Prevent DMA attacks from untrusted devices

Lu Baolu
Intel Corporation

Agenda

- A quick summary of what we have done.
- Issues and TODO list.
  - Where to flag an untrusted device?
  - We need to bring back ATS for untrusted devices.
  - Device driver uses full-page buffers to alleviate performance regression caused by bounce buffer.
  - Loose strict mode for untrusted devices.

A quick summary of what we've done.

- Identify and mark untrusted devices.
- Force IOMMU on if untrusted devices exist.
- Disable ATS on untrusted devices.
- Enforce strict mode on untrusted devices.
- Use bounce page for untrusted devices

TODO list

- Currently, an untrusted device is marked by a bit in struct pci_device, do we need to move it to struct device to remove the pci dependency?

```
struct pci_dev {
    /*
    * Devices marked being untrusted are the ones that can potentially
    * execute DMA attacks and similar. They are typically connected
    * through external ports such as Thunderbolt but not limited to
    * that. When an IOMMU is enabled they should be getting full
    * mappings to make sure they cannot access arbitrary memory.
    */
    unsigned int      untrusted:1;
}
```

TODO list

- Currently, ATS is disabled from IOMMU end if untrusted devices exist. But some features, like SVA, depends on ATS. It seems that we must bring it back with secure ATS support.

TODO list

- Currently, strict mode is enforced on untrusted devices. We need to remove it with bounce buffer enhancement.

TODO list

- Currently, bounce buffer is used if a driver applies full-page buffers and use only a fragment for a single DMA mapping. This is unnecessary and performance hurt as the result. Need to introduce a DMA attribute set by driver if bounce buffer is unnecessary.

INTEL OPEN SOURCE TECHNOLOGY CENTER | 01.org

@twitter handle