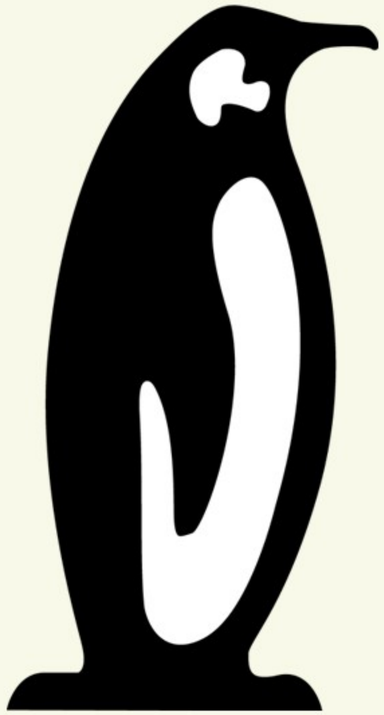


Linux Plumbers Conference

Richmond, Virginia | November 13-15, 2023

Linux
Plumbers
Conference | Richmond, VA | Nov. 13-15, 2023

IBM

Kernel handling of CPU and memory hot un/plug events for crash

- Sourabh Jain
sourabhjain@linux.ibm.com

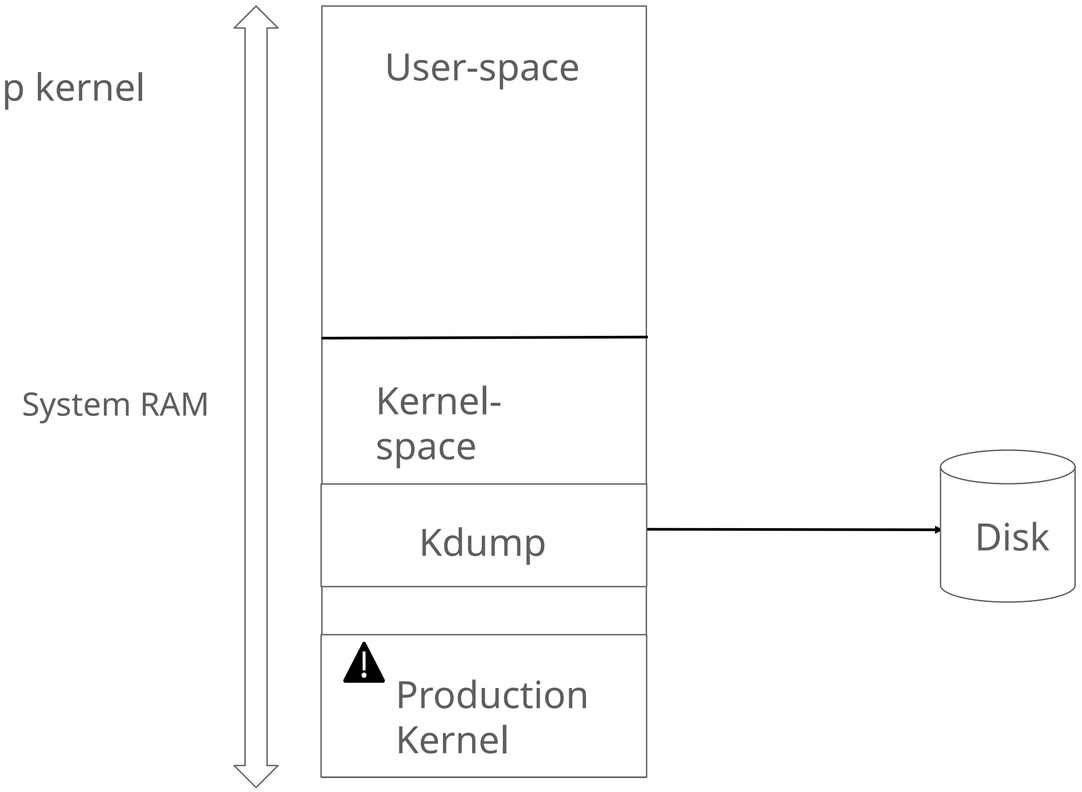
Agenda

- Kdump overview
- Impact of CPU or Memory hotplug on Kdump
- Existing solution and its shortcomings
- Proposed solution
- Implementation details
- Steps to add architecture support

Kdump overview

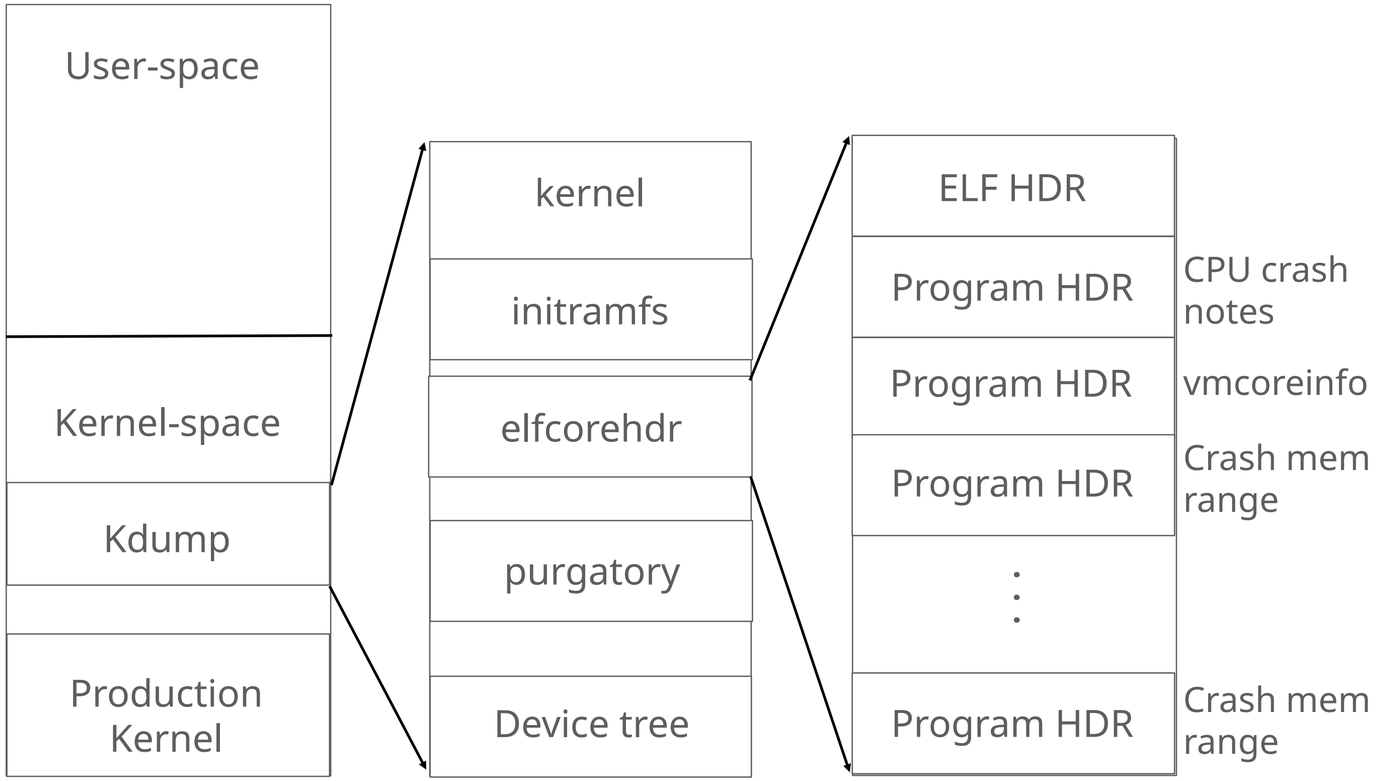
Linux kernel feature for capturing kernel crash dump

- Kdump is loaded into a reserved area
- On Production kernel crash, boot the kdump kernel
- Export kernel dump as /proc/vmcore
- Save the kernel dump to disk



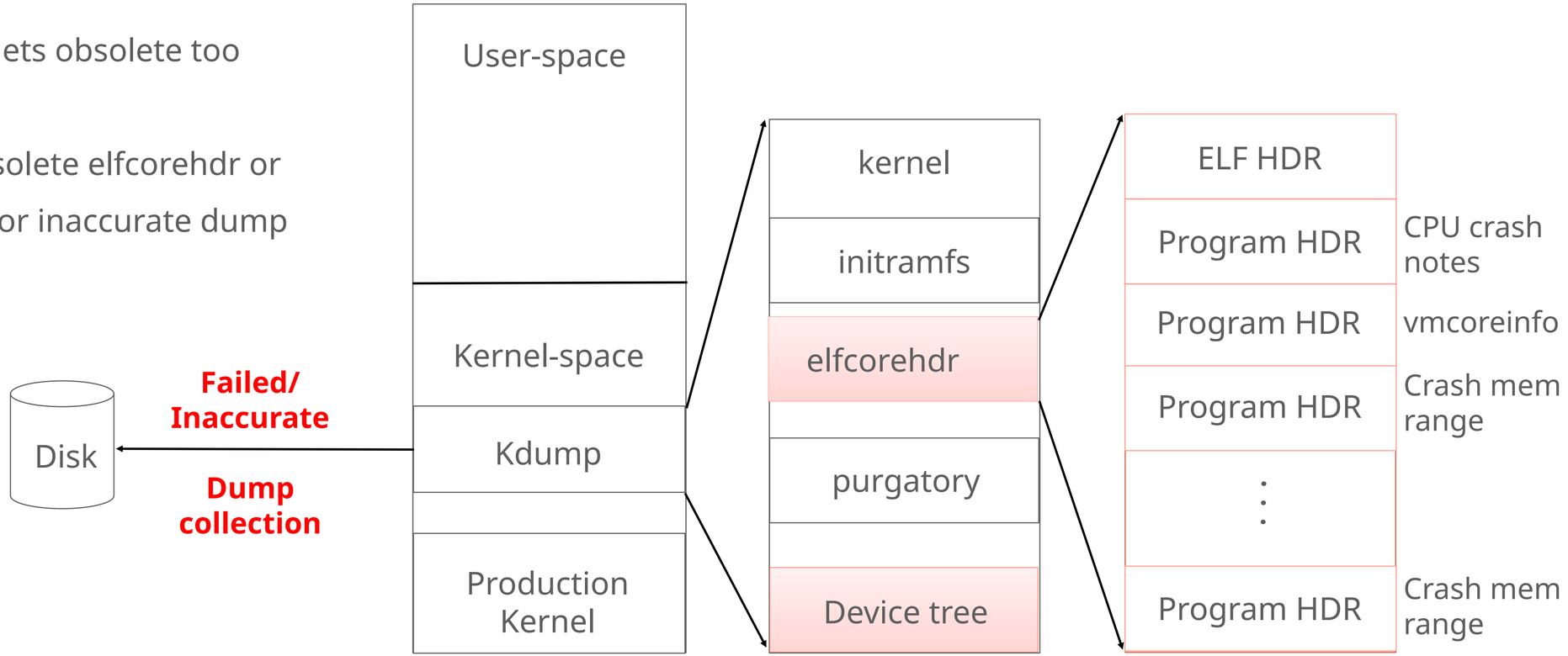
Kdump overview

- Kdump has multiple components/segments, including the kernel, initramfs, elfcorehdr, purgatory, etc
- Elfcorehdr shares dump image information between kernels
- Purgatory is an ELF relocatable object that runs between the production kernel and the kdump kernel
- There can be additional architecture-specific kdump components, for example, device tree on PowerPC architecture



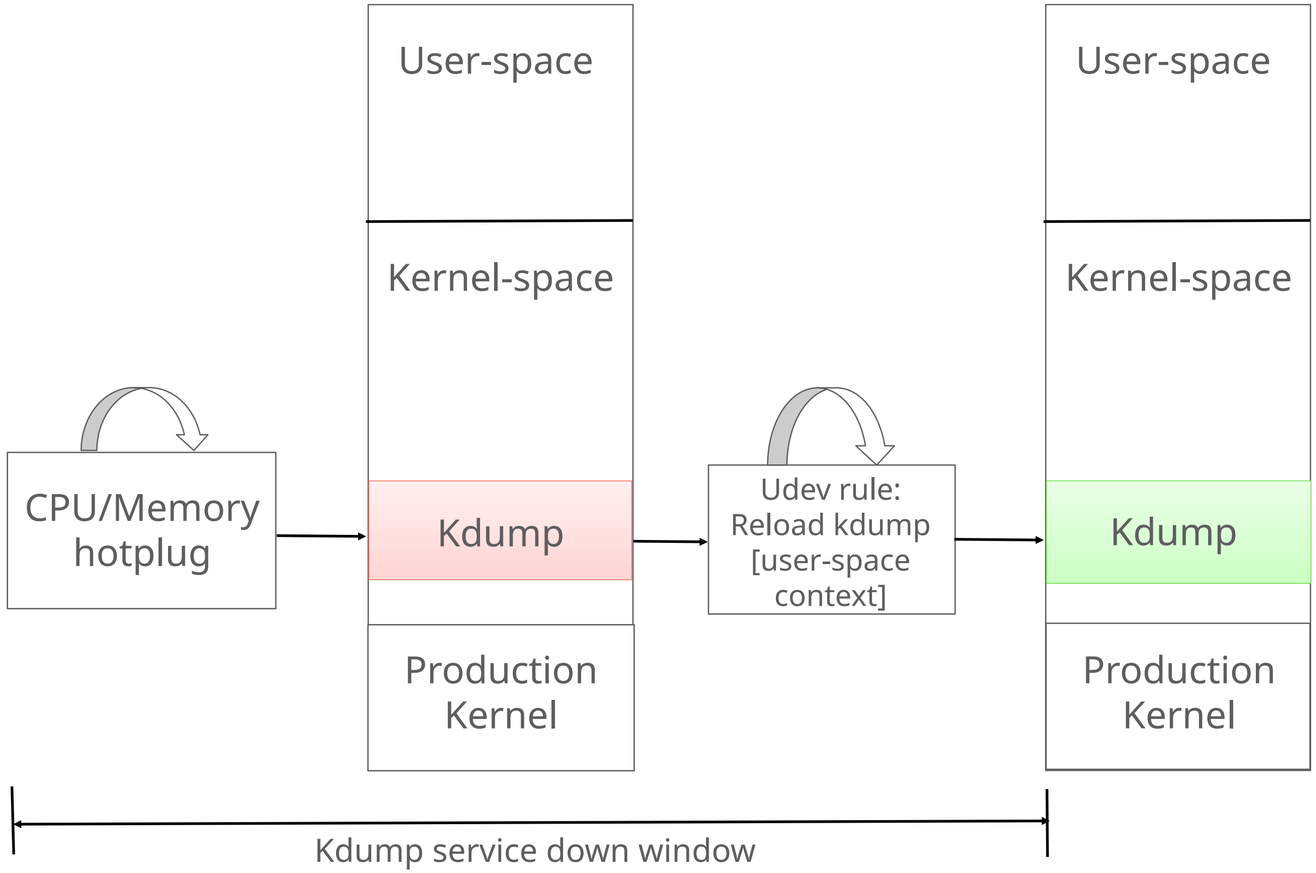
Impact of CPU and Memory hotplug on Kdump

- On CPU or Memory hotplug events, elfcorehdr gets obsolete
- On PowerPC, device tree gets obsolete too
- Dump collection using obsolete elfcorehdr or device tree leads to failed or inaccurate dump collection



Existing solution

- Monitor CPU and Memory hotplug events in user-space using udev rules
- Prepare and load all kdump components on every CPU/Memory hotplug events



… and its shortcomings

- Loading all kdump components for each hotplug event is inefficient
- Impacts serviceability, leaves a large window where kdump service is down
- Situation is worse when hotplug events happen in bulk
- udev rules are prone to race condition

Proposed solution

- Handle CPU/Memory hotplug events in the kernel
- Selectively recreate/update relevant kdump components in kernel itself

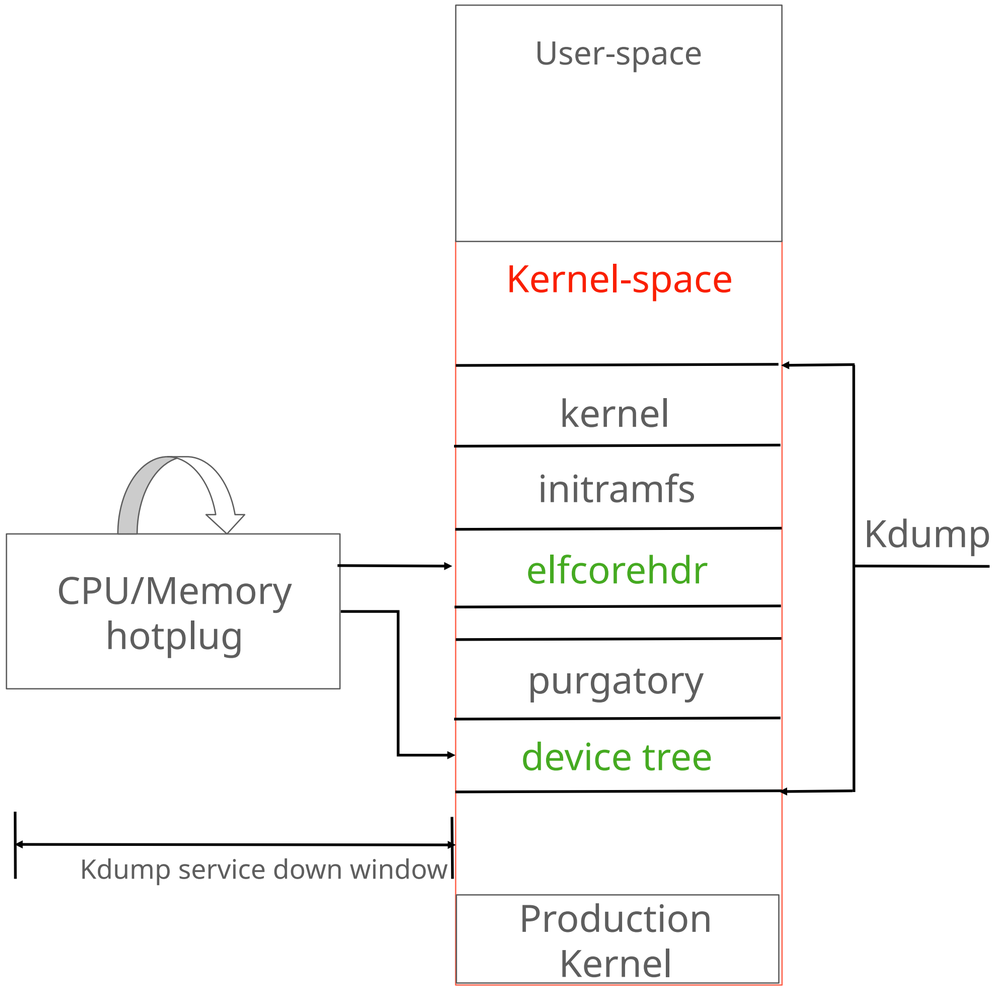
Key gain

- Time consuming kdump service reload is avoided
- Minimal or NO kdump service downtime due CPU/memory hotplug events

Results..

Hotplug type/ solution	Existing solution	Proposed solution
CPU hotplug	~1sec	~4ms
Memory hotplug	~1sec	~0.03ms

PowerPC single core/LMB



Implementation details:

There are two system calls to load kdump

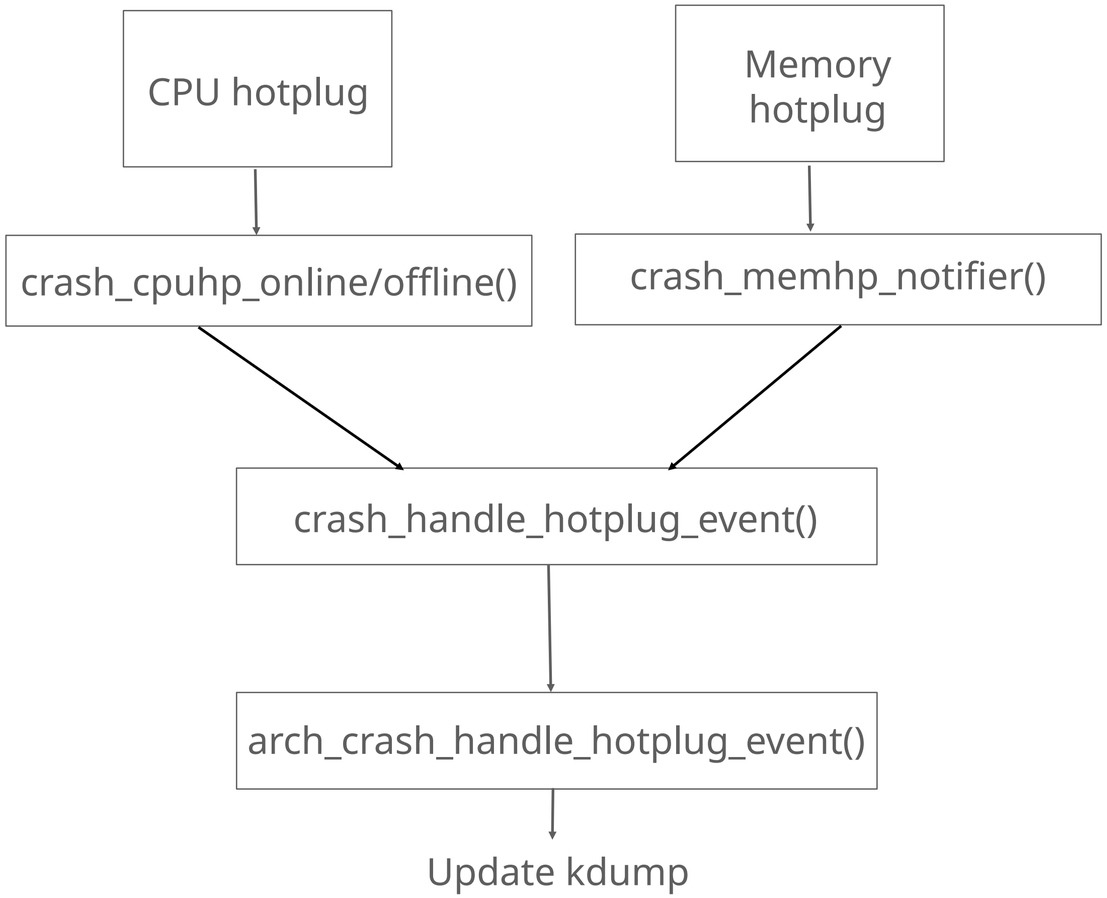
- kexec_file_load: kdump is prepared in kernel (when –s option is passed to kexec tool)
- kexec_load: kdump is prepared in user-space (when –c option is passed to kexec tool)

Implementation details: kernel

- Introduced a crash hotplug handler to handle CPU/Memory add/remove events
- Registered callbacks to get notified on CPU/Memory add/remove events
- Following sysfs nodes are added to advertise this feature to user-space

  /sys/devices/system/cpu/crash_hotplug

  /sys/devices/system/memory/crash_hotplug
- Skipped SHA calculation for a kdump component [elfcorehdr]
- CONFIG_CRASH_HOTPLUG added to enable/disable this feature



commit 24726275612140af6b1c0afc7c6611ad66233207
Author: Eric DeVolder <eric.devolder@oracle.com>
Date: Mon Aug 14 17:44:40 2023 -0400

crash: add generic infrastructure for crash hotplug support

Implementation details: user-space [kexec-tools]

Following changes done in kexec-tools to enable this feature for kexec_load system call

- Introduced a new option *"–-hotplug"* to use this feature
- Elfcorehdr prepared with additional buffer space to hold more resources
- New kexec flag KEXEC_UPDATE_ELFCOREHDR added to let kernel know that it is safe to update elfcorehdr

NOTE: *"--hotplug"* option must be passed to kexec-tools to use this feature with kexec_load system call

Steps to add architecture support

- Implement arch_crash_handle_hotplug_event() to update kdump components on CPU/Memory hotplug events

  Following kdump components updated on x86 and PowerPC, based on the type of hotplug event

  ### X86

  - CPU add/remove: no action
  - Memory add/remove: recreate elfcorehdr

  ### PowerPC

  - CPU remove: no action
  - CPU add: add new CPU to device tree
  - Memory add/remove: recreate elfcorehdr

- Implement arch_crash_hotplug_[cpu | memory]_support() functions to report value `1` to below sysfs nodes

  /sys/devices/system/cpu/crash_hotplug and /sys/devices/system/memory/crash_hotplug

- During kdump load allocate additional memory and skip SHA calculation for elfcorehdr and other necessary kdump components
- Enable ARCH_SUPPORTS_CRASH_HOTPLUG config
- Update kdump udev rules to prevent kdump reloading when this feature is enabled.

  For instance, on Fedora, add the following lines at the top of /usr/lib/udev/rules.d/98-kexec.rules file

  SUBSYSTEM=="cpu", ATTRS{crash_hotplug}=="1", GOTO="kdump_reload_end"

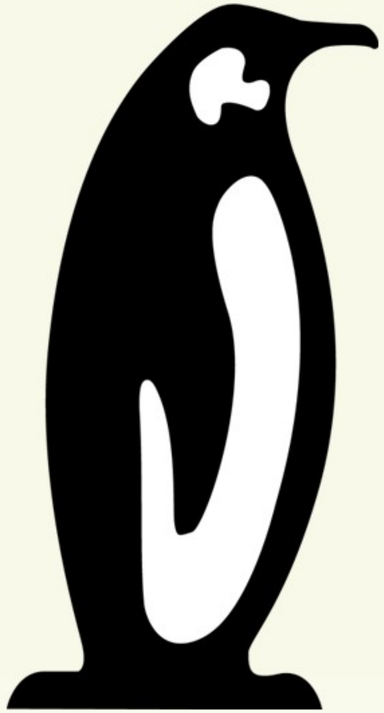
  SUBSYSTEM=="memory", ATTRS{crash_hotplug}=="1", GOTO="kdump_reload_end"

Legal Statement

- This work represents the view of the authors and does not necessarily represent the view of the employers (IBM Corporation).
- IBM and IBM (Logo) are trademarks or registered trademarks of International Business Machines in United States and/or other countries.
- Linux is a registered trademark of Linus Torvalds.
- Other company, product and service names may be trademarks or service marks of others.

IBM

Questions



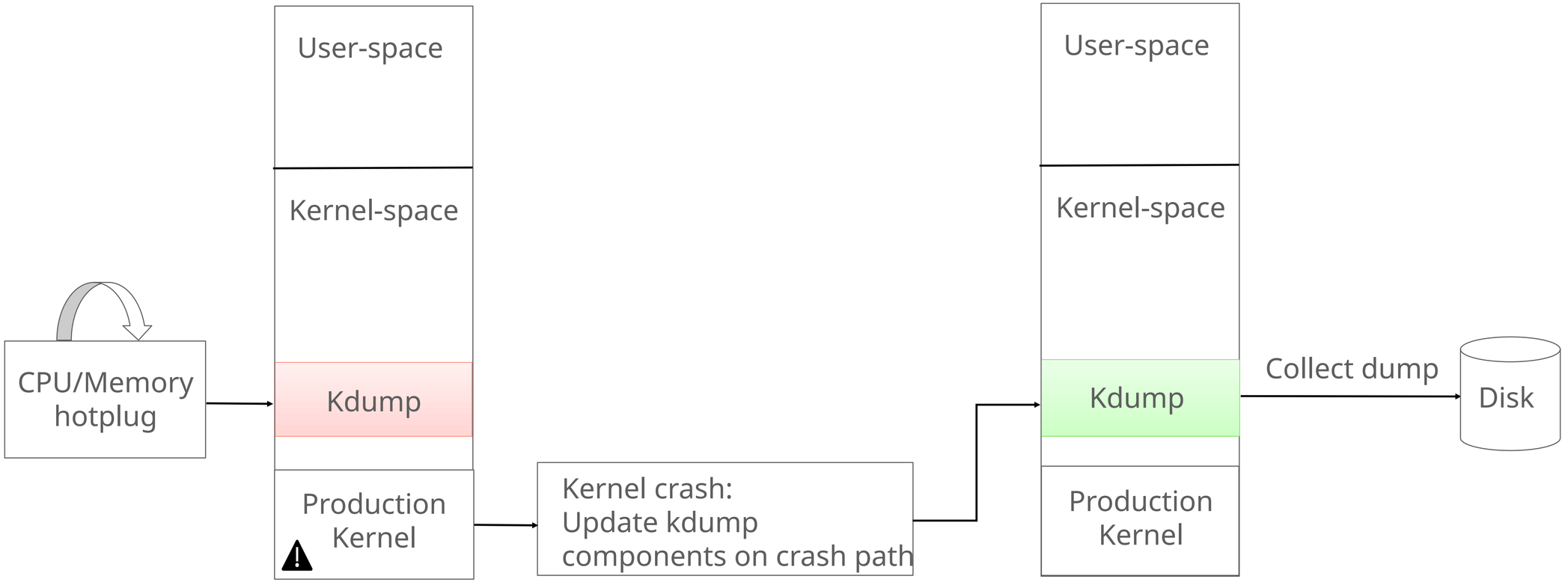
Linux Plumbers Conference

Richmond, Virginia | November 13-15, 2023

Backup

Alternative solution explored

- No kdump update on CPU/Memory hotplug events
- Update kdump components only once on crash path
- Efficient but NOT a reliable solution



Example: Impact of Memory hotplug on Kdump