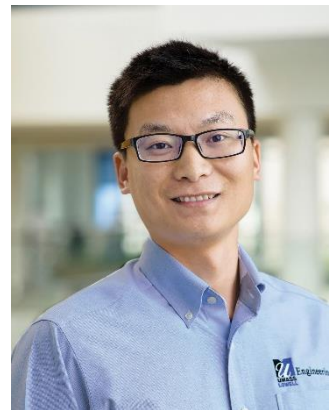


# TriCore: Parallel Triangle Counting on GPUs

Yang Hu



Hang Liu



H. Howie Huang



The George Washington University  
University of Massachusetts Lowell

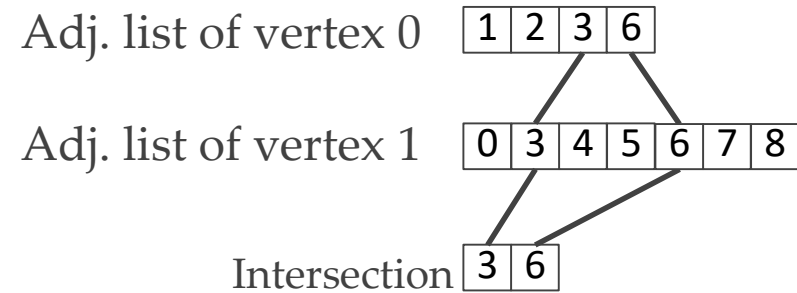
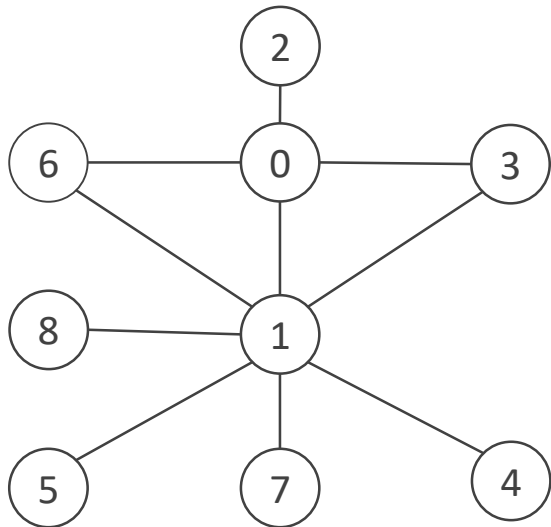
# Triangle Counting

- Applications
  - Social media
  - Web
  - E-commerce
  - Spam detection
  - ...



# Background: Triangle Counting Algorithm

- Intersection on each edge



$O(|E|^2)$  operations and  
memory access

- Two Triangles (0,1,3) and (0,1,6)
  - Start from edge (0,1)
  - Compare the neighbors of both vertices
  - Find shared neighbors 3 and 6

---

# Motivation

---

- Why is GPU good for triangle counting?
- Numerous of cores
  - Nvidia K40c GPUs: 2,880 CUDA
- High memory bandwidth
  - 288GB/s in K40c vs. 68GB/s on Xeon

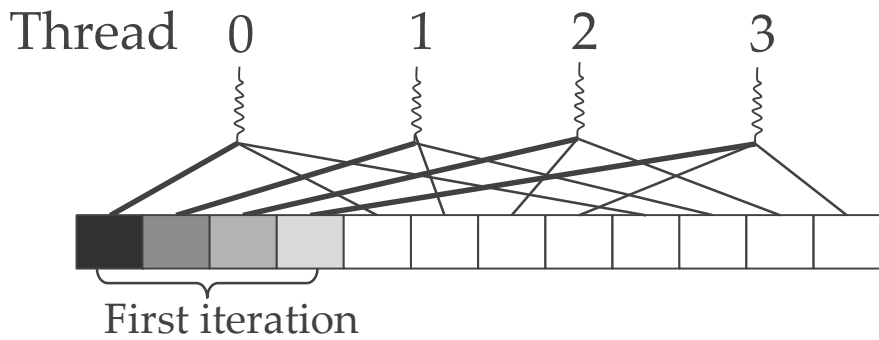
---

# Challenges

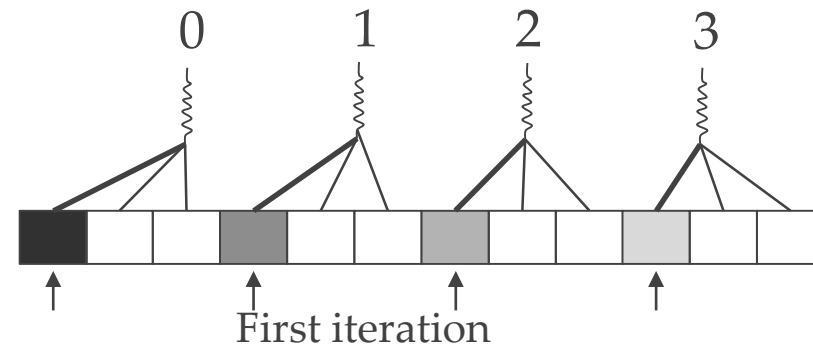
---

- SIMD: a group of threads execute same instruction
  - Warp = 32 threads
  - Low thread utilization
- Small cache
  - Low data utilization

# GPU Memory Access Patterns

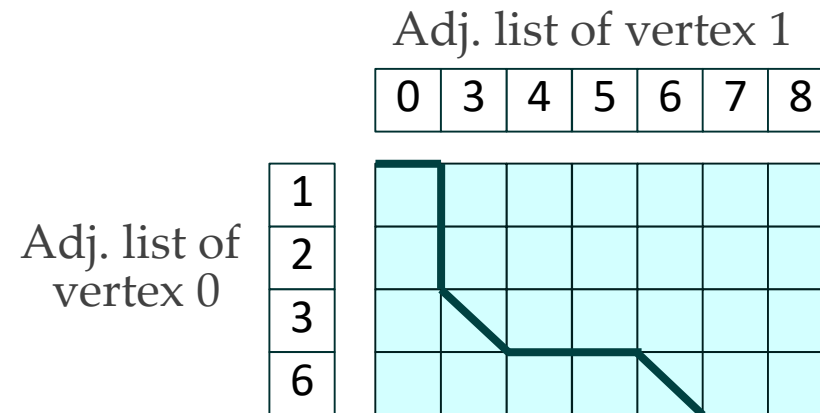
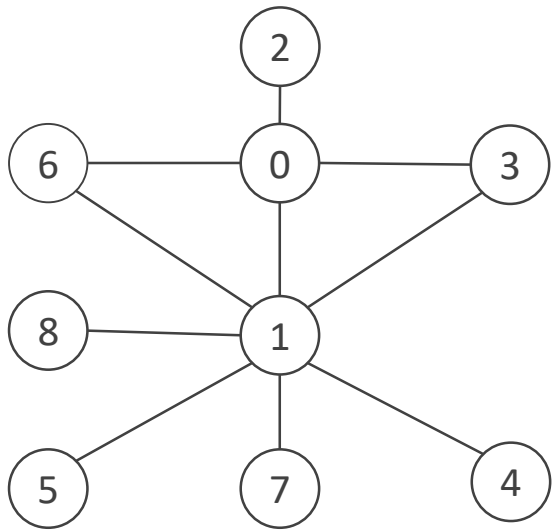


(a) Coalesced access



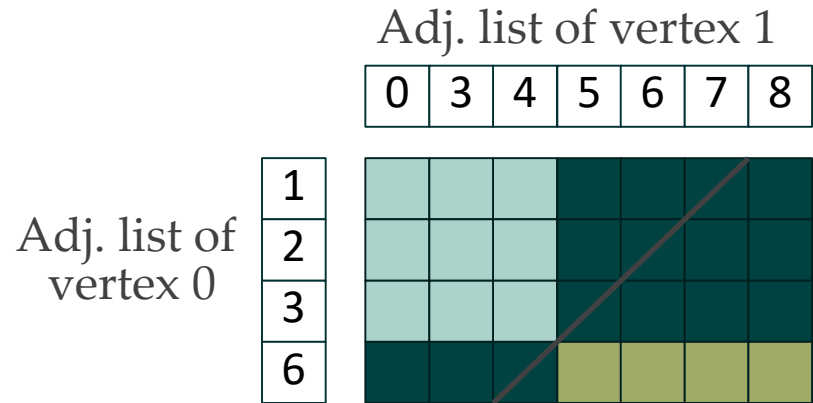
(b) Strided access

# Example: Merge-based Intersection

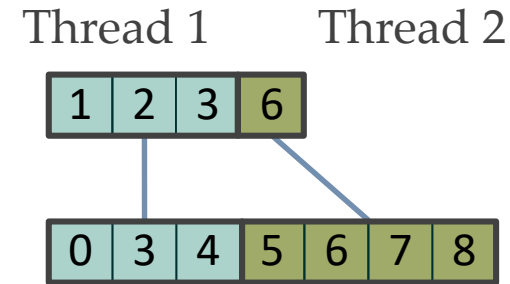


$O(m+n)$ ,  $m$  and  $n$  are the lengths of two lists

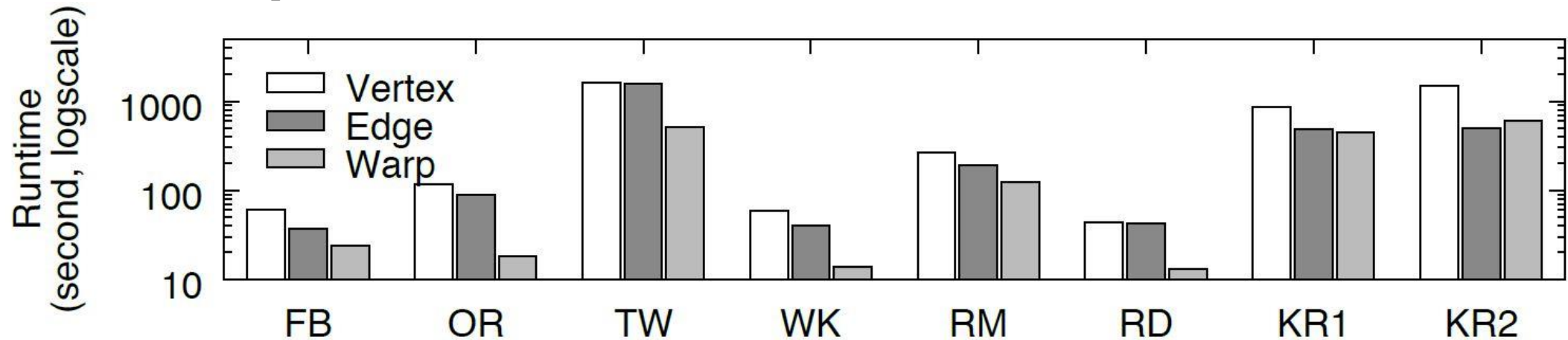
# Merge-based Algorithm for GPU



(a) Step I: Partition the workload



(b) Step II: Execution



Computation overhead and strided memory access

---

# Techniques

---

1. Binary search-based GPU algorithm
2. Shared memory caching

# Binary Search Based Triangle Counting

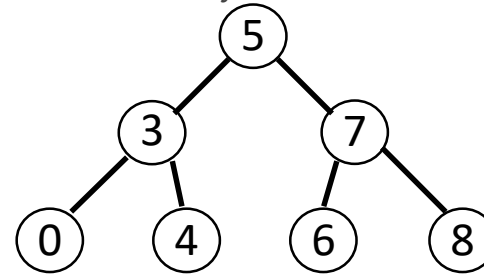
Neighbors of vertex 1

0	3	4	5	6	7	8
---	---	---	---	---	---	---

Neighbors of vertex 0

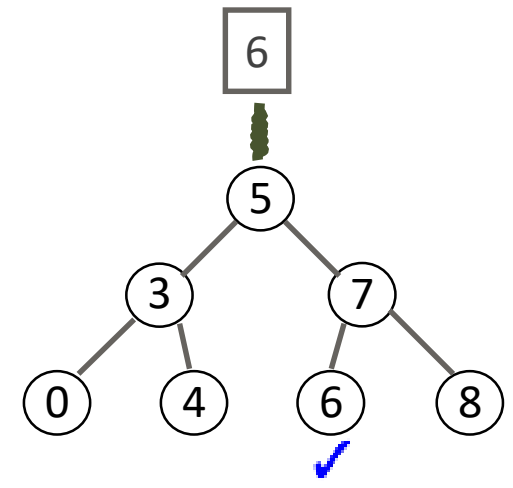
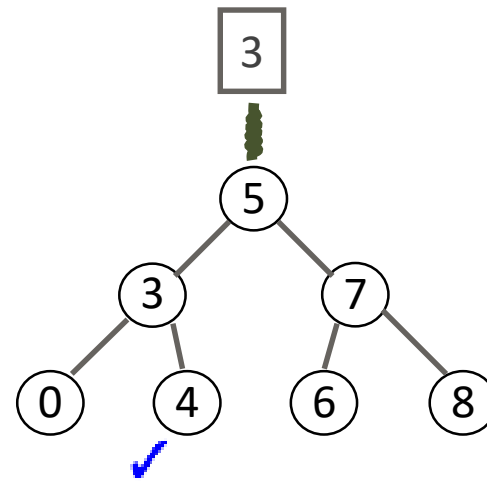
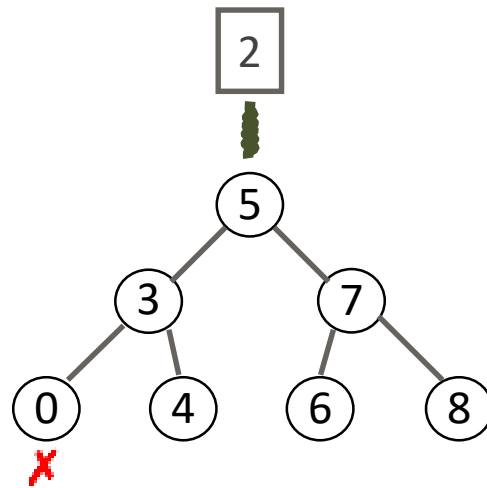
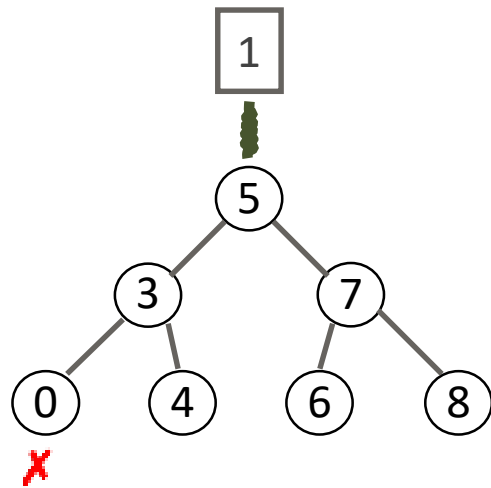
1	2	3	6
---	---	---	---

Binary Search Tree



Lookup keys

1	2	3	6
---	---	---	---



---

# Binary Search Based Triangle Counting

---

Neighbors of vertex u 

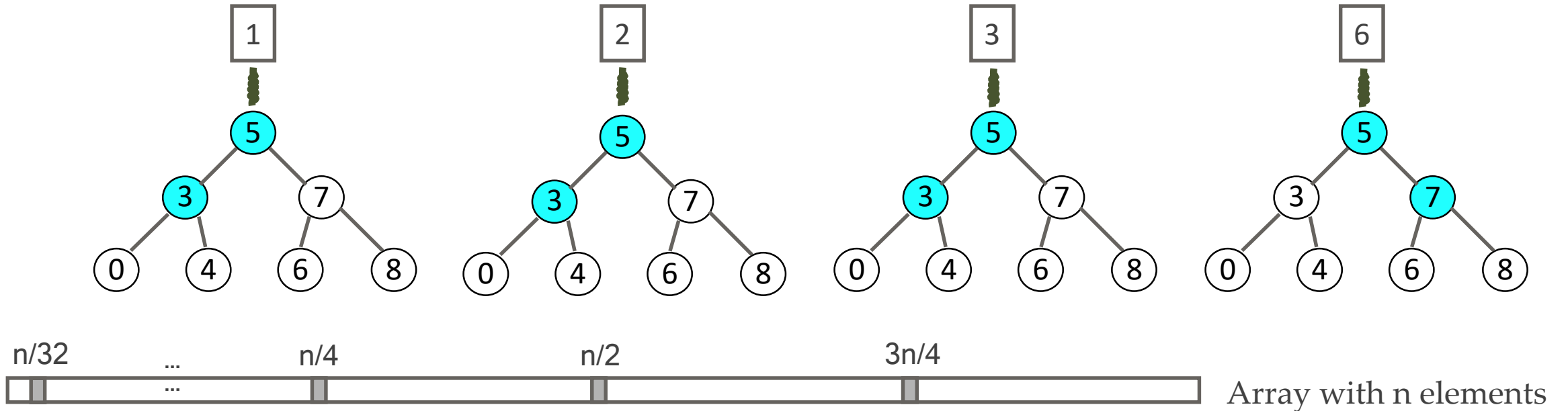
32	32	lookup list
----	----	-------------

Neighbors of vertex v 

Binary search list
--------------------

- Each edge is assigned to a warp (=32 threads)
- In each iteration 32 lookups in parallel
- **Fine-grained parallelization on GPUs**
- **Better memory access**

# Shared Memory Caching



- We cache 31 elements in shared memory for each edge
- 2-stage search

---

# TriCore vs. Merge

---

We assume  $m$  and  $n$  are the lengths of the two lists,  $m < n$

- Both  $m$  and  $n$  are small
  - No additional cost to enable warp level parallelization
- Small  $m$  and big  $n$ 
  - $m \cdot \log(n) < m+n$
- Both  $m$  and  $n$  are big
  - Contribute the majority of time cost
  - Coalesced memory access

---

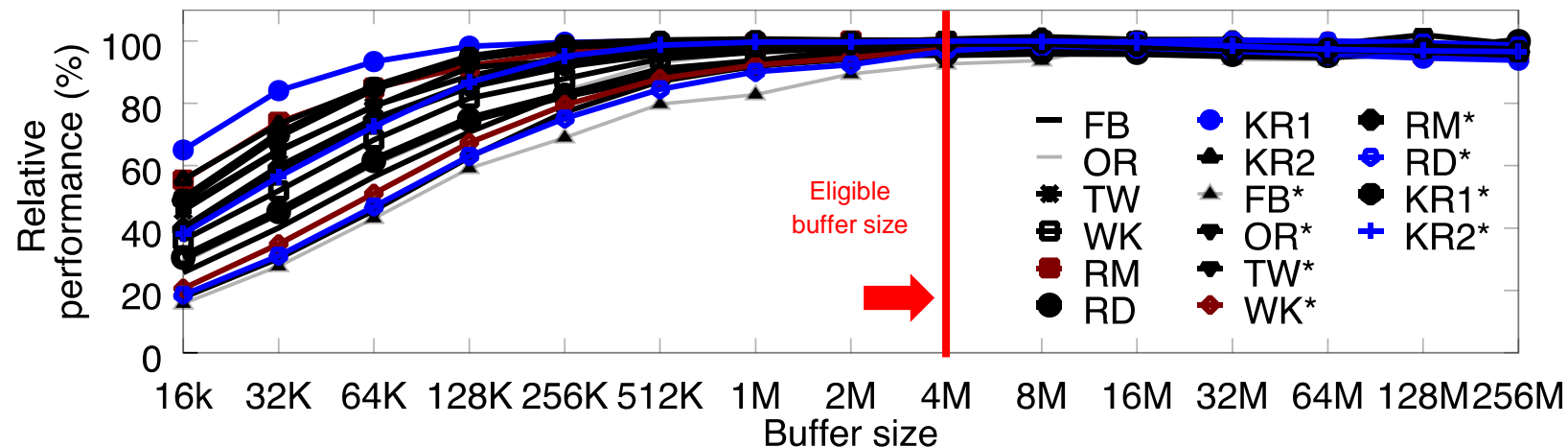
# Scalable TriCore

---

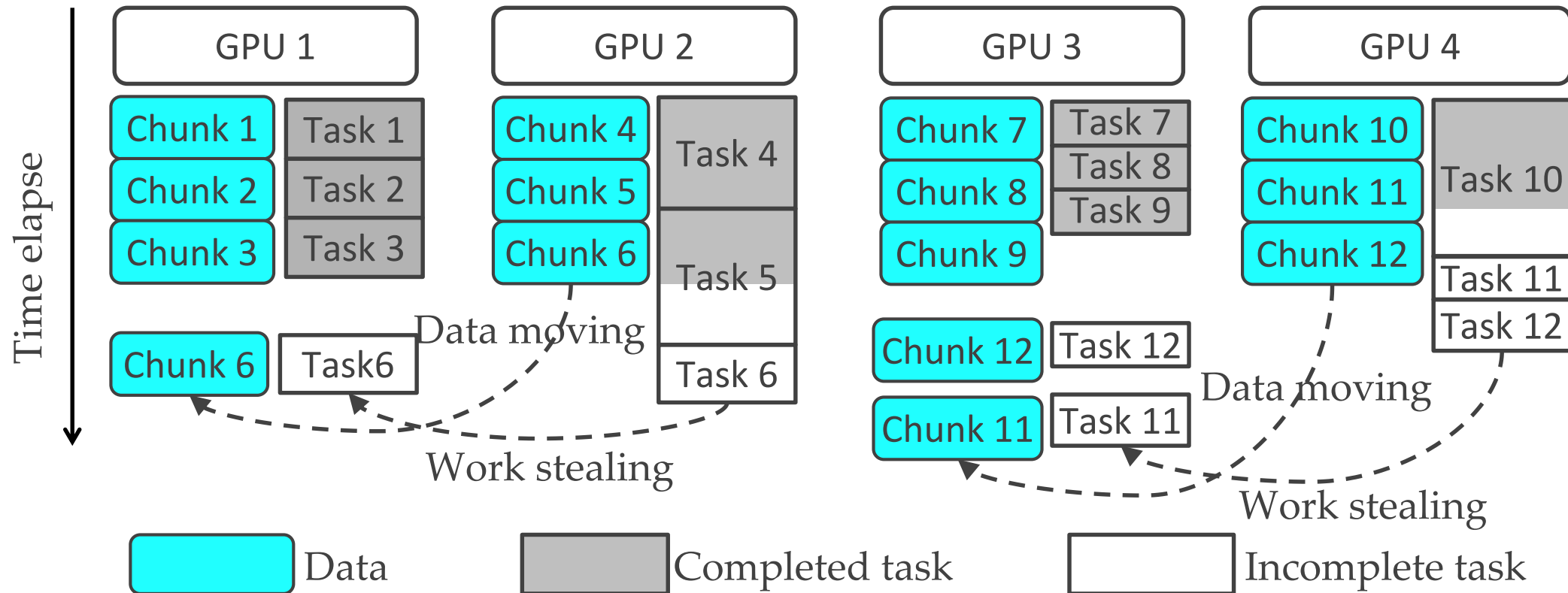
1. Edge list streaming buffer
2. Multiple GPUs

# Edge List Streaming Buffer

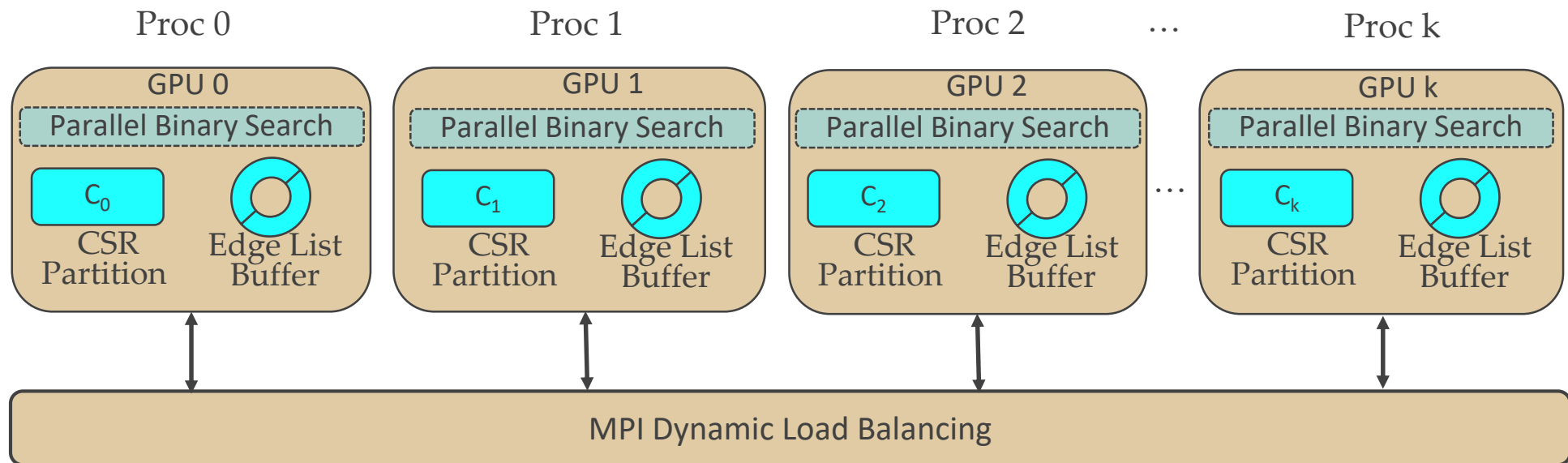
- To support edge-centric or finer granularity, a stand alone edge list format is required
- A ring buffer to load edge list iteratively
- The buffer size has to be big enough to saturate the GPU threads



# TriCore on GPUs: work stealing



# TriCore on GPUs: MPI



---

# Experiments

---

- Environments
  - CPU with 16 to 48 cores
  - K40 (x6) and Titan X Pascal GPU (x2)
  - Distributed environment with up to 32 K20 GPUs
- Datasets
  - Real and synthetic graphs with hundreds million to 1.4 billion edges
  - Big graphs with 17 to 34 billion edges

---

# Experiments

---

- Price of the CPU and GPUs used

Processors	Price (\$)
Dual socket Intel Xeon E5-2650v2 (16-core, 8-core/socket)	900
Dual socket Intel Xeon E5-2683 (28-core, 14-core/socket)	3,800
Quad socket Intel Xeon E7-8857v2 (48-core, 12-core/socket)	12,000
Tesla K20c GPU	800
Tesla K40c GPU	1,256
Titan X Pascal GPU	1,200

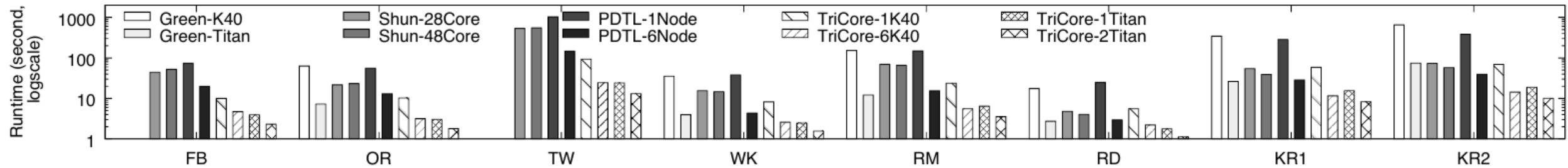
# Experiments

- Datasets

Name	Abbr.	Description	$ V $	$ E $	Triangle
Facebook	FB	Facebook user to friend link	96M	620M	3B
Orkut	OR	Orkut online social network	8M	327M	223M
Twitter	TW	Twitter follower network	41M	1.4B	34B
Wikipedia	WK	Links between Wikipedia pages	11M	258M	10B
RMAT	RM	R-mat (scale 22, degree 64)	4M	253M	2.1B
Random	RD	GTgraph: uniform degree	4M	511M	349K
Kronecker1	KR1	Kronecker (scale 22, degree 64)	4M	242M	5.3B
Kronecker2	KR2	Kronecker (scale 25, degree 16)	33M	523M	22B
Gsh-2015	GSH	Web graph	988M	33.2B	1.78T
Kron-30-16	KR3	Kronecker (scale 30, degree 16)	1.07B	17.0B	2.3T
Kron-31-16	KR4	Kronecker (scale 31, degree 16)	2.14B	34.1B	1.07T

# Experiments

## Comparison vs. state-of-the-art research projects



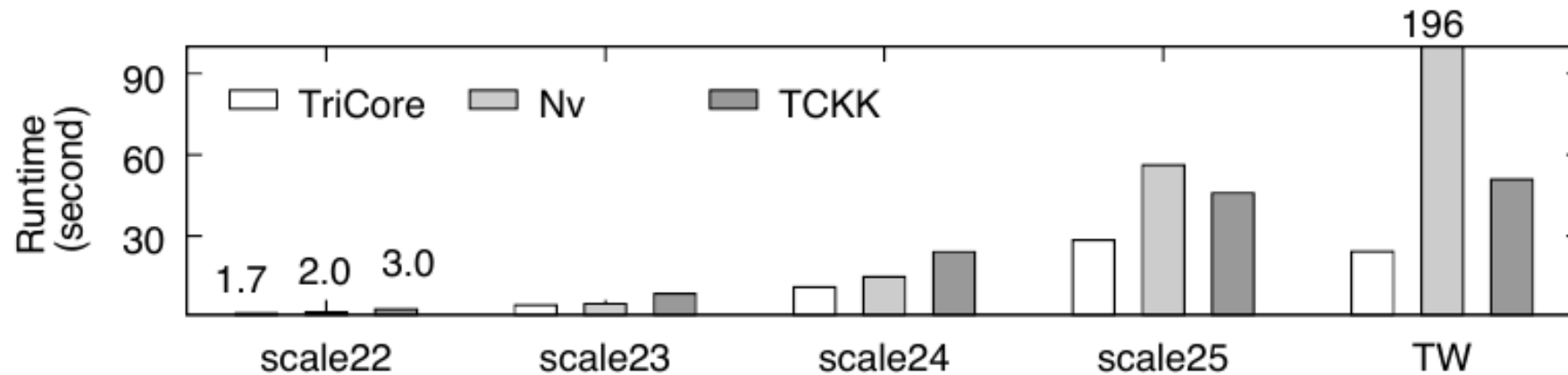
vs. GPU work: 5.9x on single K40, 2.2x on single Titan X vs. Green

vs. CPU work: on K40 (\$1256) 2.5x vs. Shun on 28-core (\$3800), on Titan X (\$1200)  
8.5x vs. Shun on 48-core (\$12000)

vs. distributed: on 6 K40 GPUs 6.5x vs. PDTL on 6 machines (16-core)

# Experiments

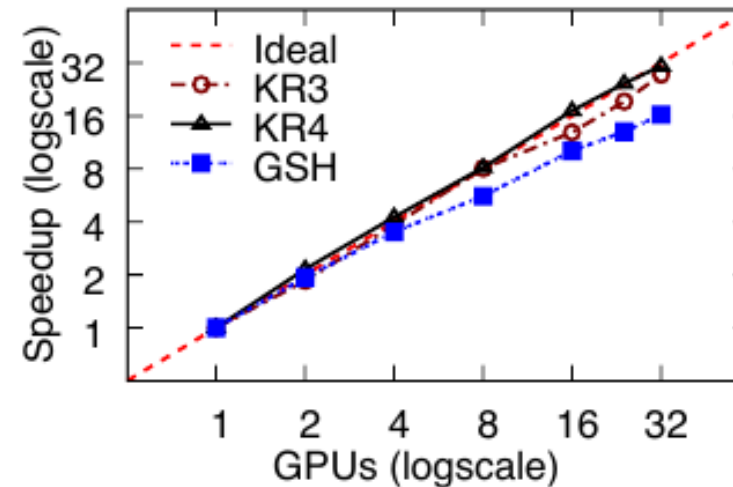
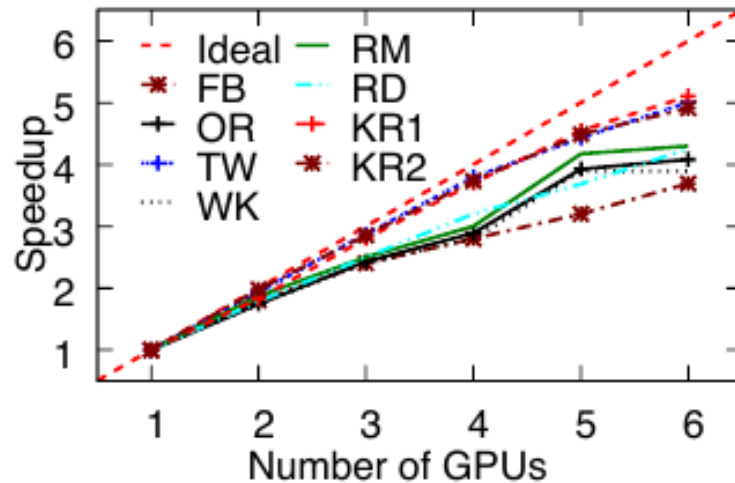
Comparison vs. graph challenge 2017 champions



2.7x and 1.9x vs. Nv and TCKK

# Experiments

## Scalability on multiple GPUs



4.4x speedup on 6 GPUs over single GPU

24x speedup on 32 GPUs over single GPU

---

# Acknowledgement

---



Please check out our graph code repository at  
<http://www.github.com/iHeartGraph>