Graphics tracing with Perfetto

Introducing gfx-pps

Antonio Caggiano
Consultant Software Engineer
antonio.caggiano@collabora.com
Overview

- GPU hardware counters
- Perfetto
- Gfx-pps
- Capture a trace
GPU hardware counters

Exposed by the GPU
- Updated every cycle/frame
- Reset after reading
- Logically grouped
- Different units
GPU hardware counters

Mali Midgard counters

- Job Manager
- Shader Core
- Tiler
- L2 Cache
Panfrost - Hardware counters ioctls

- drmIoctl( fd, DRM_IOCTL_PANFROST_PERFCNT_ENABLE, &perfcnt );
- drmIoctl( fd, DRM_IOCTL_PANFROST_PERFCNT_DUMP, &dump );
Profiling, tracing, trace analysis

- CPU, events, GPU, memory, ...
- Ftrace, procfs, sysfs
- Perfetto.dev
Perfetto

Service-based

• Service
• Producer
• Consumer
Custom Data Source

Amalgamated C++ sources

• virtual DataSource::OnSetup( args )
• virtual DataSource::OnStart( args )
• virtual DataSource::OnStop( args )
• DataSource::Trace( callback )
Gfx-pps

• Perfetto producers
  - Midgard hardware counters via Panfrost
  - Weston debug timeline

• Helper tool gpu-perf-cnt
  - Dump counters
  - List counters names
Download binaries

perfetto

master
Capture a trace

> traced

> traced_probes

> producer-gpu

> perfetto --txt -c gpu.cfg -o trace
Anatomy of a cfg

buffers {
    size_kb: 1024
    fill_policy: RING_BUFFER
}

data_sources {
    config {
        config {
            name: "gpu.metrics" // name of the data source
            gpu_counter_config {
                counter_period_ns: 1000000000 // sample every second
            }
        }
    }
}

duration_ms: 16000 // trace for 16 seconds
Trace visualizer

- Available at ui.perfetto.dev
- Modern web technologies
- Once opened it works fully offline
<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute tasks</td>
<td>0.25M</td>
</tr>
<tr>
<td>Compute threads</td>
<td>5K</td>
</tr>
<tr>
<td>Compute usage</td>
<td>50</td>
</tr>
<tr>
<td>Frag cycles no tile</td>
<td>0.25M</td>
</tr>
<tr>
<td>Frag dummy threads</td>
<td>25K</td>
</tr>
<tr>
<td>Frag num tiles</td>
<td>2.5M</td>
</tr>
<tr>
<td>Frag primitives</td>
<td>2.5K</td>
</tr>
<tr>
<td>Frag primitives dropped</td>
<td>0</td>
</tr>
<tr>
<td>Frag quad rasterized</td>
<td>2.5K</td>
</tr>
<tr>
<td>Frag quads EZS killed</td>
<td>0</td>
</tr>
<tr>
<td>Frag quads EZS test</td>
<td>5</td>
</tr>
<tr>
<td>Frag threads</td>
<td>0.25K</td>
</tr>
<tr>
<td>Frag threads LZS killed</td>
<td>5M</td>
</tr>
<tr>
<td>Frag threads LZS test</td>
<td>0.5M</td>
</tr>
<tr>
<td>Frag transaction eliminations</td>
<td>0.5M</td>
</tr>
<tr>
<td>Fragment usage</td>
<td>1</td>
</tr>
<tr>
<td>Front-facing triangles</td>
<td>50K</td>
</tr>
<tr>
<td>GPU active</td>
<td>5M</td>
</tr>
<tr>
<td>IRQ active</td>
<td>1M</td>
</tr>
<tr>
<td>JSO usage</td>
<td>100</td>
</tr>
<tr>
<td>JS1 usage</td>
<td>100</td>
</tr>
<tr>
<td>JS2 usage</td>
<td>0</td>
</tr>
<tr>
<td>L2 external read stalls</td>
<td>75K</td>
</tr>
<tr>
<td>L2 external read usage</td>
<td>10</td>
</tr>
<tr>
<td>L2 external reads</td>
<td>25M</td>
</tr>
<tr>
<td>L2 external reads</td>
<td>25M</td>
</tr>
<tr>
<td>L2 external write stalls</td>
<td>5K</td>
</tr>
<tr>
<td>L2 external write usage</td>
<td>0.25K</td>
</tr>
</tbody>
</table>