Implementing Hot Plug in NVMe Systems

BIOS, Timeouts, and All-1s

Wesley Yung (Microsemi)
Hot Plug Relationship with USB

Hot Add

Hot Remove
Hot Plug Relationship with NVMe

Hot Add
- Single
- Engaged
- Divorced
- It's Complicated
- Separated
- In a Relationship
- Married

Hot Remove
- Single
- Engaged
- Divorced
- It's Complicated
- Separated
- In a Relationship
- Married
Let’s Talk Hot Add
Why Is Hot Add Complicated?

- On boot, BIOS scans topology
- With no devices connected, BIOS stops at switch DS P2P
- BIOS may reserve a BDF and memory for each DS P2P
Why Is Hot Add Complicated?

- When NVMe is added, OS scans the bus
- If no BDF is reserved and/or no memory is reserved, NVMe drivers won’t load
Solutions to Hot Add

- Expose all hot-pluggable slots in your switch (P2P bridges)

- Program your BIOS to pre-allocate BDF and memory for every slot
  - NVMe is write-based and therefore does not need a large BAR
Solutions to Hot Add

- When drives are added
  - OS loads NVMe drivers and uses pre-allocated BDF and memory

- Drivers are properly loaded in the OS
Let’s Talk Hot Remove

Hot Add
- Single
- Engaged
- Divorced
- It’s Complicated
- Separated
- In a Relationship
- Married

Hot Remove
Why Is Hot Remove Complicated?

- Drive is removed
- Surprise down error is generated, which causes FATAL_ERROR
- Host NMI due to FATAL_ERROR
Solutions for Hot Remove

- Downstream port containment (DPC)
  - Triggers on unmasked, uncorrectable errors and shuts down the port
- Blocks new transactions destined for the port
- Logs surprise down but blocks FATAL_ERROR
Solutions for Hot Remove

- What about traffic already in flight?
  - Posted transactions are discarded
  - Non-posted transactions are trickier, as they require a completion

- Host keeps a timer for completions
  - If no completion is returned in time, then CTO
  - This can lead to kernel panic
Why Is Hot Remove Complicated?

- What about traffic already in flight?
  - Posted transactions are discarded
  - Non-posted transactions are trickier, as they require a completion
- Host keeps a timer for completions
  - If no completion is returned in time, then CTO
  - This can lead to kernel panic
Solutions for Hot Remove

- The switch can keep track of outstanding completions

- If a completion times out, the switch can *synthesize* one for the host
  - This is called completion timeout synthesis (CTS)

- Drivers that are aware of all-1s will unload
Solutions for Hot Remove

- Too good to be true?
- It was!
  - Prior to Kernel 4.7, DPC wasn’t even supported
  - Prior to Kernel 4.11, all-1s was entirely **not** supported in NVMe or PCIe service drivers!
  - Drivers that were not all-1s aware went off the rails

→ Before all-1s support
↑ After adding all-1s support

1,000s of IOs reduced to ~20
## Solutions for Hot Plug

<table>
<thead>
<tr>
<th>Contribution (Intel/Fbk/MSCC)</th>
<th>Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>New</strong> PCIe downstream port containment (DPC) driver</td>
<td>4.7</td>
</tr>
<tr>
<td>Enhancements and optimizations to PCI driver</td>
<td>4.11</td>
</tr>
<tr>
<td>Recognizing all-1s as a missing device on key config registers</td>
<td></td>
</tr>
<tr>
<td>Enhancements and optimizations to AER driver</td>
<td>4.4</td>
</tr>
<tr>
<td>Caching of extended capability pointers</td>
<td></td>
</tr>
<tr>
<td>Enhancements to NVMe driver</td>
<td>4.7</td>
</tr>
<tr>
<td>Recognizing all-1s as a missing device</td>
<td></td>
</tr>
<tr>
<td>Cleaning up after hot remove without further IO</td>
<td></td>
</tr>
<tr>
<td>Enhancements and fixes in the block multi-queue driver</td>
<td>4.7</td>
</tr>
<tr>
<td>Dealing with errors returned on IO following surprise removal</td>
<td></td>
</tr>
</tbody>
</table>
Other Considerations

• Slow add

• U.2 form factor has the PRSNT# pin as first to mate
• This pin will trigger presence state change to the host that will, in turn, enable power to the slot
• It is possible to enable power to a slot before the U.2 is fully docked
Other Considerations

• Slow remove

• U.2 form factor has the PRSNT# pin as first to mate (last to disengage)
• PCIe lanes will undock first leaving PCIe LTSSM to stay in recovery up to 24 ms before PRSNT# unconnected
• During this time, TLPs are flowing
  • Therefore, host CTO needs to be >24 ms (as a rule)
  • Short cutting LTSSM recovery can help, too
Other Considerations

• Host timeouts should be scaled to account for system congestion and slow removal

• Similarly, NVMe drive timeouts should be scaled to, as well

• An NVMe CTO generally results in link down
Summary

• Hot plug of NVMe is complicated
  • But we’re almost there
• Hot add is solved
  • With switch-exposed P2P bridges with BIOS pre-allocation
• Hot remove is solved
  • With OS updates to support all-1s and large host and NVMe drive timeouts, and enhanced with switch fast link down