserved at kernel boot time
  - Crash note mapping for VMcore is set up at kexec-load time  
`/sys/devices/system/cpu/cpu*/crash_notes`

# Implementation Problems: Crash Notes

- Problem
  - Per-cpu memory reserved at kernel boot time
  - Crash note mapping for VMcore is set up at kexec-load time  
/sys/devices/system/cpu/cpu\*/crash\_notes
  - Written to at crash-time for each cpu

# Implementation Problems: Crash Notes

- Problem
  - Per-cpu memory reserved at kernel boot time
  - Crash note mapping for VMcore is set up at kexec-load time  
/sys/devices/system/cpu/cpu\*/crash\_notes
  - Written to at crash-time for each cpu
  - When using xen, the domain U kernels see virtual cpus
    - May only be a subset of physical cpus
    - Physical to virtual mapping may change at any time

# Implementation Problems: Crash Notes

- Problem

- Per-cpu memory reserved at kernel boot time
- Crash note mapping for VMCore is set up at kexec-load time  
/sys/devices/system/cpu/cpu\*/crash\_notes
- Written to at crash-time for each cpu
- When using xen, the domain U kernels see virtual cpus
  - May only be a subset of physical cpus
  - Physical to virtual mapping may change at any time

- Solution

- Leave crash notes as-is in the domain 0 kernel
  - The virtual cpus at the time of kexec-load will be visible in the VMCore
  - The virtual cpus at the time of crash will be saved, if the domain 0 kernel crashes

# Implementation Problems: Crash Notes

- Problem
  - Per-cpu memory reserved at kernel boot time
  - Crash note mapping for VMCore is set up at kexec-load time  
/sys/devices/system/cpu/cpu\*/crash\_notes
  - Written to at crash-time for each cpu
  - When using xen, the domain U kernels see virtual cpus
    - May only be a subset of physical cpus
    - Physical to virtual mapping may change at any time
- Solution
  - Leave crash notes as-is in the domain 0 kernel
    - The virtual cpus at the time of kexec-load will be visible in the VMCore
    - The virtual cpus at the time of crash will be saved, if the domain 0 kernel crashes
  - A separate area is reserved by the hypervisor to save per-cpu crash-note-like information
    - Saved regardless of if it is the hypervisor or domain 0 kernel that crash
    - Visible as ELF sections in the VMCore
    - But a customised tool is needed anyway

# Implementation Problems: Crash Notes

- Problem

- Per-cpu memory reserved at kernel boot time
- Crash note mapping for VMCore is set up at kexec-load time  
/sys/devices/system/cpu/cpu\*/crash\_notes
- Written to at crash-time for each cpu
- When using xen, the domain U kernels see virtual cpus
  - May only be a subset of physical cpus
  - Physical to virtual mapping may change at any time

- Solution

- Leave crash notes as-is in the domain 0 kernel
  - The virtual cpus at the time of kexec-load will be visible in the VMCore
  - The virtual cpus at the time of crash will be saved, if the domain 0 kernel crashes
- A separate area is reserved by the hypervisor to save per-cpu crash-note-like information
  - Saved regardless of if it is the hypervisor or domain 0 kernel that crash
  - Visible as ELF sections in the VMCore
  - But a customised tool is needed anyway
- N.B: This is probably broken if cpus are hotplugged, regardless of Xen

# Status of the Xen Port

- Working with the port, all combinations of Xen/Linux→Xen/Linux transitions possible
- Without the port, only Linux→Xen/Linux transitions are possible
- Status

x86_32	included in Xen 3.0.4
x86_64	included in Xen 3.0.4
ia64	work in progress

# Questions?