DDI Interrupt Affinity Interfaces and PCITool Enhancements

Govinda Tatti

Solaris Platform IO Software

Sun Microsystems Inc.

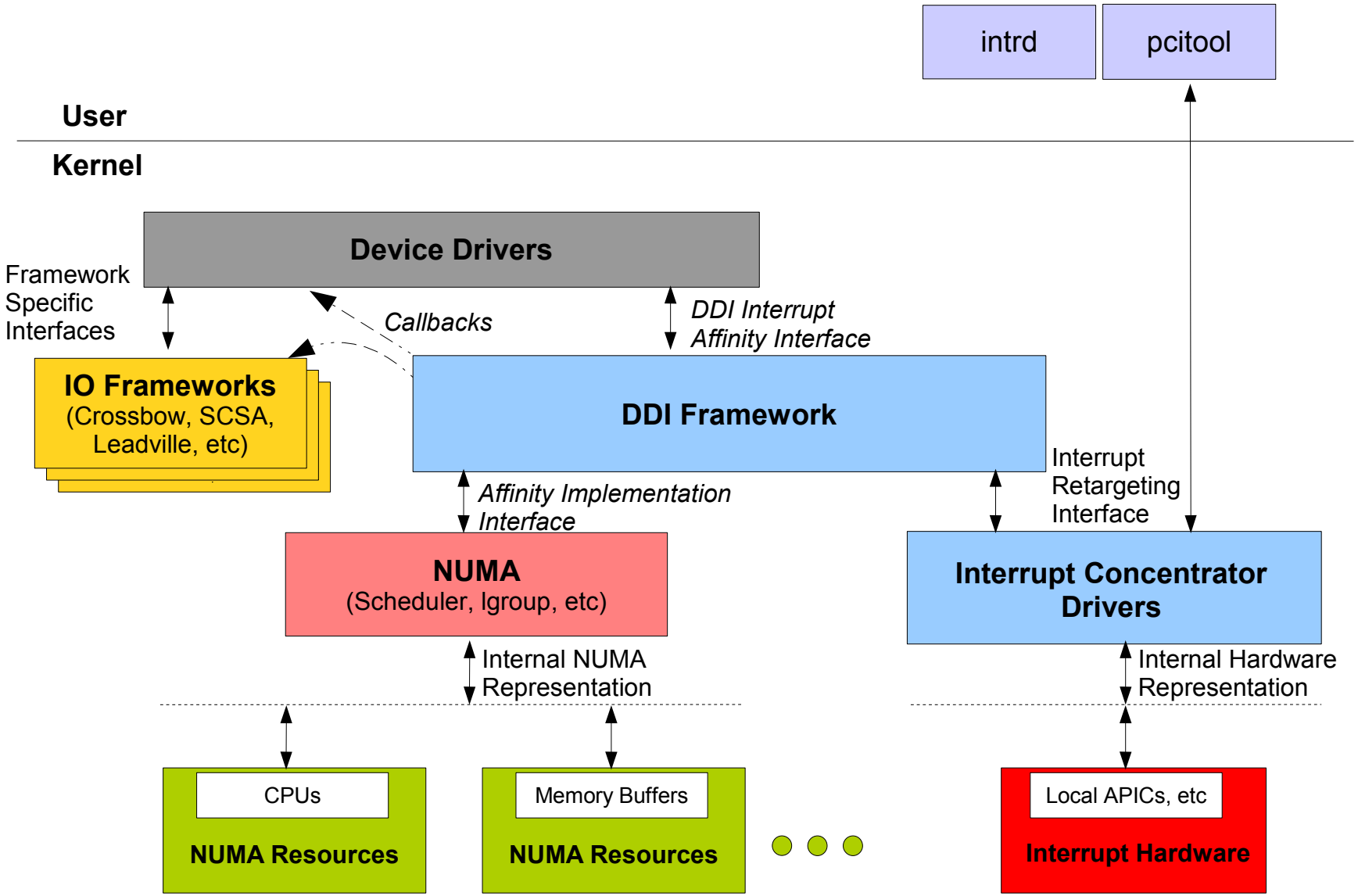
Session Outline

- Background
- Overview
- Affinity Interfaces
- PCITool Enhancements
- MSI/X retarget support for SPARC
- Interface table for ARC review
- Current status
- References

Background

- Project Goals
  - Provide simple API to allow a device driver to get and set the current interrupt target CPU
  - Provide MSI/X retarget option for PCITool users
- Key Requirement
  - Required by Crossbow, Emulex and PAE teams for their performance improvement and publishing benchmark results
- Long term Goals
  - Replace simple interfaces with hint or preference based interfaces
  - DDI interrupt framework and platform specific implementation will be modified to query the NUMA-IO framework for optimal interrupt target CPU

Overview


intrd
pcitool
User
Kernel
Device Drivers
Framework Specific Interfaces
Callbacks
DDI Interrupt Affinity Interface
IO Frameworks (Crossbow, SCSA, Leadville, etc)
DDI Framework
Affinity Implementation Interface
Interrupt Retargeting Interface
NUMA (Scheduler, lgroup, etc)
Interrupt Concentrator Drivers
Internal NUMA Representation
Internal Hardware Representation
CPUs
NUMA Resources
Memory Buffers
NUMA Resources
Local APICs, etc
Interrupt Hardware

Affinity Project - Changes

- A new function (ddi_intr_get_affinity(9f)) to return the interrupt target CPU for a given DDI interrupt handle h
- A new function (ddi_intr_set_affinity(9f)) to set the interrupt target CPU for a given DDI interrupt handle h
- Modify ddi_intr_get_cap(9f) function to return the new capability flag DDI_INTR_FLAG_RETARGETABLE indicating all the interrupts are retargetable for the current interrupt type in use

DDI Affinity Interfaces

```
typedef    processorid_t          ddi_intr_target_t;

int        ddi_intr_get_affinity(ddi_intr_handle_t h,
           ddi_intr_target_t *tgt_p);

int        ddi_intr_set_affinity(ddi_intr_handle_t h,
           ddi_intr_target_t tgt);
```

Affinity interfaces - Constraints

- Set affinity limitations for certain interrupt types
  - Fixed or INTx interrupts could be either exclusive or sharable depending on hardware. Because there is no good way to detect that, the current implementation will refuse any set affinity requests for INTx interrupts
  - On x86 platforms, multiple MSI interrupts of a single PCI function need to be rerouted together since all MSI interrupts share the same MSI address (same CPU number). Hence the current x86 implementation will refuse any set affinity requests for MSI interrupts
- Interrupt handle state
  - On x86 platform, interrupt is only bound to CPU when the handle is in ENABLE state. That means, drivers need to have called ddi_intr_enable(9f) or ddi_intr_block_enable(9f) before they call the affinity interfaces

Affinity interfaces – Constraints (Cont ..)

- CPU offline considerations
  - When a CPU is offlined, all of the interrupts targeting it are re-targeted. OS will pick any set of the surviving CPUs for re-targeting. The OS is under no obligation to maintain drivers' interrupt affinity preferences
  - If the driver is interested in maintaining optimal CPU targeting, it should monitor its interrupt CPU bindings on a regular basis or subscribe to CPU offline/online events
  - Future phase of this project will provide a callback to drivers to report various interrupt retarget events using the callback interface described in PSARC/2008/628

PCITool Enhancements

- pcitool pci@<unit-address> -i <ino#> | all [-r [-c] | -w <cpu#> [-g]] [-v] [-q]
  - Some minor changes to PCITool since the current syntax is not complied with existing uderland guidelines
- pcitool pci@<unit-address> -m <msi#> | all [-r [-c] | -w <cpu#> [-g]] [-v] [-q]
  - Adding a new “-m” option to retrieve and reroute the interrupt target CPU for a given MSI/X on SPARC platforms. This option is not supported on x86

MSI/X retarget support for SPARC

- On SPARC platforms, an interrupt mondo (INO) is mapped to an INTx or Event Queue (EQ). Currently, we can migrate any interrupt mondo associated with INTx or EQ from one CPU to another
- MSI/Xs are mapped to an Event Queue(EQ), but currently we cannot migrate any MSI/Xs from one CPU/EQ to another. Here is the proposal which talks about MSI/X migration

  http://pcie.sfbay/intr/docs/misc/msix_retarget_proposal_sparc.txt

Interface table for ARC review

Interface	Stability	Comments
ddi_intr_target_t	project private	Interrupt target CPU
ddi_intr_get_affinity (9f)	project private	Get interrupt target CPU
ddi_intr_set_affinity (9f)	project private	Set interrupt target CPU
DDI_INTR_FLAG_RETARGETABLE	project private	Return this new flag (RO) to ddi_intr_get_cap(9f) callers if current interrupt type in use is retargetable
pcitool (1m)	project private	Minor syntax changes. Plus, added new -m option for MSI/Xs

Reference

- Interrupt Project Webpage
  - http://pcie.sfbay/intr
- Specs
  - PSARC/2009/340 – DDI Interrupt Affinity Interfaces and PCITool Enhancements
  - SPARC MSI/X Retarget proposal
  - X86 Interrupt retarget proposal
- ARC cases
  - **PSARC/2004/253 Advanced DDI Interrupt Interface**
  - **PSARC/2008/628 Interrupt Resource Management**
- Aliases
  - mailto:ddi-intr-iteam@sun.com - Solaris Interrupt Framework Development Team