“Greening” Storage Through I/O Virtualization

Richard Solomon
Host Interface Architect
LSI Corporation
PCI Express® I/O Virtualization (IOV)
  - Overview
  - Single Root IOV
  - Multi-Root IOV
  - Address Translation Services
Using IOV with storage to reduce system power
  - Virtualization as a power saver
  - Multi-Root IOV for Blade Servers
PCI Express

I/O Virtualization
IOV Overview

- Makes one device “look” like multiple devices
- Generally motivated by cost
- Seek performance within the cost envelope
From an adapter point of view:

- One physical device looks like multiple devices
- Virtual devices appear completely independent
  - May occupy different PCI memory ranges
  - May have different settings for various PCI Configuration registers
- Need to keep cross-”device” traffic isolated
- Each virtual device *may* be a multi-function device
From a system point of view:

- "System Image" is a real or virtual system of CPU(s), Memory, O/S, I/O, etc
  - Multiples may run on one or more sets of hardware
    - E.g. VMware running Win32 & Linux on a single CPU
    - E.g. Blade server running multi-OS each on a single blade

- Each "System Image" (SI) needs to "see" its own PCI hierarchy
  - Even if NO end devices are actually shared
  - Only its "portion" of shared end devices
Attachment of existing PCIe Base components
- Root Complexes, Switches, Endpoints, and Bridges
- A solution to use a combination of existing base and IOV-aware components:
  - Single Root capabilities are a superset of the PCIe Base specification
  - Multi-Root capabilities are superset of the Single Root capabilities
  - IOV-capable components are backwards-compatible with existing software
  - Although some or all of the new IOV capabilities may not be supported in these circumstances

“Concentric Circles” model
Fits into existing PCI hierarchies today
- Single and Multi-CPU boxes with traditional single point of attachment to PCI
- Same address space – partitioned and allocated “above” the Root Complex
  - Uses RequesterID (formerly the Bus/Device/Function field) in packets to track back to a System Image
- Existing or absolutely minimally changed Root Complex (i.e. chipset) and Switch silicon
- New Endpoint silicon
Presumes existence of a Virtualization Intermediary (VI) aka a Hypervisor

Direct result of “don’t change the chipset!” philosophy

Opens market to lots of existing or simply-derived systems

E.g. might need new BIOS or chipset revision

Shifts substantial burden to software
Multi-Root IOV (MR-IOV)

- Most obvious example is a blade server with a PCIe “backplane”
- New PCIe hierarchy construct - (mini) fabric
  - Logically partitions the hierarchy into multiple Virtual Hierarchies (VHs) all sharing the same physical hierarchy
- Targets “small” systems (16-32 Root Ports likely max)
  - Workgroup saying “Our yardstick is a yardstick” i.e. the typical implementation is a system occupying not more than about 3 feet cubed
  - Architected to allow larger, but not optimized that way
Multi-Root IOV (MR-IOV)

- Existing or absolutely minimally changed Root Complex (i.e. chipset) silicon
- New Switch silicon
  - Allows for use of existing or minimally changed switches in a reduced capacity in certain places
- New Endpoint silicon
- Management model
Address Translation Services (ATS)

- Defines a set of transactions PCIe components can use to exchange & share translated addresses
- PCIe addresses map to system physical addresses based on the “identity” of the agent using them
  - Each SI appears to use the entire address space
  - System’s I/O MMU does translation of “normal” addresses
    - Expensive in performance terms
    - Impossible to size I/O MMU’s TLB or cache for all applications
- ATS-aware devices can translate an address range and bypass I/O MMU
Address Translation Services (ATS)

- New PCIe commands for translation Requests, Completions, and Invalidations
- Modified PCIe TLP header (2 bits) to indicate whether a given address is pre-translated or not
- Implementation is optional even for IOV devices
- Disabled after a reset, and may be enabled by ATS aware system software
- Usage generally implies system “trust” of the device and its driver as ATS-enabled devices can bypass I/O MMU address protections
Endpoint (Device) Impact

- Every System Image (SI) should see its own Virtual Function (VF)
- For $n$ Virtual Functions:
  - The device needs $n$ sets of configuration space
  - The device needs $n$ sets of registers
  - The device *may* need $n$ sets of internal logic
## Example MR-IOV Configuration Map

<table>
<thead>
<tr>
<th>VRn</th>
<th>Configuration Space</th>
<th>B/D/F “RID”</th>
</tr>
</thead>
<tbody>
<tr>
<td>VFn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VFn</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VF2</th>
<th>Configuration Space</th>
<th>B/D/F “RID”</th>
</tr>
</thead>
<tbody>
<tr>
<td>VF2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PF0</th>
<th>Configuration Space</th>
<th>B/D/F “RID”</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BF</th>
<th>Configuration Space</th>
<th>B/D/F “RID”</th>
</tr>
</thead>
<tbody>
<tr>
<td>BF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PFm</th>
<th>Configuration Space</th>
<th>B/D/F “RID”</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFm</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example SR-IOV Address Map

- Configuration Space
  - VF4
    - Configuration Space
    - B/D/F “RID”
  - VF3
    - Configuration Space
    - B/D/F “RID”
  - VF2
    - Configuration Space
    - B/D/F “RID”
  - VF1
    - Configuration Space
    - B/D/F “RID”
  - PF/V/F0
    - Configuration Space
    - B/D/F “RID”

- Memory Space
  - 1MB
  - 1MB
  - 1MB
  - 1MB
  - 1MB
Saving Power with IOV
Server-level virtualization

- Replace $n$ physical servers with a single one
- Reduced chassis and power-supply count
...or does it?
Blade Servers
Blade Servers

- Many blades have 2+ drives for mirrored O/S boot
- Each blade has less “horsepower” due to space, power, and cooling costs of those drives
- 2 drives * 8 blades = 16 drives / chassis!
Blade Server with IOV Boot Disks

- 6 drives per chassis (RAID5 + 1 hot spare) vs 16
- Could double CPU count per blade
  - 28 (7*4) vs 16 (8*2)
Blade Server with IOV Storage

- Replace External Storage array with IOV RAID set(s)
- Replace (small) SAN with IOV RAID set(s)
  - Presume doubled CPU capacity per blade via removal of on-blade drives
Blade Server with IOV Storage
Futures...

MR-IOV over PCI Express Cable

8-16 Core Blade On A Chip

8-16 Core Blade On A Chip

MR-IOV SSD Array

MRA PCIe Switch
Thank You For Attending

“Greening” Storage Through I/O Virtualization