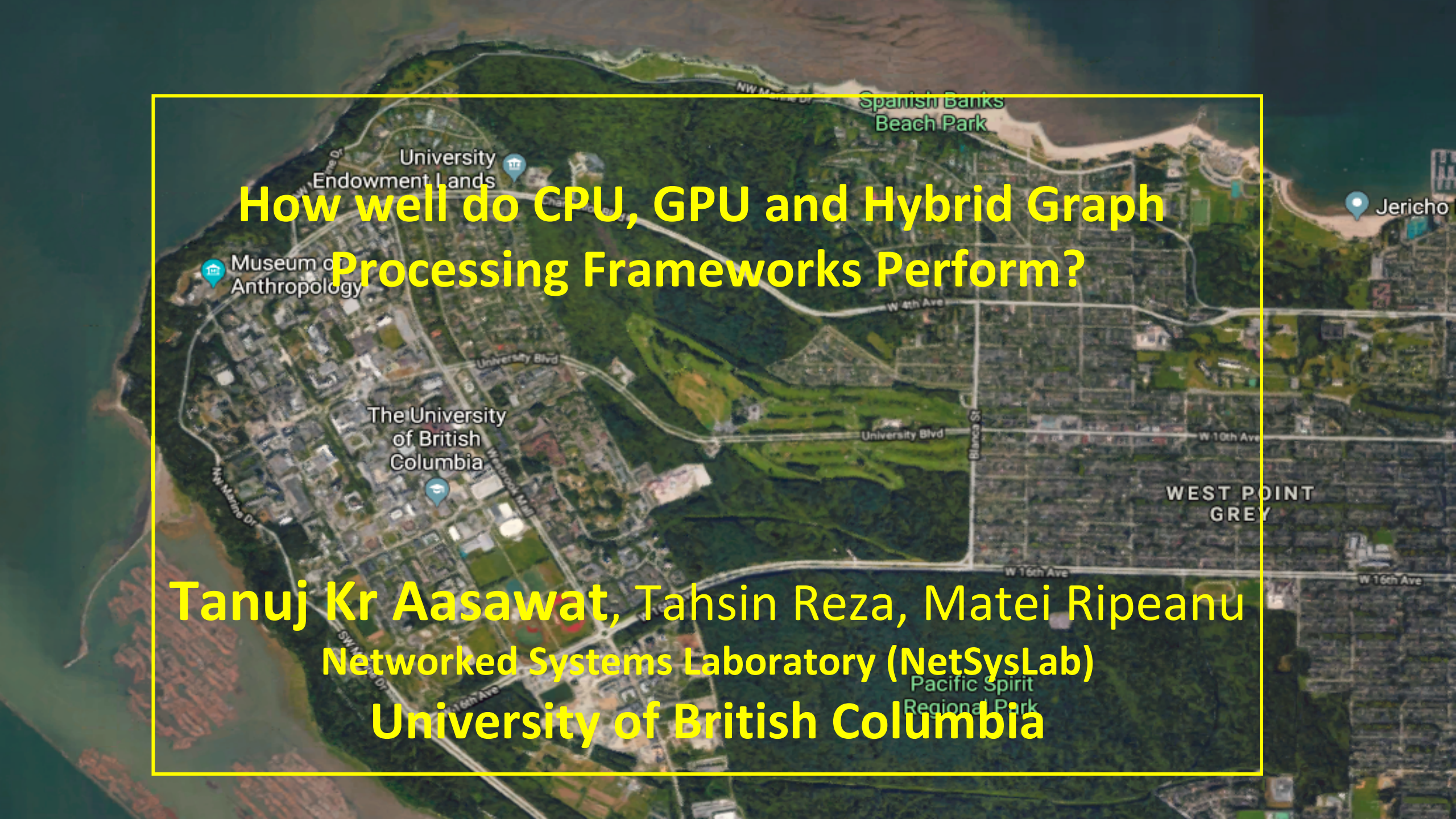
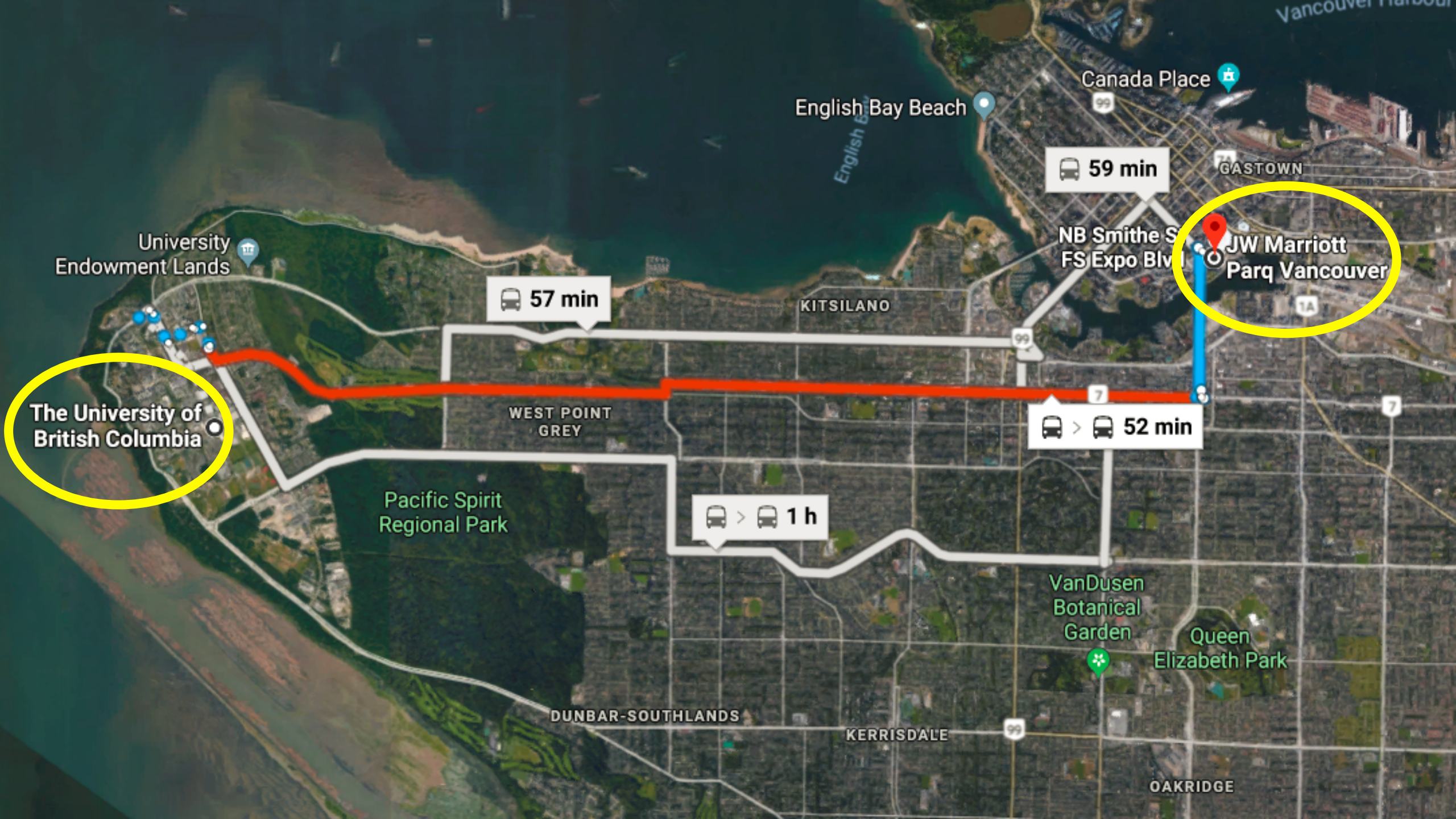


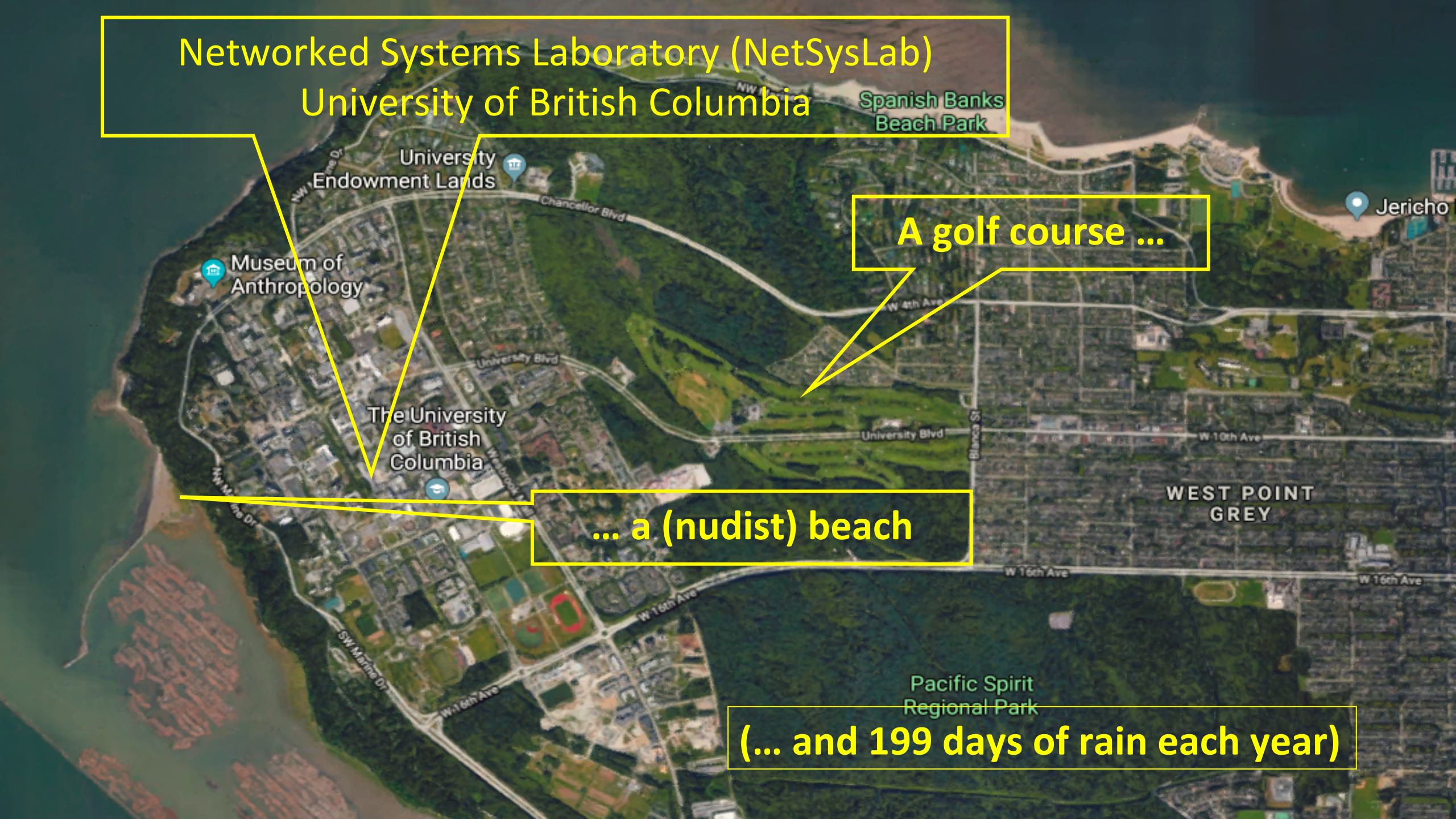
How well do CPU, GPU and Hybrid Graph Processing Frameworks Perform?

Tanuj Kr Aasawat, Tahsin Reza, Matei Ripeanu

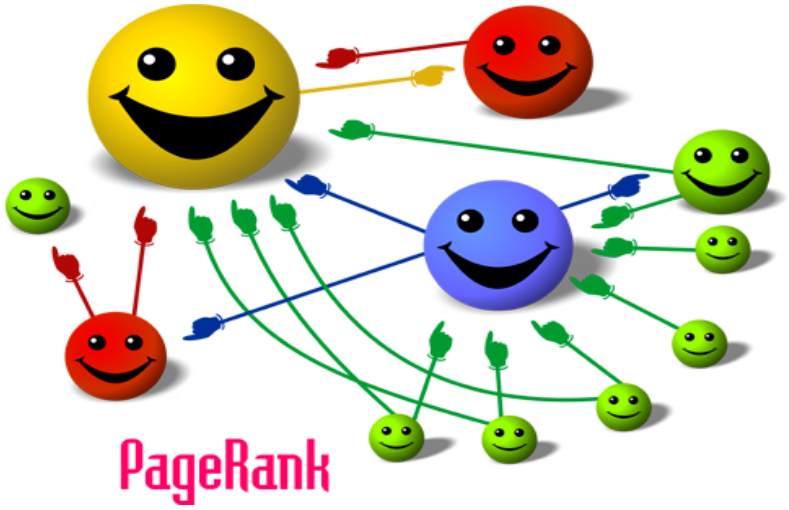
Networked Systems Laboratory (NetSysLab)

University of British Columbia


Canada Place
English Bay Beach
59 min
GASTOWN
NB Smithe S
FS Expo Blvd
JW Marriott
Parq Vancouver
University
Endowment Lands
57 min
KITSILANO
The University of
British Columbia
WEST POINT
GREY
52 min
Pacific Spirit
Regional Park
1 h
VanDusen
Botanical
Garden
Queen
Elizabeth Park
DUNBAR-SOUTHLANDS
KERRISDALE
OAKRIDGE

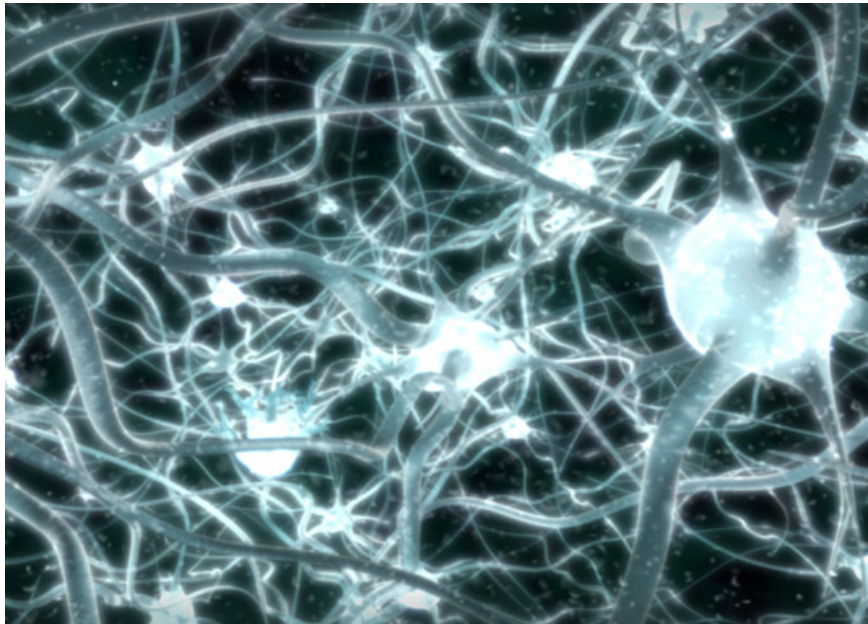

Networked Systems Laboratory (NetSysLab)
University of British Columbia
Spanish Banks Beach Park
University Endowment Lands
Chancellor Blvd
Jericho
A golf course …
Museum of Anthropology
W 4th Ave
University Blvd
The University of British Columbia
University Blvd
Blanca St
W 10th Ave
WEST POINT GREY
… a (nudist) beach
W 16th Ave
W 16th Ave
W 16th Ave
SW Marine Dr
Pacific Spirit Regional Park
(… and 199 days of rain each year)

Graphs are Everywhere

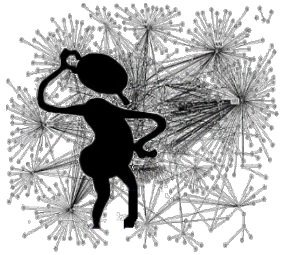


1B users
150B friendships

100B neurons
700T connections



Challenges in Graph Processing



Poor locality

Data-dependent memory access patterns

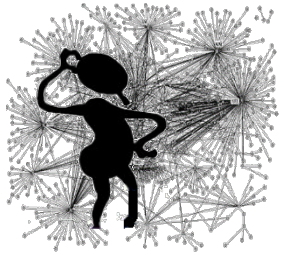
Low compute-to-memory access ratio

Graph500 “mini” graph requires 128 GB.

Large memory footprint

Varying degrees of parallelism (both intra- and inter- stage)

Processing Elements Characteristics



	CPUs	GPUs
		
Poor locality		
Data-dependent memory access patterns	Large Caches	Caches
Low compute-to-memory access ratio		Massive hardware multithreading
Large memory footprint	**>1TB**	**~16GB**
Varying degrees of parallelism (both intra- and inter- stage)		

Graph500 “mini” graph requires 128 GB.

Assemble a hybrid platform?

Graph Processing Frameworks

Programming Model

(Vertex Programming/Linear Algebra)

Architecture

(Single-node or ~~Distributed~~)

CPU/GPU/Hybrid

High Performance

Motivation

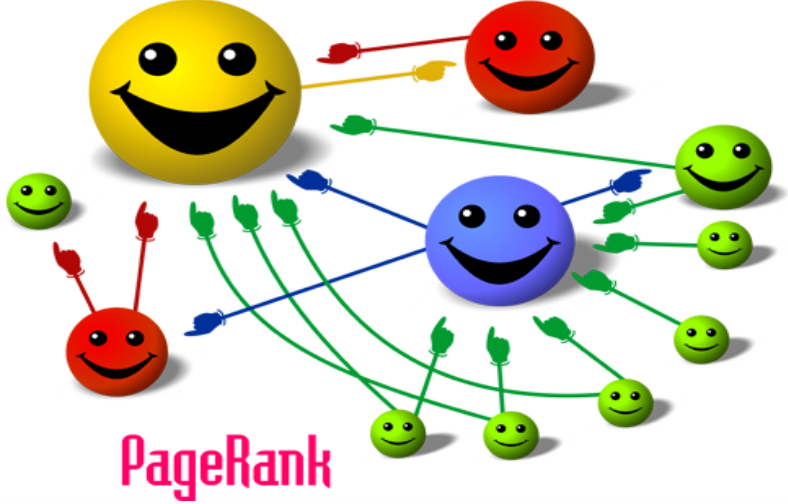
How architecture and programming model combination improves **performance** and **efficiency** of the system as a whole?

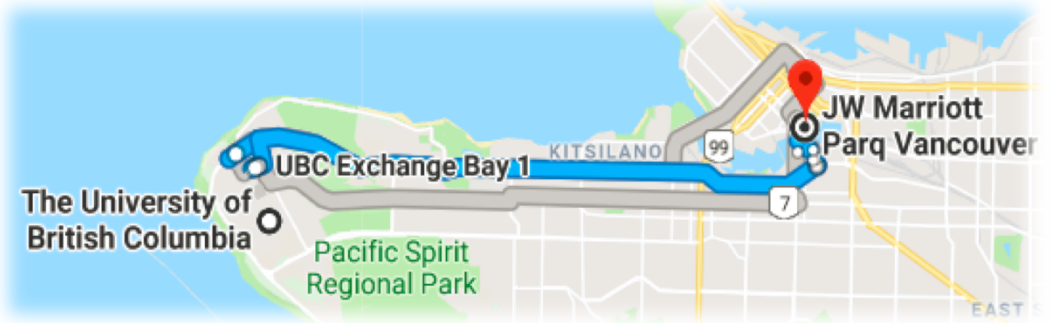
Graph Processing Frameworks

Framework	Architecture	Programming Model
Galois **UTexas, Austin**	CPU	Vertex Programming
GraphMat **Intel**	CPU + Distributed	Linear Algebra
Gunrock **UC, Davis**	Multi - GPU	Vertex Programming
Nvgraph **Nvidia**	GPU	Linear Algebra
Totem **UBC**	CPU + multi-GPU	Vertex Programming

Benchmark Algorithms

- PageRank
  - Ranking web pages
  - Compute intensive
- Single Source Shortest Paths (SSSP)
  - IP routing, Transportation networks
- Breadth-First Search (BFS)
  - Finding connected component, subroutine
  - Memory intensive





Evaluation Metrics

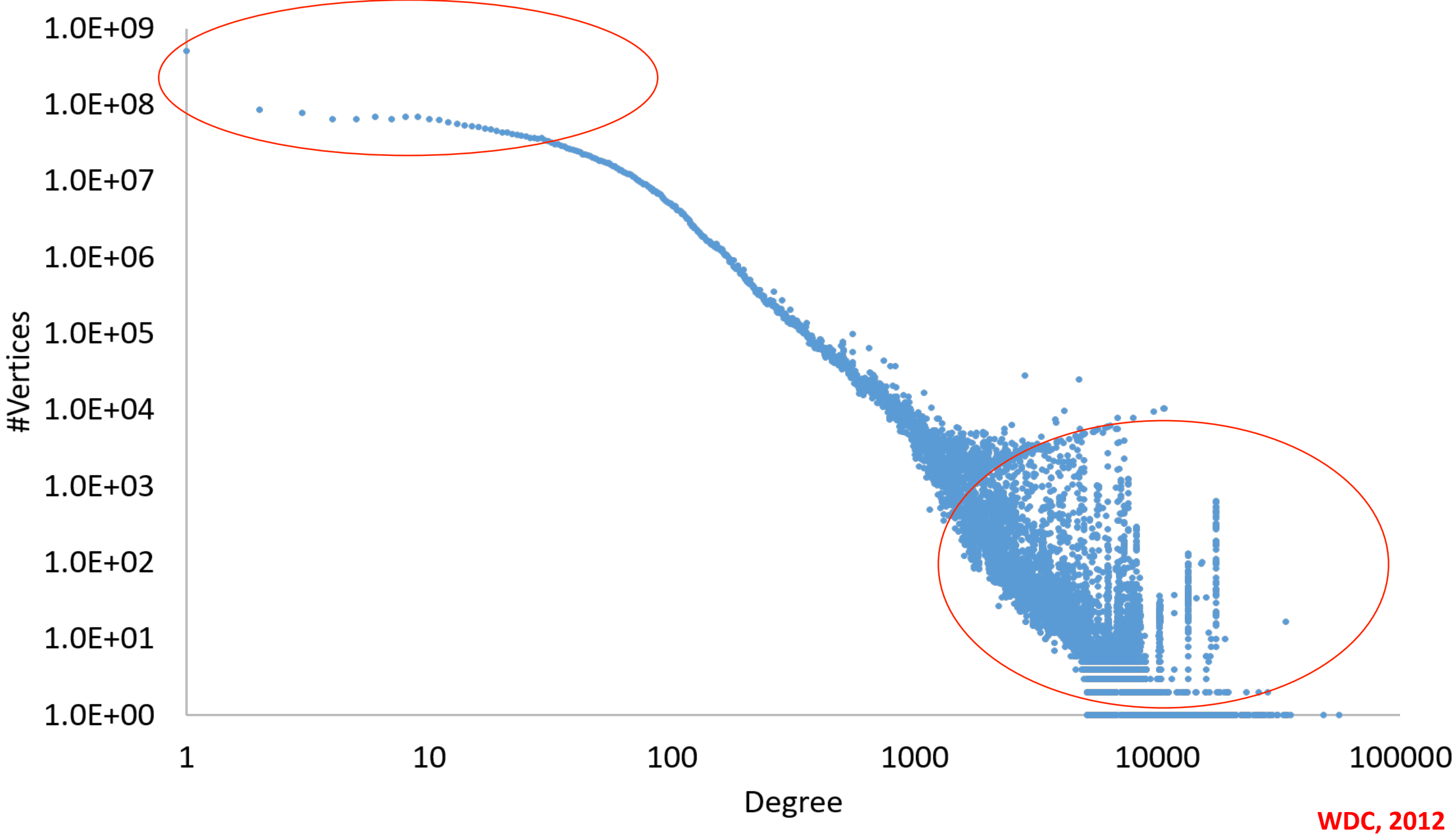
- Raw Performance
  - Traversed Edges Per Second (TEPS): *Traversed Edges / Execution Time*
- Energy Consumption
  - *Average Power consumed * Execution Time*
- Scalability
  - Strong scaling w.r.t processing units

Testbed Characteristics

	System 1
CPU	2x Intel Xeon E5-2695 v3 (Haswell)
#CPU Cores	28
Host Memory	**512 GB DDR4**
L3 Cache	70 MB
PCIe	3.0 – x16
GPU	2x Nvidia Tesla K40c
GPU Thread Count	2880
GPU Memory	**12 GB**

Datasets

	Graph	#Vertices	#Edges	Max Degree	Avg. Degree
Real World	Com-Orkut	3 M	234 M	33,313	78
	liveJournal	4.8 M	68 M	20,292	14
	Road-USA	28.8 M	47.9 M	9	1.6
	Twitter	52 M	3.9 B	3,691,240	75
Synthetic	RMAT22	4 M	128 M	168,729	32
	RMAT23	8 M	256 M	272,808	32
	RMAT24	16 M	512 M	439,994	32
	RMAT27	128 M	4 B	3,910,241	32


1.0E+09
1.0E+08
1.0E+07
1.0E+06
1.0E+05
1.0E+04
1.0E+03
1.0E+02
1.0E+01
1.0E+00
#Vertices
1
10
100
1000
10000
100000
Degree
WDC, 2012

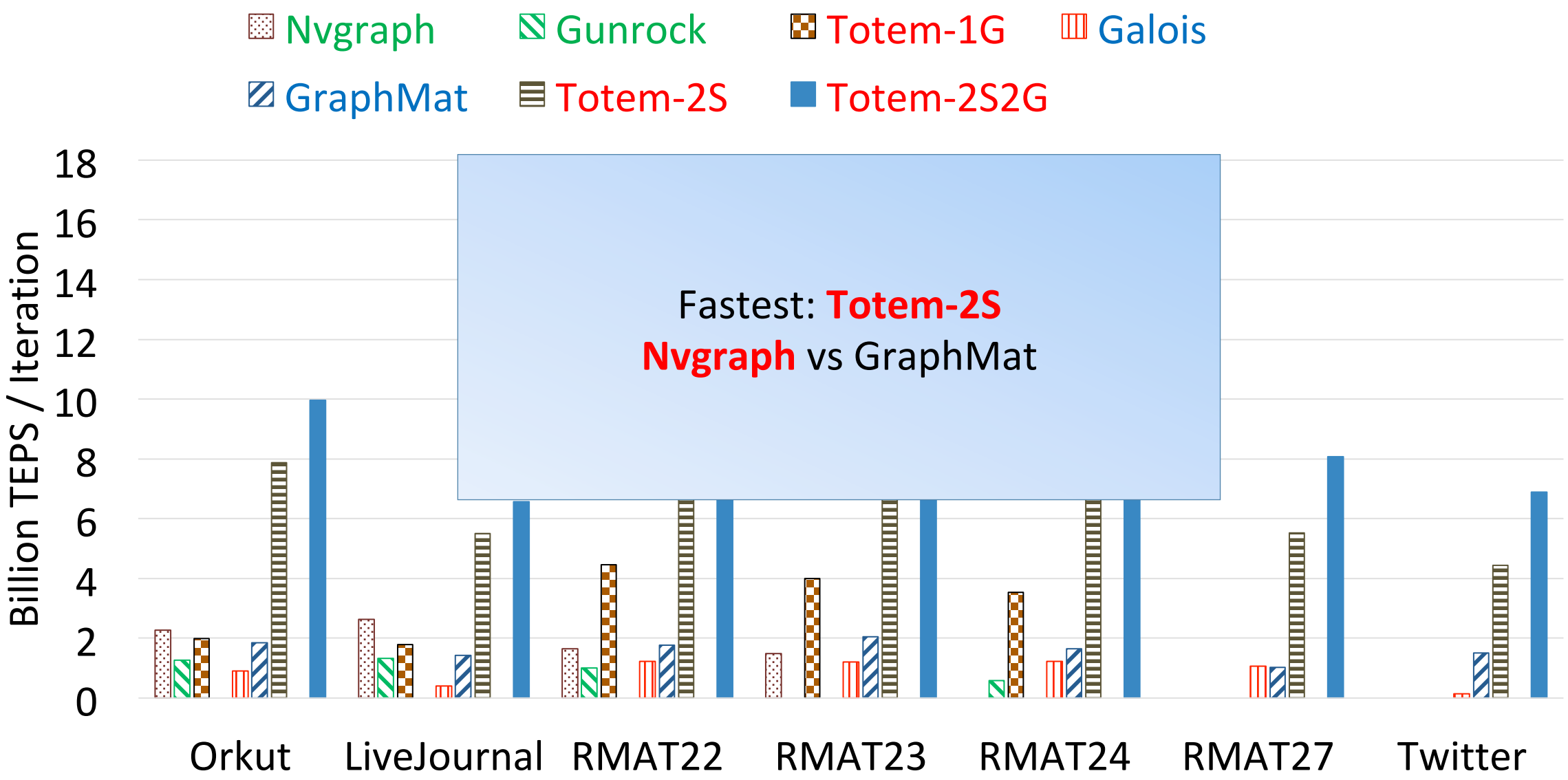
Memory Consumption

Framework	Memory layout	PageRank	SSSP	BFS
Nvgraph	CSC (PageRank, SSSP) and CSR (BFS)	1,159 (1.8x)	**1,111 (1.0x)**	**683 (1.0x)**
Gunrock	CSR and COO	**641 (1.0x)**	1,582 (1.4x)	1,443 (2.1x)
Galois	CSR	1,599 (2.5x)	2,074 (1.9x)	1,432 (2.1x)
GraphMat*	DCSC	**2,818 (4.4x)**	**2,786 (2.5x)**	**2,980 (4.4x)**
Totem-2S	CSR	1,275 (2.0x)	2,198 (2.0x)	1,282 (1.9x)
Totem-2S2G	CSR	1,628 (2.5x)	2,587 (2.3x)	1,658 (2.4x)

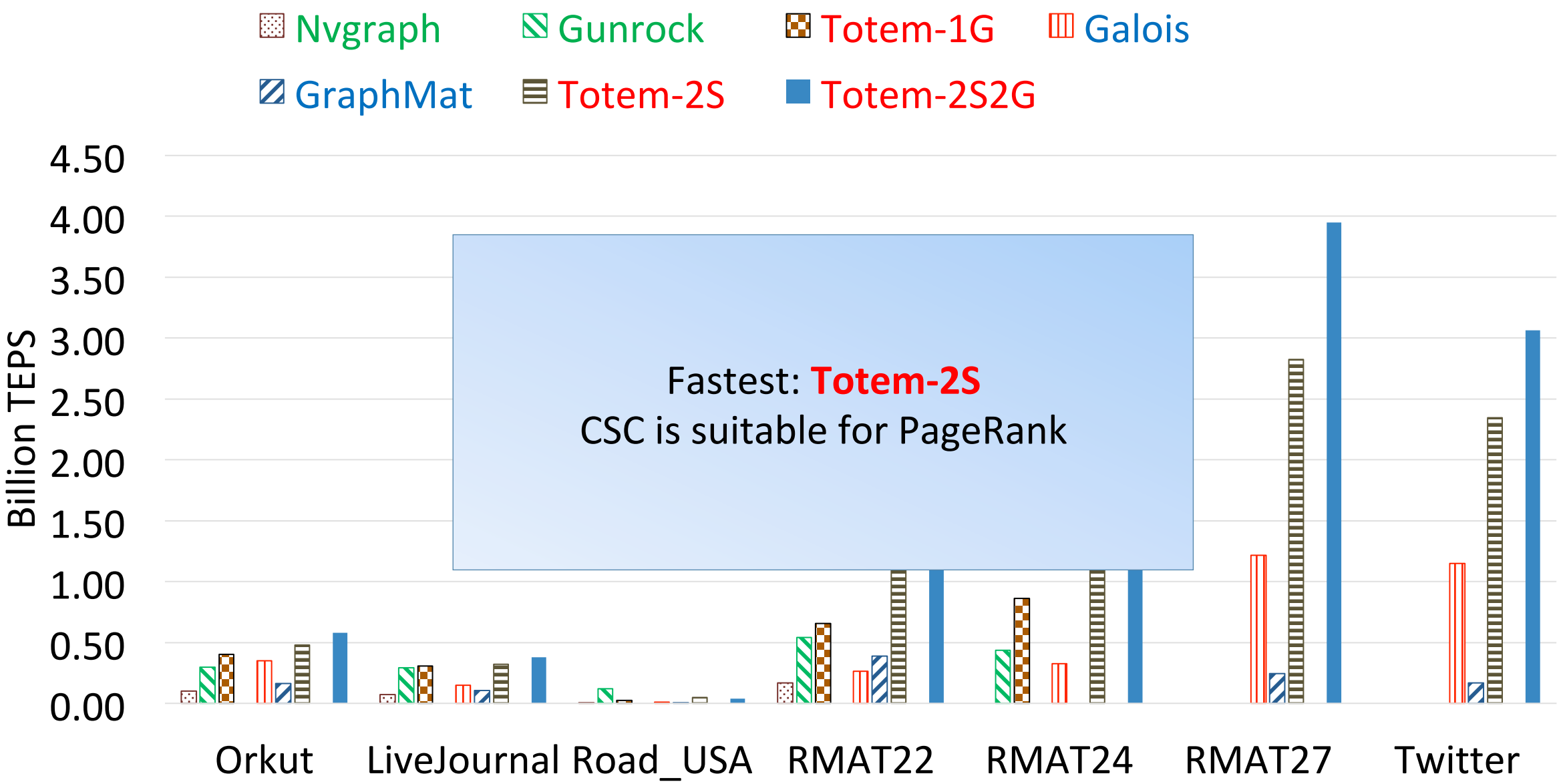
9,354 MB during pre-processing step

Memory Consumption (in MB) for RMAT22 graph (edge list size: 512 MB)

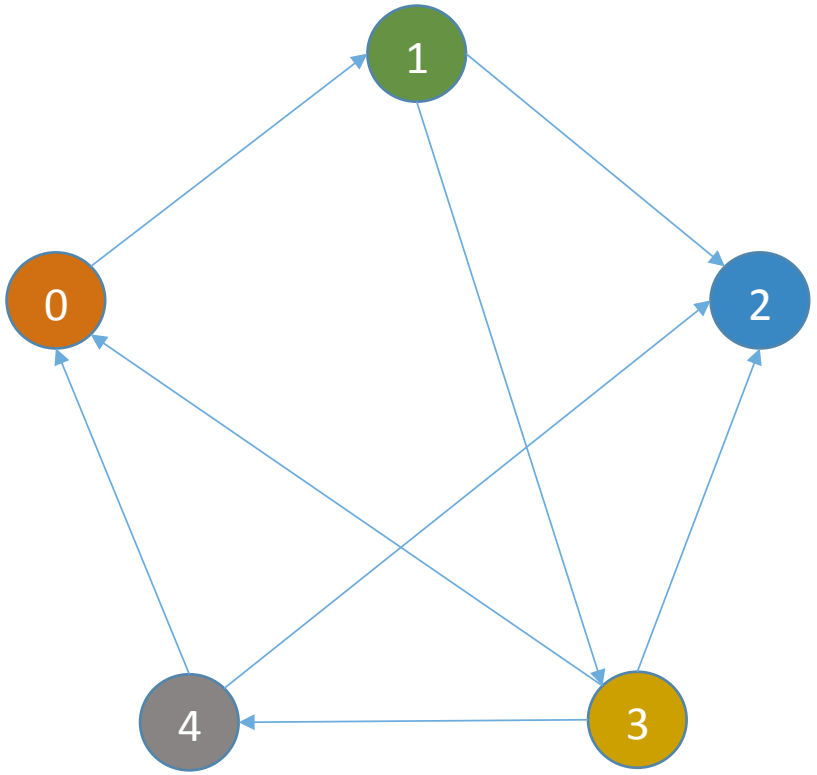
Experimental Results
1. Raw Performance - PageRank



Experimental Results
1. Raw Performance - SSSP



Graph Layout in Memory



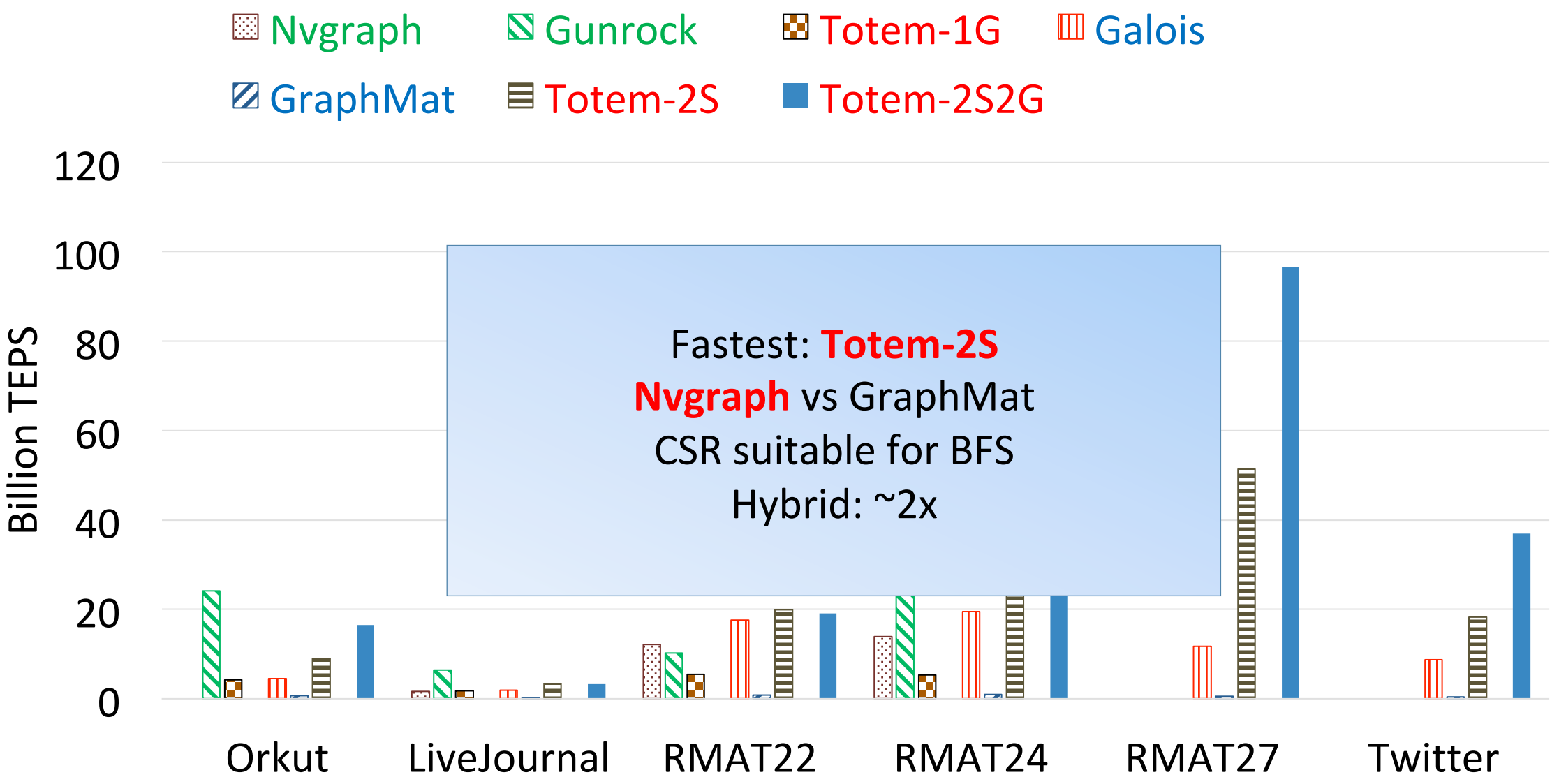
CSR Representation

rowPtr	0	1	3	3	6	8	
VertexId	0	1	2	3	4	5*	
edgeList	1	2	3	0	2	4	0	2
	0	1	2	3	4	5	6	7

CSC Representation

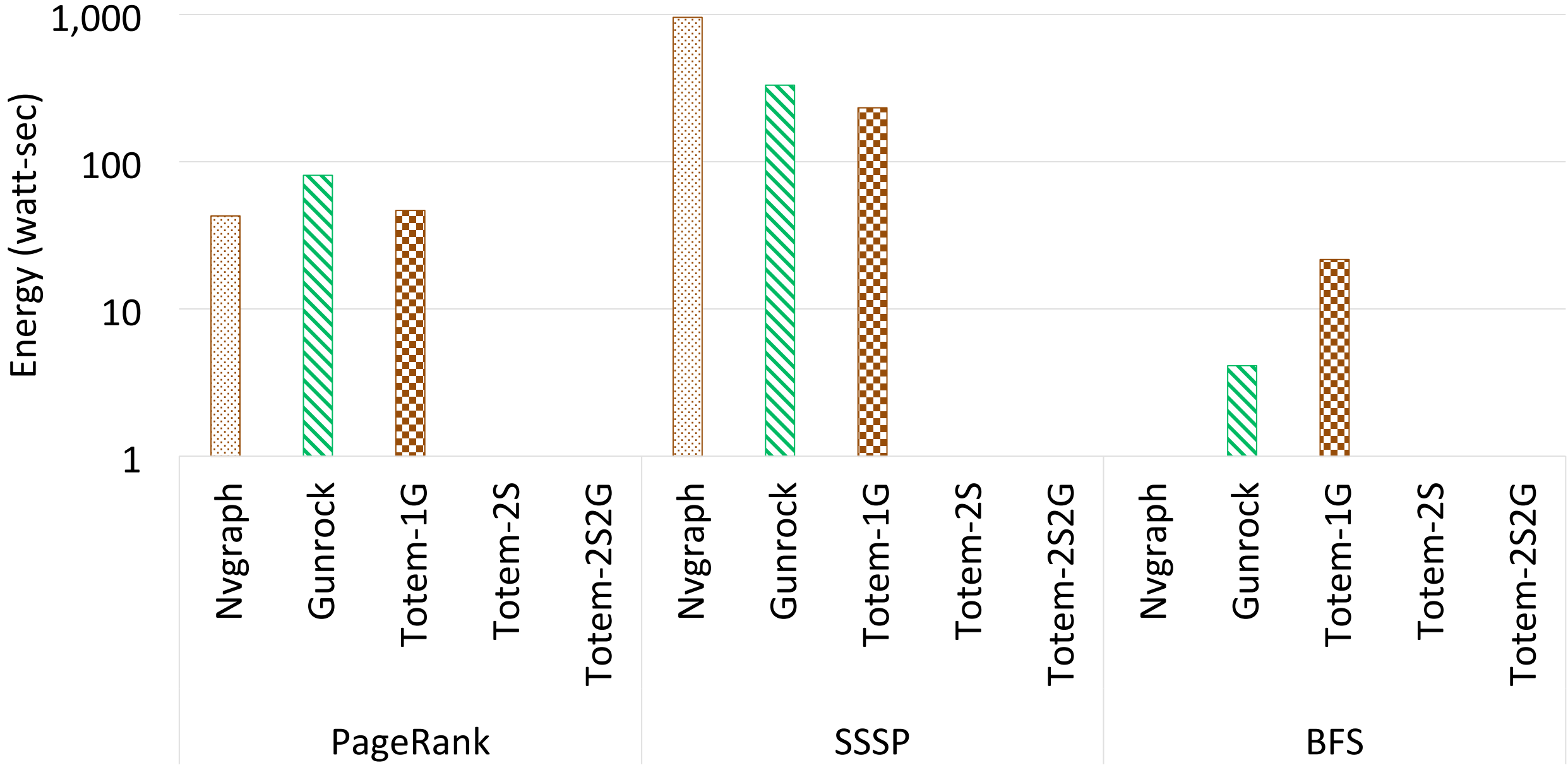
colPtr	0	2	3	6	7	8
VertexId	0	1	2	3	4	5*
edgeList	3	4	0	1	3	4	1	3
	0	1	2	3	4	5	6	7

Experimental Results
1. Raw Performance - BFS



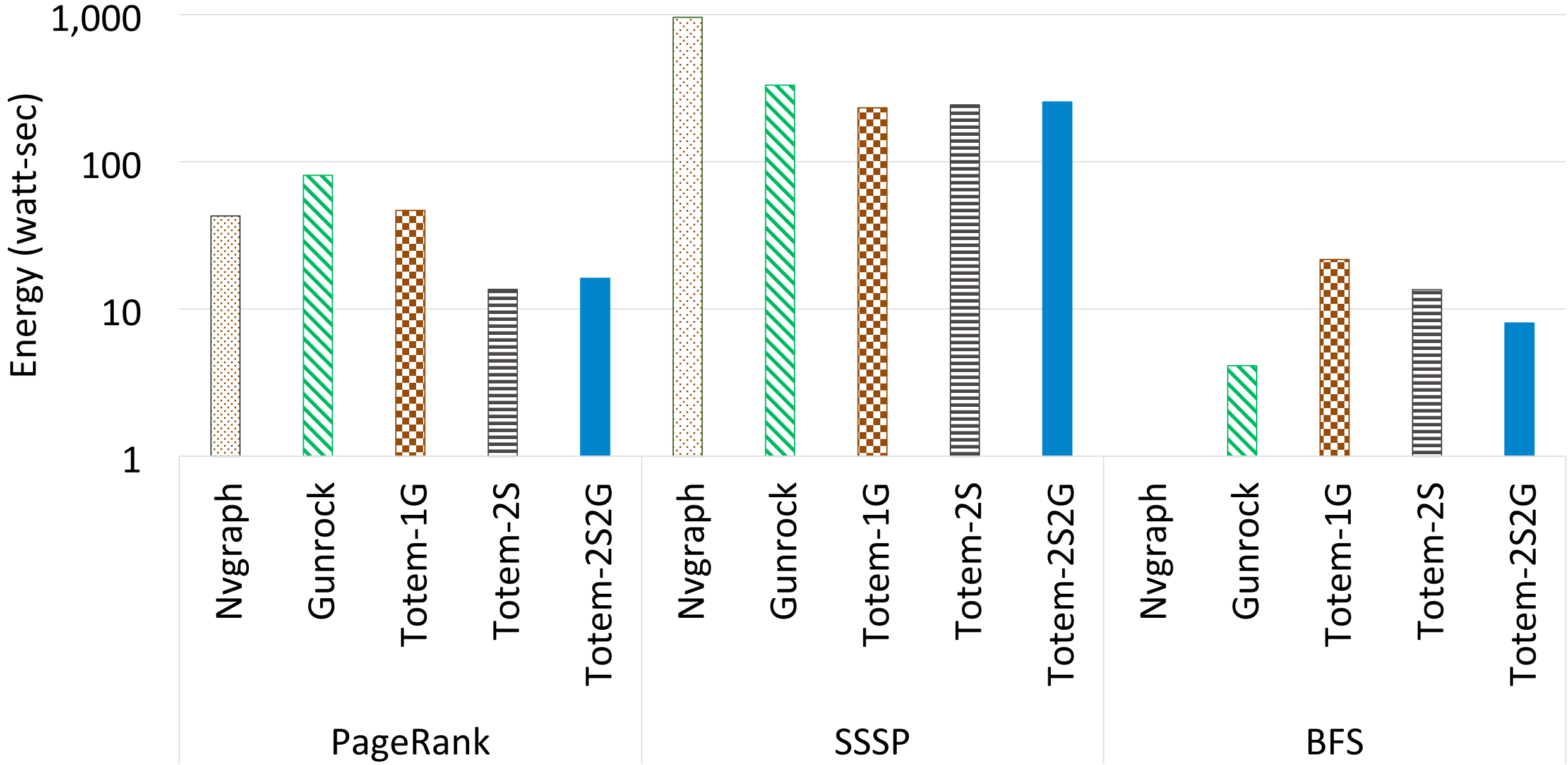
Experimental Results

2. Energy Consumption – GPU Frameworks – Orkut Workload



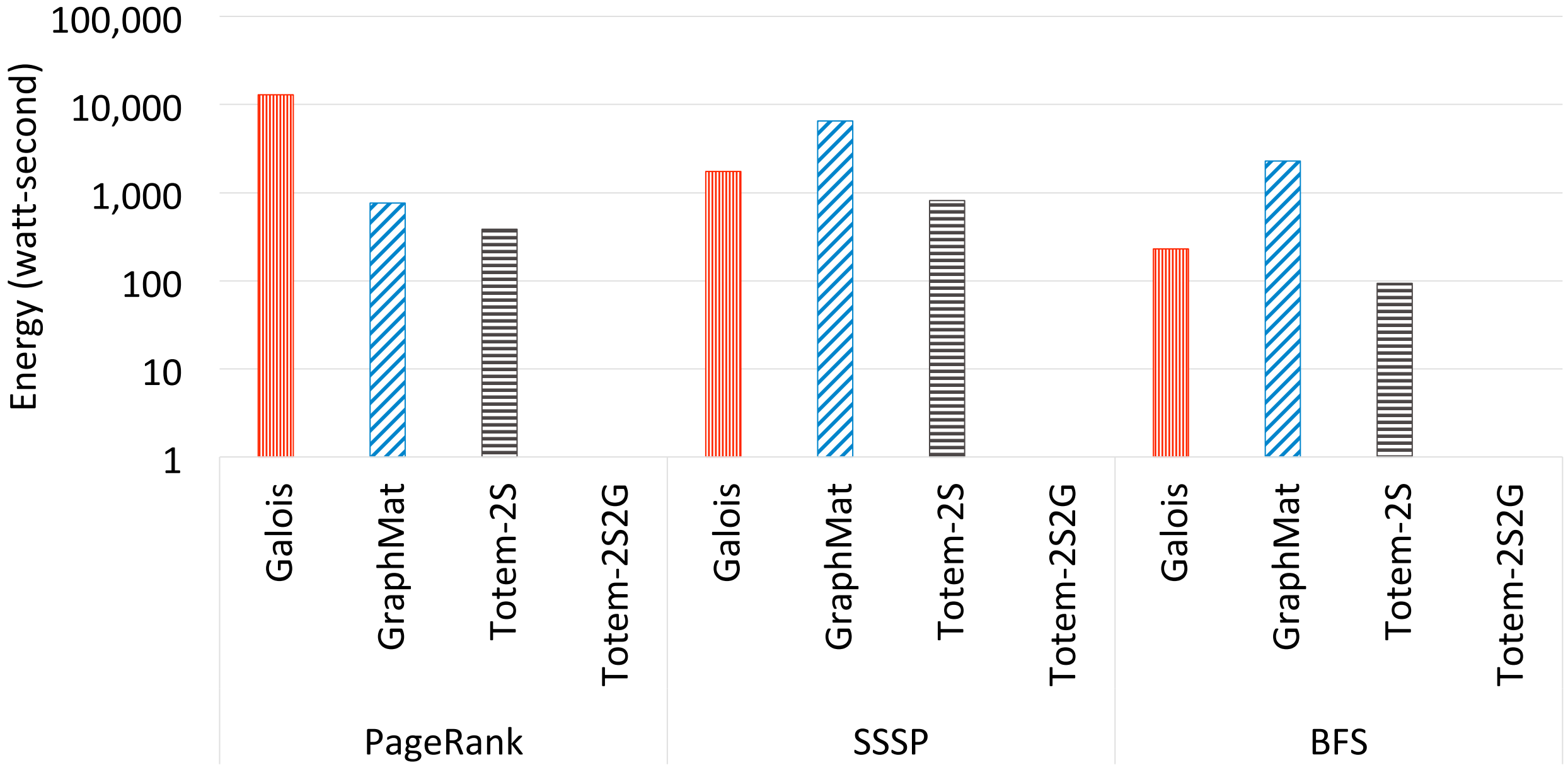
Experimental Results

2. Energy Consumption – GPU Frameworks – Orkut Workload



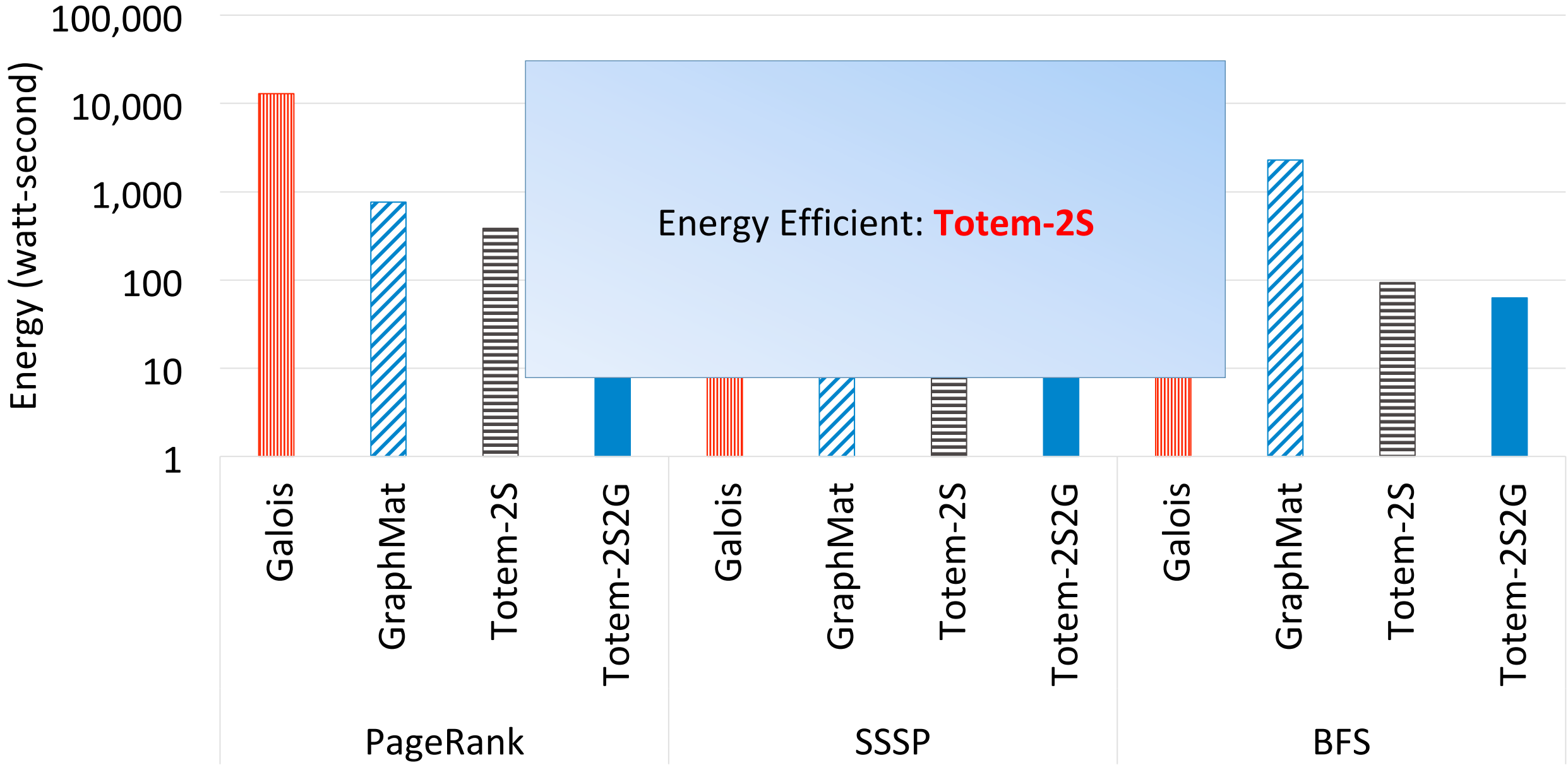
Experimental Results

2. Energy Consumption – CPU Frameworks – Twitter Workload



Experimental Results

2. Energy Consumption – CPU Frameworks – Twitter Workload

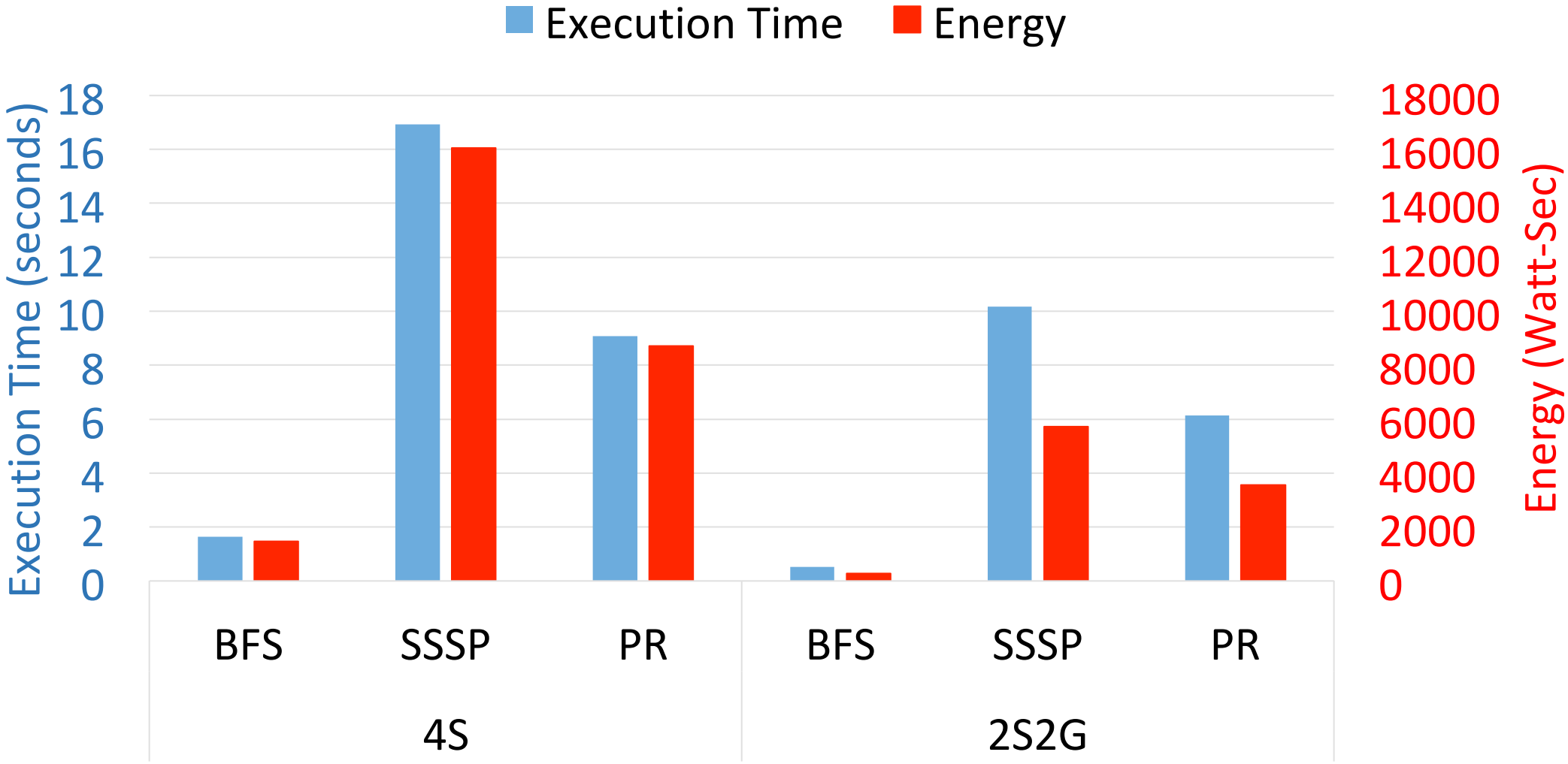


Summary

- GPU + Linear Algebra | CPU + Vertex programming = Good Match
- GPU based frameworks: ?
- CPU based frameworks: Totem-2S
- Totem Hybrid: Greenest
- CSC ⟷ PageRank
- CSR ⟷ BFS, SSSP

Discussion

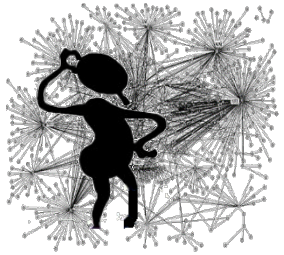
Does hybrid have the future potential?



Totem-4S vs Totem-2S2G for RMAT30 (edge list size: 128 GB)

4S Machine: 4x Intel Xeon E7-4870 v2 (Ivy bridge), with **1,536 GB** memory

Hybrid Graph Processing



Graph Processing

Poor locality

Data-dependent memory access patterns

Low compute-to-memory access ratio

Large memory footprint

Varying degrees of parallelism (both intra- and inter- stage)

CPUs

Large Caches + summary data structures

>1TB

High Degree

GPUs

Caches + summary data structures

Massive hardware multithreading

16GB!

Low Degree

Questions

code@: **netsyslab.ece.ubc.ca**