IO-MMUs

James Bottomley

SteelEye Technology

18 July 2005

State of Play

- IO-MMU translates addresses between the CPU and the Bus
- Used to bridge the gap when the device on the bus can't address all of memory.
- Also used to reduce or eliminate the need for Scatter-Gather (Virtual Merging).
- IOMMUs come in most enterprise computing systems: PA-RISC, SPARC, ia64 (zx1, altix).
- GARTS (opt in) come in most others: AMD64, PPC.
- Except Intel: x86, EM64T, ia64 (intel chipset).

IOMMU Operation

- Linux API well defined (DMA-API.txt and DMA-mapping.txt).
- Missing items:
  1. Bypass
  2. Descriptor mode setting
- Building Scatter-Gather (SG) lists is done separately by different kernel subsystems (Block, Net, Char devices).
  - Feature lack and Code Duplication.
- Also need solid API for non-IOMMU systems (Intel is extremely prevalent).

Bypass

- Having IOMMU between bus and device can be a waste if the device can see all of memory
  - additional translations slow transfers down.
- Need to lift IOMMU out of the way for capable devices.
- Only way to do this for block devices is to turn off virtual merging.
  - Virtual merging allows a multi-element SG list to be reduced sometimes to a single transfer by doing separate mappings on the scattered pages.

Descriptor Settings

- Most devices have to use wide descriptors to access full 64 bits
- Causes lack of efficiency (twice as long to load descriptors) special address modes to access ¿ 32 bit memory (e.g. PCI DAC cycle).
- Even if the device is 64 bit capable, could be faster to use 32 bit mode.
- Definitely in a system with $<$ 4GB memory, want to use the 32 bit descriptors.
- use `dma_get_required_mask()` for this.

Bouncing

- Most comprehensive is block: can do virtual and physical merging, contains API to describe SG list to driver and to device after IOMMU mapping.
- Problem: Non block devices wishing to use this can't (e.g. net and Char)
- Net has own (separate) SG system
- Every char device has to roll their own
  - Although some silently attach to non-exposed block queues to do this.
- SWIOTLB: essentially a bouncing system that sits right in the device driver courtesy of `dma_map_sg()`.

Systems without IOMMUs

- Still need to allocate memory for kernel operations that falls within the dma_mask.
- Only choices for allocation are `GFP_KERNEL` or `GFP_DMA` (old isa 24 bit mask).
- Quite a few devices (aacraid) have 31 bit masks.
- Even block devices need hacky systems to identify the need to add this flag
  - SCSI `unchecked_isa_dma` flag.
- Could do with a better interface, like `kmalloc_mask()` or `kmalloc_dev()`.