Revealing Critical Loads and Hidden Data Locality in GPGPU Applications

Gunjae Koo, Hyeran Jeon and Murali Annavaram
Memory Latency Still Matters

- Quick context switch in GPUs
  - Hide latency by switching to ready warp
  - Not enough for memory operations

Latency from global memory = 100s ~ 1000s cycles
Memory Traffic Matters

• Many threads in a GPU
  – Warp (wavefront) = group of multiple threads
  – Coalescer merges memory requests located in a cache line
  – High pressure from many threads of irregular applications

Tesla M2050
- 1536 threads / SM
- 16KB L1 cache

To interconnection
Outline

• Applications and Environment
• Classification of Load Instructions
• Impacts on Memory Traffic
• Impacts on Latency
• Data Locality among CTAs
Outline

• Applications and Environment
• Classification of Load Instructions
• Impacts on Memory Traffic
• Impacts on Latency
• Data Locality among CTAs
## Applications

- **15 applications with 3 categories**

<table>
<thead>
<tr>
<th>Name</th>
<th>Fraction of global loads</th>
<th>Name</th>
<th>Fraction of global loads</th>
<th>Name</th>
<th>Fraction of global loads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear algebra</td>
<td></td>
<td>Image processing</td>
<td></td>
<td>Graph</td>
<td></td>
</tr>
<tr>
<td>2mm</td>
<td>18.10 %</td>
<td>htw</td>
<td>8.56%</td>
<td>bfs</td>
<td>1.17%</td>
</tr>
<tr>
<td>gaus</td>
<td>3.04 %</td>
<td>mriq</td>
<td>0.03%</td>
<td>sssp</td>
<td>5.66%</td>
</tr>
<tr>
<td>grm</td>
<td>24.75 %</td>
<td>dwt</td>
<td>2.41%</td>
<td>ccl</td>
<td>5.78%</td>
</tr>
<tr>
<td>lu</td>
<td>6.65 %</td>
<td>bpr</td>
<td>3.71%</td>
<td>mst</td>
<td>1.19%</td>
</tr>
<tr>
<td>spmv</td>
<td>11.73 %</td>
<td>srad</td>
<td>3.57%</td>
<td>mis</td>
<td>0.19%</td>
</tr>
</tbody>
</table>
## Experimental Environment

**Configuration of GPGPU-sim**

<table>
<thead>
<tr>
<th>Component</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tesla C2050 (GPGPU-sim)</strong></td>
<td></td>
</tr>
<tr>
<td>Core</td>
<td>14 SMs @ 1.15GHz, 32 SIMT width</td>
</tr>
<tr>
<td>Memory</td>
<td>GDDR5 @ 1.5GHz</td>
</tr>
<tr>
<td>L1D cache</td>
<td>16KB, 128B line, 4-way, 64 MSHRs</td>
</tr>
<tr>
<td>L2 cache</td>
<td>Unified, 768KB, 128B line, 8-way</td>
</tr>
<tr>
<td>RF</td>
<td>128 KB</td>
</tr>
<tr>
<td>Shared memory</td>
<td>48 KB</td>
</tr>
<tr>
<td>ROP latency</td>
<td>120 cycles</td>
</tr>
<tr>
<td>DRAM latency</td>
<td>100 cycles</td>
</tr>
</tbody>
</table>
Outline

• Applications and Environment
• Classification of Load Instructions
• Impacts on Memory Traffic
• Impacts on Latency
• Data Locality among CTAs
Classification of Load

• Deterministic (D)
  – Originated from parameterized data (thread ids, block ids, data from *ld.param*)
  –Adjacent threads are grouped in a warp → highly coalesced

• Non-deterministic (ND)
  – Originated from unpredictable data (*ld.global, ld.local, ld.shared, ld.tex*)
Classification of Load

- Deterministic load

```c
int tid = blockIdx.x * MAX_THREADS_PER_BLOCK + threadIdx.x;
if (tid < no_of_nodes && g_graph_mask[tid]) {
    g_graph_mask[tid] = false;
    for (int i = g_graph_nodes[tid].starting; ... ) {
        int id = g_graph_edges[i];
        if (!g_graph_visited[id])
            ...
    }
}
```
Classification of Load

• Non-deterministic load

```c
int tid = blockIdx.x * MAX_THREADS_PER_BLOCK + threadIdx.x;
if (tid < no_of_nodes && g_graph_mask[tid]) {
    g_graph_mask[tid] = false;
    for (int i = g_graph_nodes[tid].starting; ...) {
        int id = g_graph_edges[i];
        if (!g_graph_visited[id])
            ...
    }
}
```
Classification of Load

• Fraction of global load

– Even irregular graph applications have 50% of D loads
Outline

• Applications and Environment
• Classification of Load Instructions
• **Impacts on Memory Traffic**
• Impacts on Latency
• Data Locality among CTAs
Impact on Memory Traffic

- Requests per warp and active thread
  - D loads are usually highly coalesced
  - ND loads are uncoalesced (e.g. bfs ≈ 0.8 req/thread)
    - More requests
Impact on Memory Traffic

- **L1 data cache cycles**
  - grm, htw: high hit ratio → efficient data reuse in L1 cache
  - Reservation fail = resource is not enough to treat requests
Outline

• Applications and Environment
• Classification of Load Instructions
• Impacts on Memory Traffic
• **Impacts on Latency**
• Impacts on Cache Miss Rate
• Data Locality among CTAs
Impact on Latency

- Turnaround time

- ND loads has longer turnaround time
- Source of delays: reservation fails, wasted cycles in L2 and DRAMs
Impact on Latency

- Turnaround time by number of requests

  - Turnaround time increases by the number of requests of warp
Impact on Latency

- Breakdown of turnaround time

- Turnaround time increased due to
  - L1 reservation fails
  - Load imbalance among global memory channels
Outline

• Applications and Environment
• Classification of Load Instructions
• Impacts on Memory Traffic
• Impacts on Latency
• Impacts on Cache Miss Rate
• Data Locality among CTAs
Data Locality

- CTA assignment in GPUs
Data Locality

• Ratio of shared data space

  – Data is shared by multiple CTAs
    • On average 50% of memory accesses for shared address space
Data Locality

• Frequency of CTA distances

- Data is shared by neighbor CTAs
- Specific distances can be observed
- Long tail for graph applications

Graph applications

Linear algebra

Image processing
Further Detailed in the Paper

• More information for applications
• Impact on cache miss ratio
• Discussion for further work
Conclusion

• Detailed characteristics of load instructions
  – Deterministic: highly-coalesced, load density and cache hit ratio matters
  – Non-deterministic: uncoalesced, consumes resource, longer turnaround time

• Data locality
  – Data is shared by multiple neighbor CTAs
Thank you

Gunjae Koo (gunjae.koo@usc.edu)
Revealing Critical Loads and Hidden Data Locality in GPGPU Applications

Gunjae Koo, Hyeran Jeon and Murali Annavaram

This work was supported by DARPA-PERFECT-HR0011-12-2-0020, NSF-CAREER-0954211 and NSF-0834798.