FIREHOSE,PAGERANK AND NVGRAPH

GPU ACCELERATED ANALYTICS

CHESAPEAKE LARGE SCALE DATA ANALYTICS, OCT 2016

Joe Eaton Ph.D.

NVIDIA

Agenda

Accelerated Computing

FireHose

PageRank end-to-end

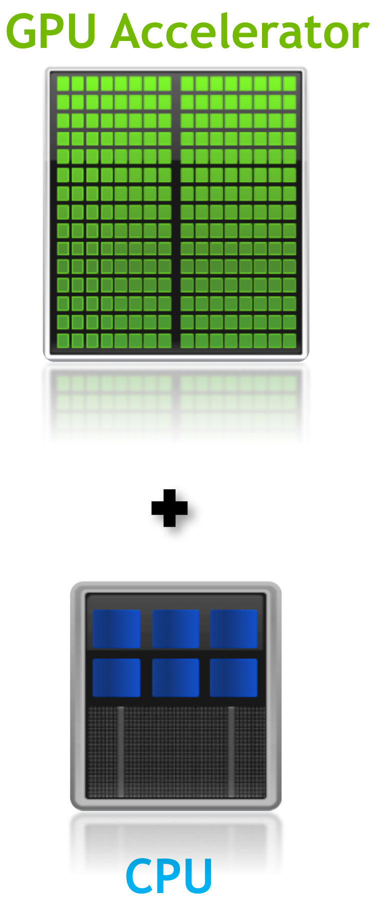
nvGRAPH

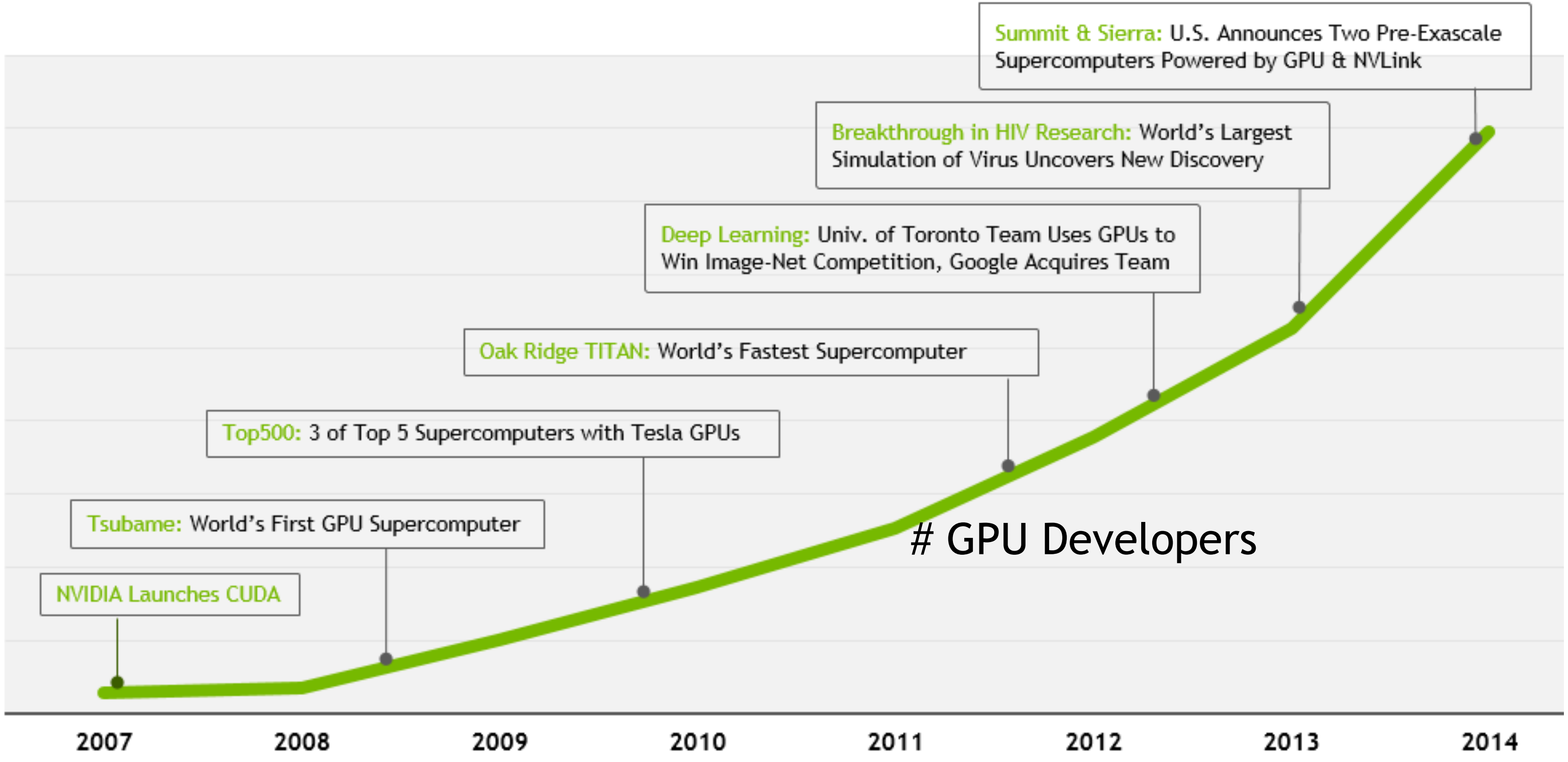
Coming Soon

Conclusion

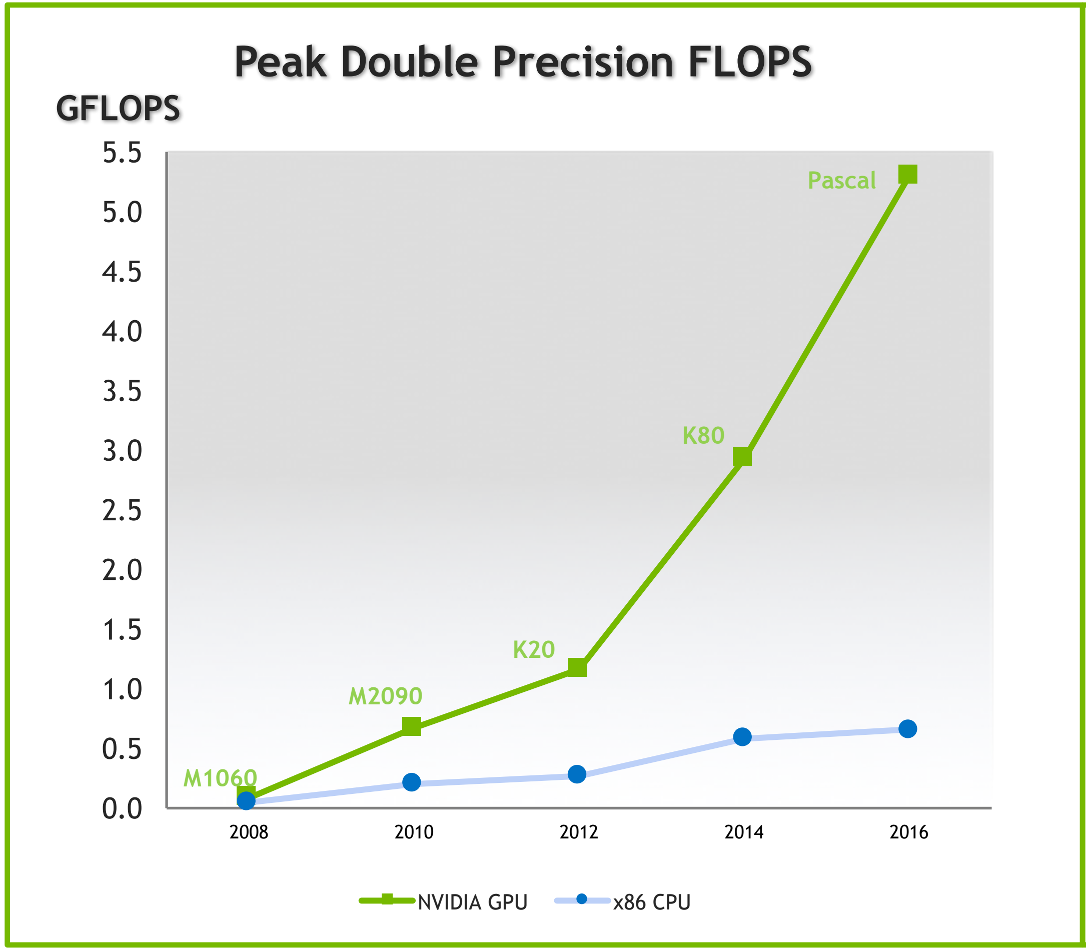
ACCELERATED COMPUTING

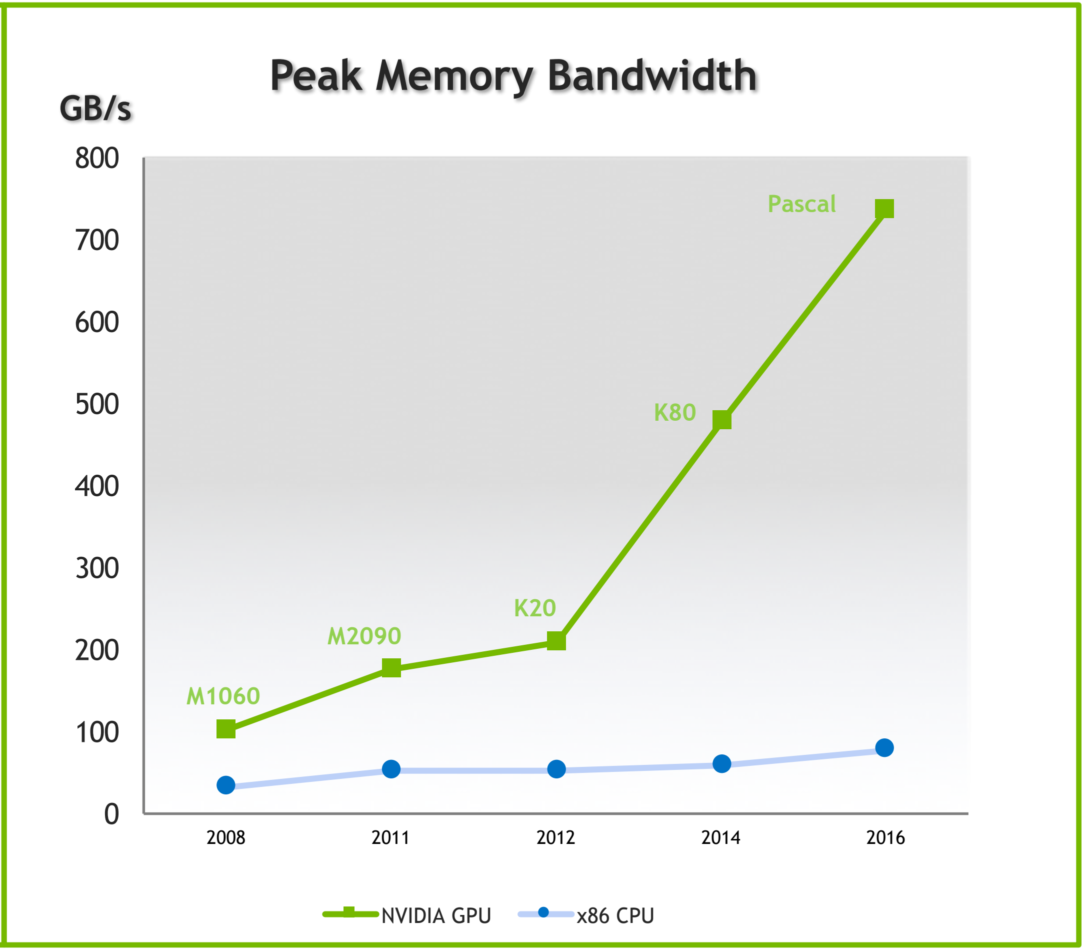
10x Performance & 5x Energy Efficiency





PERFORMANCE GAP CONTINUES TO GROW





FIREHOSE BENCHMARK

GPU Processing of Streaming Data

Volume and Velocity of some big data tasks do not allow for store & analyze.

Strong need to analyze **on-the-fly**, continuous stream of data, without trip to disk first.

Applications in Network monitoring - Social and Cyber – are growing in importance

Firehose

Suite of tasks measuring best-effort processing of UDP packets at high data rates.

Compare processing software and hardware
Quantitive & Qualitative

FIREHOSE BENCHMARK

GPU Processing of Streaming Data

Sandia National Laboratories

Firehose Parts

Generator streams UDP packets
- not throttled by Analyzer
- only a few ops per datum

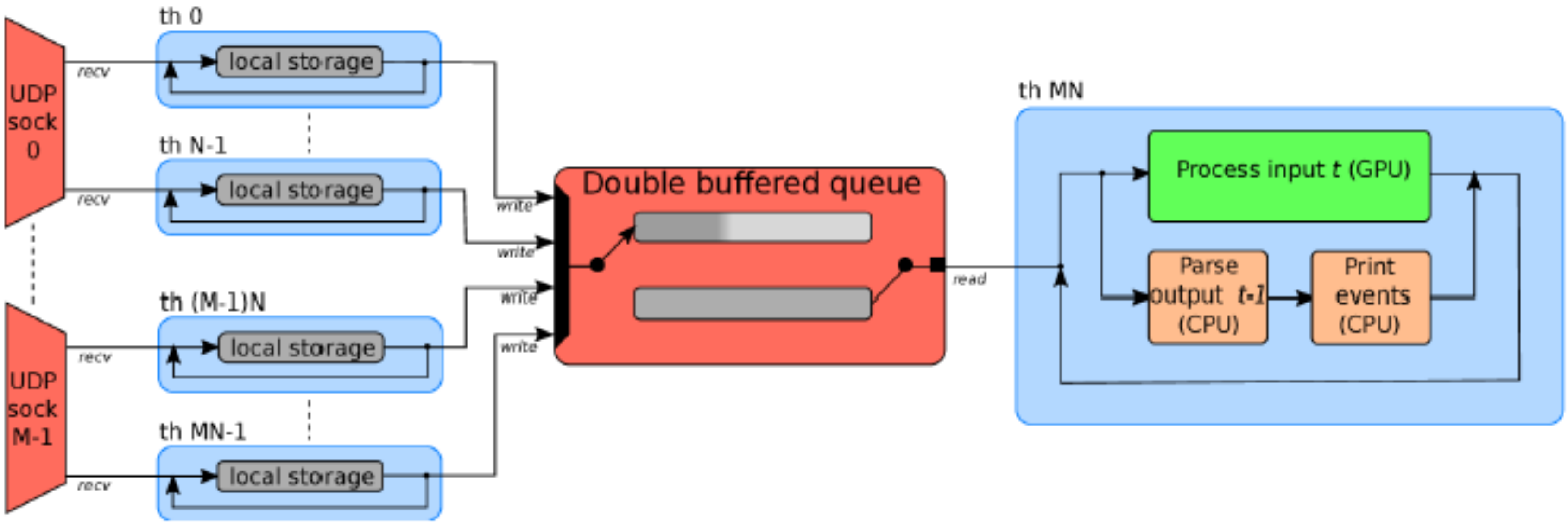
Analyzer may not be able to keep up, measure success rate

3 Firehose Benchmarks

- Power-law anomaly detection
- Active power-law anomaly detection
- Two-level anomaly detection

FIREHOSE CUDA IMPLEMENTATION

Analytics are implemented as pThreads + CUDA apps

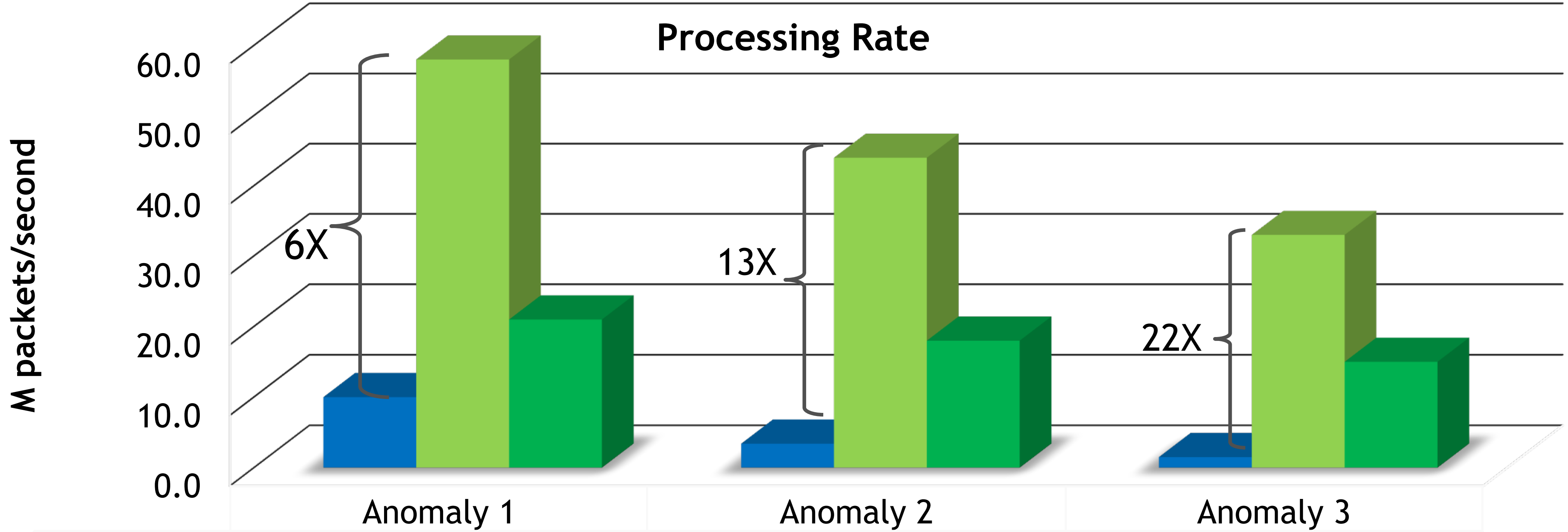


N worker threads for each UDP socket

Threads read data and insert into double-buffered queue.

When buffer filled it is sent to one of K GPUs for processing

FIREHOSE PROCESSING RATE



	Anomaly 1	Anomaly 2	Anomaly 3
CPU	10.0	3.4	1.5
4xMaxwell	58	44	33
1xPascal	21	18	15

CPU: Dual Xeon, 16 core X5690

PAGERANK PIPELINE BENCHMARK

Graph Analytics Benchmark

Proposed by MIT LL.

Apply supercomputing benchmarking methods to create scalable benchmark for big data workloads.

Four different phases that focus on data ingest and analytic processing.

Reference code for serial implementations available on GitHub.
https://github.com/NVIDIA/PRBench

PAGERANK PIPELINE BENCHMARK

4 Stage Graph Analytics Benchmark

LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Stage 1 – Generate graph (not timed)

Stage 2 – Read graph from disk, sort edges, write back to disk

Stage 3 – Read sorted edge list, generate normalized adjacency matrix for graph

Stage 4 – Run 20 iterations of Pagerank algorithm (power method)

Stage 2 tests I/O

Stage 3 tests I/O + compute

Stage 4 tests compute

SPEEDUP VS REFERENCE C++

Scale	K0(99%)	K1(90%)	K2(80%)	K3(0%)
16	2.8x	1.1x	4.6x	5.9x
17	2.6x	1.3x	5.0x	10.3x
18	2.9x	1.3x	6.3x	14.1x
19	3.2x	1.5x	7.6x	14.0x
20	3.3x	1.5x	8.7x	11.8x
21	3.2x	1.5x	9.2x	9.8x
22	3.4x	1.5x	9.9x	9.8x

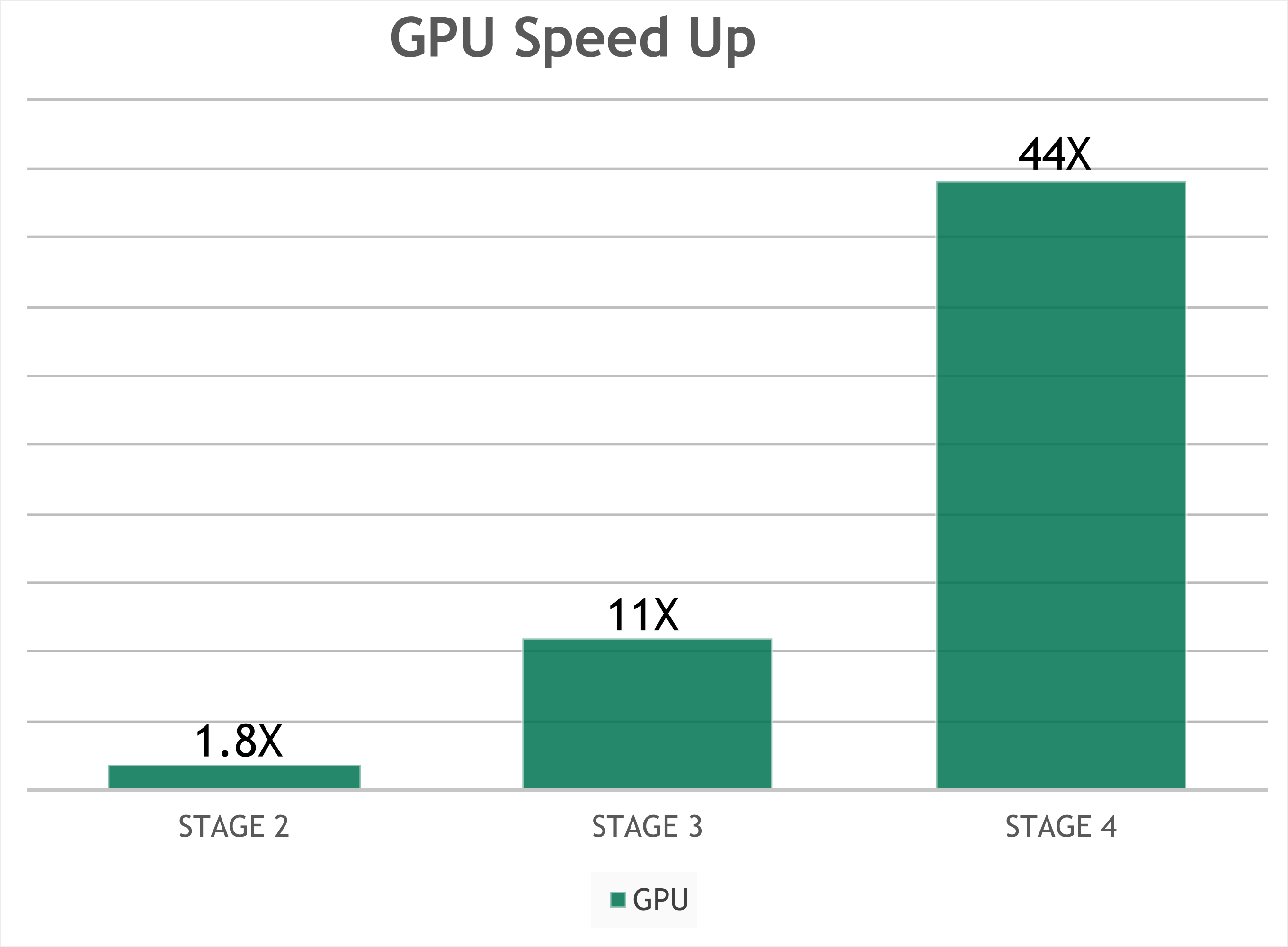
PAGERANK PIPELINE RESULTS

Single P100 GPU

Speed up versus MIT LL reference C++ implementation

GPU: Tesla P100

CPU: Intel Xeon E5 2690 3.0 GHz



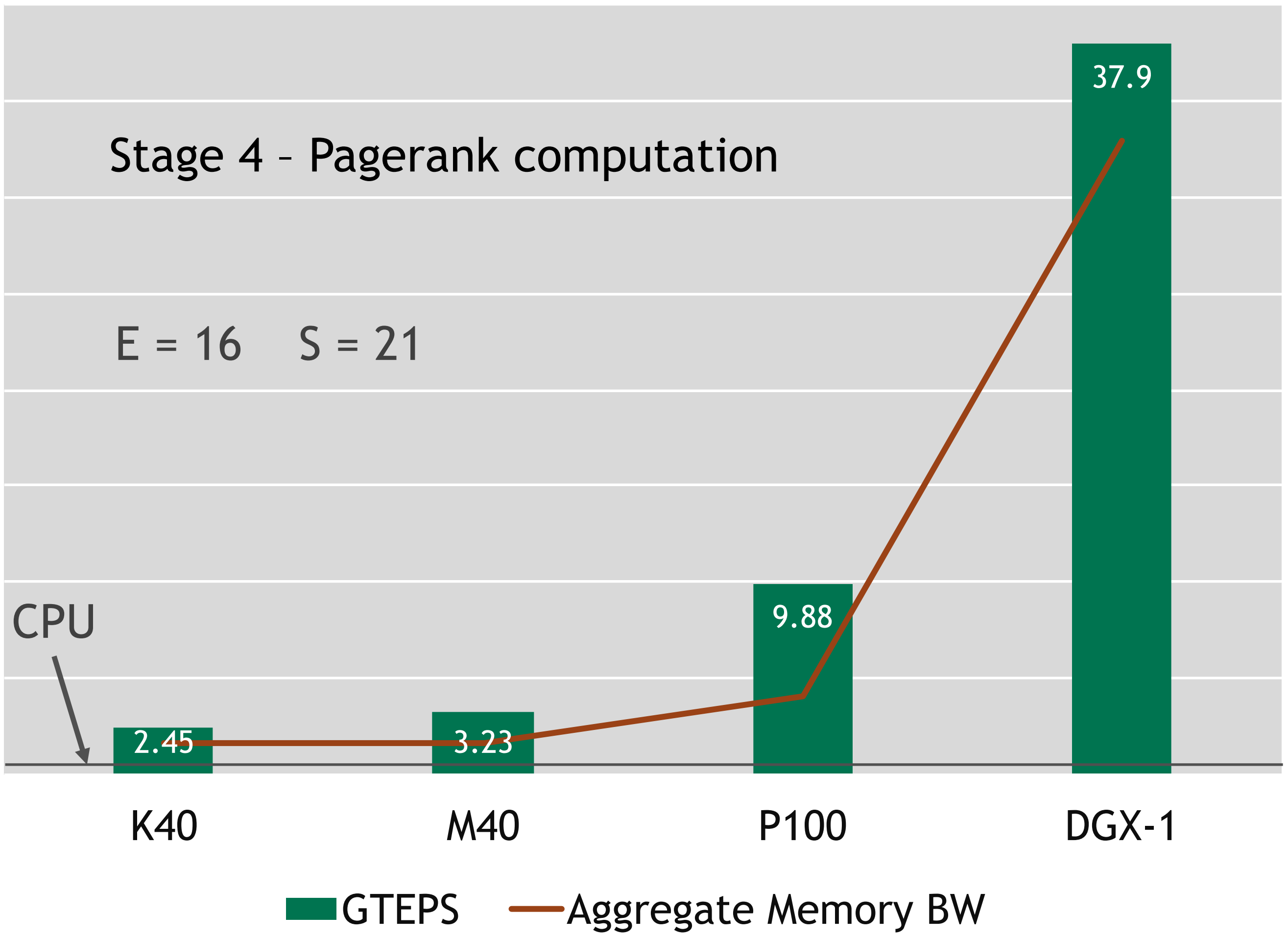
PAGERANK PIPELINE RESULTS

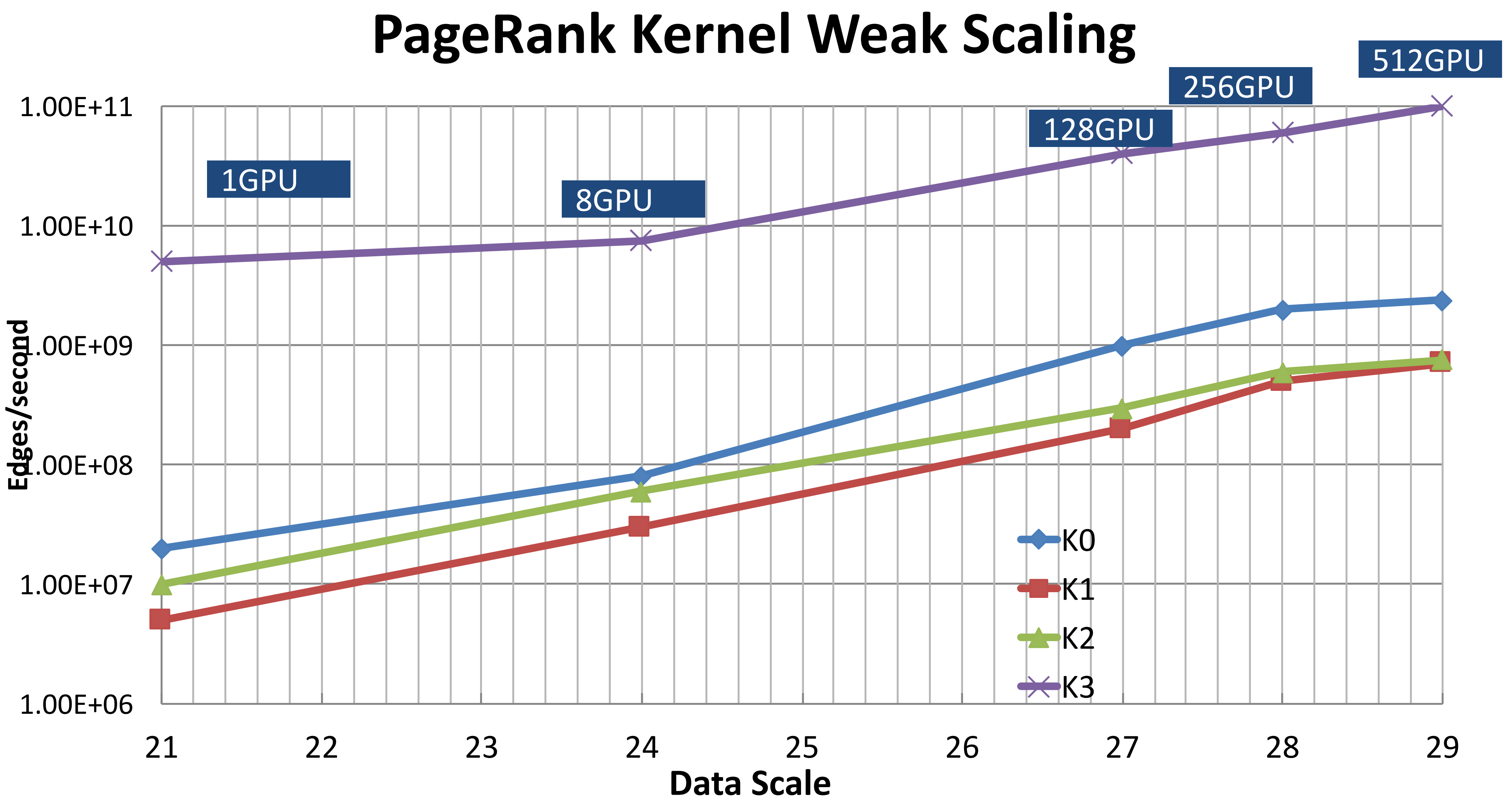
Comparing Across GPUs

Memory bandwidth
most important

GTEPS = billions edges/sec

$\# \text{ edges} = E * 2^S$



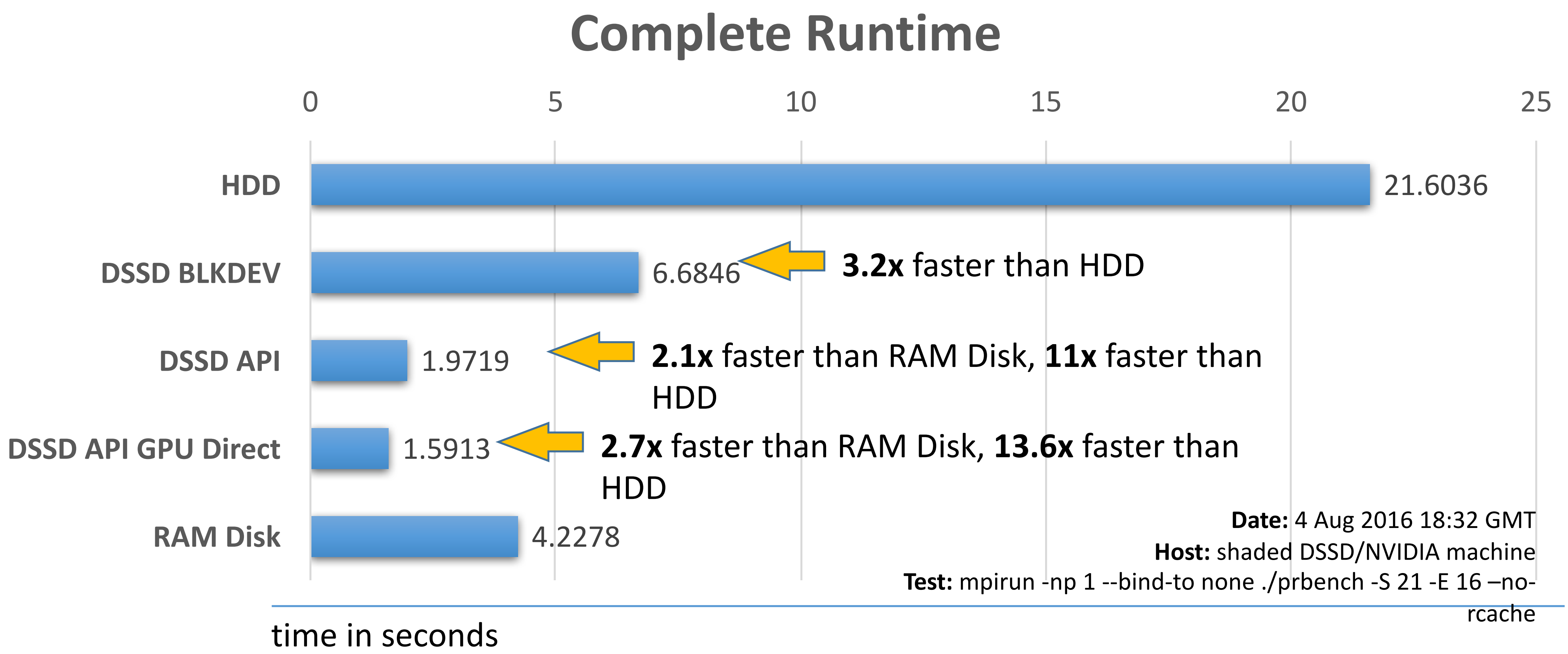

PageRank Kernel Weak Scaling
1GPU
8GPU
128GPU
256GPU
512GPU
1.00E+11
1.00E+10
1.00E+09
1.00E+08
1.00E+07
1.00E+06
Edges/second
21
22
23
24
25
26
27
28
29
Data Scale
K0
K1
K2
K3

DSSD+NVIDIA PageRank Results

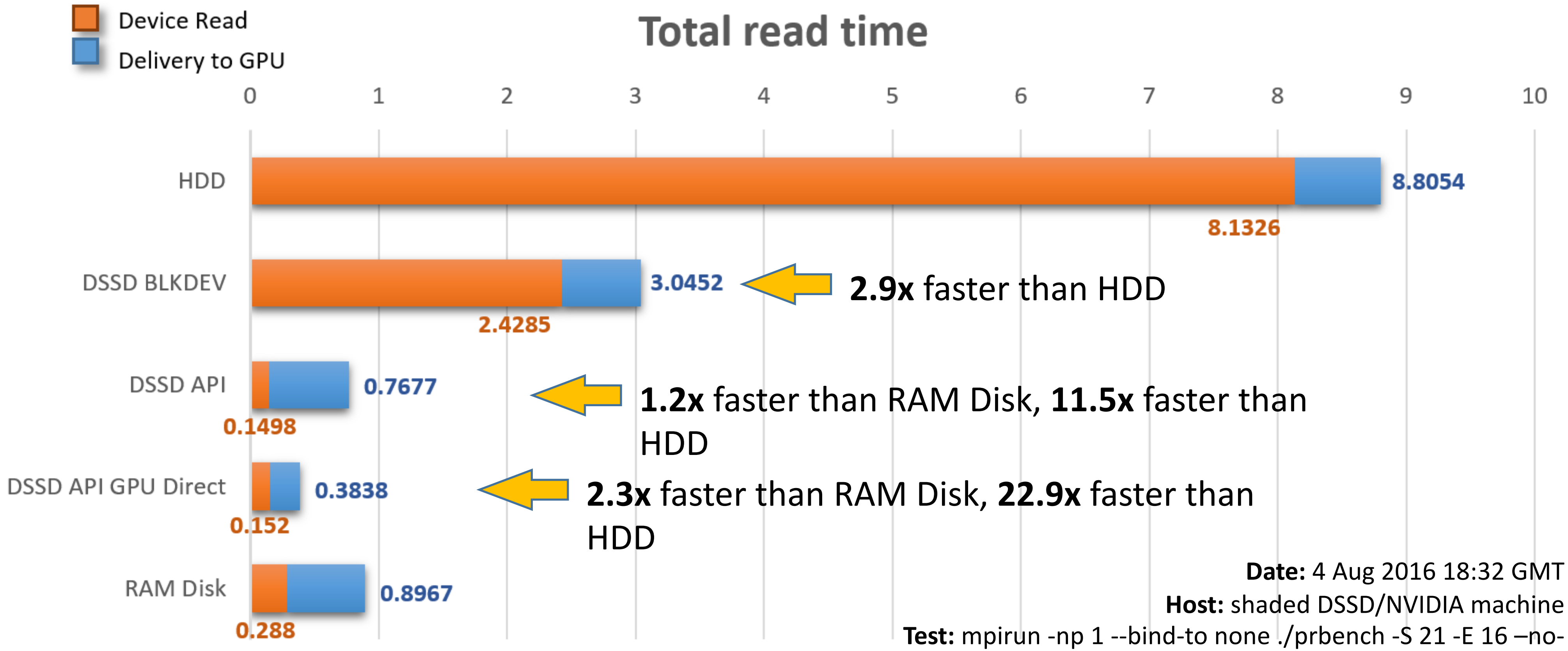
Key Takeaways

- Out of the box (no change to test code) D5 BLK speed is **comparable to RAM Disk**
- Using API (minimal code change) D5 is **2.7x faster** than RAM Disk
- D5 Advantages:
  - Device speed
  - Direct-to-device API
  - Direct transfer to GPU
  - Shared between machines
  - High Capacity

DSSD+NVIDIA PageRank Results



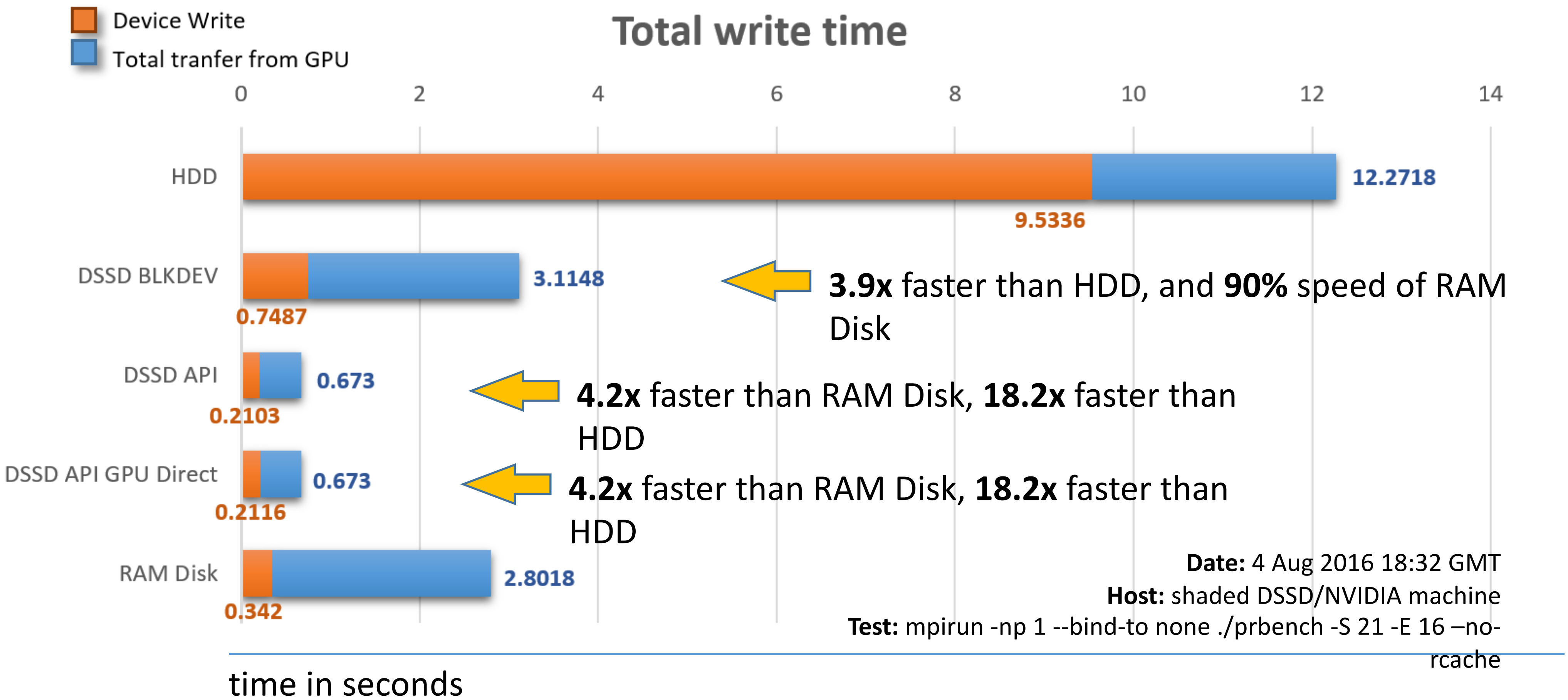
DSSD+NVIDIA PageRank Results



Date: 4 Aug 2016 18:32 GMT
Host: shaded DSSD/NVIDIA machine
Test: mpirun -np 1 --bind-to none ./prbench -S 21 -E 16 –no-rcache

time in seconds

DSSD+NVIDIA PageRank Results



3.9x faster than HDD, and **90%** speed of RAM Disk

4.2x faster than RAM Disk, **18.2x** faster than HDD

4.2x faster than RAM Disk, **18.2x** faster than HDD

Date: 4 Aug 2016 18:32 GMT
Host: shaded DSSD/NVIDIA machine
Test: mpirun -np 1 --bind-to none ./prbench -S 21 -E 16 –no-rcache

GRAPHS ARE FUNDAMENTAL

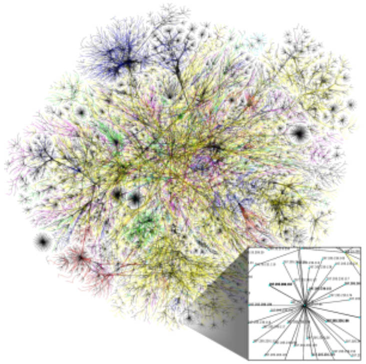
Tight connection between data and graphs

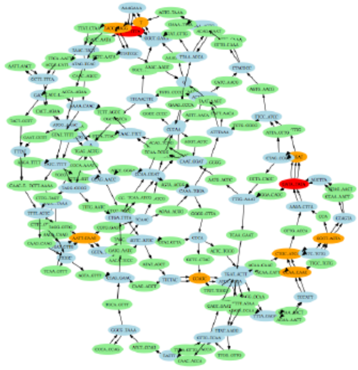
Data View	Graph View
Data Element/ Entity	Graph Vertex
Entity Attributes	Vertex labels
Binary Relation (1 to 1)	Graph Edge
N-ary Relation (many to 1)	Hypergraph edge
Relation Attributes	Edge labels
Group of relations over entities	Sets of Vertices and Edges

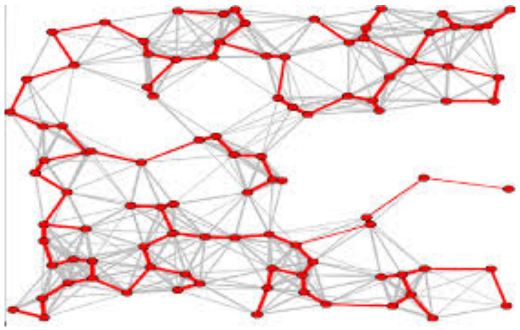
NVGRAPH

Easy Onramp to GPU Accelerated Graph Analytics

- GPU Optimized Algorithms
- Reduced cost & Increased performance
- Standard formats and primitives
  Semi-rings, load-balancing
- Performance Constantly Improving







nvGRAPH

Accelerated Graph Analytics

nvGRAPH for high performance graph analytics

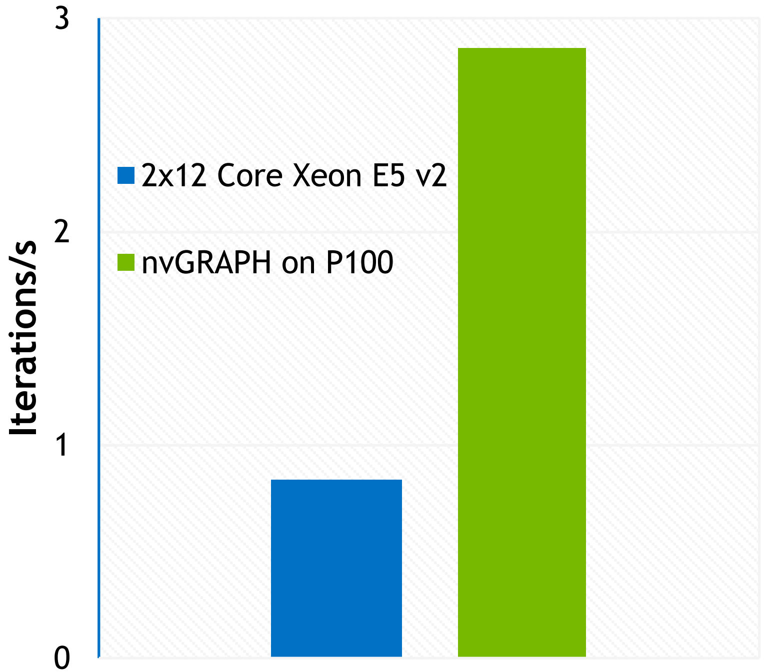
Deliver results up to 3x faster than CPU-only

Solve graphs with up to 2 Billion edges on a single GPU (M40)

Accelerates a wide range of graph analytics applications:

PageRank	Single Source Shortest Path	Single Source Widest Path
Search	Robotic Path Planning	IP Routing
Recommendation Engines	Power Network Planning	Chip Design / EDA
Social Ad Placement	Logistics & Supply Chain Planning	Traffic sensitive routing

nvGRAPH: 3.4x Speedup



PageRank on Twitter 1.5B edge dataset

nvGraph on P100

GraphMat on
2 socket 12-core Xeon E5-2697 v2 CPU,@ 2.70 GHz

developer.nvidia.com/nvgraph

Motivating example

Power law graph: wiki2003.bin

455,436 vertices (n)

2,033,173 edges (nnz)

sparsity = 4.464234

Cusparse csrmv time: 8.05 ms

Merge Path csrmv time: 1.08 ms

~7.45x faster!

PSG Cluster, K40

SEMI-RINGS

Definition / Axioms

Set R with two binary operators: + and * that satisfy:

1. (R, +) is associative, commutative with additive identity 0

   (0 + a = a)

2. (R, *) is associative with multiplicative identity 1

   (1 * a = a)

3. Left and Right multiplication is distributive over addition

4. Additive identity 0 = multiplicative null operator

   (0 * a = a * 0 = 0)

SEMI-RINGS
Examples

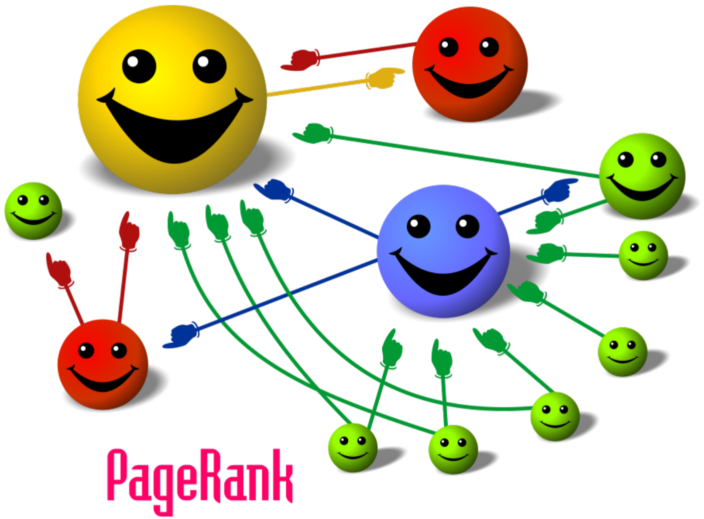
SEMIRING	SET	PLUS	TIMES	$\underline{0}$	$\underline{1}$
Real	$\mathbb{R}$	+	*	0	1
MinPlus	$\mathbb{R} \cup \{-\infty, \infty\}$	min	+	∞	0
MaxMin	$\mathbb{R} \cup \{-\infty, \infty\}$	max	min	-∞	∞
Boolean	$\{0, 1\}$	∨	∧	0	1

APPLICATIONS

Pagerank (+, *)

- Ideal application: runs on web and social graphs
- Each iteration involves computing: $y = A\,x$
- Standard csrmv
- PlusTimes Semiring
- α = 1.0 (multiplicative identity)
- β = 0.0 (multiplicative nullity)

//sw/gpgpu/naga/src/pagerank.cpp



APPLICATIONS

Single Source Shortest Path (min, +)

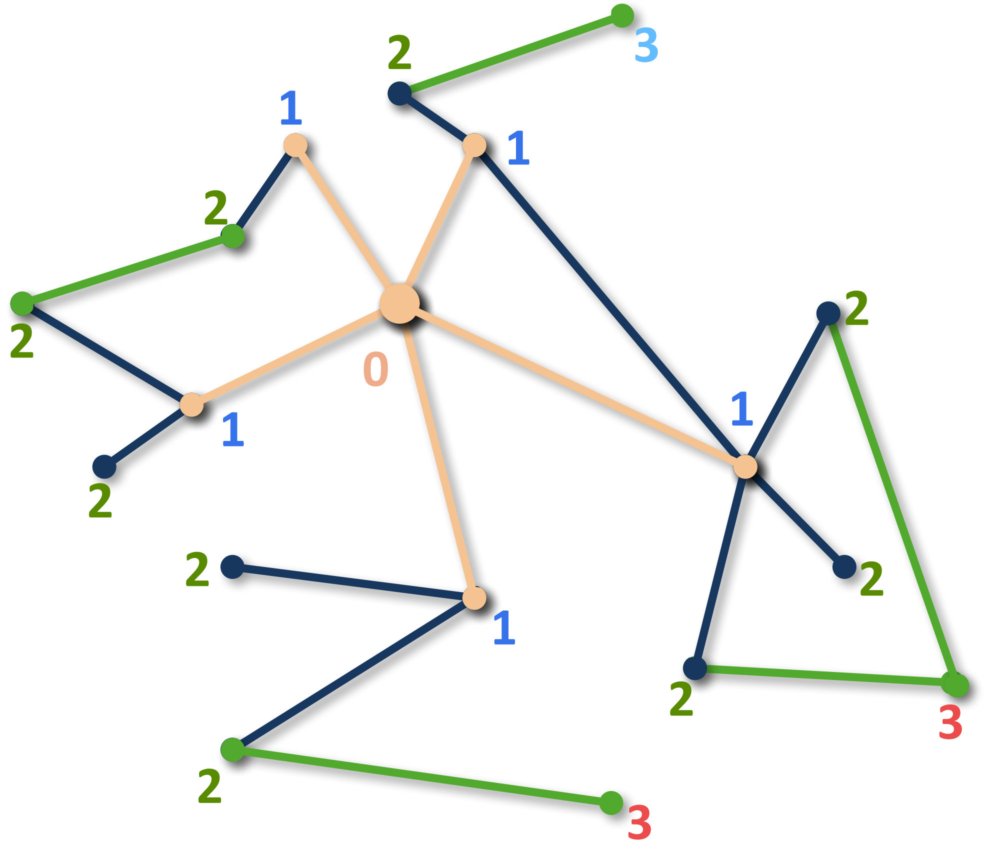
Common Usage Examples:

Path-finding algorithms:

- Navigation
- Modeling
- Communications Network

Breadth first search building block

Graph 500 Benchmark

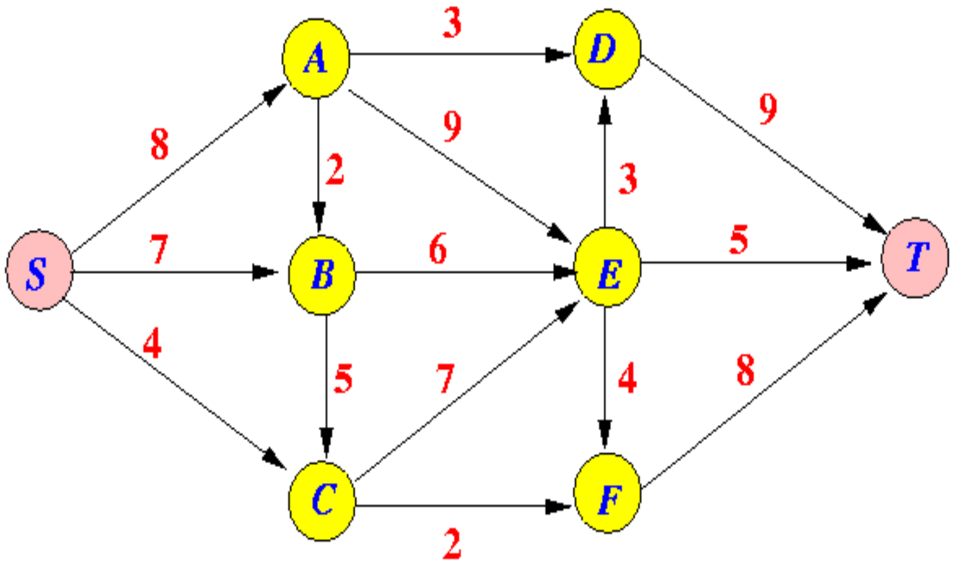


APPLICATIONS

Widest Path (max,min)

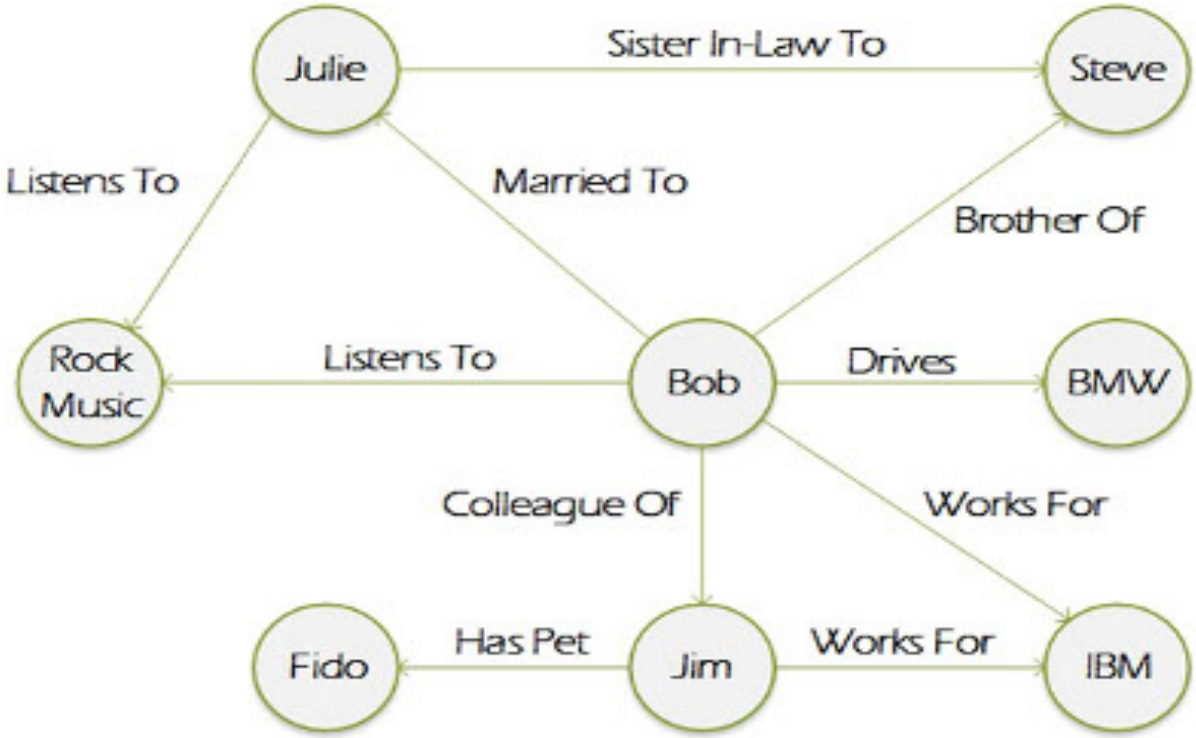
Common Usage Examples:

- Maximum bipartite graph matching
- Graph partitioning
- Minimum cut
- Common application areas:
    - power grids
    - chip circuits



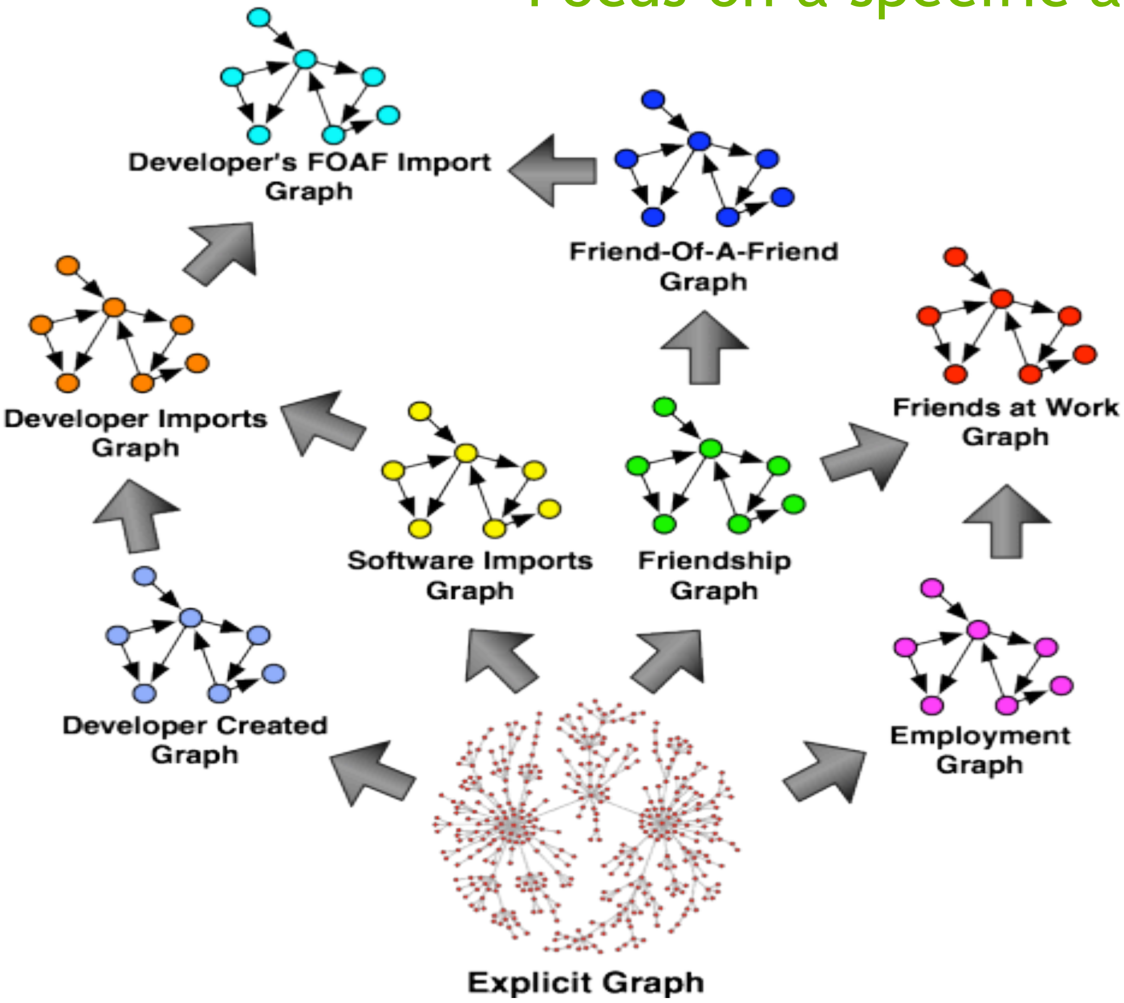
PROPERTY GRAPHS

Many simple graphs overlaid



SUBGRAPH EXTRACTION

Focus on a specific area



COMING SOON

Features in next release

Partitioning

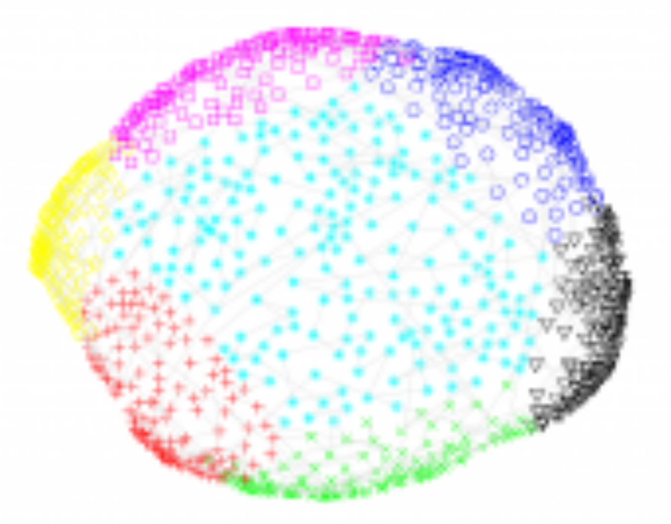
Clustering

BFS

Graph Contraction

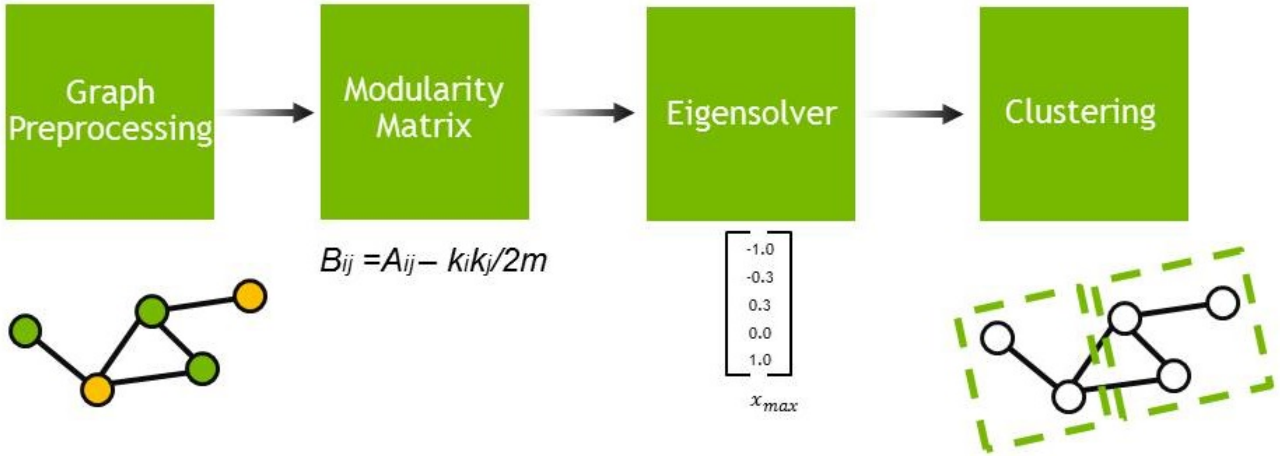
PARTITIONING AND CLUSTERING

Spectral Min Edge Cut Partition



Modularity maximization (spectral)

Example 2 partitions



BREADTH FIRST SEARCH

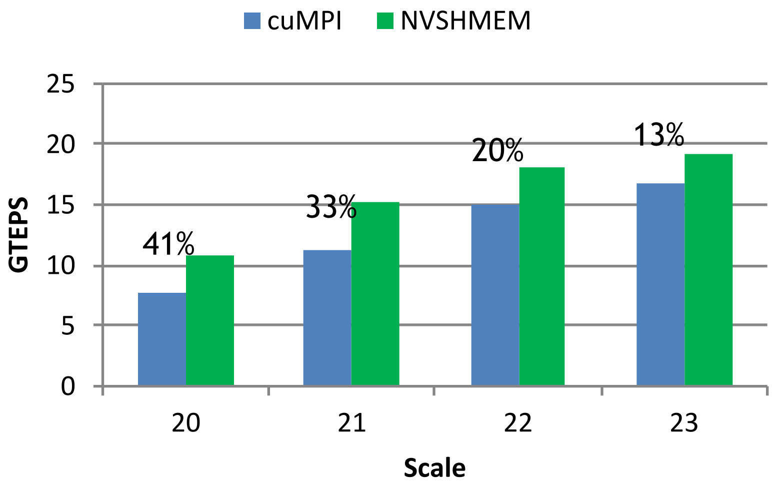
Key subroutine in several graph algorithms, naturally leads to random access

MPI Version implementations: pack or use a bitmap to exchange frontier at end of each step

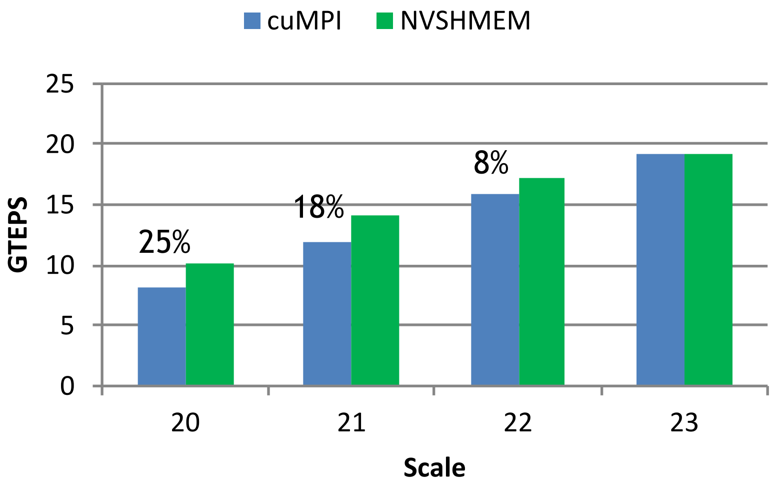
NVSHMEM version: directly updates the frontier map at target using atomics

Benefits with smaller graphs (likely behavior with strong scaling)

4 P100 GPUs alltoall connected with NVLink

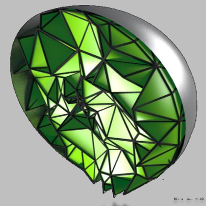


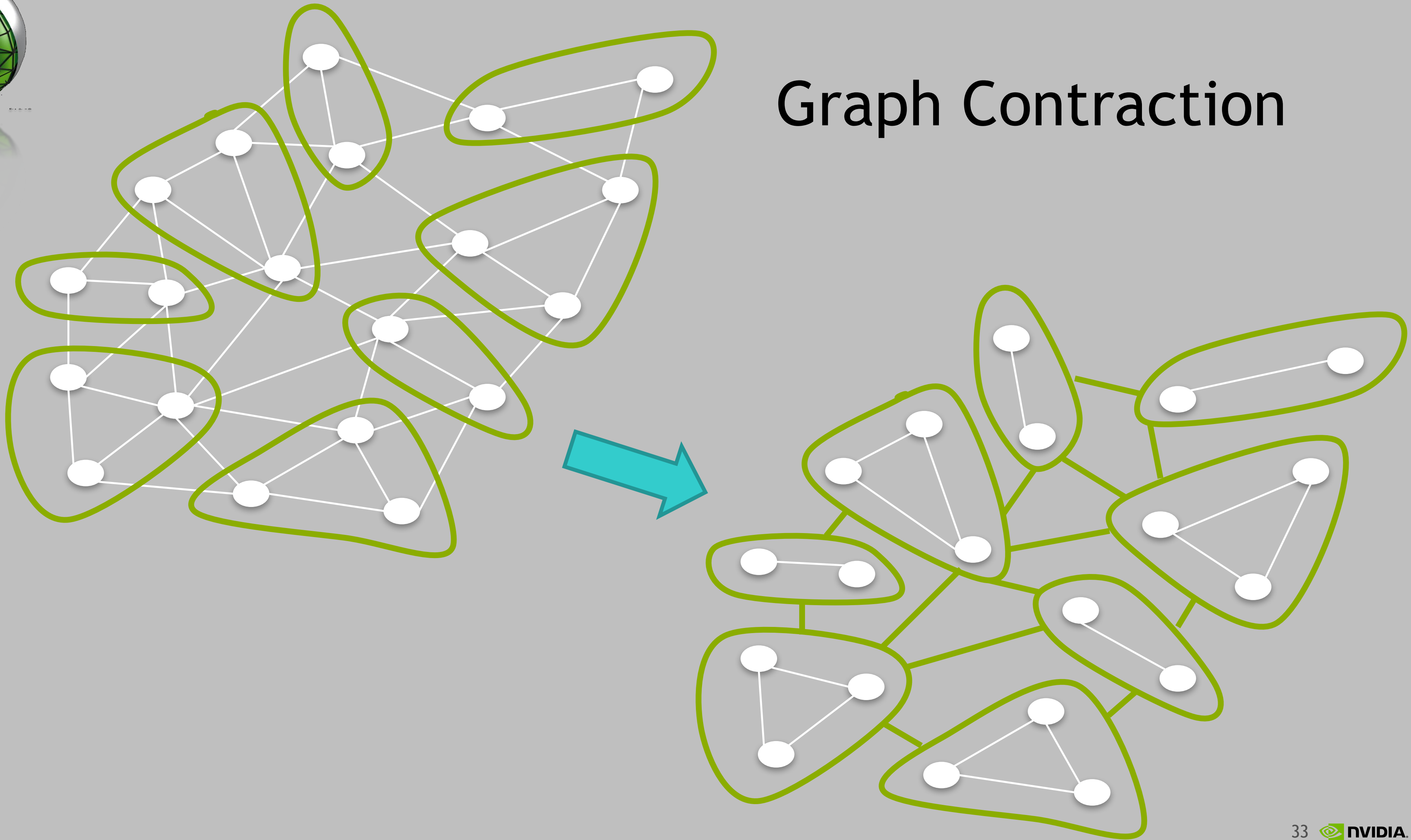
1x4 Process Grid



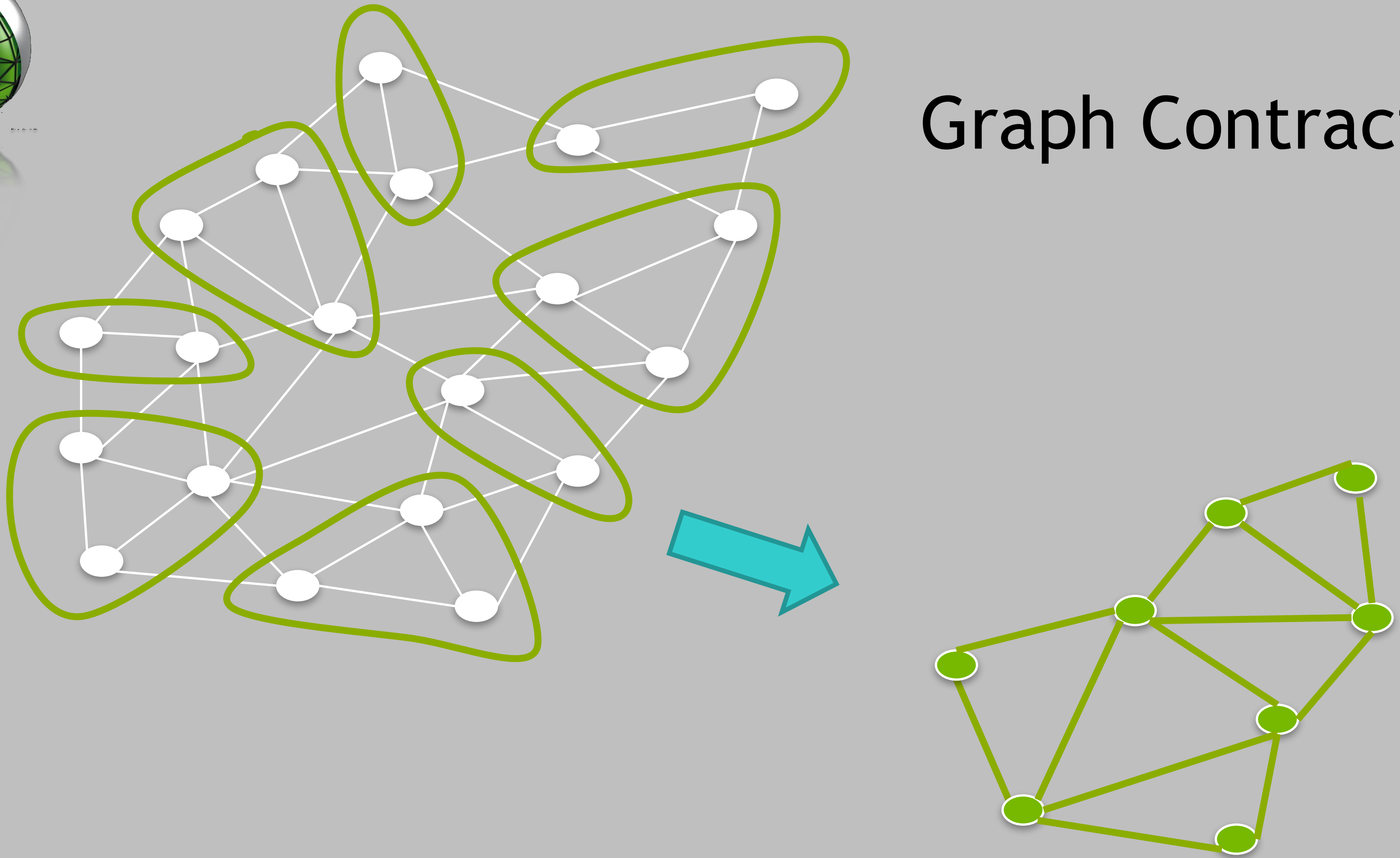
2x2 Process Grid

Graph Contraction





Graph Contraction



DYNAMIC GRAPHS

cuSTINGER brings STINGER to GPUs

Oded Green presented at HPEC 2016

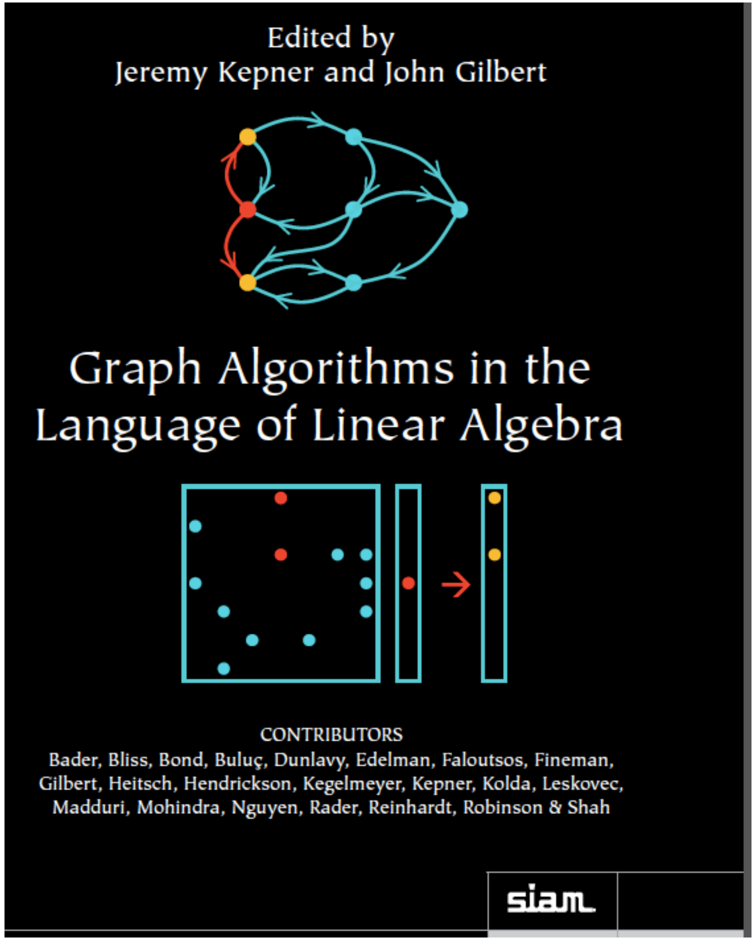
cuSTINGER: Supporting Dynamic Graph Algorithms for GPUs

https://www.researchgate.net/publication/308174457

GRAPHBLAS

nvGRAPH is working toward a full GraphBLAS implementation

Semi-rings are a start



TOWARD REAL TIME BIG DATA ANALYTICS

GPUs enable the next generation of in-memory processing

	DUAL BROADWELL SERVER	NVIDIA DGX-1 SERVER	GPU PERFORMANCE INCREASE
Aggregate Memory Bandwidth	150 GB/s	5760 GB/s	38 X
Aggregate SP FLOPS	4 TF	85 TF	21 X

Single DGX-1 server provides the compute capability of dozens of dual-cpu servers

ACCELERATED DATABASE TECHNOLOGY

Big data ISVs moving to the accelerated model

SQL

No SQL

Graph

SUMMARY

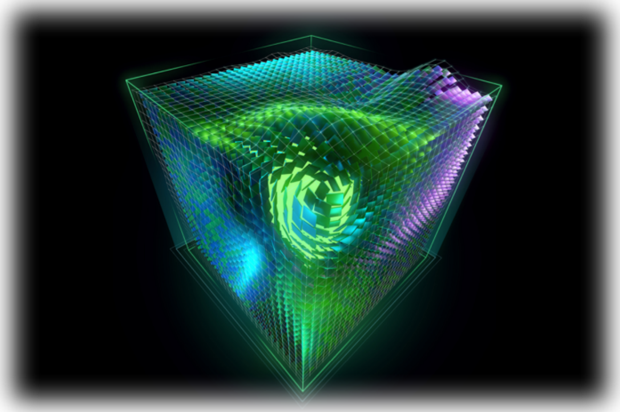
GPUs for High Performance Data Analytics

GPU computing is not just for scientists anymore!

GPUs excel at in-memory analytics.

- Streaming – Firehose
- Graph – Pagerank Pipeline
- Analytics - nvGRAPH

Savings in infrastructure size & cost by using GPU servers versus standard dual-CPU server. Database In-Memory performance 20-100x at practical scale



REFERENCES

GPU Processing of Streaming Data: A CUDA implementation of the Firehose benchmark

Mauro Bisson, Massimo Bernaschi, Massimiliano Fatica   IEEE HPEC 2016

NVIDIA corporation & Italian Research Council

A CUDA Implementation of the Pagerank Pipeline Benchmark

Mauro Bisson, Everett Phillips, Massimiliano Fatica   IEEE HPEC 2016

NVIDIA corporation

NVIDIA