Integrated PCIe Monitoring and Tracing Facilities

Yicong Yang <yangyicong@hisilicon.com>
Introduction
PCIe Performance Monitoring Unit
PCIe Tune and Trace Device
Potential Scenarios
Open questions
Introduction

• Why - Lack for (or limited) PCIe link analyzing and tuning method
  − There are various tools for Monitoring and tracing CPUs, but almost none for peripherals, especially PCIe
  − Limited method for debugging PCIe link. The common PCIe analyzer is expensive, complex, invasive and hard to setup.
  − The performance is not optimized since we deploy same configuration for all the cases, like buffer allocation

• How - Integrated monitor and tracing facilities
  − We already have uncore PMUs for DDR, interconnections, etc. This time we bring it to peripherals
  − TLP tracing modules are embedded in our controller
  − Interface for tuning the PCIe link configuration dynamically
  − Integrated in the Root Complex as iEPs, no need for extra setups
Introduction

PCIe Performance Monitoring Unit

PCIe Tune and Trace Device

Potential Scenarios

Open questions
PCIe Performance Monitoring Unit

- One PMU for each PCIe core
- Several Counters for monitor the Link Events:
  - Bandwidth
  - Latency
  - Bus utilization
  - Buffer occupancy
  - ...
- Conditional count by the filters
  - Count certain Package length
  - Count certain Root Port or Endpoint
# perf stat -e hisi_pcie0_core2/rx_mrd_flux,port=0x1,trig_len=4,trig_mode=1,thr_mode=1,thr_len=0x4/

**Usage**

**Link event**
- `port=xxx`: Events downstream certain Root Port
- `BDF=xxx`: Events from certain Endpoint

**Target Filter**
- `port=xxx`: Events downstream certain Root Port
- `BDF=xxx`: Events from certain Endpoint

**Trigger Filter**
- `trig_mode={0,1}`: 0-greater, 1-smaller
- `trig_len=N`: trigger when one TLP length greater/smaller than $2^N$ DWord

**Threshold Filter**
- `thr_mode={0,1}`: 0-greater, 1-smaller
- `thr_len=N`: count the TLP whose length greater/smaller than $2^N$ DWord

---

Performance counter state for 'system wide':

```
5984  hisi_pcie0_core2/rx_mrd_flux,port=0x1,trig_len=4,trig_mode=1,thr_mode=1,thr_len=0x4,thr_mode=1/
11.030560400 seconds time elapsed```
Introduction
PCIe Performance Monitoring Unit
PCIe Tune and Trace Device
Potential Scenarios
Open questions
PCle Tune and Trace Device

- One PTT for each PCle core
- TLP header tracer for
  - TLPs downstream certain Root Port or of certain Requester ID
  - Certain Type, P, NP or CPL
  - Certain direction, inbound or outbound
- Dynamically tuning the Link configurations of the PCle core
# perf record -e hisi_ptt0_2\filter=0x80001\type=1\direction=1\format=1/

<table>
<thead>
<tr>
<th>Filter</th>
<th>Type</th>
<th>Direction</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x8xxxx:</td>
<td>Trace TLP headers of either posted TLP, Non-Posted TLP, Completion or all.</td>
<td>Trace TLP headers of either inbound, outbound or both.</td>
<td>The desired data format of the traced TLP headers. Can be either 4DW or 8DW.</td>
</tr>
<tr>
<td>0x0xxxx:</td>
<td>Trace TLPs with Requester ID xxxx.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Raw Data Format

4DW format

```
bits [31:30] [ 29:25 ][24][23][22][21] [ 20:11 ] [ 10:0 ]
```

- `Dw0`: Fmt][ Type ][T9][T8][TH][SO][ Length ][ Time ]
- `Dw1`: Header DW1
- `Dw2`: Header DW2
- `Dw3`: Header DW3

```
0x1fffffff || Reserved (0x7ff)
```

- `Dw0`: Prefix
- `Dw1`: Header DW0
- `Dw2`: Header DW1
- `Dw3`: Header DW2
- `Dw4`: Header DW3
- `Dw5`: Reserved (0x0)
- `Dw6`: Time

8DW format

```
# perf report -D -i perf.data
```

Decoding and Display by `perf report`

```
00000000: 08 00 00 00 Prefix
00000004: 01 00 00 00 Header DW0
00000008: 04 00 00 00 Header DW1
0000000c: 40 00 00 00 Header DW2
00000010: 10 40 00 00 Header DW3
```

Header DWx with same definition in the Spec

```
4DW data format
8DW data format
```
• TLP header tracing is implemented by hardware wiring, so merely very little overhead
• It’s deliberately designed to have four >4MiB DMA buffers, so we won’t miss one packet even under the full bandwidth
The link events are exported as sysfs attributes. Currently support tuning the QoS of certain TLP packets (P, NP, CPL) and the buffer level of certain direction (Inbound, outbound).

- The value of the tune event is an abstract level, like 0 for a low level and 2 for a high level.
- The real RC configurations are hidden behind the level and be set/read indirectly, with assistance of IMU.
Introduction
PCIe Performance Monitoring Unit
PCIe Tune and Trace Device
Potential Scenarios
Open questions
Potential Scenarios

• Monitor the status and utilization of the PCIe link
  - Monitor the bandwidth, latency, buffer util, PM status etc. Help profiling and evaluating the IO applications.

• tracing for validating and monitoring, quick and convenient
  - Finding the hardware bugs, for example out of order TLPs
  - Track the access order and help debug the driver
  - Help the error locating and handling, complement to the AER, etc

• Tune the PCIe performance
  - If already know the access pattern, tune the link directly
  - Otherwise, monitor the link statistic and tune accordingly
# scp 4MiB.dat root@192.110.110.1:/tmp/

PCIe PMU or/with PTT trace

Profiling

feedback

Tuning

# echo 2 > tx_alloc_buf_level

PTT tune

Profilong the Buffer allocation success count
0x0009 – Rx Buffer. 0x0109 – Tx Buffer
In this case Tx buffer allocation is less likely to success

The Tx Buffer allocation success count increased.

Performance counter stats for 'system wide':

41774 hisi_pcie0_core2/port=0x1, event=0x0009/
  (50.09%)
6746 hisi_pcie0_core2/port=0x1, event=0x0109/
  (50.00%)

Performance counter stats for 'system wide':

35252 hisi_pcie0_core2/port=0x1, event=0x0009/
  (50.05%)
7676 hisi_pcie0_core2/port=0x1, event=0x0109/
  (49.80%)
Introduction
PCIe Performance Monitoring Unit
PCIe Tune and Trace Device
Potential Scenarios
Open questions
Open questions

- need feedbacks for the design, usage and future plan. What we want and what'll be more helpful?

- possible for more platforms and devices, like PMU/PTT for a switch?
  - We already have PMU for HiSilicon HNS3 network card.

- Will it be helpful if we extend the tracing from TLPs to DLLPs, etc?

- Will it be possible and helpful to make it standardization? For either specification or software framework.

- etc.
• PCIe Performance Monitor Units


Driver support: https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/drivers/perf/hisilicon/hisi_pcie_pmu.c?h=v5.19

• PCIe Tune and Trace device (Request for review!)

Driver support: https://lore.kernel.org/lkml/20220816114414.4092-1-yangyicong@huawei.com/

Perf support: https://lore.kernel.org/lkml/20220816125757.60302-1-yangyicong@huawei.com/
Thanks!