IOMMUs

Muli Ben-Yehuda
Jon D Mason

Poorly presented by
Jimi Xenidis
What is an IOMMU?

- Provides two functions
  - Translation
  - Isolation
IOMMU Translation

- Creates a unique address space
  - Call it an “IO Address Space”
  - Could be same as on host processor
IOMMU Isolation

- Restricts device addressability
  - Desirable for Hypervisors
  - Allows unprivileged domains to have direct device access
Why you want one?

- **Pros**
  - Extends the addressability of a device
  - Scatter/Gather coalescing
  - Device Isolation

- **Cons**
  - Performance?
  - Yet another MMU to manage
What we are working with

- Calgary and DART
  - TCE based
  - Single IO address space per MMU
  - Usually on each host bridge
Its just a frame array

- Easy to understand and access

```c
struct tce {
    int rw:2;
    int dev:6;
    int mfn:24;
};
```

- Easy to change

```c
tce_table[max_io_fn];
invalidate_tce_entry(io_fn);
```
Dom0 Interfaces:

- **IOMMU_DETECTED**
  - Tell Xen what and where the IOMMU is
- **CREATE_IO_SPACE**
  - Associate an TCE space for a device
- **DESTROY_IO_SPACE**
DomU Interfaces

- **X86**
  - `u64 do_iommu_map(
u64 ioaddr, u64 mfn, u32
access, u32 bdf, u32 size)`
  - `int do_iommu_unmap(
u64 ioaddr, u32 bdf, u32 size)`

- **PowerPC**
  - `int tce_put(u32 bdf, u32 idx,
struct_tce tce)`
  - `int tce_stuff(u32 bdf, u32
idx, struct_tce tce, int
count)`
Other IOMMUs

What Muli and Jon had to say

- AMD IOV and Intel VT-d

  - Provides translation and isolation
  - Devices are assigned into a protection domain with a set of I/O page tables defining the allowed memory addresses.
  - Before a DMA transfer begins, the IOMMU intercepts the access and checks its cache (IOTLB) and (if necessary) the I/O page tables for that device, based on the devices Bus/Dev/Func.
  - Can be arranged in a topology of IOMMUs
  - I/O page tables maintained in system memory by host software; with AMD's implementation, the page table format is compatible with the MMU's page table format.