

The Xen Port of Kexec / Kdump

A short introduction and status report

Magnus Damm Simon Horman

VA Linux Systems Japan K.K.
www.valinux.co.jp/en/

Xen Summit, September 2006

Outline

Introduction to Kexec

- What is Kexec?

- Kexec Examples

- Kexec Overview

Introduction to Kdump

- What is Kdump?

- Kdump Kernels

- The Crash Utility

Xen Porting Effort

- Kexec under Xen

- Kdump under Xen

- The Dumpread Tool

- Partial Dumps

- Current Status

Outline

Introduction to Kexec

- What is Kexec?

- Kexec Examples

- Kexec Overview

Introduction to Kdump

- What is Kdump?

- Kdump Kernels

- The Crash Utility

Xen Porting Effort

- Kexec under Xen

- Kdump under Xen

- The Dumpread Tool

- Partial Dumps

- Current Status

What is Kexec?

"kexec is a system call that implements the ability to shutdown your current kernel, and to start another kernel. It is like a reboot but it is independent of the system firmware..."

Configuration help text in Linux-2.6.17

What is Kexec?

"kexec is a system call that implements the ability to shutdown your current kernel, and to start another kernel. It is like a reboot but it is independent of the system firmware..."

Configuration help text in Linux-2.6.17

Kexec allows you to reboot from Linux into *any* kernel...

What is Kexec?

"kexec is a system call that implements the ability to shutdown your current kernel, and to start another kernel. It is like a reboot but it is independent of the system firmware..."

Configuration help text in Linux-2.6.17

Kexec allows you to reboot from Linux into *any* kernel...

...as long as the new kernel doesn't depend on the BIOS for setup.

Kexec Examples

Below are a few examples on how to use Kexec to reboot into. . .

Kexec Examples

Below are a few examples on how to use Kexec to reboot into. . .

Linux

- ▶ `kexec vmlinux --append="root=/dev/hda3"`

Kexec Examples

Below are a few examples on how to use Kexec to reboot into. . .

Linux

- ▶ `kexec vmlinux --append="root=/dev/hda3"`
- ▶ `kexec bzImage --append="ip=on" --initrd=initramfs.cpio.gz`

Kexec Examples

Below are a few examples on how to use Kexec to reboot into. . .

Linux

- ▶ `kexec vmlinuz --append="root=/dev/hda3"`
- ▶ `kexec bzImage --append="ip=on" --initrd=initramfs.cpio.gz`

Xen

Kexec Examples

Below are a few examples on how to use Kexec to reboot into. . .

Linux

- ▶ `kexec vmlinux --append="root=/dev/hda3"`
- ▶ `kexec bzImage --append="ip=on" --initrd=initramfs.cpio.gz`

Xen

- ▶ `kexec -t multiboot-x86 /xen.gz`

Kexec Examples

Below are a few examples on how to use Kexec to reboot into. . .

Linux

- ▶ `kexec vmlinux --append="root=/dev/hda3"`
- ▶ `kexec bzImage --append="ip=on" --initrd=initramfs.cpio.gz`

Xen

- ▶ `kexec -t multiboot-x86 /xen.gz
--append="/xen.gz com1=115200,8n1,0x3f8"`

Kexec Examples

Below are a few examples on how to use Kexec to reboot into. . .

Linux

- ▶ `kexec vmlinuz --append="root=/dev/hda3"`
- ▶ `kexec bzImage --append="ip=on" --initrd=initramfs.cpio.gz`

Xen

- ▶ `kexec -t multiboot-x86 /xen.gz`
`--append="/xen.gz com1=115200,8n1,0x3f8"`
`--module="/vmlinuz console=ttyS0,115200 ip=on"`

Kexec Examples

Below are a few examples on how to use Kexec to reboot into. . .

Linux

- ▶ `kexec vmlinuz --append="root=/dev/hda3"`
- ▶ `kexec bzImage --append="ip=on" --initrd=initramfs.cpio.gz`

Xen

- ▶ `kexec -t multiboot-x86 /xen.gz`
`--append="/xen.gz com1=115200,8n1,0x3f8"`
`--module="/vmlinuz console=ttyS0,115200 ip=on"`
`--module="/initramfs.cpio.gz"`

Kexec Overview

Kexec is a combination of kernel code and user space code:

- ▶ Linux kernel support available through `CONFIG_KEXEC`.
- ▶ `kexec-tools` provides the user space tool `kexec`.
 - ▶ <http://www.xmission.com/~ebiederm/files/kexec/>

Kexec Overview

Kexec is a combination of kernel code and user space code:

- ▶ Linux kernel support available through `CONFIG_KEXEC`.
- ▶ `kexec-tools` provides the user space tool `kexec`.
 - ▶ <http://www.xmission.com/~ebiederm/files/kexec/>

Supported architectures:

- ▶ i386, x86_64, PowerPC/PPC, s390, SH are all in Linux-2.6.17.
- ▶ ia64 support is currently under development.

Kexec Overview

Kexec is a combination of kernel code and user space code:

- ▶ Linux kernel support available through `CONFIG_KEXEC`.
- ▶ `kexec-tools` provides the user space tool `kexec`.
 - ▶ <http://www.xmission.com/~ebiederm/files/kexec/>

Supported architectures:

- ▶ i386, x86_64, PowerPC/PPC, s390, SH are all in Linux-2.6.17.
- ▶ ia64 support is currently under development.

Development:

- ▶ Discussions take place on the fastboot mailing list.
 - ▶ <https://lists.osdl.org/mailman/listinfo/fastboot>
- ▶ Many patches available for `kexec-tools`.

Outline

Introduction to Kexec

- What is Kexec?

- Kexec Examples

- Kexec Overview

Introduction to Kdump

- What is Kdump?

- Kdump Kernels

- The Crash Utility

Xen Porting Effort

- Kexec under Xen

- Kdump under Xen

- The Dumpread Tool

- Partial Dumps

- Current Status

What is Kdump?

Kdump is a Kexec-based crash dumping solution.

What is Kdump?

Kdump is a Kexec-based crash dumping solution.

What is a crash dump then?

What is Kdump?

Kdump is a Kexec-based crash dumping solution.

What is a crash dump then? A crash dump is similar to a core dump:

What is Kdump?

Kdump is a Kexec-based crash dumping solution.

What is a crash dump then? A crash dump is similar to a core dump:

- ▶ A core dump represents the contents of a process.
 - ▶ User space register contents and virtual memory.
- ▶ A crash dump represents the contents of the kernel.
 - ▶ Register contents and *physical* memory.

What is Kdump?

Kdump is a Kexec-based crash dumping solution.

What is a crash dump then? A crash dump is similar to a core dump:

- ▶ A core dump represents the contents of a process.
 - ▶ User space register contents and virtual memory.
- ▶ A crash dump represents the contents of the kernel.
 - ▶ Register contents and *physical* memory.

Kdump is used to extract a crash dump from a crashed machine.

What is Kdump?

Kdump is a Kexec-based crash dumping solution.

What is a crash dump then? A crash dump is similar to a core dump:

- ▶ A core dump represents the contents of a process.
 - ▶ User space register contents and virtual memory.
- ▶ A crash dump represents the contents of the kernel.
 - ▶ Register contents and *physical* memory.

Kdump is used to extract a crash dump from a crashed machine.

The `crash` utility is later on used to analyze the crash dump.

Kdump Kernels

A Kdump enabled setup requires two kernels.

Kdump Kernels

A Kdump enabled setup requires two kernels.

- ▶ Primary kernel:
 - ▶ Regular Linux kernel.

- ▶ Secondary kernel:
 - ▶ “Crash kernel” which is used to retrieve the crash dump.
 - ▶ Started by the primary kernel when a panic occurs.

Kdump Kernels

A Kdump enabled setup requires two kernels.

- ▶ Primary kernel:
 - ▶ Regular Linux kernel.
 - ▶ Booted with “crashkernel=” to reserve a physical memory window.
- ▶ Secondary kernel:
 - ▶ “Crash kernel” which is used to retrieve the crash dump.
 - ▶ Started by the primary kernel when a panic occurs.

Kdump Kernels

A Kdump enabled setup requires two kernels.

- ▶ Primary kernel:
 - ▶ Regular Linux kernel.
 - ▶ Booted with “crashkernel=” to reserve a physical memory window.
- ▶ Secondary kernel:
 - ▶ “Crash kernel” which is used to retrieve the crash dump.
 - ▶ Started by the primary kernel when a panic occurs.
 - ▶ Runs in the reserved physical address window.

Kdump Kernels

A Kdump enabled setup requires two kernels.

- ▶ Primary kernel:
 - ▶ Regular Linux kernel.
 - ▶ Booted with “crashkernel=” to reserve a physical memory window.
 - ▶ Configured with `CONFIG_KEXEC=y`.
- ▶ Secondary kernel:
 - ▶ “Crash kernel” which is used to retrieve the crash dump.
 - ▶ Started by the primary kernel when a panic occurs.
 - ▶ Runs in the reserved physical address window.

Kdump Kernels

A Kdump enabled setup requires two kernels.

- ▶ Primary kernel:
 - ▶ Regular Linux kernel.
 - ▶ Booted with “crashkernel=” to reserve a physical memory window.
 - ▶ Configured with `CONFIG_KEXEC=y`.
- ▶ Secondary kernel:
 - ▶ “Crash kernel” which is used to retrieve the crash dump.
 - ▶ Started by the primary kernel when a panic occurs.
 - ▶ Runs in the reserved physical address window.
 - ▶ Configured with `CONFIG_CRASH_DUMP=y`.
 - ▶ `CONFIG_PHYSICAL_START` needs to match crashkernel option.

Kdump Kernels

A Kdump enabled setup requires two kernels.

- ▶ Primary kernel:
 - ▶ Regular Linux kernel.
 - ▶ Booted with “`crashkernel=`” to reserve a physical memory window.
 - ▶ Configured with `CONFIG_KEXEC=y`.
- ▶ Secondary kernel:
 - ▶ “Crash kernel” which is used to retrieve the crash dump.
 - ▶ Started by the primary kernel when a panic occurs.
 - ▶ Runs in the reserved physical address window.
 - ▶ Configured with `CONFIG_CRASH_DUMP=y`.
 - ▶ `CONFIG_PHYSICAL_START` needs to match `crashkernel` option.
 - ▶ The file `/proc/vmcore` contains the crash dump.

Kdump Kernels

A Kdump enabled setup requires two kernels.

- ▶ Primary kernel:
 - ▶ Regular Linux kernel.
 - ▶ Booted with “`crashkernel=`” to reserve a physical memory window.
 - ▶ Configured with `CONFIG_KEXEC=y`.
- ▶ Secondary kernel:
 - ▶ “Crash kernel” which is used to retrieve the crash dump.
 - ▶ Started by the primary kernel when a panic occurs.
 - ▶ Runs in the reserved physical address window.
 - ▶ Configured with `CONFIG_CRASH_DUMP=y`.
 - ▶ `CONFIG_PHYSICAL_START` needs to match `crashkernel` option.
 - ▶ The file `/proc/vmcore` contains the crash dump.

A patched `kexec-tools` is required to use Kdump.

The Crash Utility

The `crash` utility is used to analyze crash dumps.

The Crash Utility

The `crash` utility is used to analyze crash dumps.

By this time the crash dump file `/proc/vmcore` has been copied to some place where we have access to it.

The Crash Utility

The `crash` utility is used to analyze crash dumps.

By this time the crash dump file `/proc/vmcore` has been copied to some place where we have access to it.

The `crash` utility can then be used on the crash dump file to . . .

- ▶ Extract dmesg from the primary kernel.
- ▶ List running processes.
- ▶ Backtrace and debug.
- ▶ . . .

The Crash Utility

The `crash` utility is used to analyze crash dumps.

By this time the crash dump file `/proc/vmcore` has been copied to some place where we have access to it.

The `crash` utility can then be used on the crash dump file to . . .

- ▶ Extract dmesg from the primary kernel.
- ▶ List running processes.
- ▶ Backtrace and debug.
- ▶ . . .

More information about `crash` can be found at:

http://people.redhat.com/anderson/crash_whitepaper/

Outline

Introduction to Kexec

- What is Kexec?

- Kexec Examples

- Kexec Overview

Introduction to Kdump

- What is Kdump?

- Kdump Kernels

- The Crash Utility

Xen Porting Effort

- Kexec under Xen

- Kdump under Xen

- The Dumpread Tool

- Partial Dumps

- Current Status

Kexec under Xen

The Xen port of Kexec allows you to reboot the machine from dom0. . .

Kexec under Xen

The Xen port of Kexec allows you to reboot the machine from dom0. . .
. . . in the same way that you reboot using Kexec under Linux.

Kexec under Xen

The Xen port of Kexec allows you to reboot the machine from dom0. in the same way that you reboot using Kexec under Linux.

- ▶ Kexec under Xen reboots the *physical* machine.
 - ▶ This means that the hypervisor and all domains go away.
- ▶ Kexec can reboot into Xen or into a Linux kernel.

Kexec under Xen

The Xen port of Kexec allows you to reboot the machine from dom0...
... in the same way that you reboot using Kexec under Linux.

- ▶ Kexec under Xen reboots the *physical* machine.
 - ▶ This means that the hypervisor and all domains go away.
- ▶ Kexec can reboot into Xen or into a Linux kernel.

The `kexec-tools` used under Xen are the same as for Linux.

Kdump under Xen

Kdump under Xen is similar to the standard Linux implementation.

Kdump under Xen

Kdump under Xen is similar to the standard Linux implementation.

- ▶ Both dom0 kernel panic and hypervisor panic are supported.

Kdump under Xen

Kdump under Xen is similar to the standard Linux implementation.

- ▶ Both dom0 kernel panic and hypervisor panic are supported.
 - ▶ dom0 loads the secondary “crash kernel”.
 - ▶ The secondary kernel starts if dom0 panics *or* Xen panics.

Kdump under Xen

Kdump under Xen is similar to the standard Linux implementation.

- ▶ Both dom0 kernel panic and hypervisor panic are supported.
 - ▶ dom0 loads the secondary “crash kernel”.
 - ▶ The secondary kernel starts if dom0 panics *or* Xen panics.
- ▶ The physical memory range is reserved in the hypervisor.

Kdump under Xen

Kdump under Xen is similar to the standard Linux implementation.

- ▶ Both dom0 kernel panic and hypervisor panic are supported.
 - ▶ dom0 loads the secondary “crash kernel”.
 - ▶ The secondary kernel starts if dom0 panics *or* Xen panics.
- ▶ The physical memory range is reserved in the hypervisor.
 - ▶ Range is reserved using Xen command line options `kdump_megabytes` and `kdump_megabytes_base`.

Kdump under Xen

Kdump under Xen is similar to the standard Linux implementation.

- ▶ Both dom0 kernel panic and hypervisor panic are supported.
 - ▶ dom0 loads the secondary “crash kernel”.
 - ▶ The secondary kernel starts if dom0 panics *or* Xen panics.
- ▶ The physical memory range is reserved in the hypervisor.
 - ▶ Range is reserved using Xen command line options `kdump_megabytes` and `kdump_megabytes_base`.

Remember `/proc/vmcore`?

Kdump under Xen

Kdump under Xen is similar to the standard Linux implementation.

- ▶ Both dom0 kernel panic and hypervisor panic are supported.
 - ▶ dom0 loads the secondary “crash kernel”.
 - ▶ The secondary kernel starts if dom0 panics *or* Xen panics.
- ▶ The physical memory range is reserved in the hypervisor.
 - ▶ Range is reserved using Xen command line options `kdump_megabytes` and `kdump_megabytes_base`.

Remember `/proc/vmcore`? It is used in the case of Xen too.

Kdump under Xen

Kdump under Xen is similar to the standard Linux implementation.

- ▶ Both dom0 kernel panic and hypervisor panic are supported.
 - ▶ dom0 loads the secondary “crash kernel”.
 - ▶ The secondary kernel starts if dom0 panics *or* Xen panics.
- ▶ The physical memory range is reserved in the hypervisor.
 - ▶ Range is reserved using Xen command line options `kdump_megabytes` and `kdump_megabytes_base`.

Remember `/proc/vmcore`? It is used in the case of Xen too. The secondary “crash” kernel interface is unchanged.

Kdump under Xen

Kdump under Xen is similar to the standard Linux implementation.

- ▶ Both dom0 kernel panic and hypervisor panic are supported.
 - ▶ dom0 loads the secondary “crash kernel”.
 - ▶ The secondary kernel starts if dom0 panics *or* Xen panics.
- ▶ The physical memory range is reserved in the hypervisor.
 - ▶ Range is reserved using Xen command line options `kdump_megabytes` and `kdump_megabytes_base`.

Remember `/proc/vmcore`? It is used in the case of Xen too.

The secondary “crash” kernel interface is unchanged.

This means that the secondary kernel used as Linux “crash kernel” can be reused under Xen without modification.

The Dumpread Tool

The `dumpread` tool extracts information from Xen crash dumps.

The Dumpread Tool

The `dumpread` tool extracts information from Xen crash dumps.

Supported crash dump operations:

The Dumpread Tool

The `dumpread` tool extracts information from Xen crash dumps.

Supported crash dump operations:

- ▶ Extract the hypervisor.

The Dumpread Tool

The `dumpread` tool extracts information from Xen crash dumps.

Supported crash dump operations:

- ▶ Extract the hypervisor.
- ▶ Get a list of domains.

The Dumpread Tool

The `dumpread` tool extracts information from Xen crash dumps.

Supported crash dump operations:

- ▶ Extract the hypervisor.
- ▶ Get a list of domains.
- ▶ `dom0` and `domU` extraction.

The Dumpread Tool

The `dumpread` tool extracts information from Xen crash dumps.

Supported crash dump operations:

- ▶ Extract the hypervisor.
- ▶ Get a list of domains.
- ▶ `dom0` and `domU` extraction.
 - ▶ Extracted domains can then be analyzed with the `crash` utility.

The Dumpread Tool

The `dumpread` tool extracts information from Xen crash dumps.

Supported crash dump operations:

- ▶ Extract the hypervisor.
- ▶ Get a list of domains.
- ▶ `dom0` and `domU` extraction.
 - ▶ Extracted domains can then be analyzed with the `crash` utility.

`Dumpread` fully supports `i386` and `i386/PAE`.

Basic `x86_64` support is in place too.

The Dumphread Tool

The `dumphread` tool extracts information from Xen crash dumps.

Supported crash dump operations:

- ▶ Extract the hypervisor.
- ▶ Get a list of domains.
- ▶ `dom0` and `domU` extraction.
 - ▶ Extracted domains can then be analyzed with the `crash` utility.

`Dumphread` fully supports `i386` and `i386/PAE`.

Basic `x86_64` support is in place too.

It is possible to use the `crash` utility directly on Xen crash dumps.

Extraction of `domU` is however not supported by `crash`.

The Dumpread Tool

The `dumpread` tool extracts information from Xen crash dumps.

Supported crash dump operations:

- ▶ Extract the hypervisor.
- ▶ Get a list of domains.
- ▶ `dom0` and `domU` extraction.
 - ▶ Extracted domains can then be analyzed with the `crash` utility.

`Dumpread` fully supports `i386` and `i386/PAE`.

Basic `x86_64` support is in place too.

It is possible to use the `crash` utility directly on Xen crash dumps.

Extraction of `domU` is however not supported by `crash`.

More information about `dumpread` can be found at:

<http://people.valinux.co.jp/~moriwaka/dumpread/>

Partial Dumps

In the secondary kernel `/proc/vmcore` points to the crash dump.

Partial Dumps

In the secondary kernel `/proc/vmcore` points to the crash dump.
This file includes all physical memory - excluding the reserved window.

Partial Dumps

In the secondary kernel `/proc/vmcore` points to the crash dump. This file includes all physical memory - excluding the reserved window.

The standard procedure is to copy `/proc/vmcore` somewhere.

Partial Dumps

In the secondary kernel `/proc/vmcore` points to the crash dump. This file includes all physical memory - excluding the reserved window.

The standard procedure is to copy `/proc/vmcore` somewhere.

- ▶ Probably OK during development (on smaller machines).
- ▶ Not suitable for large production machines.

Partial Dumps

In the secondary kernel `/proc/vmcore` points to the crash dump. This file includes all physical memory - excluding the reserved window.

The standard procedure is to copy `/proc/vmcore` somewhere.

- ▶ Probably OK during development (on smaller machines).
- ▶ Not suitable for large production machines.

Wanted: A small tool that extracts parts of `/proc/vmcore`.

Partial Dumps

In the secondary kernel `/proc/vmcore` points to the crash dump. This file includes all physical memory - excluding the reserved window.

The standard procedure is to copy `/proc/vmcore` somewhere.

- ▶ Probably OK during development (on smaller machines).
- ▶ Not suitable for large production machines.

Wanted: A small tool that extracts parts of `/proc/vmcore`.

`/proc/vmcore` provides enough information for such a tool already.

Partial Dumps

In the secondary kernel `/proc/vmcore` points to the crash dump. This file includes all physical memory - excluding the reserved window.

The standard procedure is to copy `/proc/vmcore` somewhere.

- ▶ Probably OK during development (on smaller machines).
- ▶ Not suitable for large production machines.

Wanted: A small tool that extracts parts of `/proc/vmcore`.

`/proc/vmcore` provides enough information for such a tool already.

We are talking about a rather complex tool though - it requires knowledge about kernel and hypervisor data structures. These tend to change over time too...

Current Status

The Xen port of Kexec / Kdump supports the following architectures:

	Kexec	Kexec	Kexec	Kdump	
	vmlinux	bzImage	Xen	vmlinux	Dumpread
i386	OK	OK	OK	OK	OK
i386/PAE	OK	OK	OK	OK	OK
x86_64 ¹	OK	OK	OK	OK	Limited
ia64 ²	WIP	WIP	WIP	WIP	WIP
PPC	-	-	-	-	-

¹Xen, x86_64 and vmlinux requires a patched `kexec-tools`.

²ia64 support not included in mainline Linux yet.

Current Status

The Xen port of Kexec / Kdump supports the following architectures:

	Kexec	Kexec	Kexec	Kdump	
	vmlinux	bzImage	Xen	vmlinux	Dumpread
i386	OK	OK	OK	OK	OK
i386/PAE	OK	OK	OK	OK	OK
x86_64 ¹	OK	OK	OK	OK	Limited
ia64 ²	WIP	WIP	WIP	WIP	WIP
PPC	-	-	-	-	-

Kexec / Kdump status: i386, i386/PAE and x86_64 are fully functional.

¹Xen, x86_64 and vmlinux requires a patched `kexec-tools`.

²ia64 support not included in mainline Linux yet.

Current Status

The Xen port of Kexec / Kdump supports the following architectures:

	Kexec vmlinux	Kexec bzImage	Kexec Xen	Kdump vmlinux	Dumpread
i386	OK	OK	OK	OK	OK
i386/PAE	OK	OK	OK	OK	OK
x86_64 ¹	OK	OK	OK	OK	Limited
ia64 ²	WIP	WIP	WIP	WIP	WIP
PPC	-	-	-	-	-

Kexec / Kdump status: i386, i386/PAE and x86_64 are fully functional.

Dumpread status: x86_64 supports extracting the hypervisor for now.

¹Xen, x86_64 and vmlinux requires a patched `kexec-tools`.

²ia64 support not included in mainline Linux yet.

Current Status

The Xen port of Kexec / Kdump supports the following architectures:

	Kexec vmlinux	Kexec bzImage	Kexec Xen	Kdump vmlinux	Dumpread
i386	OK	OK	OK	OK	OK
i386/PAE	OK	OK	OK	OK	OK
x86_64 ¹	OK	OK	OK	OK	Limited
ia64 ²	WIP	WIP	WIP	WIP	WIP
PPC	-	-	-	-	-

Kexec / Kdump status: i386, i386/PAE and x86_64 are fully functional.

Dumpread status: x86_64 supports extracting the hypervisor for now.

Our team is currently focusing on ia64 support.

¹Xen, x86_64 and vmlinux requires a patched `kexec-tools`.

²ia64 support not included in mainline Linux yet.

Summary

- ▶ The Xen port of Kexec reboots the *entire* physical machine.
- ▶ Kdump under Xen triggers a crash dump from Xen *and* dom0.
- ▶ i386 and x86_64 are ready *now*. ia64 is under development.

Any questions?