LinuxCon Japan 2013

shaping tomorrow with you

kdump: Improve reliability on kernel switching

May 30th, 2013

Takao Indoh
Fujitsu Limited

Background

FUJITSU

- Fujitsu has been working for kdump to improve its reliability
- Now kdump is very reliable, but still has problems
  - Recently IOMMU becomes important for KVM
    - PCI passthrough, SR-IOV
  - Kdump fails if IOMMU is enabled
- Need to eradicate remaining problems of kdump

Theme of this presentation

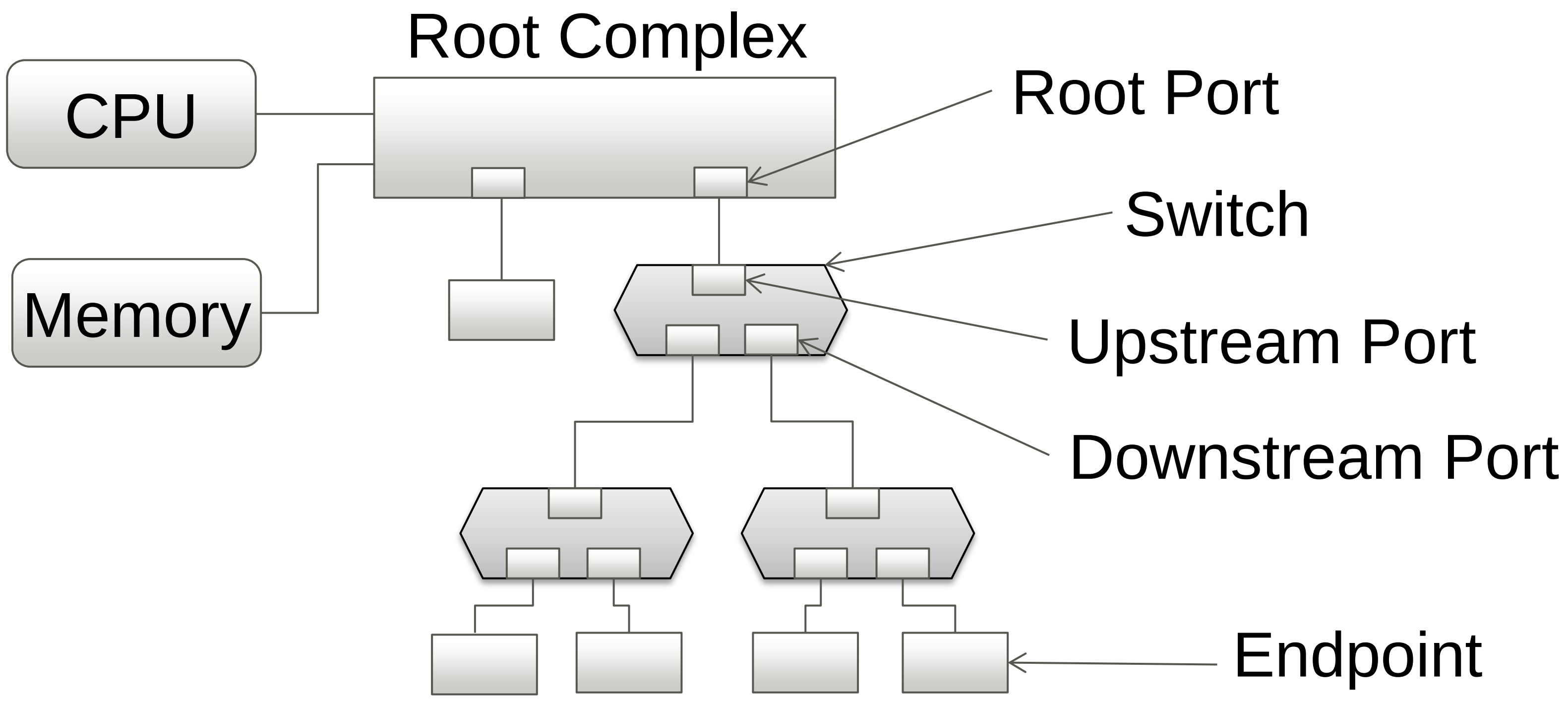
- Theme: How we can improve kdump reliability?
  - Why does kdump fail?
  - How can we fix it?

Table of Contents

Technical terms

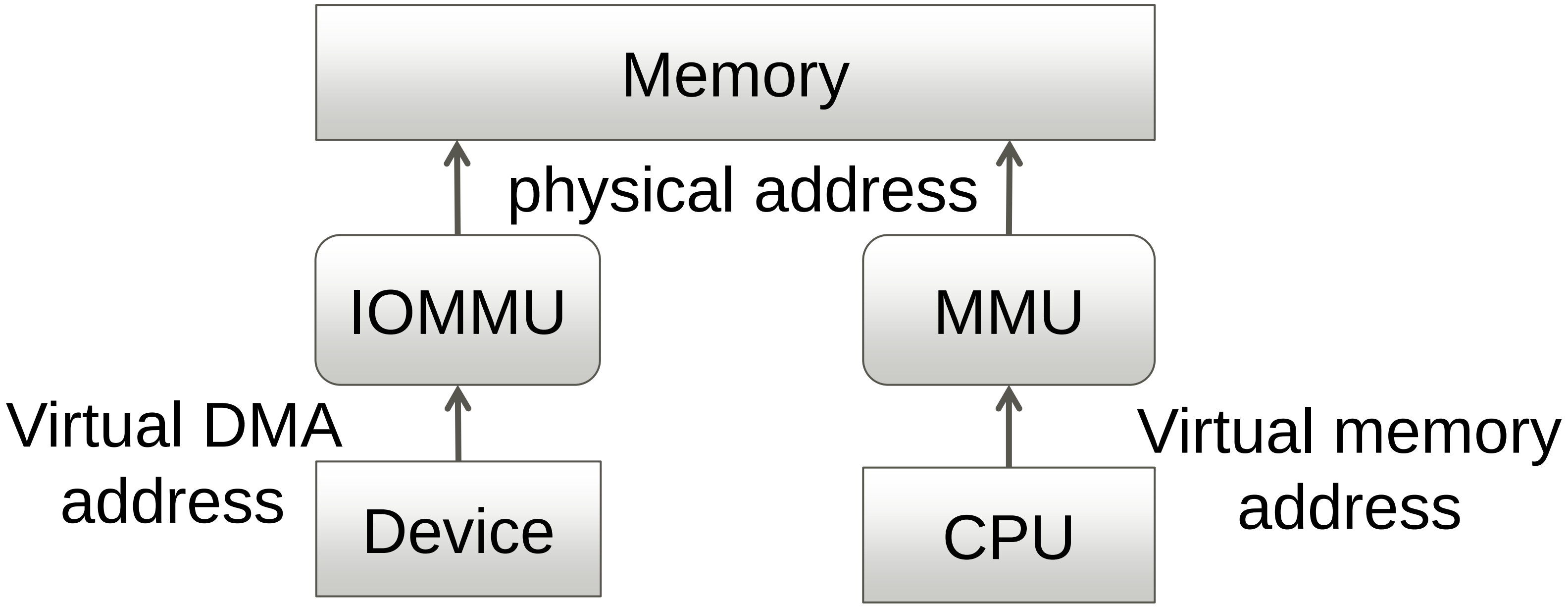
PCI express

- High performance, general purpose I/O interconnect [01]



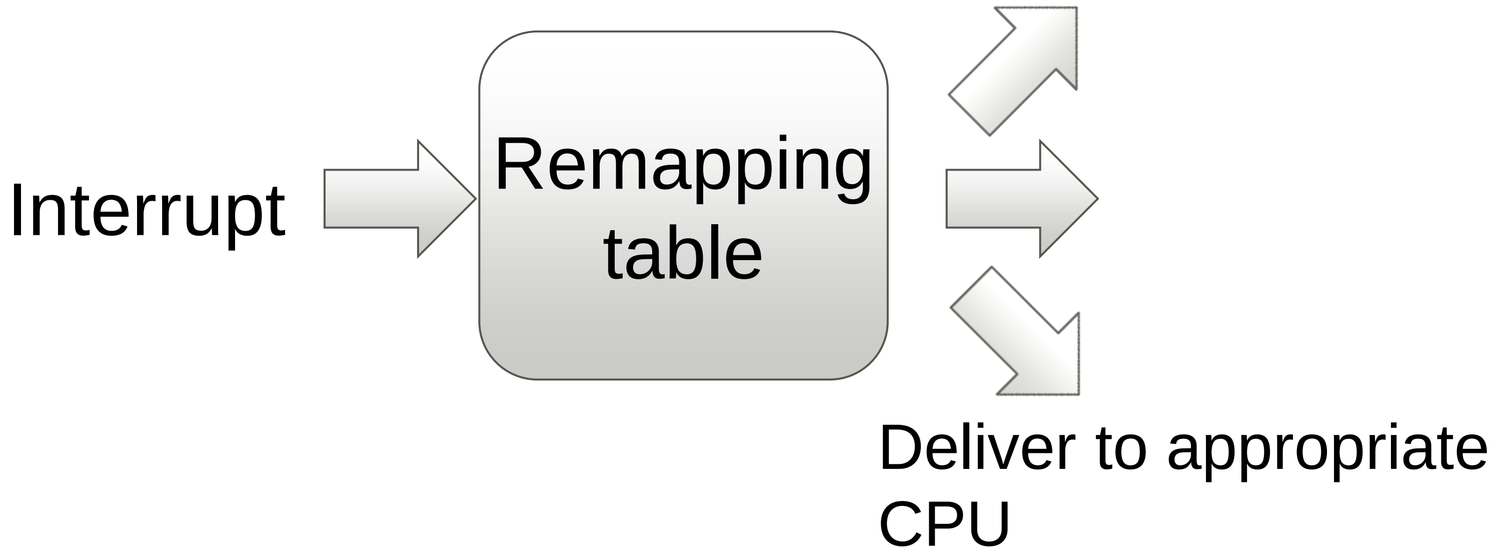
IOMMU

- input/output memory management unit
- Convert device address to physical address on DMA.



Interrupt remapping

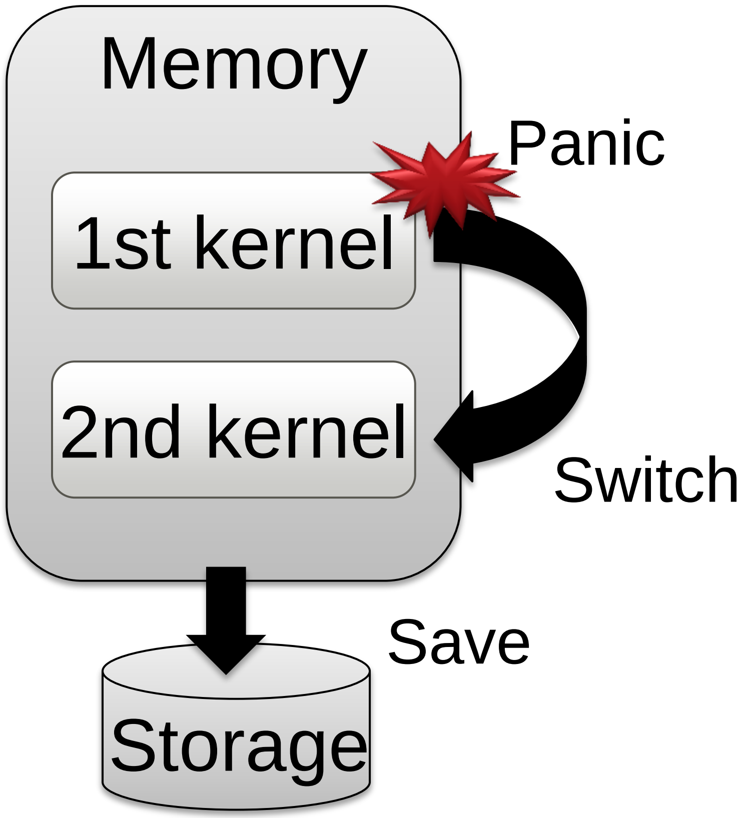
- Intel VT-d feature
- The interrupt-remapping architecture enables system software to control and censor external interrupt requests [02]



Overview of kdump

What is kdump?

- Kdump - Kexec-based Kernel Crash Dumping Mechanism



1. 1st kernel boots up
2. Panic in 1st kernel
3. Switch to 2nd kernel
4. 2nd kernel boots up
5. Save memory to the storage

Advantage of kdump

- More reliable than traditional crash dump
  - Traditional dump
    - Works in insane kernel
    - Dumping in panic situation is not safe
  - Kdump
    - Works in newly booted sane kernel
    - Dumping in almost clean situation is safe!

- But there are some cases that kdump fails, why?

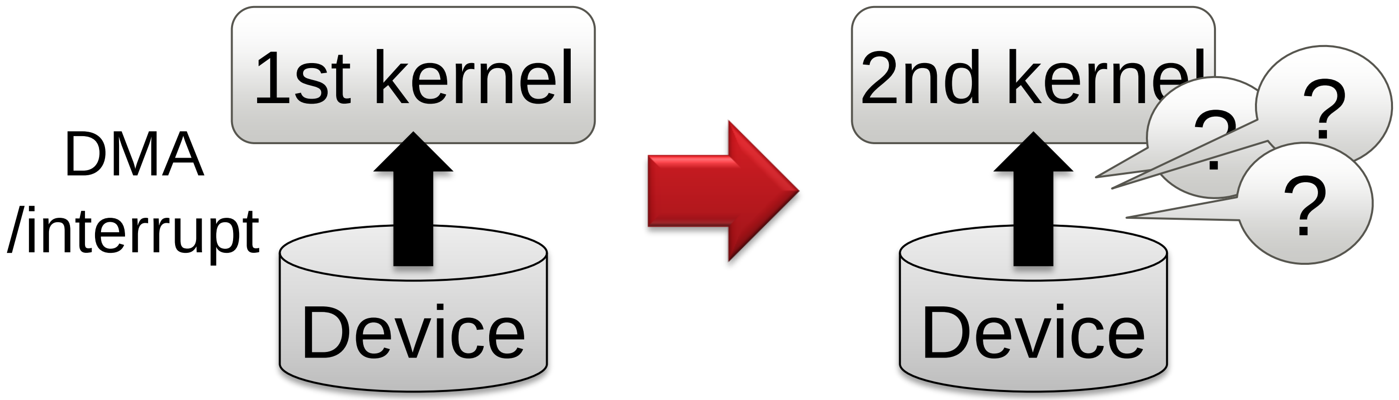
Why does kdump fail?

Reasons of kdump failure

- Human error
  - wrong configuration
  - disk to save memory is full
  - etc.
- Problem on kernel switching
  - Kdump is disturbed by DMA/interrupt from device

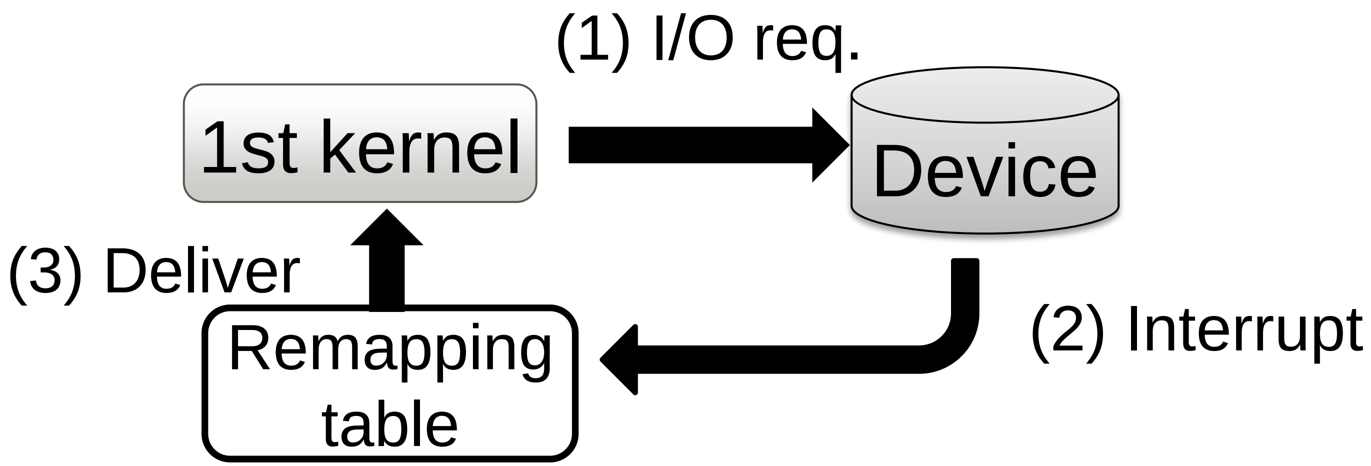
Problem on kernel switching

1. Device is working in first kernel
2. Device is still working after kernel switching, DMA/interrupt continues
3. Second kernel cannot handle it, it causes problem



Example(1)

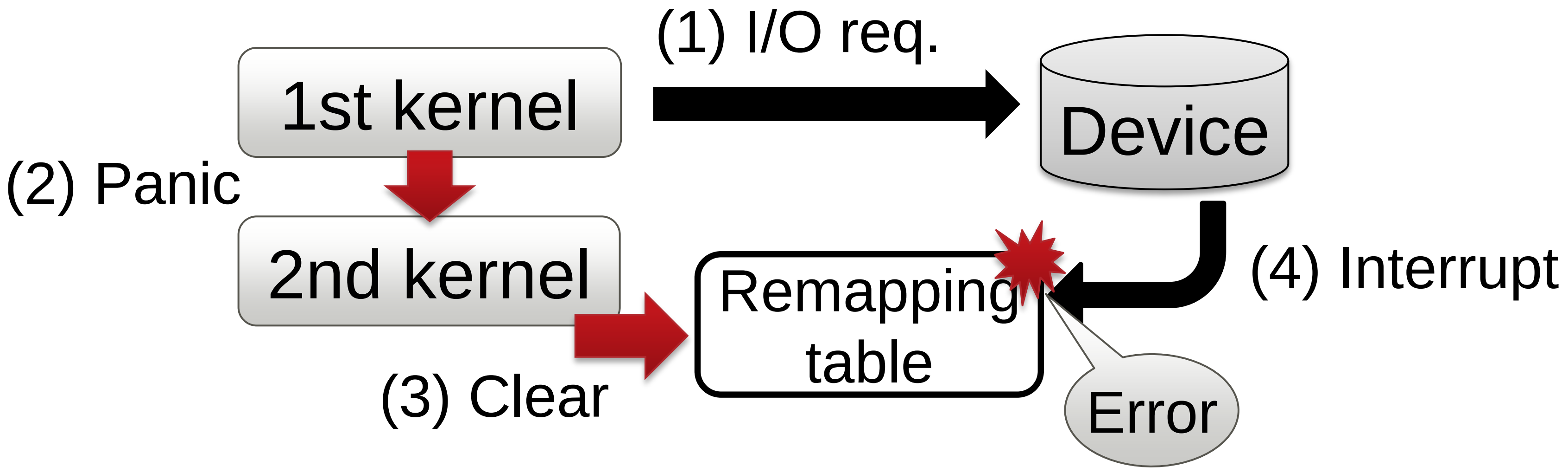
- Interrupt remapping error



- In normal case, interrupt is delivered correctly based on remapping table

Example(1)

- Interrupt remapping error



  - Remapping fault because table was cleared
  - On certain platform system hangs up due to SMI
  - Fixed by PCI quirk on Intel chipset

Example(2)

- IOMMU error
  - Almost same as interrupt remapping error

1. Table for IOMMU to convert address is cleared on second kernel boot
2. DMA by devices causes IOMMU error
3. Driver error/PCI SERR
4. Driver does not work correctly, kdump fails

  - Not fixed yet, I’m working on this.

Back to Theme

- Theme: How we can improve kdump reliability?
  - Why does kdump fail?
  - How can we fix it?

Back to Theme

- Theme: How we can improve kdump reliability?
  - Why does kdump fail?
    - Devices continue working after switching to second kernel
  - How can we fix it?

Back to Theme

- Theme: How we can improve kdump reliability?
  - Why does kdump fail?
    - Devices continue working after switching to second kernel
  - How can we fix it?
    - Stop all devices on second kernel boot

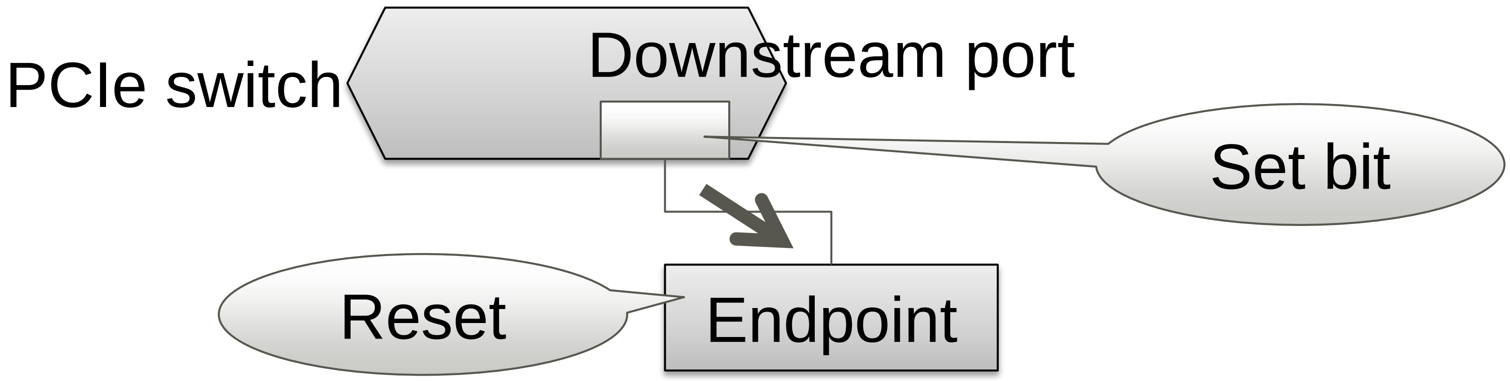
Solution

Stop devices - How?

- Clear bus master bit in PCI config reg.
  - Result: DMA stopped, but IOMMU error after driver loading
  - Driver set this bit again to enable DMA
- PCIe function level reset
  - Result: Same as above
  - This is optional, may not be supported
- PCIe hot reset
  - Result: Works!

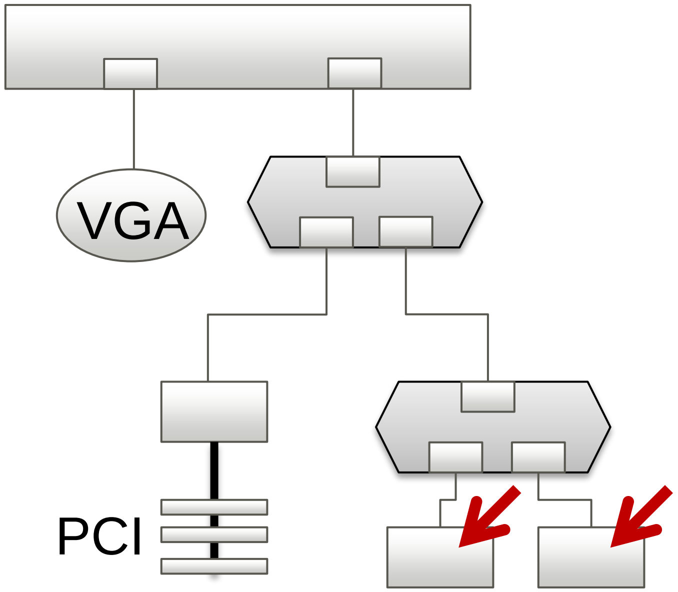
PCIe hot reset

- A reset propagated in-band across a Link using a Physical Layer mechanism [01]
- How to trigger hot reset
  - Setting the Secondary Bus Reset bit of the Bridge Control register in downstream port

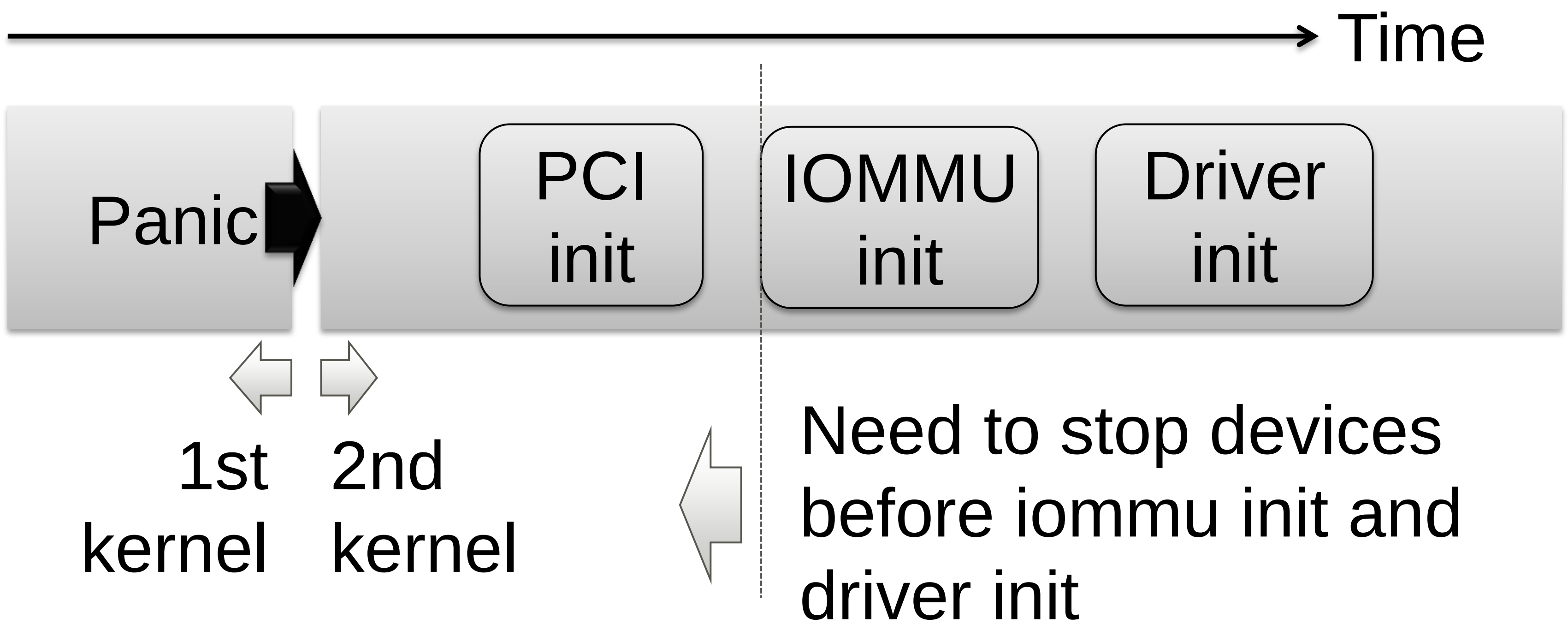


Stop devices – Which one?

- Reset all PCIe endpoints
- Don’t reset:
  - Display devices
    - A monitor blacks out when its controller is reset
  - Legacy PCI devices
    - To simplify algorithm
    - Recently not used

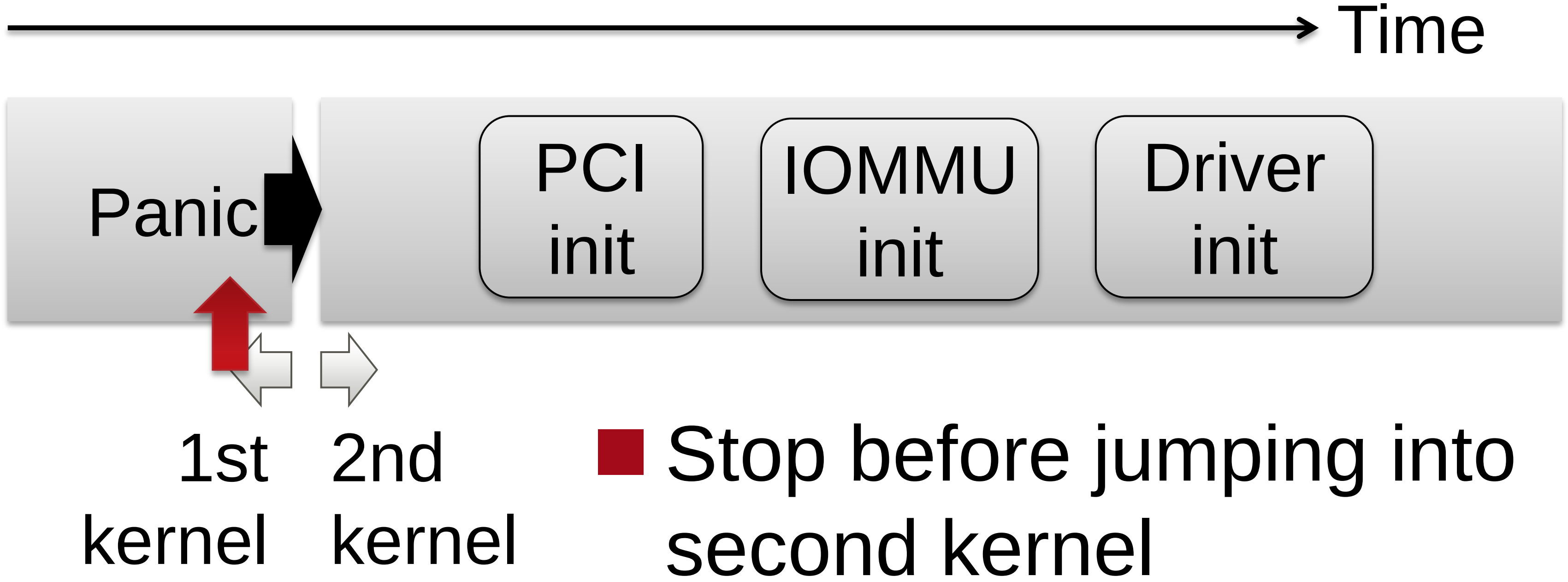


Stop devices - When?



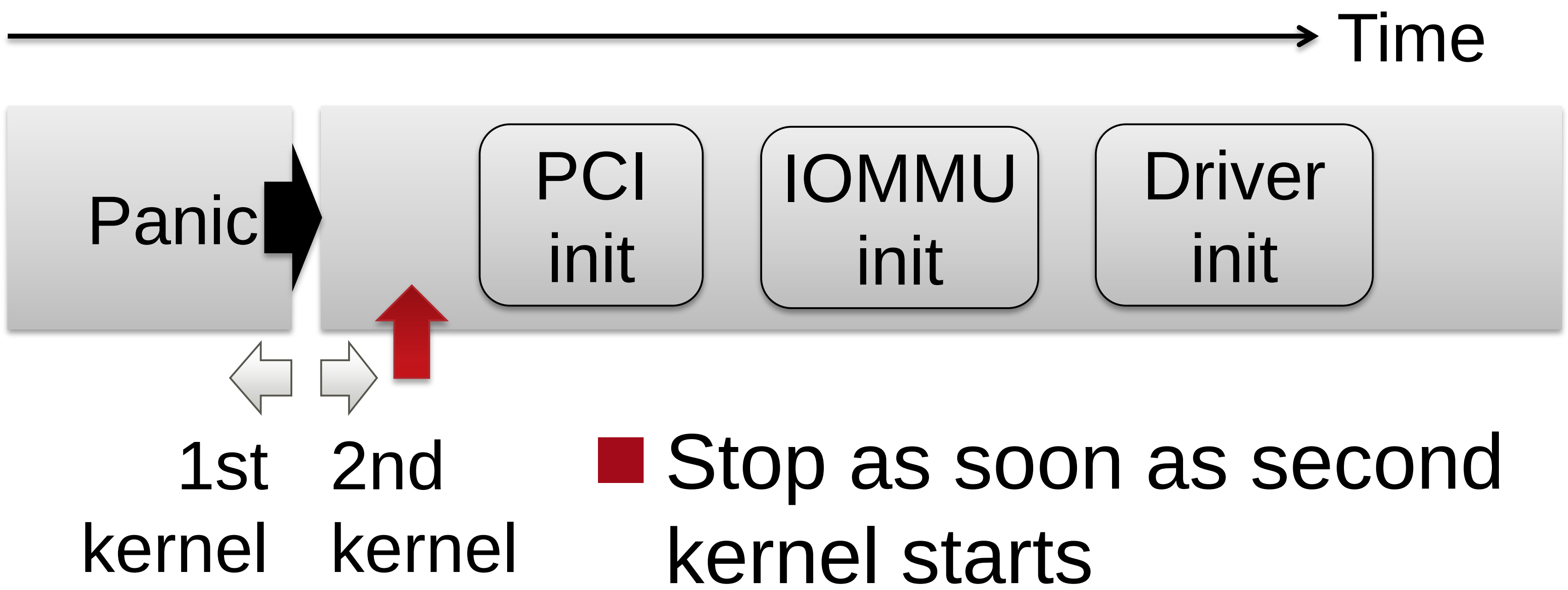
Need to stop devices before iommu init and driver init

Stop devices - When?



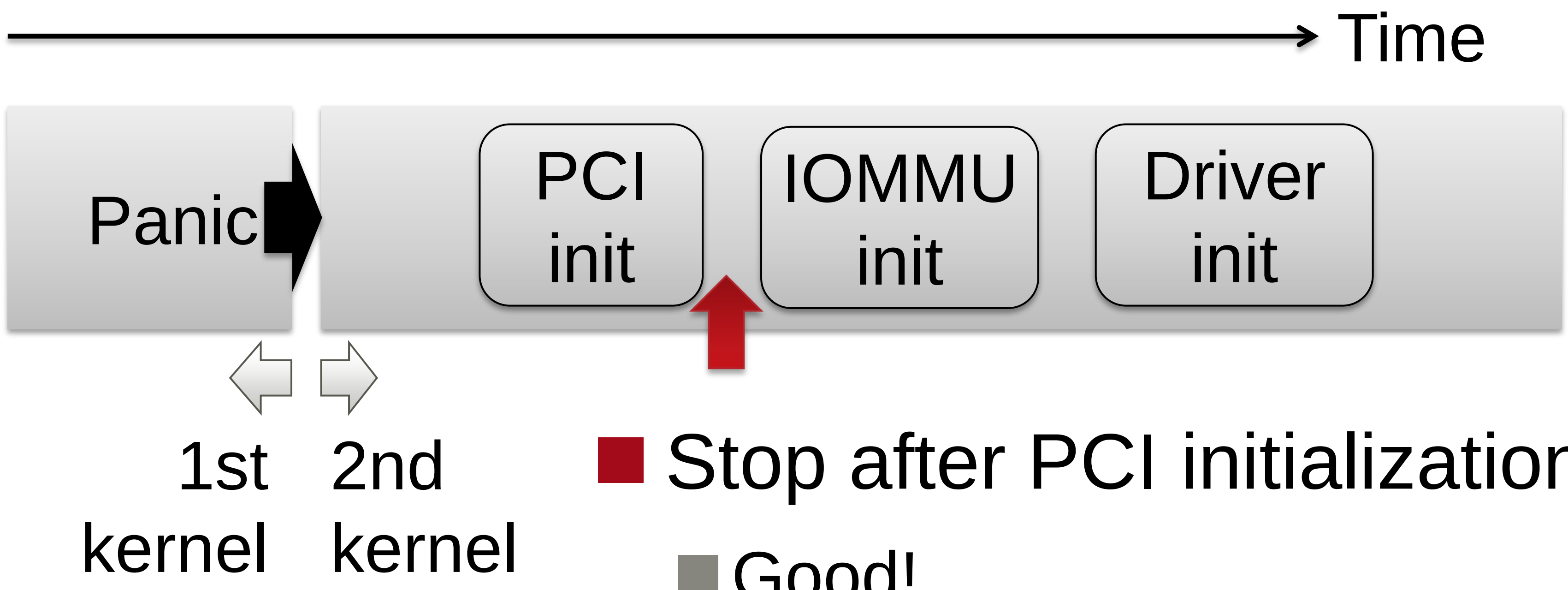
- Stop before jumping into second kernel
  - Not safe

Stop devices - When?



- Stop as soon as second kernel starts
  - No pci_dev struct , difficult to implement
  - MMCONFIG is not ready

Stop devices - When?



- Stop after PCI initialization
  - Good!

Stop devices - When?

- Another idea: Reset devices when error is detected
  - Reset devices in IOMMU error handler
  - Reset devices in AER error handler
- Result: driver does not work correctly because device is reset during driver loading

Summary of solution

- Reset devices by PCIe hot reset
- Reset all PCIe endpoints except display devices and legacy PCI devices
- Reset after PCI initialization

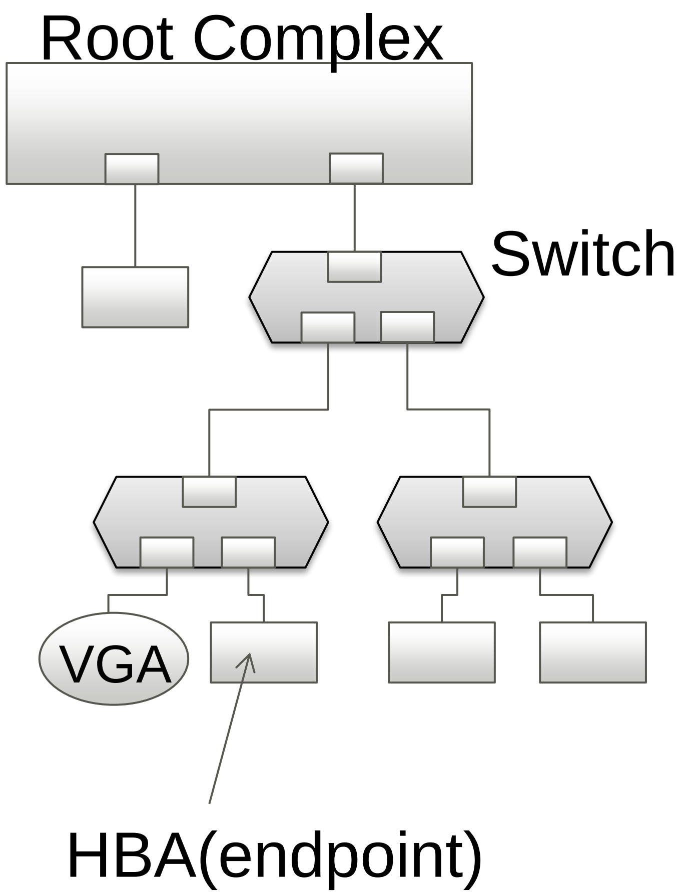
Implementation

Add new initcall handler

Type	Handler	Work
arch_initcall	pci_arch_init	Setup MMCONFIG
subsys_initcall	acpi_init	Setup PCI
fs_initcall_sync	reset_pcie_endpoints	Reset devices
rootfs_initcall	pci_iommu_init	Setup iommu
device_initcall	*	Setup driver

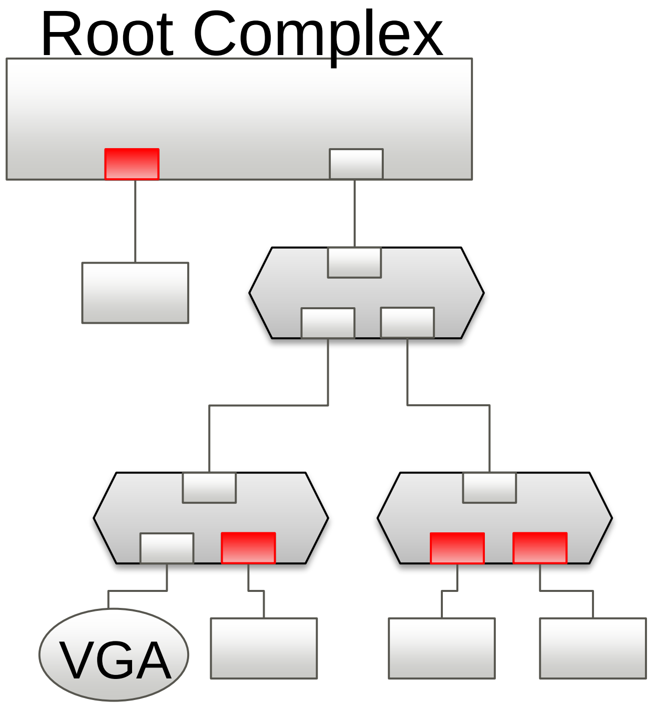
- Initcall is called in this order on boot
- New fs_initcall_sync handler is added to reset devices

reset_pcie_endpoints



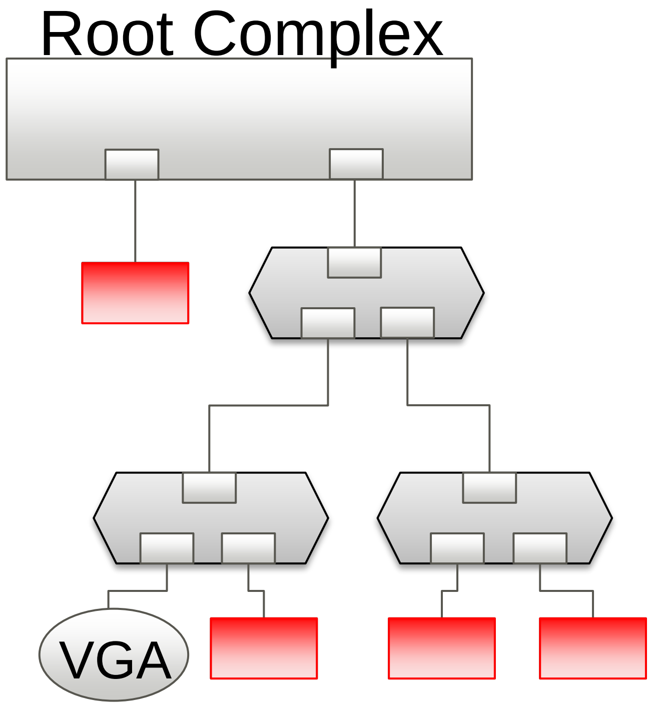
1. Find root port/downstream port whose child is endpoint except display device
2. Save PCI config regs of endpoint
3. Bus reset on the port
4. Restore PCI config regs of endpoint

reset_pcie_endpoints



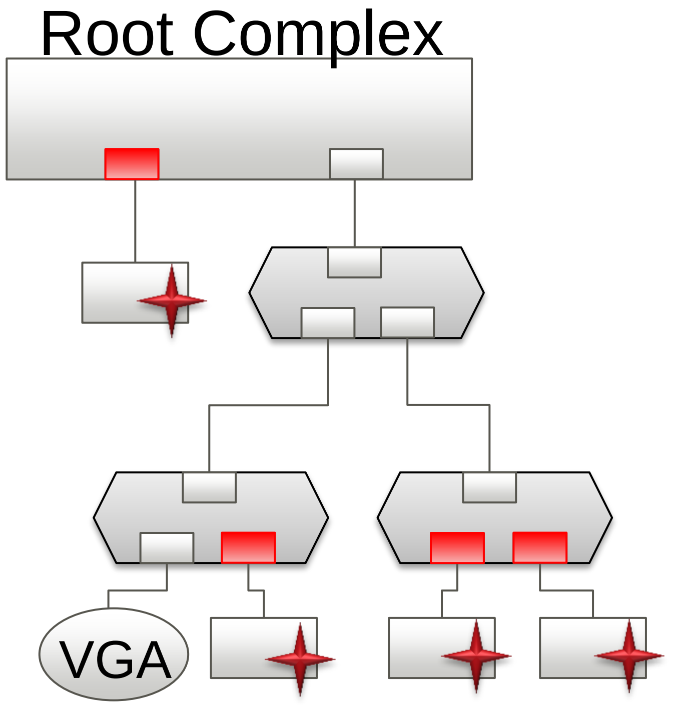
1. Find root port/downstream port whose child is endpoint except display device
2. Save PCI config regs of endpoint
3. Bus reset on the port
4. Restore PCI config regs of endpoint

reset_pcie_endpoints



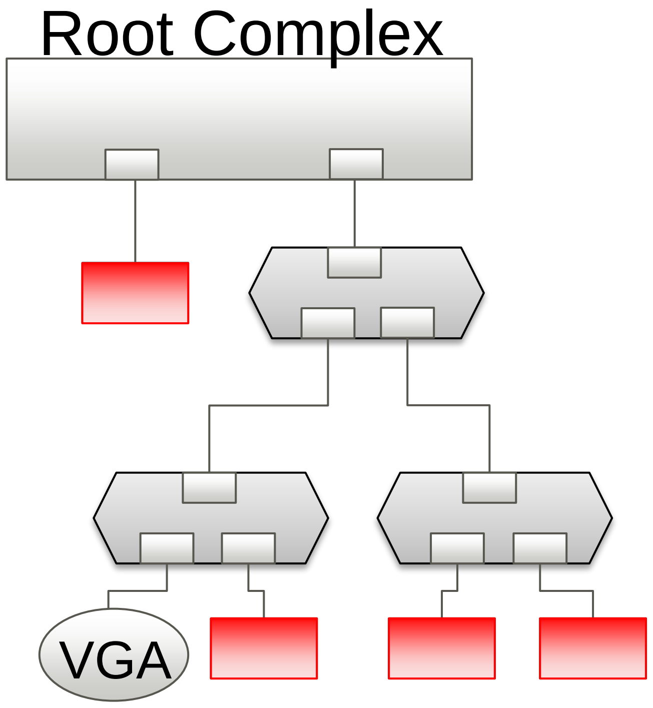
1. Find root port/downstream port whose child is endpoint except display device
2. Save PCI config regs of endpoint
3. Bus reset on the port
4. Restore PCI config regs of endpoint

reset_pcie_endpoints



1. Find root port/downstream port whose child is endpoint except display device
2. Save PCI config regs of endpoint
3. Bus reset on the port
4. Restore PCI config regs of endpoint

reset_pcie_endpoints



1. Find root port/downstream port whose child is endpoint except display device
2. Save PCI config regs of endpoint
3. Bus reset on the port
4. Restore PCI config regs of endpoint

Upstream status

- Patch posted
  - v1 https://patchwork.kernel.org/patch/2482291/
  - v2 https://patchwork.kernel.org/patch/2562841/
- Under discussion
  - Please join in discussion!
  - Please test this patch!

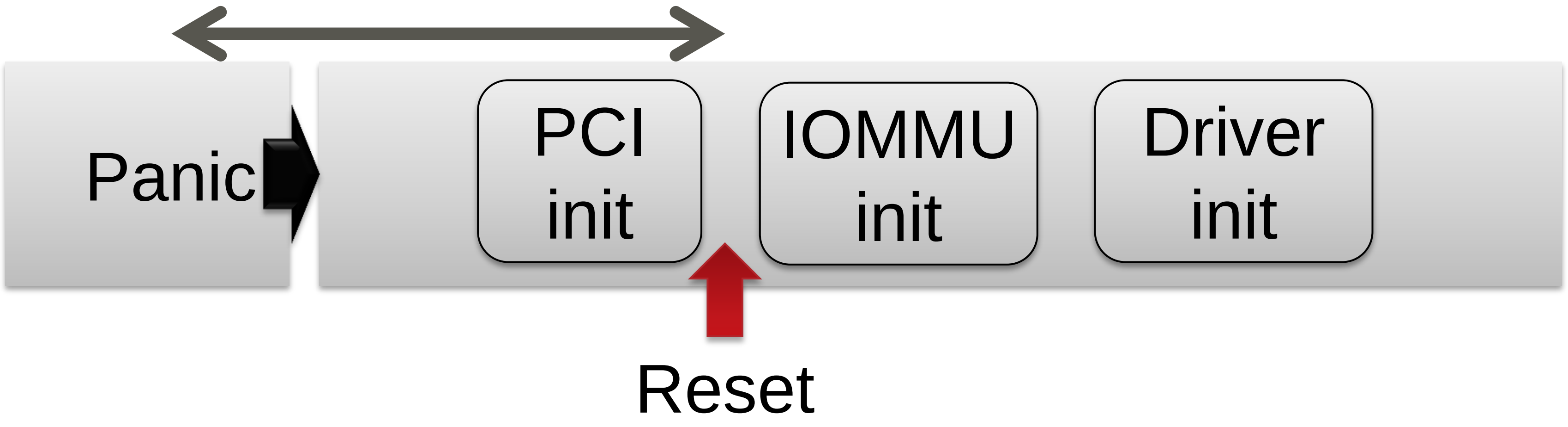
Summary

Summary

- Kdump sometimes fails because devices are working on kernel switching
- Stopping devices is needed to improve kdump reliability
- Resetting devices by PCIe hot reset after PCI initialization is effective to fix this problem
- Posted a patch to upstream, still under discussion

TODO

- More testing
  - HP engineer reported that kernel hangs up after device reset on certain platform [03]
- Improve reset timing
  - Still unstable window



References

FUJITSU

- 01
  - PCI Express® Base Specification Revision 3.0
- 02
  - Intel® Virtualization Technology for Directed I/O
- 03
  - RE: [PATCH] Reset PCIe devices to stop ongoing DMA
  - https://lkml.org/lkml/2013/4/30/211

FUJITSU

shaping tomorrow with you