Gunrock  High-Performance  Graph  Analytics  for  the  GPU  
Muhammad  Osama  —  University  of  California,  Davis
Why use GPUs for graph processing?
GPUs and Graphs

Graphs

- Found **everywhere**
  - Road & social networks, web, etc.

- Require **fast processing**
  - Memory bandwidth, computing power and GOOD software

- Becoming **very large**
  - Billions of edges

- **Irregular data access pattern** and control flow
  - Limits performance and scalability

GPUs

- Found **everywhere**
  - Data center, desktops, mobiles, etc.

- **Very powerful**
  - High memory bandwidth (900 GBps) and computing power (15.7 TFlops)

- **Limited memory** size
  - 32 GB per NVIDIA V100

- **Difficult to program**
  - Harder to optimize
What is Gunrock?
A CUDA-based graph processing library, aims for performance. State-of-the-art graph processing library.
A **CUDA**-based graph processing library, aims for **Generality**

Covers a **broad range** of graph algorithms
A CUDA-based graph processing library, aims for Programmability

Makes it easy to implement and extend graph algorithms from 1-GPU to multi-GPUs
A CUDA-based graph processing library, aims for

**Scalability**

Fits in (very) limited GPU memory space

**performance scales**

when using many GPUs
Where can you find Gunrock?
High-Performance Graph Primitives on GPUs  [https://gunrock.github.io/](https://gunrock.github.io/)

<table>
<thead>
<tr>
<th>Branch</th>
<th>Master</th>
<th>New Pull Request</th>
<th>Create New File</th>
<th>Upload Files</th>
<th>Find File</th>
<th>Clone or Download</th>
</tr>
</thead>
<tbody>
<tr>
<td>crozhan</td>
<td>Modify device intrinsics to compile on all arches</td>
<td>Latest commit 66e6226 13 days ago</td>
<td>8 months ago</td>
<td>17 days ago</td>
<td>8 months ago</td>
<td>3 months ago</td>
</tr>
</tbody>
</table>
Project’s Workflow

Release *(master)* branch

Development *(dev)* branch

Git Forking Workflow

Contribute GitHub [Issues](#) & [Pull Requests](#)

Apache 2.0 License

Code Coverage [codecov.io](#)

Central Integration [jenkins.io](#)

Documentation slate & doxygen
Project’s Workflow (cont.)

Gunrock's Roadmap
Some Stats and Stuff! (as of 01/30/2020)

- 32 Contributors (over 2500 commits)
- ~600 stars 148 forks
- NVIDIA CUDA-X: GPU Accelerated Library
- RAPIDS
How does Gunrock work?
Programming Model

- **Data-centric** abstraction
- **Bulk-synchronous** programming
Frontier

A *frontier*; group of vertices or edges
Parallel Operators

Manipulation of frontiers is an operation

- Advance
- Filter
- For
- Intersection
- Neighbor-Reduce
- ... and more.

Illustration of **Advance** Operator

*Generates* new frontier by visiting the neighbors.
Bulk-Synchronous Programming

Series of **parallel** operations separated by **global barriers**
Example application in Gunrock.
auto advance_op =
    [distances, weights] __host__ __device__ (...) -> bool {
        auto distance = distances[vertex_id] + weights[edge_id];
        auto old_distance = atomicMin(distances + neighbor_id, distance);

        if (distance < old_distance) return true;
        return false;
    };

auto filter_op =
    [labels, iteration] __host__ __device__ (...) -> bool {
        if (!util::isValid(neighbor_id)) return false;
        return true;
    };

Single-Source Shortest Path

**Launch** the lambdas within the operator call

```cpp
while (!frontier.isEmpty()) {
    oprtr::Advance<oprtr::OprtrType_V2V>(
        graph.csr(), frontier, oprtr_parameters,
        advance_op, filter_op);
}
```
Acknowledgements & Thanks!

NVIDIA AI Laboratory. UC Davis Center for GPU Graph Analytics.


Adobe Data Science Research Award. Scalability and Mutability for Large Streaming Graph Problems on the GPU.

National Science Foundation (Award OAC-1740333) SItxtin(2-SSE: Gunrock: High-Performance GPU Graph Analytics.

National Science Foundation (Award CCF-1637442) Theory and implementation of dynamic data structures for the GPU. Program: AltF---Algorithms in the Field.

National Science Foundation (Award CCF-1629657) PARAGRAPH: Parallel, Scalable Graph Analytics. XPS---Exploiting Parallelism & Scalability.

Department of Defense Advanced Research Projects Agency (DARPA) SBIR SB152-004. Many-Core Acceleration of Common Graph Programming Frameworks. Phase II: award W911NF-16-C-0020.

Department of Defense Advanced Research Projects Agency (DARPA) STTR ST138-004 (“Data-Parallel Analytics on Graphics Processing Units (GPUs)”). A High-Level Operator Abstraction for GPU Graph Analytics. Awards D14PC00023 and D15PC00010.