Subway: Minimizing Data Transfer during Out-of-GPU-Memory Graph Processing

Amir Hossein Nodehi Sabet, Zhijia Zhao, Rajiv Gupta

Computer Science and Engineering
UC Riverside
Background and Motivation

• GPUs enable massive parallelism for graph processing
  - CuSha [1]
  - Gunrock [2]
  - Tigr [3]
  - ...

• Graphs can be large and tend to grow over time
  - Web graphs
  - Social networks

• But GPU memory is limited!!
  - Out-of-GPU-Memory Graph Processing

Partition-based Graph Processing

Main Memory

Transferring

GPU Memory

Computation
A Key Observation

Ratio of active vertices (edges) is often low in most iterations

### Average Ratio of Active Edges across Iterations

<table>
<thead>
<tr>
<th>Algo.</th>
<th>friendster</th>
<th>Uk-2007</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSSP</td>
<td>9.1%</td>
<td>5.1%</td>
</tr>
<tr>
<td>BFS</td>
<td>4.1%</td>
<td>0.6%</td>
</tr>
<tr>
<td>CC</td>
<td>9.8%</td>
<td>3.2%</td>
</tr>
</tbody>
</table>
Only Load Active Edges to GPU?

Main Memory

GPU Memory

Too expensive to generate ?!
Efficient Subgraph Generation

Subway:
• a concise subgraph representation, called SubCSR
• a highly parallel algorithm for subgraph generation
• an efficient GPU-accelerated implementation
SubCSR Generation Cost

Costs: Partitioning-based vs. Subway (subgraph generation)
Takeaway

Too expensive to dynamically generate subgraphs!

Improve performance up to 28X!
Thank you

Amir Nodehi: anode001@ucr.edu or on Slack
The source code (to be posted soon): https://github.com/AutomataLab/Subway