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:

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***

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**.*

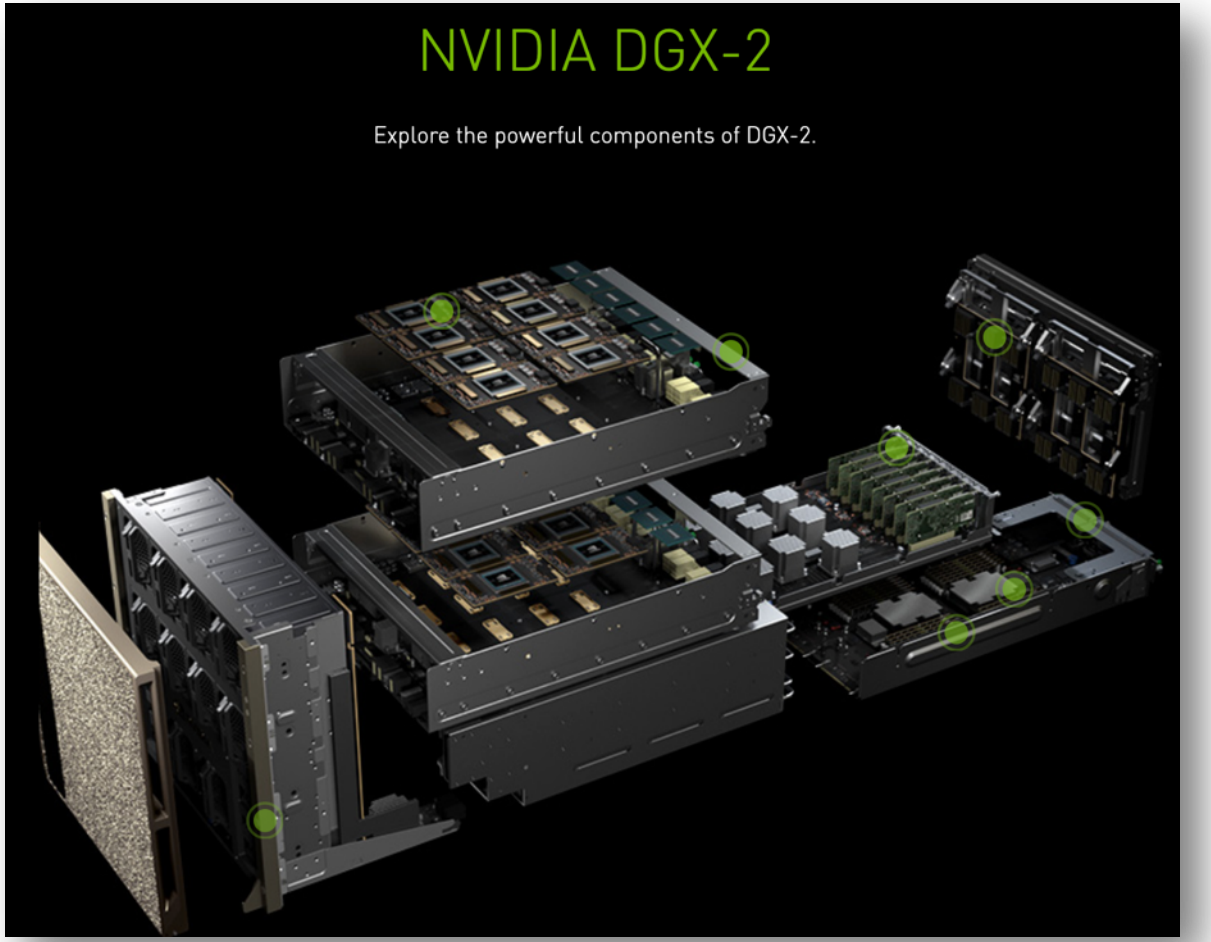
What is a GPU? (1999 Edition)

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."

https://web.archive.org/web/20160408122443/http://www.nvidia.com/object/gpu.html

What is a GPU? (2019 Edition)

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



https://www.nvidia.com/en-us/data-center/dgx-2/

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

Trying to Define the Modern GPU

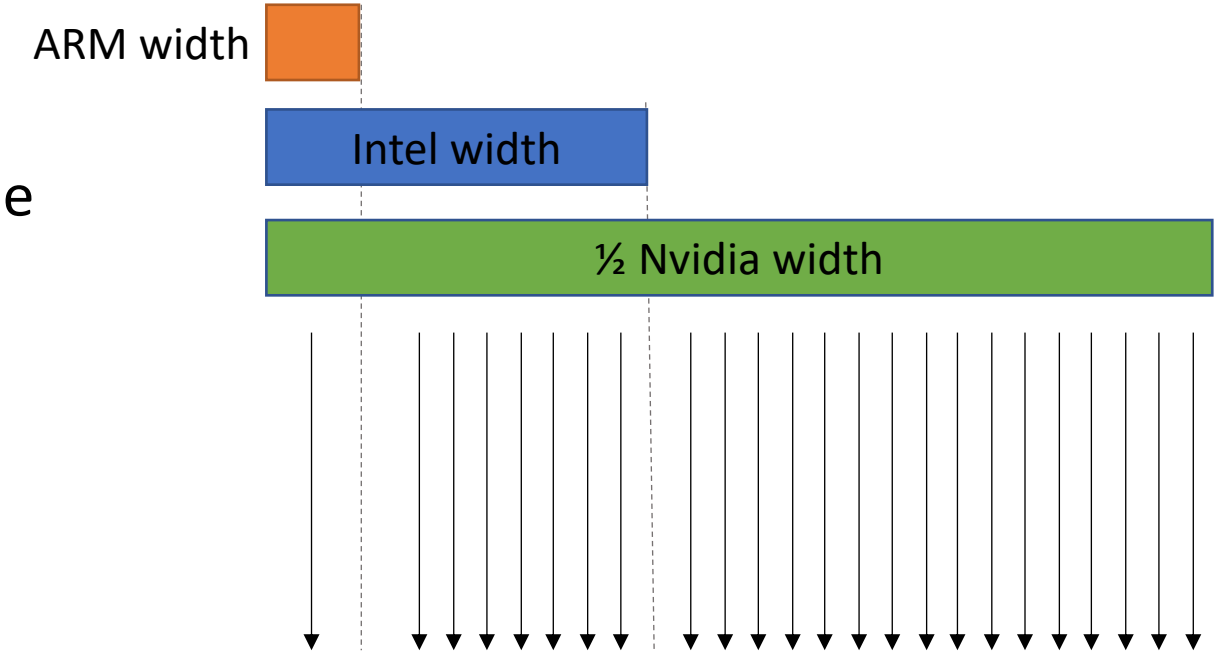
Still used for high-end graphics

Use in data centers for AI and scientific computing

Increasingly used in mobile devices

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)

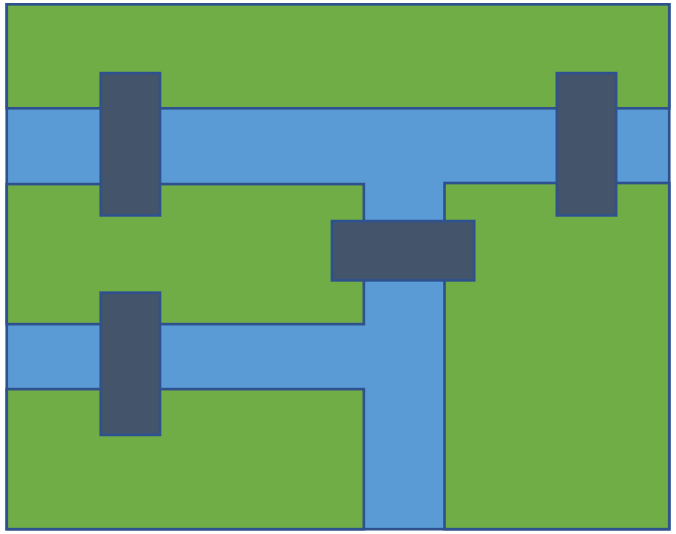
The GPU is:

An exemplar of the architectural Cambrian explosion predicted by Hennessy and Patterson's 2017 Turing award lecture "The New Golden Age of Computer Architecture"

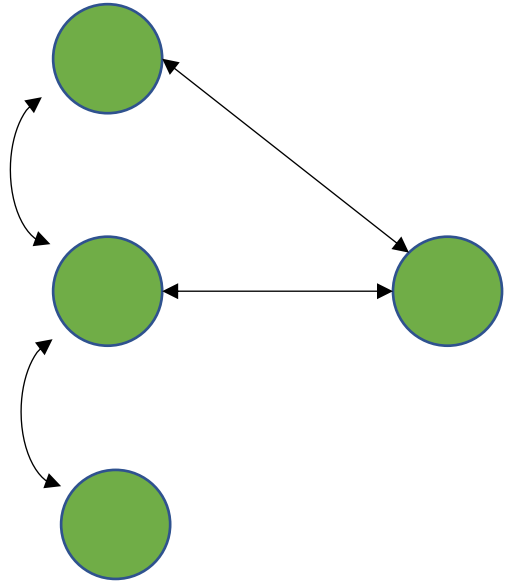
Graphs (1736 Edition)

- Euler's Königsberg Bridges

Modern day



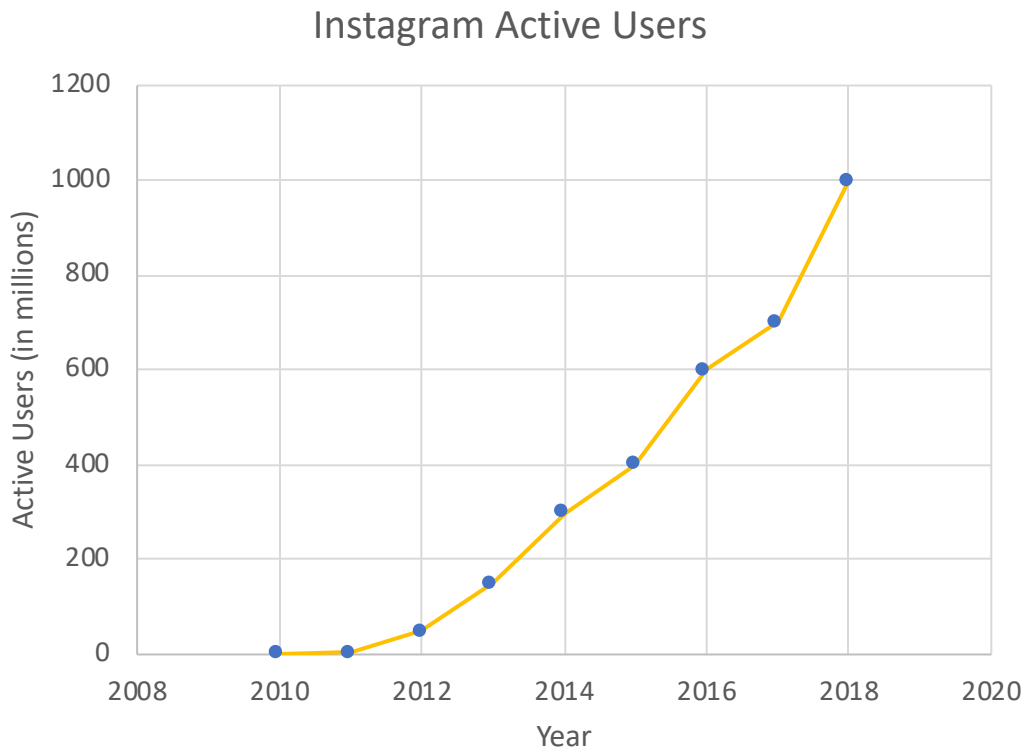
Abstract View

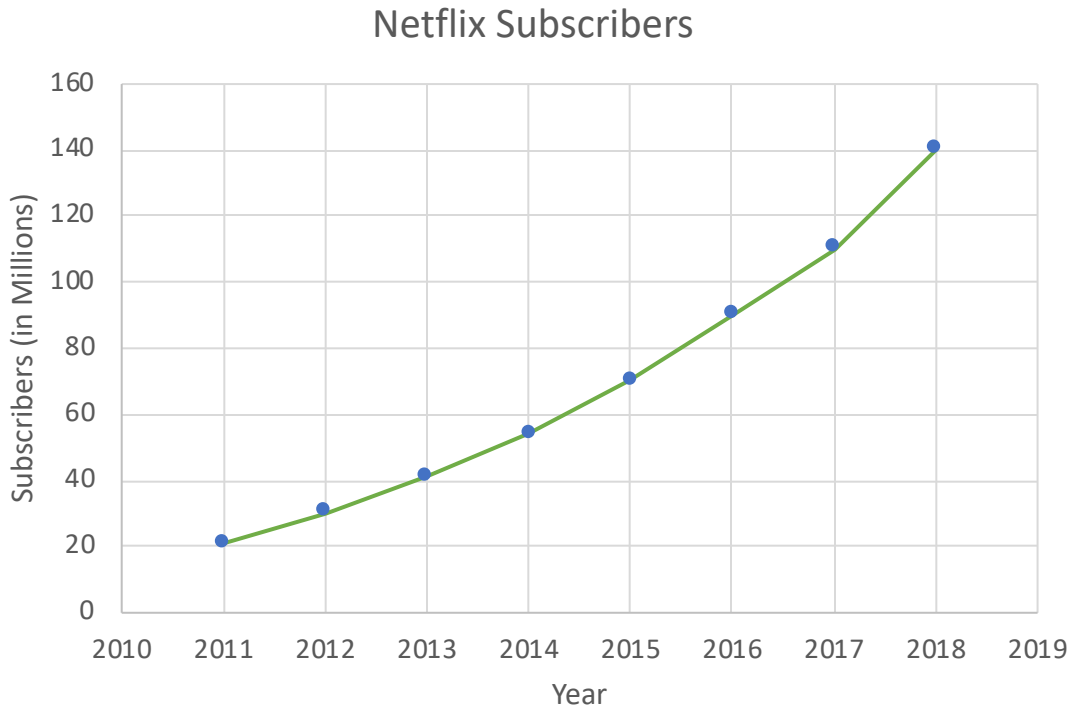


As a Graph

Graphs in 2019

- Size/Growth of modern graphs

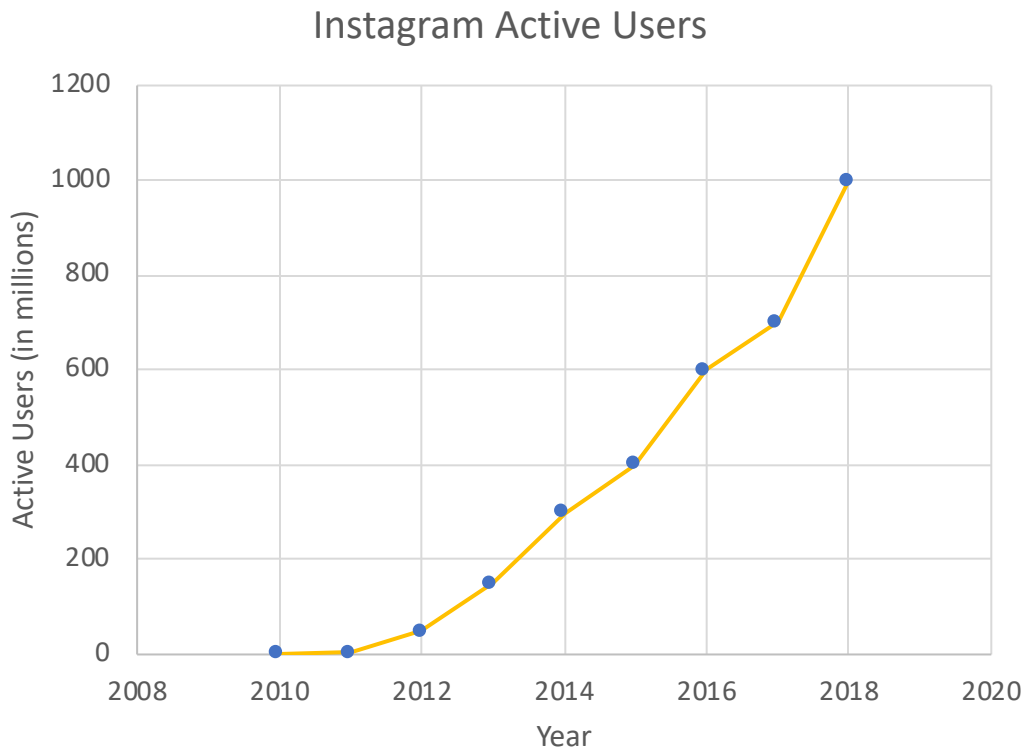


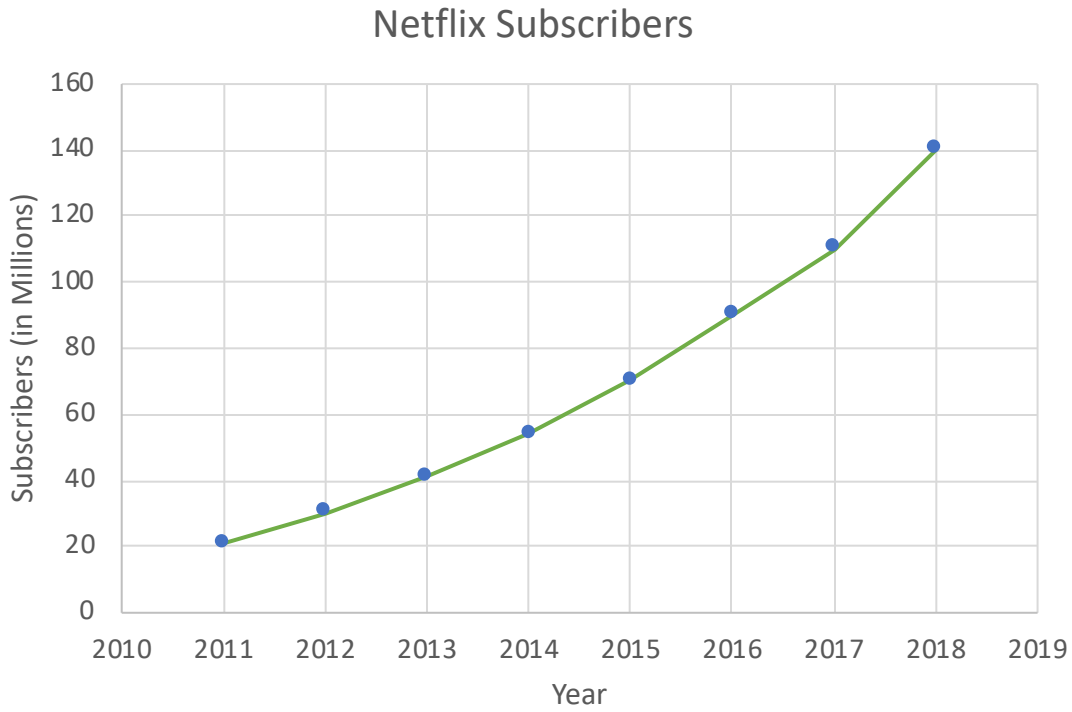


https://techcrunch.com/2018/06/20/instagram-1-billion-users/
https://www.statista.com/statistics/250934/quarterly-number-of-netflix-streaming-subscribers-worldwide/

Graphs in 2019

- Size/Growth of modern graphs



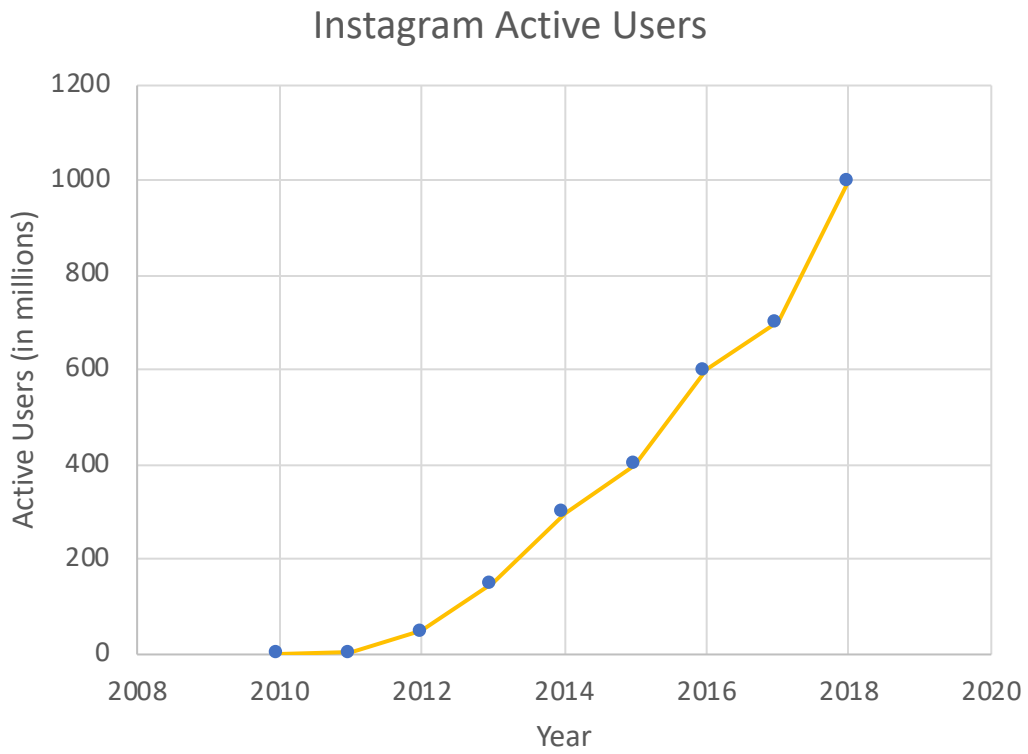


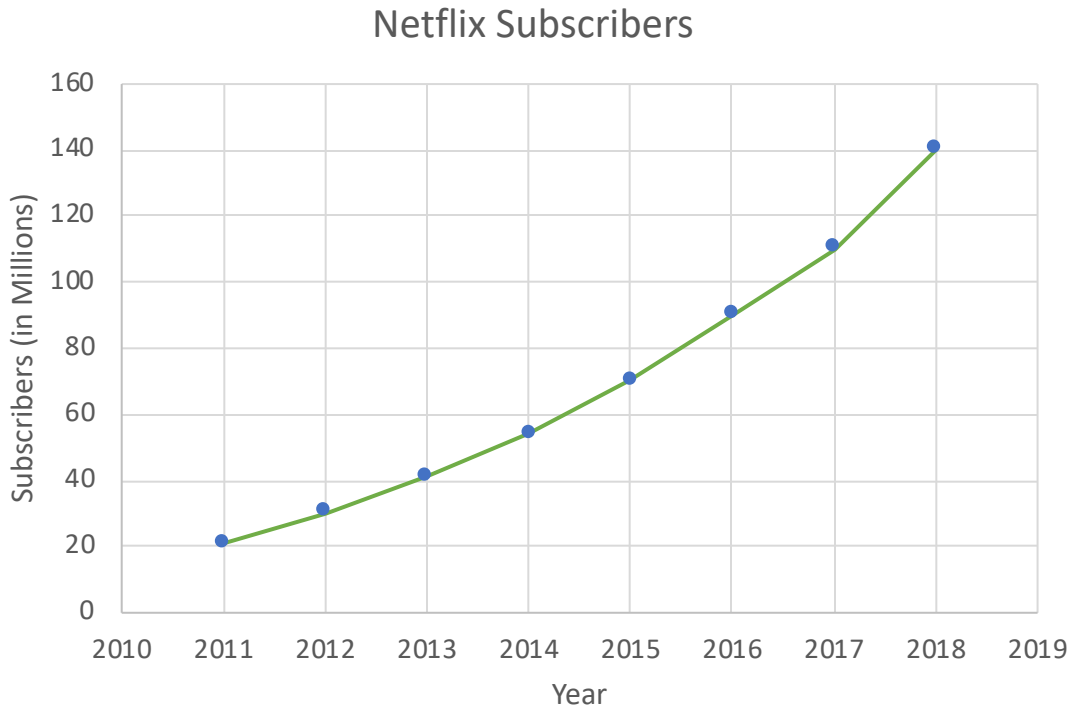
- Applications:
  - recommendation systems

https://techcrunch.com/2018/06/20/instagram-1-billion-users/
https://www.statista.com/statistics/250934/quarterly-number-of-netflix-streaming-subscribers-worldwide/

Graphs in 2019

- Size/Growth of modern graphs





- Applications:
  - recommendation systems
  - (mis)information spread

https://techcrunch.com/2018/06/20/instagram-1-billion-users/
https://www.statista.com/statistics/250934/quarterly-number-of-netflix-streaming-subscribers-worldwide/

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!***

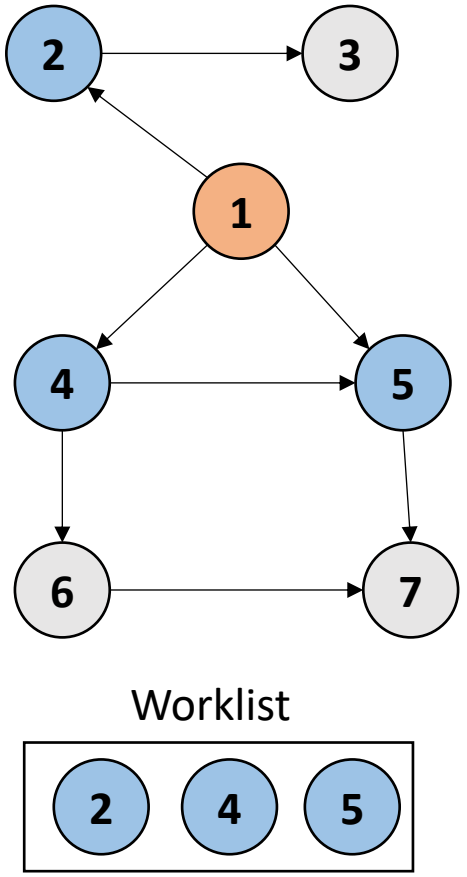
This Work

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

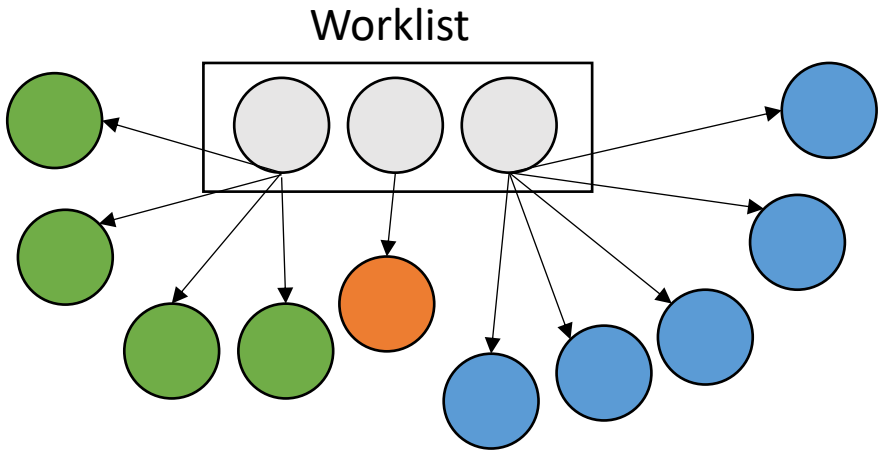
- IrGL : Pai and Pingali, OOPSLA 2016
  - Original work targets only Nvidia GPUs
- First class support for nodes, edges, worklists
- Optimizing compiler
  - Load balancing
  - On-chip synchronization
  - Atomic RMW coalescing

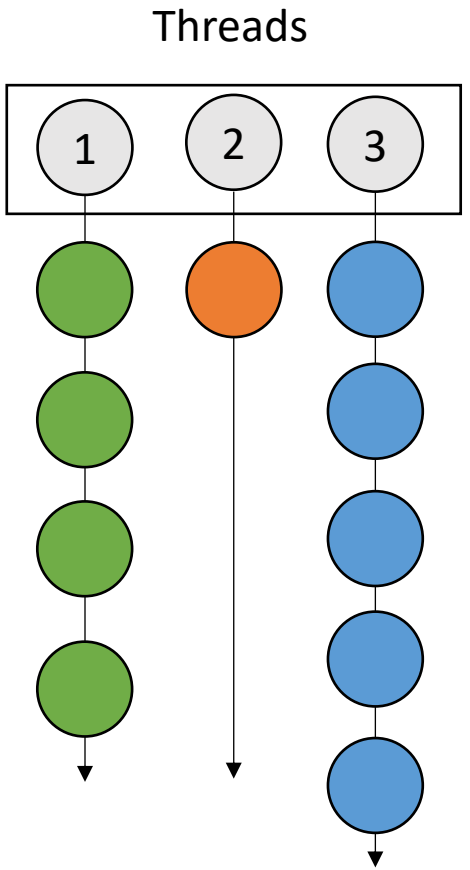


IrGL Optimizations

Load Balancing

Graphs have *irregular* parallelism leading to load imbalance



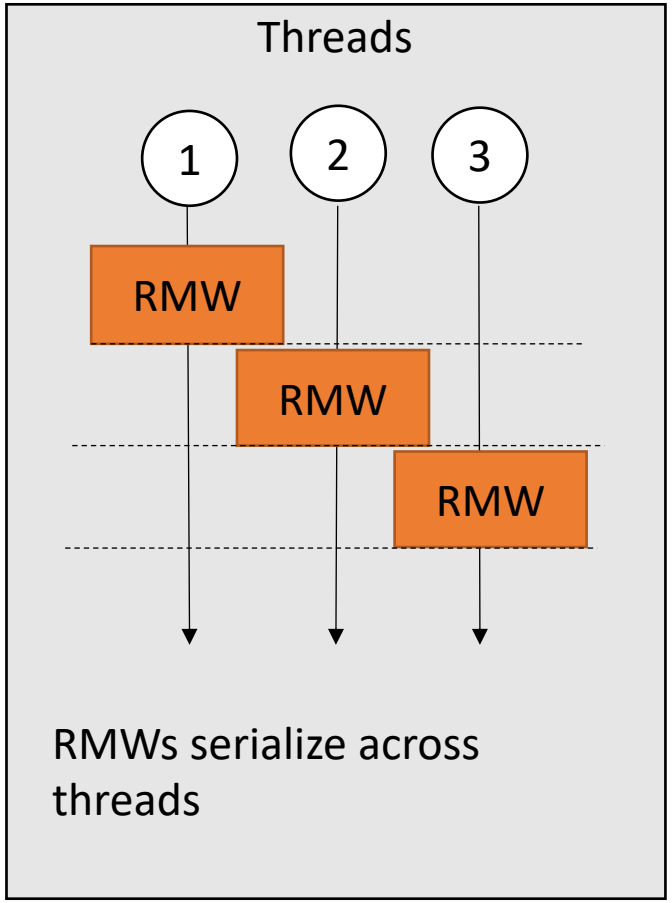


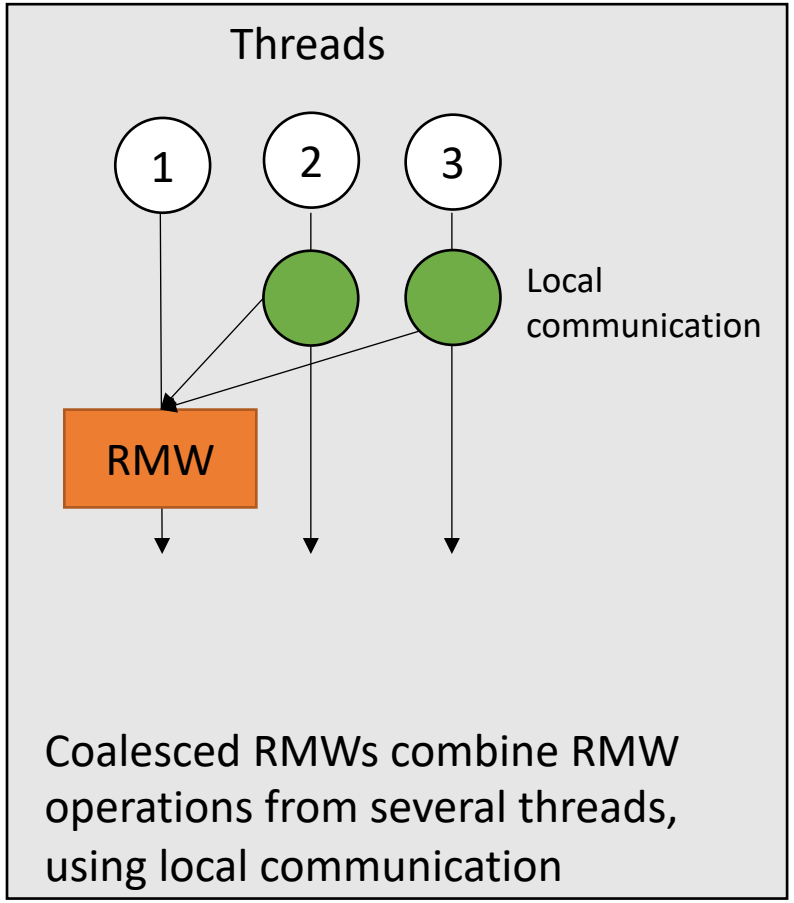
IrGL has 3 transformations to perform load balancing at 3 levels of the GPU hierarchy: Local, Subgroup, Workgroup

IrGL Optimizations

Atomic RMW Coalescing

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

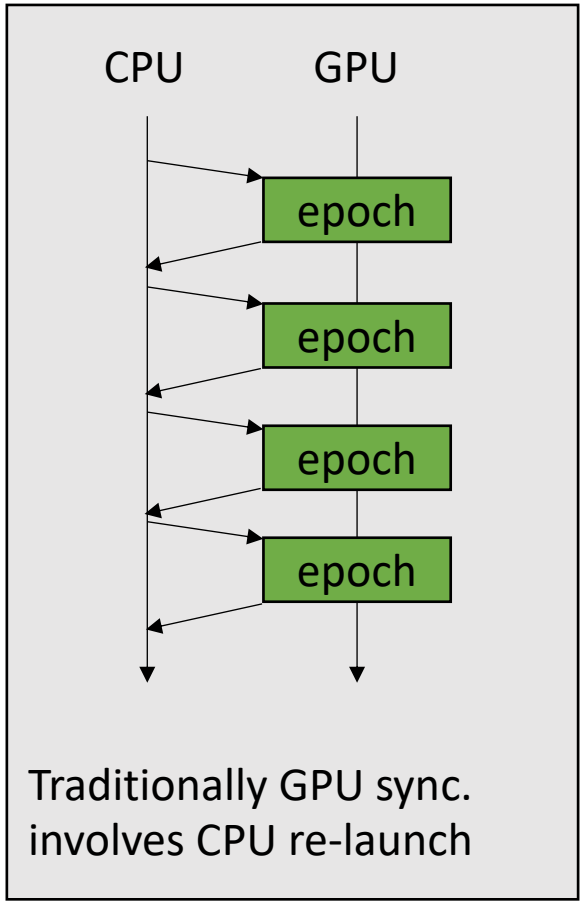




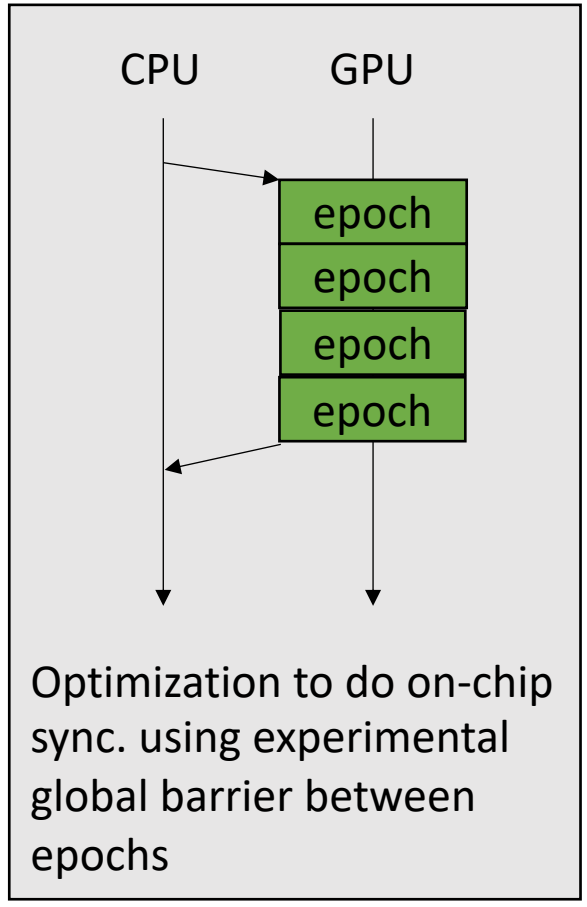
IrGL Optimizations

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

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

All combinations of above were run

Total runtime of **240 hours**

Over 10K individual runs

widest empirical study across GPUs that we are aware of!

Performance Portability

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

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

Optimization Space
(32 options)

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

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

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

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

Max Geomean

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

Highest Geomean

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

49 Slowdowns
66 Speedups,
1.18x Geomean

GPUs	# Speedups	# Slowdowns
Nvidia-Quadro	10	21
Nvidia-1080	00	16
AMD-R9	12	3
Intel-Iris	10	2
Intel-HD5500	14	2
ARM-Mali T628	20	5

Max Geomean

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

Highest Geomean

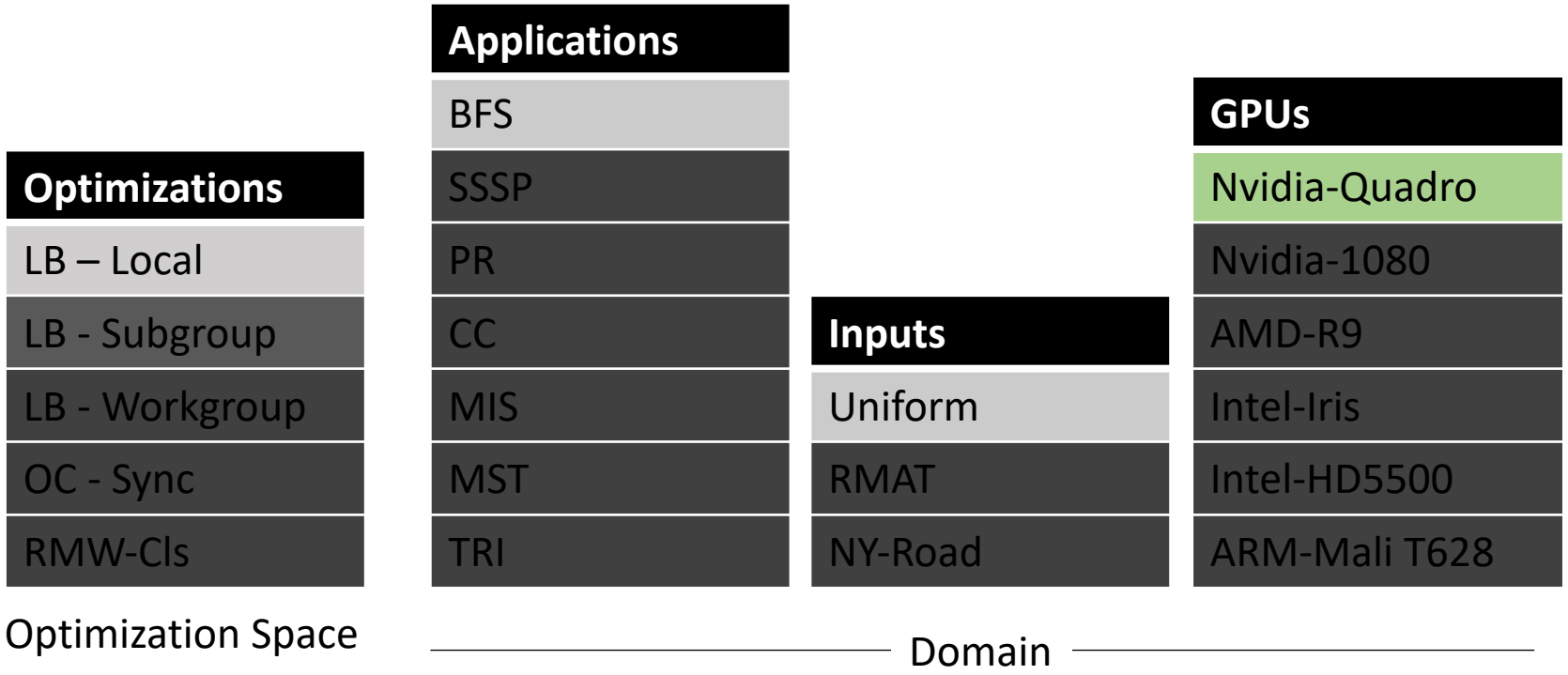
Optimizations
LB - Local
LB - Subgroup
LB - Workgroup
OC - Sync
RMW-Cls

49 Slowdowns
66 Speedups,
1.18x Geomean

GPUs	# Speedups	# Slowdowns
Nvidia-Quadro	10	21
Nvidia-1080	00	16
AMD-R9	12	3
Intel-Iris	10	2
Intel-HD5500	14	2
ARM-Mali T628	20	5

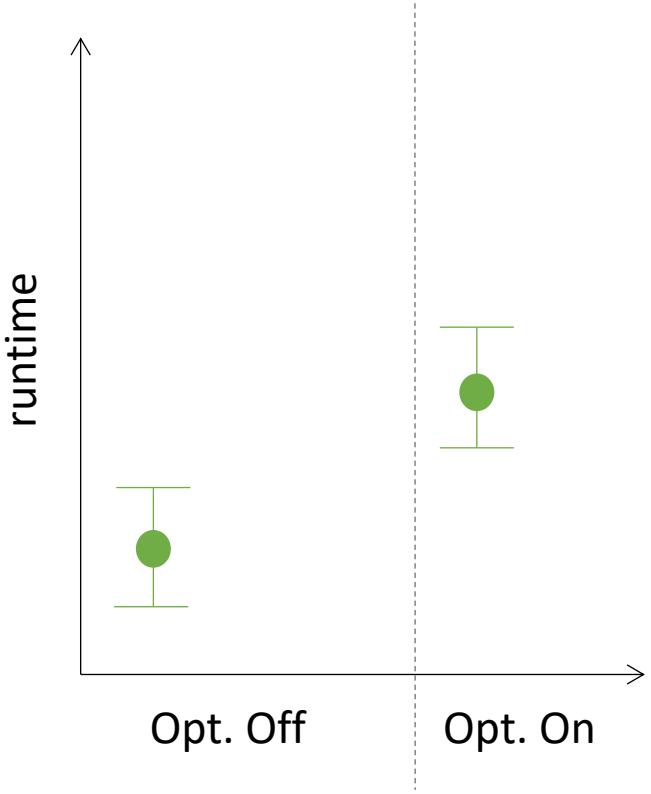
Our Approach: Rank-based

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

Optimization Space

Applications
BFS
SSSP
PR
CC
MIS
MST
TRI

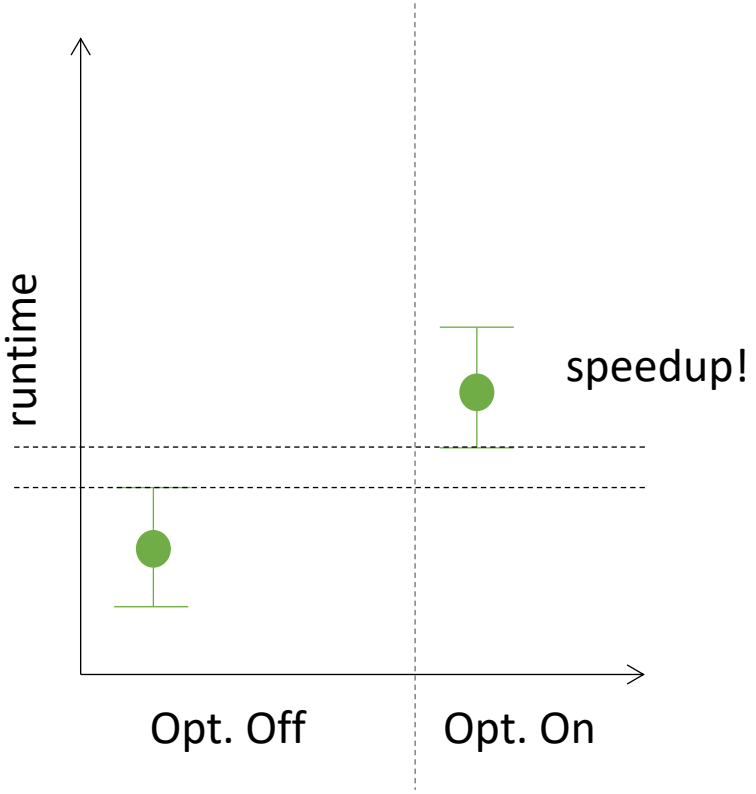
Inputs
Uniform
RMAT
NY-Road

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

Domain

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

Optimization Space

Applications
BFS
SSSP
PR
CC
MIS
MST
TRI

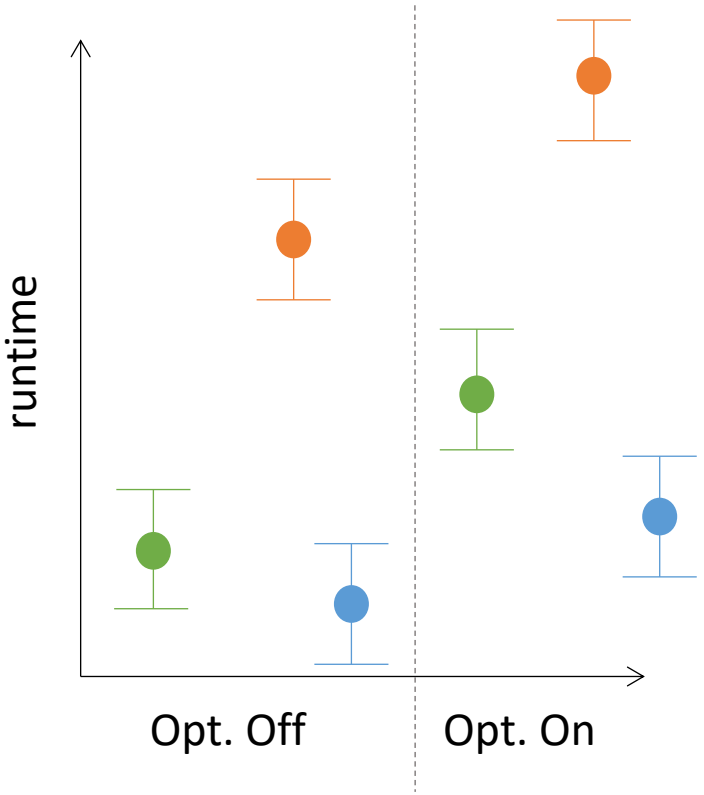
Inputs
Uniform
RMAT
NY-Road

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

Domain

Our Approach: Rank-based

Things become trickier when more chips are added



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

Optimization Space

Applications
BFS
SSSP
PR
CC
MIS
MST
TRI

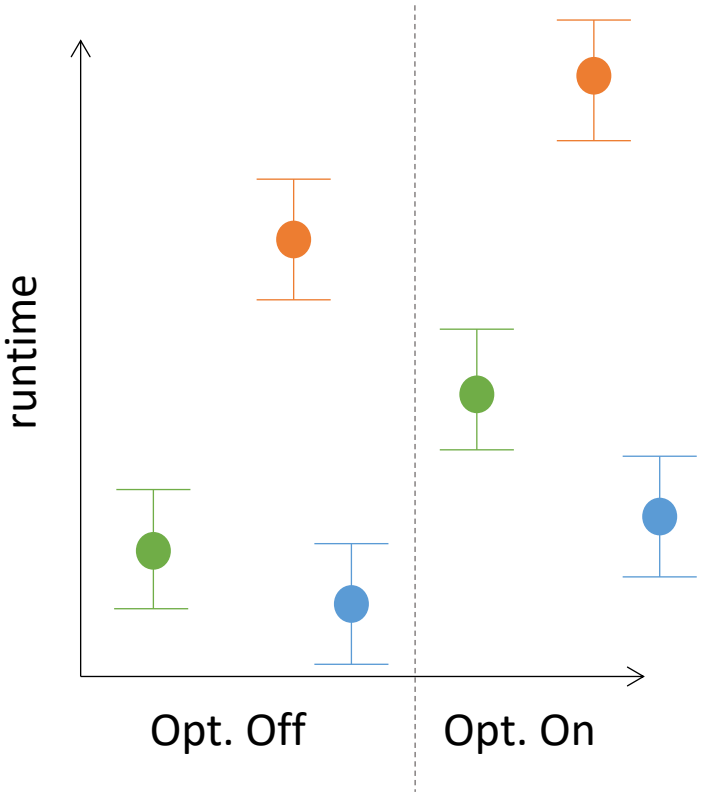
Inputs
Uniform
RMAT
NY-Road

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

Domain

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

Optimization Space

Applications
BFS
SSSP
PR
CC
MIS
MST
TRI

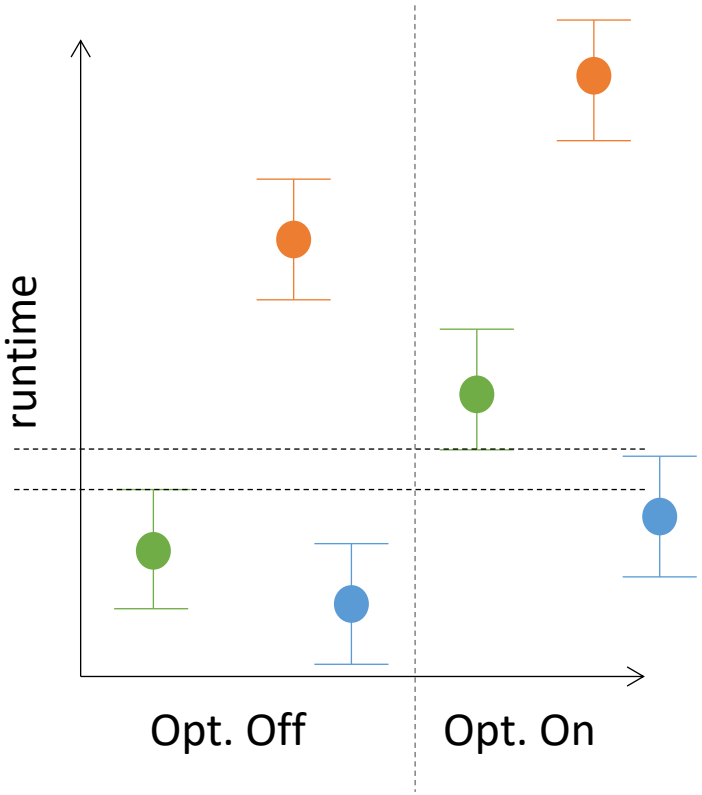
Inputs
Uniform
RMAT
NY-Road

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

Domain

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

Optimization Space

Applications
BFS
SSSP
PR
CC
MIS
MST
TRI

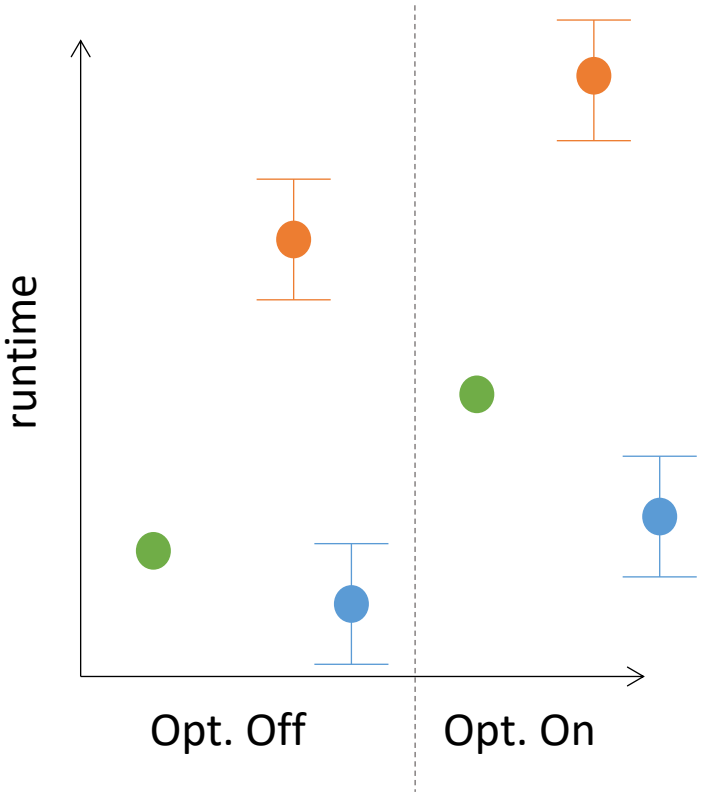
Inputs
Uniform
RMAT
NY-Road

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

Domain

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

Optimization Space

Applications
BFS
SSSP
PR
CC
MIS
MST
TRI

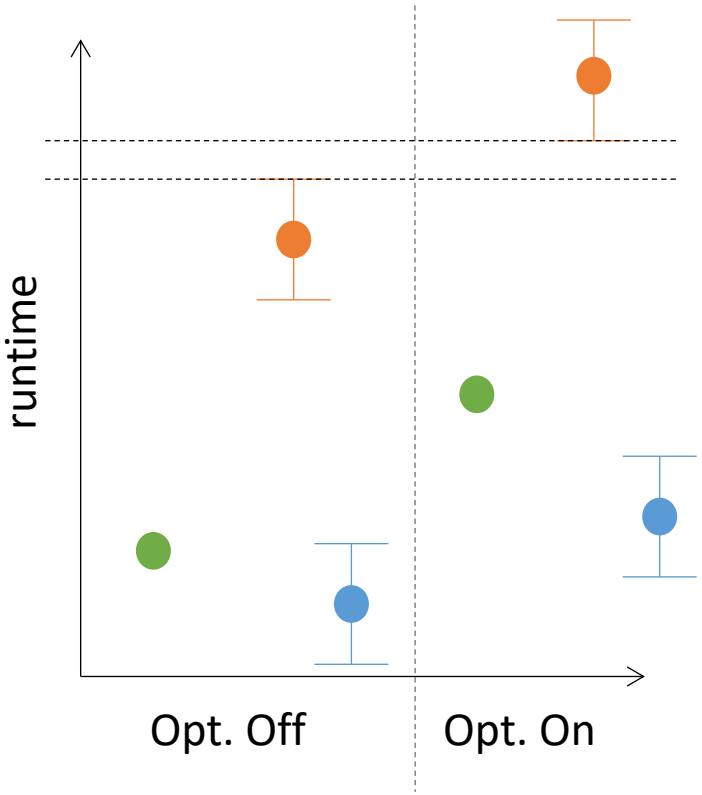
Inputs
Uniform
RMAT
NY-Road

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

Domain

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

Optimization Space

Applications
BFS
SSSP
PR
CC
MIS
MST
TRI

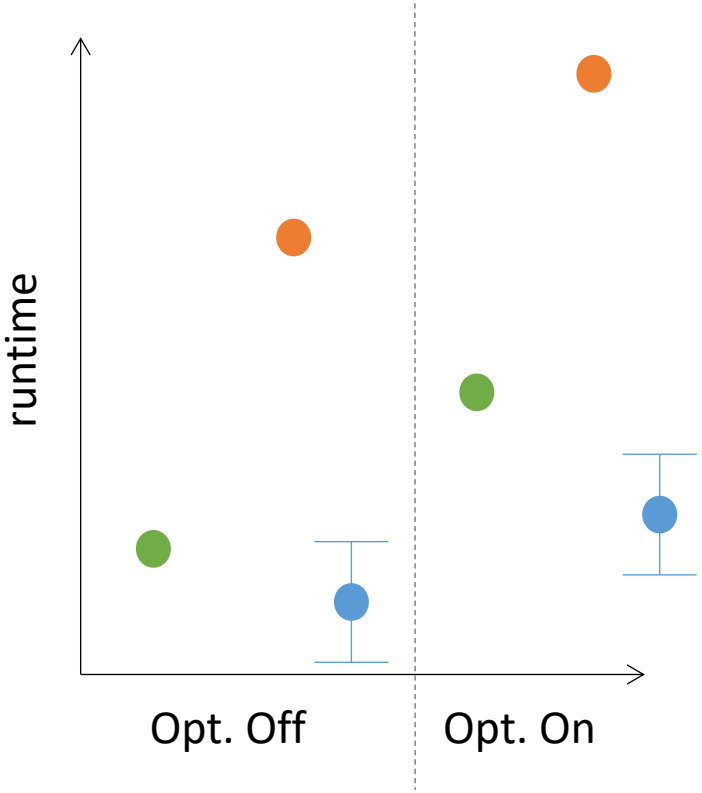
Inputs
Uniform
RMAT
NY-Road

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

Domain

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

Optimization Space

Applications
BFS
SSSP
PR
CC
MIS
MST
TRI

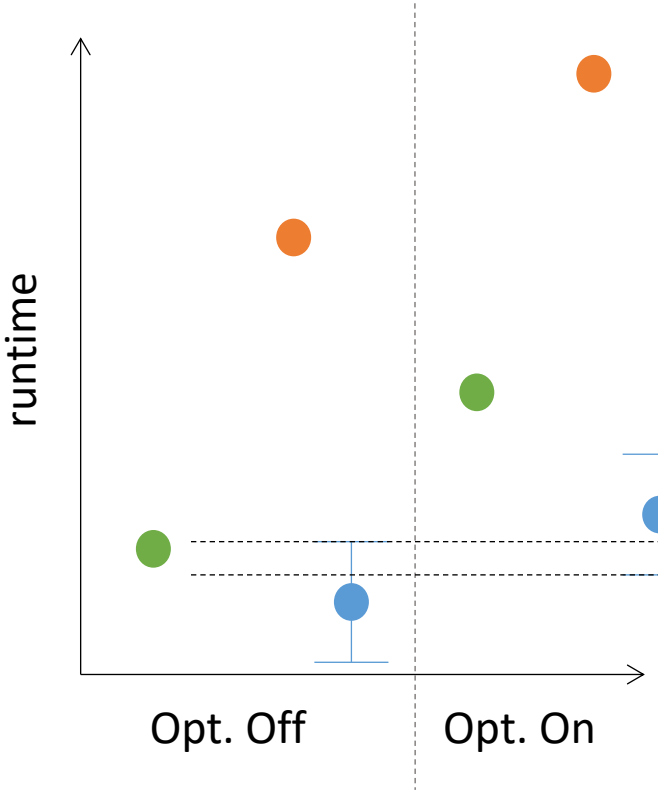
Inputs
Uniform
RMAT
NY-Road

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

Domain

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

Optimization Space

Applications
BFS
SSSP
PR
CC
MIS
MST
TRI

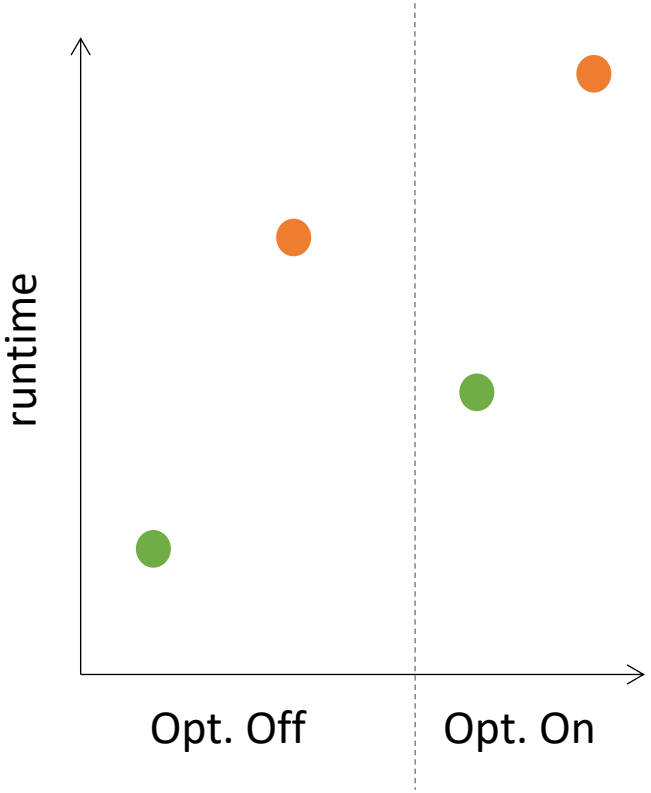
Inputs
Uniform
RMAT
NY-Road

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

Domain

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

Optimization Space

Applications
BFS
SSSP
PR
CC
MIS
MST
TRI

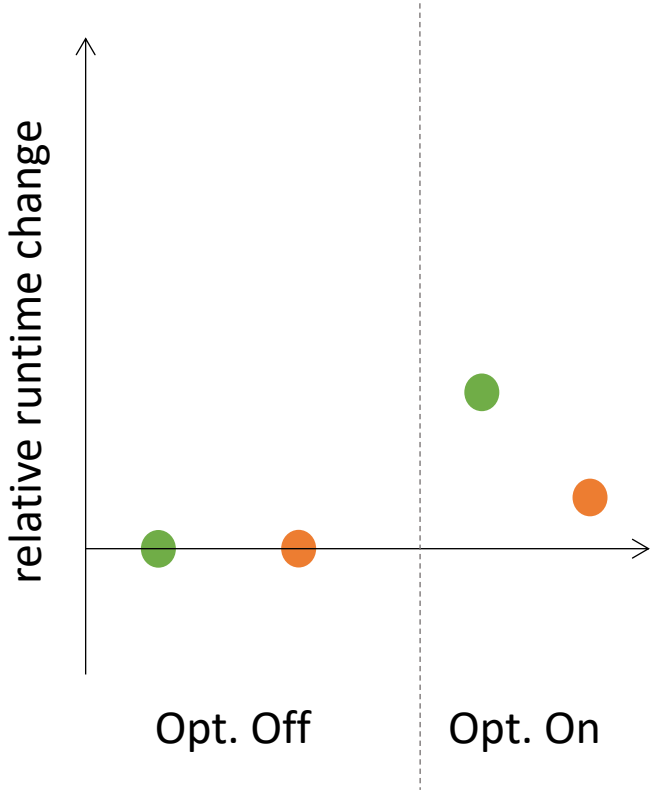
Inputs
Uniform
RMAT
NY-Road

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

Domain

Our Approach: Rank-based

Normalize with respect to Opt. Off



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

Optimization Space

Applications
BFS
SSSP
PR
CC
MIS
MST
TRI

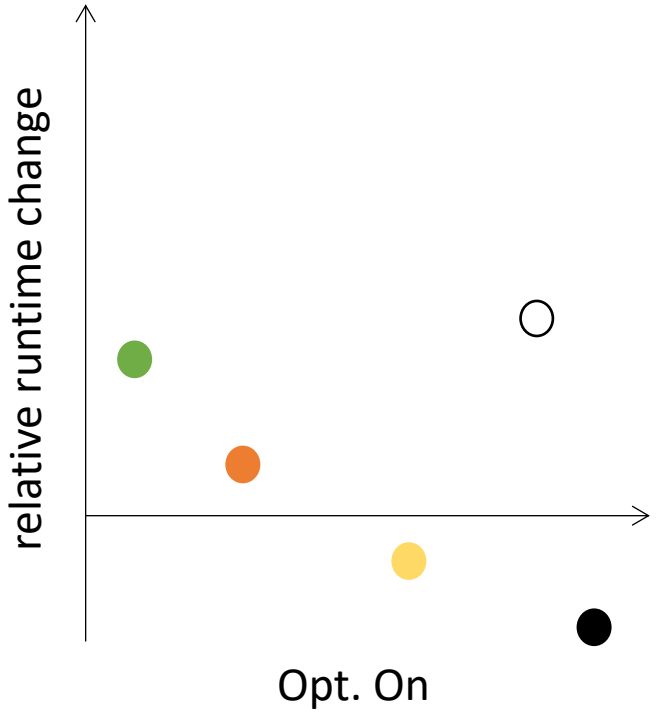
Inputs
Uniform
RMAT
NY-Road

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

Domain

Our Approach: Rank-based

Only consider relative *Opt. On* points, we can show more now visually



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

Optimization Space

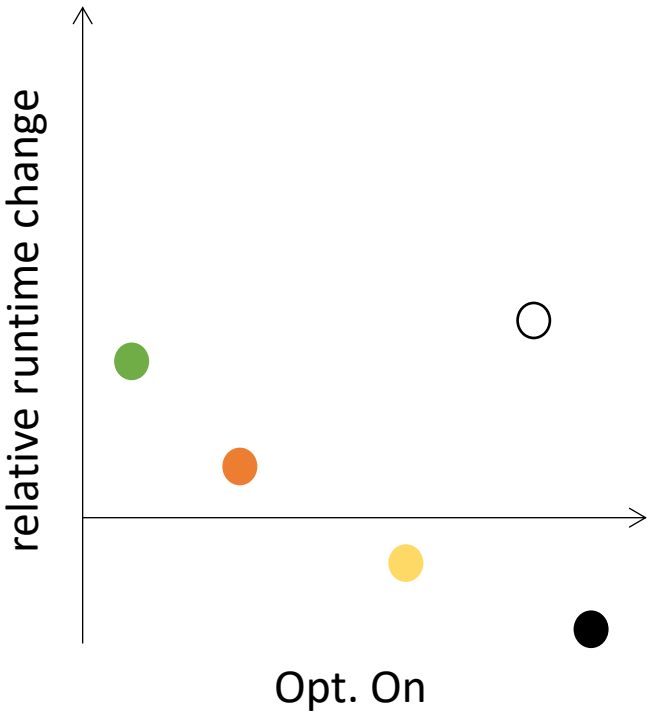
Applications
BFS
SSSP
PR
CC
MIS
MST
TRI

Inputs
Uniform
RMAT
NY-Road

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

Domain

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

Fewest slowdowns

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

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

Rank-based

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

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

GPUs	# Speedups	# Slowdowns
Nvidia-Quadro	10	21
Nvidia-1080	00	16
AMD-R9	12	3
Intel-Iris	10	2
Intel-HD5500	14	2
ARM-Mali T628	20	5

Rank-based

GPUs	# Speedups	# Slowdowns
Nvidia-Quadro	22	13
Nvidia-1080	13	07
AMD-R9	17	4
Intel-Iris	10	10
Intel-HD5500	21	12
ARM-Mali T628	20	04

Semi-specialization per GPU

- Provides 6 different optimization strategies, one per chip:

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

Semi-specialization per GPU

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

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

Semi-specialization per GPU

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

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

Semi-specialization per GPU

- Mysterious that ARM balances across subgroups…

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

Semi-specialization per GPU

- Mysterious that ARM balances across subgroups…

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

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

Conclusion

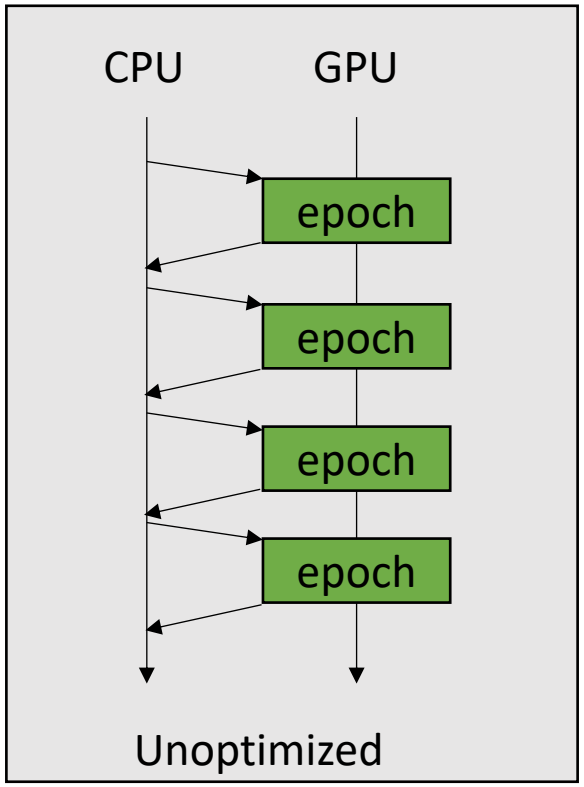
- ***GPUs*** and ***graph applications*** are important ***emerging domain***.
  - We perform a massive empirical study (240 hours across 6 different GPUs)
- Traditional ***performance portability*** fall short in this domain.
- ***Rank-based*** statistical procedures offer a new way of thinking about performance portability

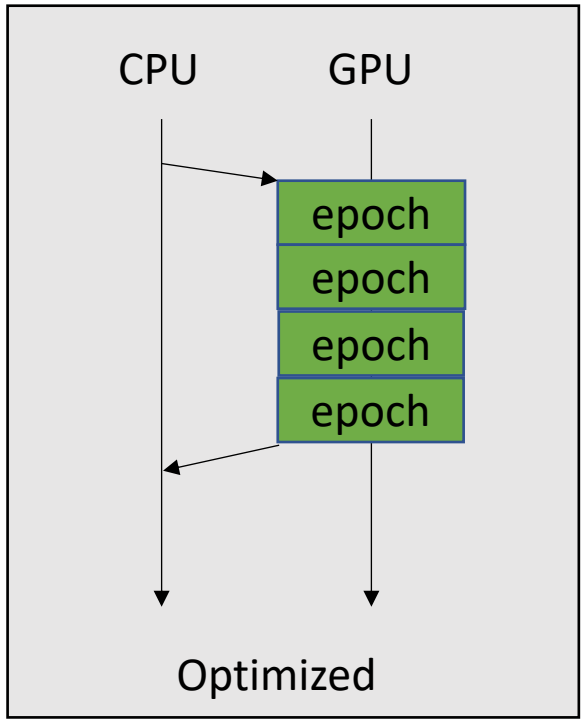
Tyler Sorensen
https://twitter.com/Tyler_UCSC
https://www.cs.princeton.edu/~ts20/

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

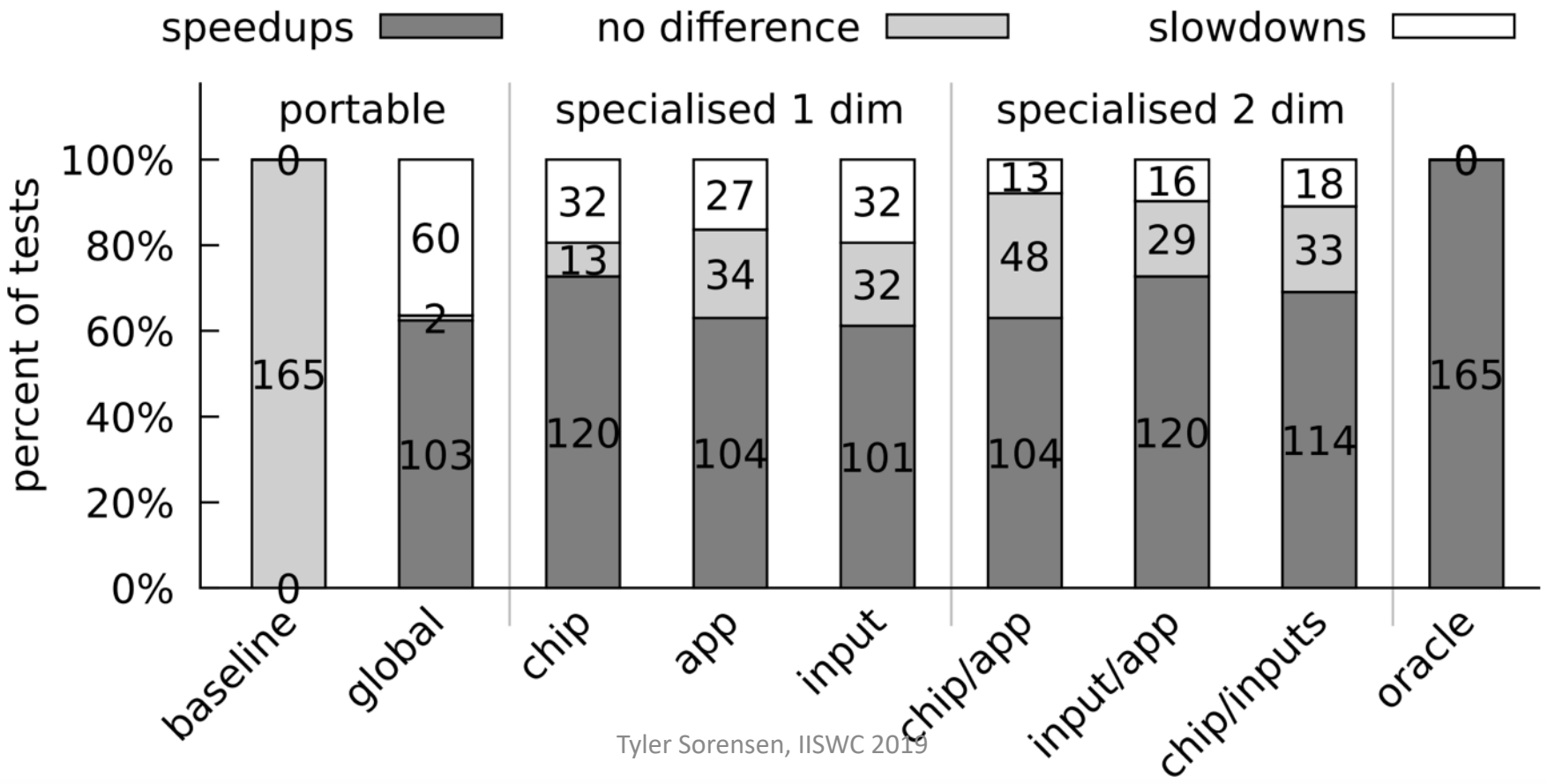
Optimizations
LB - Local
LB - Subgroup
LB - Workgroup
OC - Sync
RMW-Cls

60 Slowdowns
66 Speedups,
1.15x Geomean
6x max speedup

KHRONOS GROUP

Semi-specialization in Other Dimensions

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



Do the Least Harm

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

Fewest slowdowns

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

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

Most Slowdowns

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

195 Slowdowns
22 Speedups,
.53x Geomean

At First Glance – IrGL Optimizations

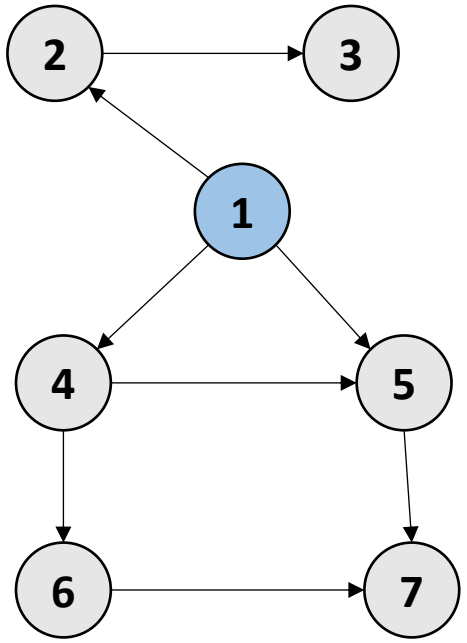
- **The Good:** Fantastic Speedups!
  - **Optimizations achieved up to a 16x speedup for AMD**
  - Speedups of over 10x on Intel chips
  - Geomean of 1.5x top speedups

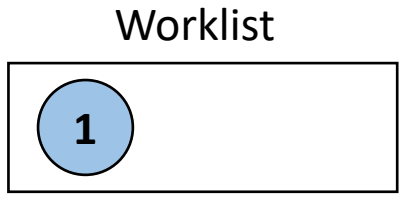
- **The Bad:** Horrible Slowdowns!
  - **Slowdowns of up to 22x on Intel GPUs for some "optimizations"**
  - Other GPUs suffered slowdowns of at least 8x

- **The Ugly:** Performance Portability?
  - How to tame this area?

A GPU Graph DSL and Compiler

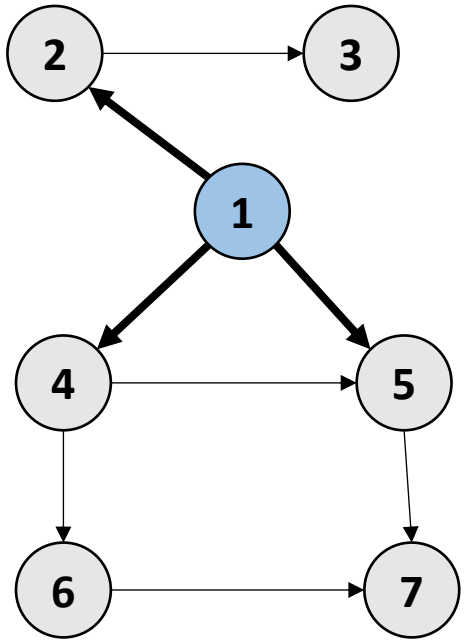
- IrGL : Pai and Pingali, OOPSLA 2016
  - Original work targets only Nvidia GPUs
- First class support for nodes, edges, worklists
- Optimizing compiler
  - Load balancing
  - On-chip synchronization
  - Atomic RMW coalescing

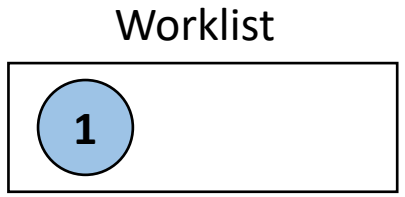




A GPU Graph DSL and Compiler

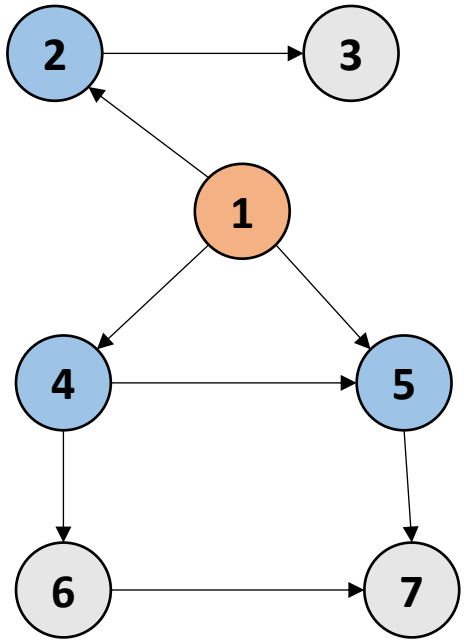
- IrGL : Pai and Pingali, OOPSLA 2016
  - Original work targets only Nvidia GPUs
- First class support for nodes, edges, worklists
- Optimizing compiler
  - Load balancing
  - On-chip synchronization
  - Atomic RMW coalescing

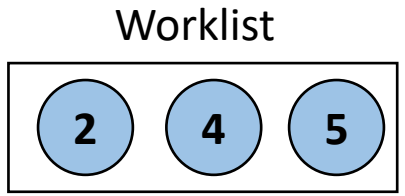




A GPU Graph DSL and Compiler

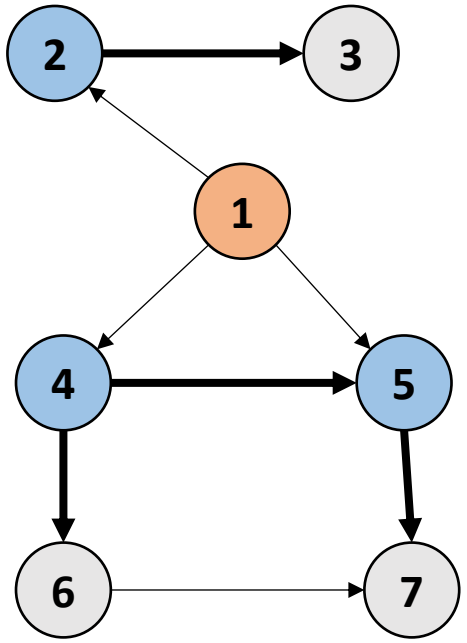
- IrGL : Pai and Pingali, OOPSLA 2016
  - Original work targets only Nvidia GPUs
- First class support for nodes, edges, worklists
- Optimizing compiler
  - Load balancing
  - On-chip synchronization
  - Atomic RMW coalescing

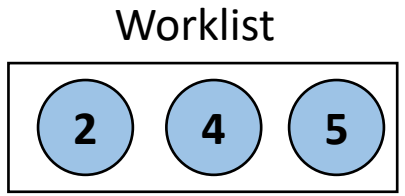




A GPU Graph DSL and Compiler

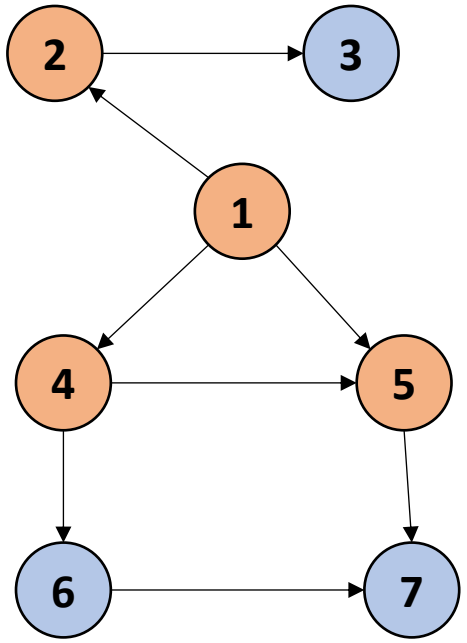
- IrGL : Pai and Pingali, OOPSLA 2016
  - Original work targets only Nvidia GPUs
- First class support for nodes, edges, worklists
- Optimizing compiler
  - Load balancing
  - On-chip synchronization
  - Atomic RMW coalescing

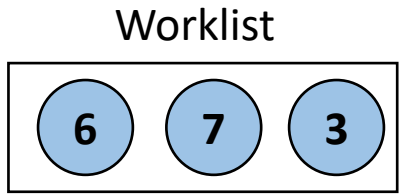




A GPU Graph DSL and Compiler

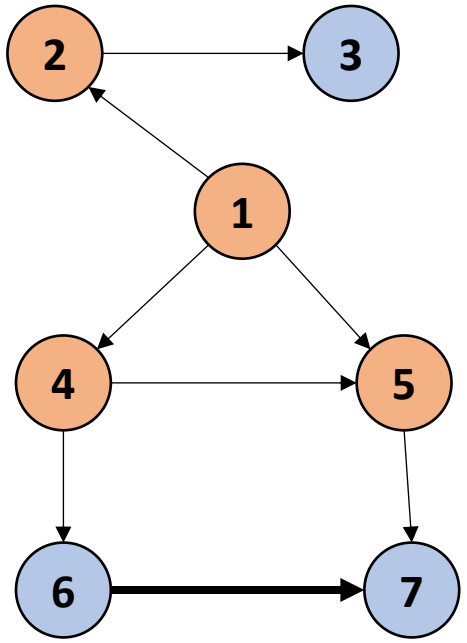
- IrGL : Pai and Pingali, OOPSLA 2016
  - Original work targets only Nvidia GPUs
- First class support for nodes, edges, worklists
- Optimizing compiler
  - Load balancing
  - On-chip synchronization
  - Atomic RMW coalescing

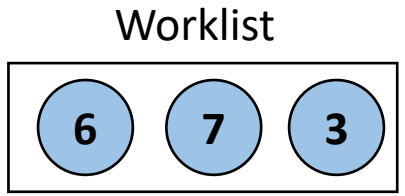




A GPU Graph DSL and Compiler

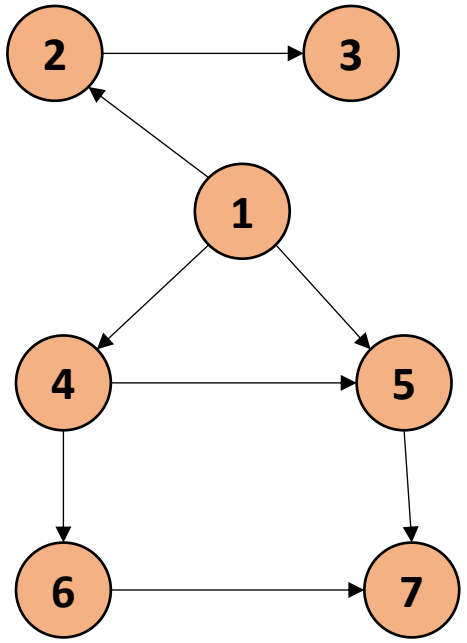
- IrGL : Pai and Pingali, OOPSLA 2016
  - Original work targets only Nvidia GPUs
- First class support for nodes, edges, worklists
- Optimizing compiler
  - Load balancing
  - On-chip synchronization
  - Atomic RMW coalescing





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



Worklist