Multi-GPU Graph Analytics

Yuechao Pan, Yangzhiao Wang, Yuduo Wu, Carl Yang and John D. Owens, University of California, Davis {ychpan, yzhwang, yuduo, ctyyang, jowens}@ucdavis.edu

Introduction - about Gunrock

Gunrock is a multi-GPU graph processing library, which targets at:

- **High performance** analytics of large graphs
- **Low programming complexity** in implementing parallel graph algorithms on GPUs

Homepage: [http://gunrock.github.io](http://gunrock.github.io)

The copyright of Gunrock is owned by The Regents of the University of California, 2016. All source code are released under Apache 2.0.

Graph algorithm as a data-centric process

Frontier: compact queue of nodes or edges

What we aimed and achieved

- **Programmability**: easy to develop graph primitives on multiple GPUs
- **Algorithm generality**: support a wide range of graph algorithms
- **Hardware compatibility**: usable on most single node GPU systems
- **Performance**: low runtime, and leverages the underlying hardware well
- **Scalability**: scalable in terms of both performance and memory usage

Multi-GPU Framework

What the programmer needs to specify:

- Core single-GPU primitives
- Data to communicate
- How to combine remote and local data
- Stop condition

What the framework takes care:

- Split frontiers
- Package data
- Push to remote GPUs
- Merge local & received data
- Manage GPUs

Optimizations

- Direction-optimizing traversal
- Compute / communication overlap
- Just enough memory allocation

All tailored to support multi-GPU environment better

Future Work

- extending Gunrock onto multiple nodes
- Asynchronous graph algorithms
- Other partitioning methods
- more algorithms

Results

<table>
<thead>
<tr>
<th>Graph</th>
<th>ref.</th>
<th>ref. hw.</th>
<th>ref. perf.</th>
<th>our hw.</th>
<th>our perf.</th>
<th>comp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>com-orkit (1M, 117M, UD)</td>
<td>Bisson [3]</td>
<td>1 x K20x x 4</td>
<td>2.67 GTEPS</td>
<td>4 x K40</td>
<td>11.42 GTEPS</td>
<td>5.33 X</td>
</tr>
<tr>
<td>com-Friendfinder (100k, 117M)</td>
<td>Bisson [3]</td>
<td>1 x K20x x 6</td>
<td>15.68 GTEPS</td>
<td>4 x K40</td>
<td>14.1 GTEPS</td>
<td>0.90 X</td>
</tr>
<tr>
<td>kron_{n25_16} (UD, 117M)</td>
<td>Bernaschi [4]</td>
<td>1 x K20x x 4</td>
<td>~ 3.3 GTEPS</td>
<td>4 x K40</td>
<td>30.8 GTEPS</td>
<td>23.7 X</td>
</tr>
<tr>
<td>kron_{n25_16} (UD, 117M)</td>
<td>Bernaschi [4]</td>
<td>1 x K20x x 6</td>
<td>~ 3.2 GTEPS</td>
<td>6 x K40</td>
<td>31.0 GTEPS</td>
<td>9.69 X</td>
</tr>
<tr>
<td>kron_{n25_32} (UD, 107M, UD)</td>
<td>Fu [5]</td>
<td>2 x K20 x 32</td>
<td>22.7 GTEPS</td>
<td>4 x K40</td>
<td>32.0 GTEPS</td>
<td>1.41 X</td>
</tr>
<tr>
<td>kron_{n23_32} (UD, 256M, UD)</td>
<td>Fu [5]</td>
<td>2 x K20 x 20</td>
<td>6.3 GTEPS</td>
<td>4 x K40</td>
<td>7.9 GTEPS</td>
<td>4.43 X</td>
</tr>
<tr>
<td>kron_{n24_32} (UD, 107M, UD)</td>
<td>Liu [6]</td>
<td>2 x K40 x 41</td>
<td>8.8 GTEPS</td>
<td>18 GTEPS</td>
<td>87.7 GTEPS</td>
<td>5.38 X</td>
</tr>
<tr>
<td>kron_{n24_32} (UD, 107M, UD)</td>
<td>Liu [6]</td>
<td>2 x K40 x 41</td>
<td>8.8 GTEPS</td>
<td>18 GTEPS</td>
<td>87.7 GTEPS</td>
<td>5.38 X</td>
</tr>
<tr>
<td>twitter-mpi (32M, 668, D)</td>
<td>Bebej [7]</td>
<td>1 x K40 x 16</td>
<td>0.2242 sec</td>
<td>3 x K40</td>
<td>94.31 ms</td>
<td>2.38 X</td>
</tr>
</tbody>
</table>

Graphs with 3.62B edges processed using 6 GPUs

> 2.63 X, 2.57 X, 2.00 X, 1.96 X, and 3.86 X geometric mean speedups over 16 datasets for BFS, SSPC, CC, BC, and PR

- graphs with 3.62B edges processed using 6 GPUs

Acknowledgements

This work was funded by the DARPA MOSAIC program under ARPA Contract N66001-15-1-4880 and by NSF Award CCF-1528128 and CCF-1527011.

References


Graph algorithms with 3.62B edges processed using 6 GPUs