Hardware-assisted software tracing

Adrien Vergé

DORSAL lab
Computer Engineering Departement
December 11, 2013
Introduction

Software tracing has non-zero side-effects:

- interrupts
- system events
  - cache flushing, etc.
- may add some latency

Hardware tracing:

- observes the system: almost zero overhead
- provides dedicated resources (buffers...)

Hardware-assisted software tracing
Hardware-assisted tracing

Use of hardware components
  ▶ on chip

Advantages:
  ▶ dedicated circuit for tracing
  ▶ very detailed info
    ▶ from the source
  ▶ customizable events

source: arm.com
Studied platforms

ARM (Coresight)

Intel (BTS)

Freescale (Performance Monitor)
Hardware-assisted software tracing

DORSAL, Polytechnique Montréal
Coresight and LTTng

- **Embedded Trace Buffer**: stores trace generated by different sources retrieved on-chip or off-chip.

- **System Trace Macrocell**: stores software events many "channels".

- **Program Trace Macrocell**: stores execution path (executed instructions sequence).

**Automatic Timestamping**: cycle-precise.
Coresight: STM

System Trace Macrocell

- dedicated hardware to store software events
- timestamped
  - correlated with other events
  - cycle-precise
- multisource trace in a single stream
  - optimized bandwidth
Coresight: ETM/PTM

Embedded/Program Trace Macrocell

- monitors the core’s internal busses
  - no burden on performance
- hardware triggers
  - start tracing only when needed
- hardware filters
  - output only what needed
- data compression
  - complete sequence of executed instructions: approx. 1 bit / cycle
Coresight: ETB

Embedded Trace Buffer

- dedicated buffer (on Pandaboard: 8 kiB)
- gives time to defer tracing
- multiplex data from different sources
- on-chip and off-chip connection (JTAG)
STM + ETB: results

Compare to LTTng-UST

```c
for (i = 0; i < 1000000; i++)
    tracepoint(int, smallvalue);

for (i = 0; i < 1000000; i++)
    tracepoint(string, "a longer message");
```
Tracing overhead in a simple loop

- No tracing (LTTng stopped)
- LTTng-UST
- STM + ETB

<table>
<thead>
<tr>
<th>iterations</th>
<th>no tracing (LTTng stopped)</th>
<th>LTTng-UST</th>
<th>STM + ETB</th>
</tr>
</thead>
<tbody>
<tr>
<td>100k</td>
<td></td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>1M</td>
<td></td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>10M</td>
<td></td>
<td>0.01</td>
<td>0.01</td>
</tr>
</tbody>
</table>

-90% savings at 100k iterations.
-91% savings at 1M iterations.
-91% savings at 10M iterations.
Tracing overhead in a simple loop

- **no tracing (LTTng stopped)**
- **LTTng-UST**
- **STM + ETB**

<table>
<thead>
<tr>
<th>trace data size (B)</th>
<th>time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>20</td>
<td>15</td>
</tr>
<tr>
<td>40</td>
<td>15</td>
</tr>
<tr>
<td>60</td>
<td>15</td>
</tr>
<tr>
<td>80</td>
<td>15</td>
</tr>
<tr>
<td>100</td>
<td>15</td>
</tr>
</tbody>
</table>

Hardware-assisted software tracing

DORSAL, Polytechnique Montréal
ETM + ETB: results

- equivalent in LTTng: empty tracepoint
  - cannot embed custom data

- extra info from hardware

- requires info on the program symbols
  - to set address matching trigger

- ETB: 512 bytes
  For now, benchmark on small loops
  - avoid overflow
Tracing overhead in a simple loop

- LTTng-UST (stopped)
- LTTng-UST
- no ETM
- ETM + ETB

Hardware-assisted software tracing
DORSAL, Polytechnique Montréal
Intel: BTS (Branch Trace Store)

- traces executed instructions
- logs EVERY branch
- stores everything in RAM
  - 24 bytes / branch
  - intense bus usage
- need to drain to disk
  - intense disk usage
Tracing overhead in a simple loop

<table>
<thead>
<tr>
<th>Branching Rate (branch/s)</th>
<th>Time Overhead (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4E+06</td>
<td>100</td>
</tr>
<tr>
<td>5.6E+06</td>
<td>120</td>
</tr>
<tr>
<td>7.5E+06</td>
<td>150</td>
</tr>
<tr>
<td>7.7E+06</td>
<td>200</td>
</tr>
<tr>
<td>7.9E+06</td>
<td>220</td>
</tr>
<tr>
<td>8.8E+06</td>
<td>250</td>
</tr>
<tr>
<td>8.9E+06</td>
<td>270</td>
</tr>
<tr>
<td>9.8E+06</td>
<td>300</td>
</tr>
<tr>
<td>1.1E+07</td>
<td>350</td>
</tr>
</tbody>
</table>

- LTTng-UST
- BTS with kernel support

BTS loses events

md5sum: 7.4E+06
sha256sum: 3.70E+07
gcc: 6.70E+08
Intel: Processor Trace

- traces executed instructions
- triggering, filtering and compression capabilities
  - similar to Coresight ETM
- not yet on silicon
- already generators and decoders
  - open-source

2014? 2015?
Freescale

Hardware-assisted software tracing

DORSAL, Polytechnique Montréal
Freescale: Performance Monitor

Freescale processors
- PowerPC architecture

Performance Monitor
- collection of counters
- hundreds of events
  - cache misses, type of instructions, branches...

Nexus module
- program trace
- data trace
- taken interrupts...
## Conclusion

<table>
<thead>
<tr>
<th>ARM</th>
<th>Intel</th>
<th>Freescale</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>lightweight tracepoints</em></td>
<td><em>branch tracing</em></td>
<td>work in progress...</td>
</tr>
<tr>
<td>STM + ETB ✓</td>
<td>BTS X</td>
<td></td>
</tr>
<tr>
<td><em>lightweight branch tracing</em></td>
<td><em>lightweight and configurable branch tracing</em></td>
<td></td>
</tr>
<tr>
<td>ETM + ETB ✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Hardware-assisted software tracing

DORSAL, Polytechnique Montréal
Questions?