HIGH-PERFORMANCE GRAPH PROCESSING PROGRAMMING MODEL ON THE GPU

Yangzihao Wang
January 29, 2015
University of California, Davis
Topics

Context  Related works on parallel graph processing

Current  Design and implementation of Gunrock

Future   Research problems and next steps
The trade-off between programmability and performance
CHALLENGES FOR GRAPH PROCESSING ON THE GPU

- The irregularity of data access/control flow
- The complexity of programming GPUs
Deliver the performance of GPU hardwired graph primitives with a high-level programming model that allows programmers to quickly develop new graph primitives.
A graph is an ordered pair $G = (V, E, w_e, w_v)$ comprised of a set of vertices $V$ together with a set of edges $E$, where $E \subseteq V \times V$. Our primary working set is a frontier. A vertex frontier is a subset of vertices $U \in V$ and an edge frontier a subset of edges $I \in E$. 
Most graph algorithms have two major operations:

**Traverse**  Updating a frontier by traversing in the graph or subsetting the current frontier.

**Compute**  Doing computation on edges or nodes.
Two ways to traverse:

**Advance** Generate a new frontier by visiting the neighbors of the current frontier.

**Filter** Chooses a subset of the current frontier based on programmer-specified criteria.
Two ways to traverse:

**Advance** Generate a new frontier by visiting the neighbors of the current frontier.

**Filter** Chooses a subset of the current frontier based on programmer-specified criteria.
Two ways to traverse:

**Advance** Generate a new frontier by visiting the neighbors of the current frontier.

**Filter** Chooses a subset of the current frontier based on programmer-specified criteria.
Two ways to traverse:

**Advance** Generate a new frontier by visiting the neighbors of the current frontier.

**Filter** Chooses a subset of the current frontier based on programmer-specified criteria.
Two ways to traverse:

**Advance** Generate a new frontier by visiting the neighbors of the current frontier.

**Filter** Chooses a subset of the current frontier based on programmer-specified criteria.
**METHODS: BUILD HIGH-LEVEL ABSTRACTION**

**BFS:**
- **Advance:** Update Label Value
- **Filter:** Remove Redundant

**BC:**
- **Advance:** Accumulate Sigma Value
- **Filter:** Remove Redundant

**CC:**
- **Filter:** For e=(v1,v2), assign c[v1] to c[v2]. Remove e when c[v1]==c[v2]

**SSSP:**
- **Advance:** Update Label Value
- **Filter:** Remove Redundant

**Near/Far Pile**

**PR:**
- **Advance:** Distribute PR value to Neighbors
- **Filter:** Update PR value. Remove when PR value converge

**Traversal**
**Computation**
METHODS: DEVELOP LOW-LEVEL OPTIMIZATION

- Workload Mapping for Advance
- Improving Work Efficiency
- Primitive-Specific Optimization
Methods: Develop Low-Level Optimization

Figure: Merrill et al. PPoPP’12

Block cooperative Advance of large neighbor lists;

Warp cooperative Advance of medium neighbor lists;

Per-thread Advance of small neighbor lists.
METHODS: DEVELOP LOW-LEVEL OPTIMIZATION

Input frontier: \( v_1, v_2, v_3 \)

Input queue: \( v_4, v_5, v_6, v_7 \)

Explored edges (blue ones are valid ones)

Final output frontier
Other optimizations: Priority Queues, Idempotent Operation, Optimizing Filter, and Output Frontier Storage, etc..
SOFTWARE FRAMEWORK

Application Code

Enactor
(CUDA Kernel Entry)

Cond
Apply
Functor

Graph Data
App Data
Problem

Utilities

Advance
Filter
Operators

Priority Queue

Library Code

Dynamic Cooperative
Load-balanced Partitioning
Direction Optimal

Traversal Optimizations
Performance Evaluation

- 10x better than BGL and PowerGraph;
- Always better than other programmable GPU;
- On par with Ligra and hardwired GPU.
Open Questions:

- What is the right programming model? (Expressivity, Mutable Graph, and Performance Model)
- How to expand the current framework? (To Graph BLAS, Multi-Node GPUs, and Out-of-Core)
What is the right programming model for graph processing on the GPU?
**Open Questions:**

**Expressivity** Does current model cover all operations? How difficult to express one operation using current model?

<table>
<thead>
<tr>
<th>Primitive</th>
<th>Library</th>
<th>Description</th>
<th>Gunrock Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>Help</td>
<td>Remove elements from filter</td>
<td>Filtering</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Filtering+Sort</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+Reduce+Scan+Advance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+Filtering+Sort</td>
</tr>
<tr>
<td>FS</td>
<td>Help</td>
<td>Form Supervertex</td>
<td>Advance</td>
</tr>
<tr>
<td>ANV</td>
<td>Help</td>
<td>Aggregating Neighbor Values</td>
<td>Filtering</td>
</tr>
<tr>
<td>LUV</td>
<td>Help</td>
<td>Local Update of Vertices</td>
<td>Filtering</td>
</tr>
<tr>
<td>UVUOV</td>
<td>Help</td>
<td>Update Vertices Using One Other Vertex</td>
<td>Filtering</td>
</tr>
<tr>
<td>AGV</td>
<td>Help</td>
<td>Aggregate Global Value</td>
<td>Filtering</td>
</tr>
<tr>
<td>Gather</td>
<td>PowerGraph</td>
<td>Aggregating Neighbor Values</td>
<td>Advance</td>
</tr>
<tr>
<td>Apply</td>
<td>PowerGraph</td>
<td>Local Update of Vertices</td>
<td>Filtering</td>
</tr>
<tr>
<td>Scatter</td>
<td>PowerGraph</td>
<td>Update Neighbor Values</td>
<td>Advance</td>
</tr>
</tbody>
</table>
Open Questions:

Expressivity How to support mutable graph? Two sources of mutability:

Algorithm MST, Clustering, Mesh Refinement, etc. need new operators such as mergeEdge, formSuperVertices, reshapeSubgraph, ...

Data Incrementally computation of node ranking or betweenness centrality in time-series graphs. New users and new links in Twitter’s social graph appear constantly. How do we support that?
Open Questions:

Performance Model How to build a performance model to help us improve the current programming model?

- Runtime as a function of # iterations;
- Runtime as a function of {edges, vertices, etc.} touched/traversed;
- Runtime as a function of graph parameters.
Performance Model How to build a performance model to help us improve the current programming model?

- How efficient the programming model is at exploiting parallelism?
- How performance changes as parallelism increases?
- How well we do at GPU-low-level performance metrics? (like degree of memory coalescing, branch coherence, etc.)
- How much computational work/memory bandwidth do we incur?
- How to shift workload from memory access to computation?
Graph BLAS style: How to fit in?
Sparse matrix sparse vector multiplication (SpMSpV): Linear combination of columns specified by nonzero elements of the sparse vector.
Beyond A Single GPU: How to utilize more computing power and larger memory space?

- Single-Node Multi-GPUs
- Out-of-Core
- Multi-Node GPUs
Open Questions:

Single-Node Multi-GPUs

Approach:

- Partition into subgraph, duplicate remote nodes
- Multithreaded host program with multistream support
- Reuse single-node code, partitioner as a plug-in

Issues:

- Best partitioner yet to be found
- Runtime bounded by diameter and # of iterations
- Scaling factor drops?
Single-Node Multi-GPUs

Results:

- BFS scales across multiple GPUs when #iteration is small (<10), #edge is high (>100M) and average degree is high (>80). For the best case, 3.7x speedup using 6 GPUs (kron_n21)
- SSSP 1.3x speedup using 2 GPUs, BC 2.7x speedup using 4 GPUs.
**Open Questions:**

**Out-of-Core**

**Approach:**
- Borrow the Partitioner + Single-Node Gunrock pattern
- Overlap data movement with computation

**Issue:**
- Communication cost between each data chunk would still take up the majority of the time.
Multi-Node Multi-GPUs

- Target: A scalable multi-node layer using MPI
- Approach: Add network communication operation into the single-node multi-GPU version
- Issue: Global partitioner and auxiliary arrays, make good use of NVLink.
Acknowledgment

**Gunrock team**  Yuechao Pan, Yuduo Wu, Carl Yang, Andrew Davidson, Andy Riffel and John D. Owens

**Royal Caliber team**  Erich Elsen and Vishal Vaidyanathan

**NVIDIA**  Duane Merrill, Sean Baxter, the amazing GPU cluster and all other general technical supports

**DARPA**  XDATA Program, Eric Whyne of Data Tactics Corporation, and Dr. Christopher White.
Questions?