One Size Doesn’t Fit All: Quantifying Performance Portability of Graph Applications on GPUs

Tyler Sorensen
Princeton University
UC Santa Cruz

Sreepathi Pai
University of Rochester

Alastair F. Donaldson
Imperial College London

November 4, 2019
International Symposium on Workload Characterization (IISWC)
Headlines

*GPUs* and *graph applications* are important *emerging domain*.

- We perform a massive empirical study (240 hours across 6 different GPUs)
- Using a GPU graph application DSL and optimizing compiler, we find:
GPUs and graph applications are important emerging domain.

- We perform a massive empirical study (240 hours across 6 different GPUs)
- Using a GPU graph application DSL and optimizing compiler, we find:

  Compiler optimizations can provide speedups of up to 16x
  and a geomean across the domain of 1.5x
Headlines

**GPUs** and **graph applications** are important **emerging domain**.

• We perform a massive empirical study (240 hours across 6 different GPUs)

• Using a GPU graph application DSL and optimizing compiler, we find:

  Compiler optimizations can provide **speedups** of up to **16x** and a geomean across the domain of **1.5x**

  These optimizations can also provide **slowdowns** of up to **22x**
Headlines

Traditional *performance portability* fall short for graph applications on GPUs

• Previous approaches produce trivial or biased results
Headlines

Traditional *performance portability* fall short for graph applications on GPUs

- Previous approaches produce trivial or biased results

All optimization combinations cause slowdowns **AND** speedups across the domain. **Magnitude-based approaches are biased towards more sensitive GPUs**
Headlines

*Rank-based* statistical procedures offer a new way of thinking about performance portability
**Headlines**

*Rank-based* statistical procedures offer a new way of thinking about performance portability

- Produces non-trivial performance portable optimization combination yielding a *max speedups of 6x*
- Analysis can create *semi-specialized* optimization strategies, which yield greater speedups and *performance critical insights*.  

Tyler Sorensen, IISWC 2019
What is a GPU? (1999 Edition)

The technical definition of a GPU is "a single chip processor with integrated transform, lighting, triangle setup/clipping, and rendering engines that is capable of processing a minimum of 10 million polygons per second."

What is a GPU? (2019 Edition)

• 20 years later, Nvidia’s homepage advertises GPUs without the ability to output graphics!

Trying to Define the Modern GPU

Still used for high-end graphics
Trying to Define the Modern GPU

Still used for high-end graphics

Use in data centers for AI and scientific computing

Tyler Sorensen, IISWC 2019
Trying to Define the Modern GPU

Still used for high-end graphics

Use in data centers for AI and scientific computing

Increasingly used in mobile devices

Tyler Sorensen, IISWC 2019
Trying to Define the Modern GPU

• Programmable vector lanes?
  • Nvidia GPUs have 32 threads per lane
  • Intel GPUs have 8 threads per lane
  • ARM GPUs have 1 thread per lane

• Highly parallel?
  • Nvidia GPUs execute over 10K threads concurrently
  • ARM GPUs execute 500 threads concurrently
What is a GPU?

My best definition:

• High computational efficiency goals
• SIMT programming abstraction (OpenCL)
What is a GPU?

My best definition:

• High computational efficiency goals
• SIMT programming abstraction (OpenCL)

The GPU is:

An exemplar of the architectural Cambrian explosion predicted by Hennessy and Patterson’s 2017 Turing award lecture “The New Golden Age of Computer Architecture”
Graphs (1736 Edition)

• Euler’s Königsberg Bridges
Graphs in 2019

• Size/Growth of modern graphs

Instagram Active Users

Netflix Subscribers

https://techcrunch.com/2018/06/20/instagram-1-billion-users/
Graphs in 2019

• Size/Growth of modern graphs

• Applications:
  • recommendation systems

---

https://techcrunch.com/2018/06/20/instagram-1-billion-users/
Graphs in 2019

• Size/Growth of modern graphs

https://techcrunch.com/2018/06/20/instagram-1-billion-users/

• Applications:
  • recommendation systems
  • (mis)information spread
Performance Portability: Graphs and GPUs

• Privacy at the edge
  • Recommendation systems require intimate shopping/viewing data

• Data collection and latest models in the cloud
  • Community monitoring requires constant computation and model updating

• Increasingly support for both will be required!
This Work

*Characterizing performance portability of Graph applications on GPUs*

**• We Developed:**
  - A portable backend for a GPU graph application DSL and optimizing compiler

**• We Conducted:**
  - A large empirical study, collecting 240 hours of runtime data across 6 GPU

**• We Characterized:**
  - Performance portability in this domain using a rank-based statistical method
A GPU Graph DSL and Compiler

• IrGL : Pai and Pingali, OOPSLA 2016
  • Original work targets only Nvidia GPUs

• First class support for nodes, edges, worklists

• Optimizing compiler
  • Load balancing
  • On-chip synchronization
  • Atomic RMW coalescing
IrGL Optimizations

*Load Balancing*

Graphs have *irregular* parallelism leading to load imbalance

IrGL has 3 transformations to perform load balancing at 3 levels of the GPU hierarchy: Local, Subgroup, Workgroup
IrGL Optimizations

Atomic RMW Coalescing

Graph applications require atomic RMWs to update the worklist for the next iteration.

- RMWs serialize across threads.
- Coalesced RMWs combine RMW operations from several threads, using local communication.

Tyler Sorensen, IISWC 2019
IrGL Optimizations

On-chip Synchronization

Many graph apps are iterative, requiring a global sync between iterations (epochs)

Traditionally GPU sync. involves CPU re-launch

Optimization to do on-chip sync. using experimental global barrier between epochs
## Our Empirical Study

All combinations of above were run

Total runtime of **240 hours**

**Over 10K individual runs**

---

**Applications**

- BFS
- SSSP
- PR
- CC
- MIS
- MST
- TRI

**Inputs**

- Uniform
- RMAT
- NY-Road

**GPUs**

- Nvidia-Quadro
- Nvidia-1080
- AMD-R9
- Intel-Iris
- Intel-HD5500
- ARM-Mali T628

*widest empirical study across GPUs that we are aware of!*

Tyler Sorensen, IISWC 2019
Performance Portability

• Which optimizations should be applied to provide best performance across the entire domain?

<table>
<thead>
<tr>
<th>Optimizations</th>
<th>Applications</th>
<th>Inputs</th>
<th>GPUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>LB - Local</td>
<td>BFS</td>
<td>Uniform</td>
<td>Nvidia-Quadro</td>
</tr>
<tr>
<td>LB - Subgroup</td>
<td>SSSP</td>
<td></td>
<td>Nvidia-1080</td>
</tr>
<tr>
<td>LB - Workgroup</td>
<td>PR</td>
<td></td>
<td>AMD-R9</td>
</tr>
<tr>
<td>OC - Sync</td>
<td>CC</td>
<td>RMAT</td>
<td>Intel-Iris</td>
</tr>
<tr>
<td>RMW-CIs</td>
<td>MIS</td>
<td>NY-Road</td>
<td>Intel-HD5500</td>
</tr>
<tr>
<td></td>
<td>MST</td>
<td></td>
<td>ARM-Mali T628</td>
</tr>
<tr>
<td></td>
<td>TRI</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Optimization Space
(32 options)

Domain
Do No Harm

• Only apply an optimization if it:
  • Does not provide any slowdowns across the entire domain
  • Provides at least one speedup

• Easily to query from our data set, and we found...
Do No Harm

• Only apply an optimization if it:
  • Does not provide any slowdowns across the entire domain
  • Provides at least one speedup

• Easily to query from our data set, and we found…

NOTHING!!!

• All optimizations provided at least one instance of a slowdown
Do the Least Harm

- Relaxation of Do no Harm: Select the optimization combination that caused the fewest slowdowns.

**Fewest slowdowns**

<table>
<thead>
<tr>
<th>Optimizations</th>
<th>36 Slowdowns</th>
<th>60 Speedups,</th>
<th>1.01x Geomean</th>
<th>2x max speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>LB - Local</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LB - Subgroup</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LB - Workgroup</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OC - Sync</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMW-CIs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tyler Sorensen, IISWC 2019
Max Geomean

• Select the optimization combination that provides the highest geomean across the domain

<table>
<thead>
<tr>
<th>Optimizations</th>
<th>49 Slowdowns</th>
<th>66 Speedups, 1.18x Geomean</th>
</tr>
</thead>
</table>

Highest Geomean

<table>
<thead>
<tr>
<th>GPUs</th>
<th># Speedups</th>
<th># Slowdowns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nvidia-Quadro</td>
<td>10</td>
<td>21</td>
</tr>
<tr>
<td>Nvidia-1080</td>
<td>00</td>
<td>16</td>
</tr>
<tr>
<td>AMD-R9</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>Intel-Iris</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>Intel-HD5500</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>ARM-Mali T628</td>
<td>20</td>
<td>5</td>
</tr>
</tbody>
</table>
Max Geomean

- Select the optimization combination that provides the highest geomean across the domain

**Highest Geomean**

<table>
<thead>
<tr>
<th>Optimizations</th>
<th># Speedups</th>
<th># Slowdowns</th>
</tr>
</thead>
<tbody>
<tr>
<td>LB - Local</td>
<td>10</td>
<td>21</td>
</tr>
<tr>
<td>LB - Subgroup</td>
<td>00</td>
<td>16</td>
</tr>
<tr>
<td>LB - Workgroup</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OC - Sync</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>RMW-Cls</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>ARM-Mali T628</td>
<td>20</td>
<td>5</td>
</tr>
</tbody>
</table>
Our Approach: Rank-based

<table>
<thead>
<tr>
<th>Optimization Space</th>
<th>Applications</th>
<th>GPUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>LB – Local</td>
<td>BFS</td>
<td>Nvidia-Quadro</td>
</tr>
<tr>
<td>LB - Subgroup</td>
<td>SSSP</td>
<td>Nvidia-1080</td>
</tr>
<tr>
<td>LB - Workgroup</td>
<td>PR</td>
<td>AMD-R9</td>
</tr>
<tr>
<td>OC - Sync</td>
<td>CC</td>
<td>Intel-Iris</td>
</tr>
<tr>
<td>RMW-Cls</td>
<td>MIS</td>
<td>Intel-HD5500</td>
</tr>
<tr>
<td></td>
<td>MST</td>
<td>ARM-Mali T628</td>
</tr>
<tr>
<td></td>
<td>TRI</td>
<td></td>
</tr>
</tbody>
</table>

Inputs
- Uniform
- RMAT
- NY-Road

Domain

For a single chip, app, input combination, just compare confidence intervals
Our Approach: Rank-based

For a single chip, app, input combination, just compare confidence intervals.

Optimizations:
- LB – Local
- LB - Subgroup
- LB - Workgroup
- OC - Sync
- RMW-Cls

Applications:
- BFS
- SSSP
- PR
- CC
- CC
- MIS
- MST
- TRI

Inputs:
- Uniform
- RMAT
- NY-Road

GPUs:
- Nvidia-Quadro
- Nvidia-1080
- AMD-R9
- Intel-Iris
- Intel-HD5500
- ARM-Mali T628

Optimization Space

Runtime

Opt. Off

Opt. On

Tyler Sorensen, IISWC 2019
Our Approach: Rank-based

For a single chip, app, input combination, just compare confidence intervals

- BFS
- SSSP
- PR
- CC
- MIS
- MST
- TRI

Optimizations
- LB – Local
- LB - Subgroup
- LB - Workgroup
- OC - Sync
- RMW-CIs

Applications

Inputs
- Uniform
- RMAT
- NY-Road

GPUs
- Nvidia-Quadro
- Nvidia-1080
- AMD-R9
- Intel-Iris
- Intel-HD5500
- ARM-Mali T628

 Tyler Sorensen, IISWC 2019
Our Approach: Rank-based

Applications
- BFS
- SSSP
- PR
- CC
- MIS
- MST
- TRI

Optimizations
- LB – Local
- LB - Subgroup
- LB - Workgroup
- OC - Sync
- RMW-CIs

Input
- Uniform
- RMAT
- NY-Road

GPUs
- Nvidia-Quadro
- Nvidia-1080
- AMD-R9
- Intel-Iris
- Intel-HD5500
- ARM-Mali T628

Things become trickier when more chips are added.

Tyler Sorensen, IISWC 2019
Our Approach: Rank-based

First, only consider points whose confidence intervals don’t overlap.
Our Approach: Rank-based

First, only consider points whose confidence intervals don’t overlap
Our Approach: Rank-based

First, only consider points whose confidence intervals don’t overlap

- Inputs
  - Uniform
  - RMAT
  - NY-Road

- Optimization Space
  - LB – Local
  - LB - Subgroup
  - LB - Workgroup
  - OC - Sync
  - RMW-CIs

- Application
  - BFS
  - SSSP
  - PR
  - CC
  - MIS
  - MST
  - TRI

- GPUs
  - Nvidia-Quadro
  - Nvidia-1080
  - AMD-R9
  - Intel-Iris
  - Intel-HD5500
  - ARM-Mali T628

Tyler Sorensen, IISWC 2019
Our Approach: Rank-based

First, only consider points whose confidence intervals don’t overlap.
Our Approach: Rank-based

First, only consider points whose confidence intervals don’t overlap.

Optimizations:
- BFS
- SSSP
- PR
- CC
- MIS
- MST
- TRI
- LB – Local
- LB - Subgroup
- LB - Workgroup
- OC - Sync
- RMW-CIs

Applications:
- Uniform
- RMAT
- NY-Road

GPUs:
- Nvidia-Quadro
- Nvidia-1080
- AMD-R9
- Intel-Iris
- Intel-HD5500
- ARM-Mali T628

Inputs:
- Uniform
- RMAT
- NY-Road

Optimization Space:
- Opt. Off
- Opt. On

Tyler Sorensen, IISWC 2019
Our Approach: Rank-based

First, only consider points whose confidence intervals don’t overlap.
Our Approach: Rank-based

First, only consider points whose confidence intervals don’t overlap

Optimizations
- LB – Local
- LB - Subgroup
- LB - Workgroup
- OC - Sync
- RMW-Cls

Applications
- BFS
- SSSP
- PR
- CC
- MIS
- MST
- TRI

Inputs
- Uniform
- RMAT
- NY-Road

GPUs
- Nvidia-Quadro
- Nvidia-1080
- AMD-R9
- Intel-Iris
- Intel-HD5500
- ARM-Mali T628

Optimization Space

runtime

runtime

Opt. Off

Opt. On

Tyler Sorensen, IISWC 2019
Our Approach: Rank-based

- BFS
- SSSP
- PR
- CC
- MIS
- MST
- TRI

Inputs
- Uniform
- RMAT
- NY-Road

Optimizations
- LB – Local
- LB - Subgroup
- LB - Workgroup
- OC - Sync
- RMW-CIs

Applications

GPUs
- Nvidia-Quadro
- Nvidia-1080
- AMD-R9
- Intel-Iris
- Intel HD5500
- ARM-Mali T628

Domain

Normalize with respect to Opt. Off

Tyler Sorensen, IISWC 2019
Our Approach: Rank-based

Optimizations
- LB – Local
- LB - Subgroup
- LB - Workgroup
- OC - Sync
- RMW-Cls

Applications
- BFS
- SSSP
- PR
- CC
- MIS
- MST
- TRI

Inputs
- Uniform
- RMAT
- NY-Road

GPUs
- Nvidia-Quadro
- Nvidia-1080
- AMD-R9
- Intel-Iris
- Intel HD5500
- ARM-Mali T628

Inputs
- Uniform
- RMAT
- NY-Road

Optimization Space

Opt. On

Tyler Sorensen, IISWC 2019
Our Approach: Rank-based

We now use the Mann-Whitney U test to determine if points are stochastically more likely to be above the horizontal line.

The test is non-parametric: it assumes nothing about the distribution.
Rank-based Results

- Compared to fewest slowdowns, more slowdowns, also more speedups. Higher Geomean and higher max speedup.

<table>
<thead>
<tr>
<th>Fewest slowdowns</th>
<th>Rank-based</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Optimizations</strong></td>
<td><strong>Optimizations</strong></td>
</tr>
<tr>
<td>LB - Local</td>
<td>LB - Local</td>
</tr>
<tr>
<td>LB - Subgroup</td>
<td>LB - Subgroup</td>
</tr>
<tr>
<td>LB - Workgroup</td>
<td>LB - Workgroup</td>
</tr>
<tr>
<td>OC - Sync</td>
<td>OC - Sync</td>
</tr>
<tr>
<td>RMW-Cls</td>
<td>RMW-Cls</td>
</tr>
</tbody>
</table>

- 36 Slowdowns
- 60 Speedups,
- 1.01x Geomean
- 2x max speedup

- 60 Slowdowns
- 66 Speedups,
- 1.15x Geomean
- 6x max speedup
Rank-based Results

- Compared to highest geomean: No more bias against Nvidia GPUs

**Highest Geomean**

<table>
<thead>
<tr>
<th>GPUs</th>
<th># Speedups</th>
<th># Slowdowns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nvidia-Quadro</td>
<td>10</td>
<td>21</td>
</tr>
<tr>
<td>Nvidia-1080</td>
<td>00</td>
<td>16</td>
</tr>
<tr>
<td>AMD-R9</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>Intel-Iris</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>Intel-HD5500</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>ARM-Mali T628</td>
<td>20</td>
<td>5</td>
</tr>
</tbody>
</table>

**Rank-based**

<table>
<thead>
<tr>
<th>GPUs</th>
<th># Speedups</th>
<th># Slowdowns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nvidia-Quadro</td>
<td>22</td>
<td>13</td>
</tr>
<tr>
<td>Nvidia-1080</td>
<td>13</td>
<td>07</td>
</tr>
<tr>
<td>AMD-R9</td>
<td>17</td>
<td>4</td>
</tr>
<tr>
<td>Intel-Iris</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Intel-HD5500</td>
<td>21</td>
<td>12</td>
</tr>
<tr>
<td>ARM-Mali T628</td>
<td>20</td>
<td>04</td>
</tr>
</tbody>
</table>
Semi-specialization per GPU

• Provides 6 different optimization strategies, one per chip:

<table>
<thead>
<tr>
<th>GPUs</th>
<th>LB-Local</th>
<th>LB-Subgroup</th>
<th>LB-Workgroup</th>
<th>OC - Sync</th>
<th>RMW-Cls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nvidia-Quadro</td>
<td>.86</td>
<td>.68</td>
<td>.22</td>
<td>.47</td>
<td>.07</td>
</tr>
<tr>
<td>Nvidia-1080</td>
<td>.86</td>
<td>.78</td>
<td>.32</td>
<td>.22</td>
<td>.19</td>
</tr>
<tr>
<td>AMD-R9</td>
<td>.90</td>
<td>.74</td>
<td>.18</td>
<td>.65</td>
<td>.70</td>
</tr>
<tr>
<td>Intel-Iris</td>
<td>.58</td>
<td>.63</td>
<td>.09</td>
<td>.73</td>
<td>.67</td>
</tr>
<tr>
<td>Intel-HD5500</td>
<td>.54</td>
<td>.56</td>
<td>.12</td>
<td>.63</td>
<td>.41</td>
</tr>
<tr>
<td>ARM-Mali T628</td>
<td>.47</td>
<td>.76</td>
<td>.11</td>
<td>.71</td>
<td>.12</td>
</tr>
</tbody>
</table>
Semi-specialization per GPU

• AMD has widest vector lane, it makes sense that it benefits from coalescing

<table>
<thead>
<tr>
<th>GPUs</th>
<th>LB-Local</th>
<th>LB-Subgroup</th>
<th>LB-Workgroup</th>
<th>OC - Sync</th>
<th>RMW-Cls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nvidia-Quadro</td>
<td>.86</td>
<td>.68</td>
<td>.22</td>
<td>.47</td>
<td>.07</td>
</tr>
<tr>
<td>Nvidia-1080</td>
<td>.86</td>
<td>.78</td>
<td>.32</td>
<td>.22</td>
<td>.19</td>
</tr>
<tr>
<td>AMD-R9</td>
<td>.90</td>
<td>.74</td>
<td>.18</td>
<td>.65</td>
<td><strong>.70</strong></td>
</tr>
<tr>
<td>Intel-Iris</td>
<td>.58</td>
<td>.63</td>
<td>.09</td>
<td>.73</td>
<td>.67</td>
</tr>
<tr>
<td>Intel-HD5500</td>
<td>.54</td>
<td>.56</td>
<td>.12</td>
<td>.63</td>
<td>.41</td>
</tr>
<tr>
<td>ARM-Mali T628</td>
<td>.47</td>
<td>.76</td>
<td>.11</td>
<td>.71</td>
<td>.12</td>
</tr>
</tbody>
</table>

Tyler Sorensen, IISWC 2019
Semi-specialization per GPU

- Nvidia slimmed down kernel launch overhead; no need for on-chip synchronization

<table>
<thead>
<tr>
<th>GPUs</th>
<th>LB-Local</th>
<th>LB-Subgroup</th>
<th>LB-Workgroup</th>
<th>OC - Sync</th>
<th>RMW-Cls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nvidia-Quadro</td>
<td>.86</td>
<td>.68</td>
<td>.22</td>
<td>.47</td>
<td>.07</td>
</tr>
<tr>
<td>Nvidia-1080</td>
<td>.86</td>
<td>.78</td>
<td>.32</td>
<td>.22</td>
<td>.19</td>
</tr>
<tr>
<td>AMD-R9</td>
<td>.90</td>
<td>.74</td>
<td>.18</td>
<td>.65</td>
<td>.70</td>
</tr>
<tr>
<td>Intel-Iris</td>
<td>.58</td>
<td>.63</td>
<td>.09</td>
<td>.73</td>
<td>.67</td>
</tr>
<tr>
<td>Intel-HD5500</td>
<td>.54</td>
<td>.56</td>
<td>.12</td>
<td>.63</td>
<td>.41</td>
</tr>
<tr>
<td>ARM-Mali T628</td>
<td>.47</td>
<td>.76</td>
<td>.11</td>
<td>.71</td>
<td>.12</td>
</tr>
</tbody>
</table>

Tyler Sorensen, IISWC 2019
Semi-specialization per GPU

• Mysterious that ARM balances across subgroups...

<table>
<thead>
<tr>
<th>GPUs</th>
<th>LB-Local</th>
<th>LB-Subgroup</th>
<th>LB-Workgroup</th>
<th>OC - Sync</th>
<th>RMW-Cls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nvidia-Quadro</td>
<td>.86</td>
<td>.68</td>
<td>.22</td>
<td>.47</td>
<td>.07</td>
</tr>
<tr>
<td>Nvidia-1080</td>
<td>.86</td>
<td>.78</td>
<td>.32</td>
<td>.22</td>
<td>.19</td>
</tr>
<tr>
<td>AMD-R9</td>
<td>.90</td>
<td>.74</td>
<td>.18</td>
<td>.65</td>
<td>.70</td>
</tr>
<tr>
<td>Intel-Iris</td>
<td>.58</td>
<td>.63</td>
<td>.09</td>
<td>.73</td>
<td>.67</td>
</tr>
<tr>
<td>Intel-HD5500</td>
<td>.54</td>
<td>.56</td>
<td>.12</td>
<td>.63</td>
<td>.41</td>
</tr>
<tr>
<td>ARM-Mali T628</td>
<td>.47</td>
<td><strong>.76</strong></td>
<td>.11</td>
<td>.71</td>
<td>.12</td>
</tr>
</tbody>
</table>

Tyler Sorensen, IISWC 2019
Semi-specialization per GPU

• Mysterious that ARM balances across subgroups...

<table>
<thead>
<tr>
<th>GPUs</th>
<th>LB-Local</th>
<th>LB-Subgroup</th>
<th>LB-Workgroup</th>
<th>OC - Sync</th>
<th>RMW-Cls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nvidia-Quadro</td>
<td>.86</td>
<td>.68</td>
<td>.22</td>
<td>.47</td>
<td>.07</td>
</tr>
<tr>
<td>Nvidia-1080</td>
<td>.86</td>
<td>.78</td>
<td>.32</td>
<td>.22</td>
<td>.19</td>
</tr>
<tr>
<td>AMD-R9</td>
<td>.90</td>
<td>.74</td>
<td>.18</td>
<td>.65</td>
<td>.70</td>
</tr>
<tr>
<td>Intel-Iris</td>
<td>.58</td>
<td>.63</td>
<td>.09</td>
<td>.73</td>
<td>.67</td>
</tr>
<tr>
<td>Intel-HD5500</td>
<td>.54</td>
<td>.56</td>
<td>.12</td>
<td>.63</td>
<td>.41</td>
</tr>
<tr>
<td>ARM-Mali T628</td>
<td>.47</td>
<td><strong>.76</strong></td>
<td>.11</td>
<td>.71</td>
<td>.12</td>
</tr>
</tbody>
</table>

• Turns out it is because of “memory divergence”!

Tyler Sorensen, IISWC 2019
Conclusion

- **GPUs** and **graph applications** are important **emerging domain**.
  - We perform a massive empirical study (240 hours across 6 different GPUs)

- Traditional **performance portability** fall short in this domain.

- **Rank-based** statistical procedures offer a new way of thinking about performance portability

Tyler Sorensen
https://twitter.com/Tyler_UCSC
https://www.cs.princeton.edu/~ts20/
Extra Slides Start
Impact on GPU Programming Languages

• Working with Khronos group to better specify a progress model that allows on-chip synchronization (OC-Sync)

Rank-based Global Optimizations

- LB - Local
- LB - Subgroup
- LB - Workgroup
- OC - Sync
- RMW-CIs

60 Slowdowns
66 Speedups, 1.15x Geomean
6x max speedup
Semi-specialization in Other Dimensions

- Semi-specialized optimizations for chip, application, and graph input

---

Tyler Sorensen, IISWC 2019
Do the Least Harm

• Relaxation of Do no Harm: Select the optimization combination that caused the fewest slowdowns.

<table>
<thead>
<tr>
<th>Optimizations</th>
<th>Fewest slowdowns</th>
<th>Most Slowdowns</th>
</tr>
</thead>
<tbody>
<tr>
<td>LB - Local</td>
<td>36 Slowdowns</td>
<td>195 Slowdowns</td>
</tr>
<tr>
<td>LB - Subgroup</td>
<td>60 Speedups,</td>
<td>22 Speedups,</td>
</tr>
<tr>
<td>LB - Workgroup</td>
<td>1.01x Geomean</td>
<td>.53x Geomean</td>
</tr>
<tr>
<td>OC - Sync</td>
<td>2x max speedup</td>
<td>2x max speedup</td>
</tr>
<tr>
<td>RMW-Cls</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
At First Glance – IrGL Optimizations

• **The Good:** Fantastic Speedups!
  - *Optimizations achieved up to a 16x speedup for AMD*
  - Speedups of over 10x on Intel chips
  - Geomean of 1.5x top speedups

• **The Bad:** Horrible Slowdowns!
  - *Slowdowns of up to 22x on Intel GPUs for some “optimizations”*
  - Other GPUs suffered slowdowns of at least 8x

• **The Ugly:** Performance Portability?
  - How to tame this area?
A GPU Graph DSL and Compiler

- IrGL : Pai and Pingali, OOPSLA 2016
  - Original work targets only Nvidia GPUs

- First class support for nodes, edges, worklists

- Optimizing compiler
  - Load balancing
  - On-chip synchronization
  - Atomic RMW coalescing

Tyler Sorensen, IISWC 2019
A GPU Graph DSL and Compiler

• IrGL : Pai and Pingali, OOPSLA 2016
  • Original work targets only Nvidia GPUs

• First class support for nodes, edges, worklists

• Optimizing compiler
  • Load balancing
  • On-chip synchronization
  • Atomic RMW coalescing
A GPU Graph DSL and Compiler

- IrGL : Pai and Pingali, OOPSLA 2016
  - Original work targets only Nvidia GPUs

- First class support for nodes, edges, worklists

- Optimizing compiler
  - Load balancing
  - On-chip synchronization
  - Atomic RMW coalescing

Tyler Sorensen, IISWC 2019
A GPU Graph DSL and Compiler

- IrGL : Pai and Pingali, OOPSLA 2016
  - Original work targets only Nvidia GPUs

- First class support for nodes, edges, worklists

- Optimizing compiler
  - Load balancing
  - On-chip synchronization
  - Atomic RMW coalescing
A GPU Graph DSL and Compiler

- IrGL : Pai and Pingali, OOPSLA 2016
  - Original work targets only Nvidia GPUs

- First class support for nodes, edges, worklists

- Optimizing compiler
  - Load balancing
  - On-chip synchronization
  - Atomic RMW coalescing
A GPU Graph DSL and Compiler

• IrGL : Pai and Pingali, OOPSLA 2016
  • Original work targets only Nvidia GPUs

• First class support for nodes, edges, worklists

• Optimizing compiler
  • Load balancing
  • On-chip synchronization
  • Atomic RMW coalescing
A GPU Graph DSL and Compiler

• IrGL : Pai and Pingali, OOPSLA 2016
  • Original work targets only Nvidia GPUs

• First class support for nodes, edges, worklists

• Optimizing compiler
  • Load balancing
  • On-chip synchronization
  • Atomic RMW coalescing