Collaborative (CPU+GPU) Algorithms for Triangle Counting and Truss Decomposition

Vikram S. Mailthody[1], Ketan Date[2], Zaid Qureshi[3], Carl Pearson[1], Rakesh Nagi[2], Jinjun Xiong[4], Wen-mei Hwu[1]

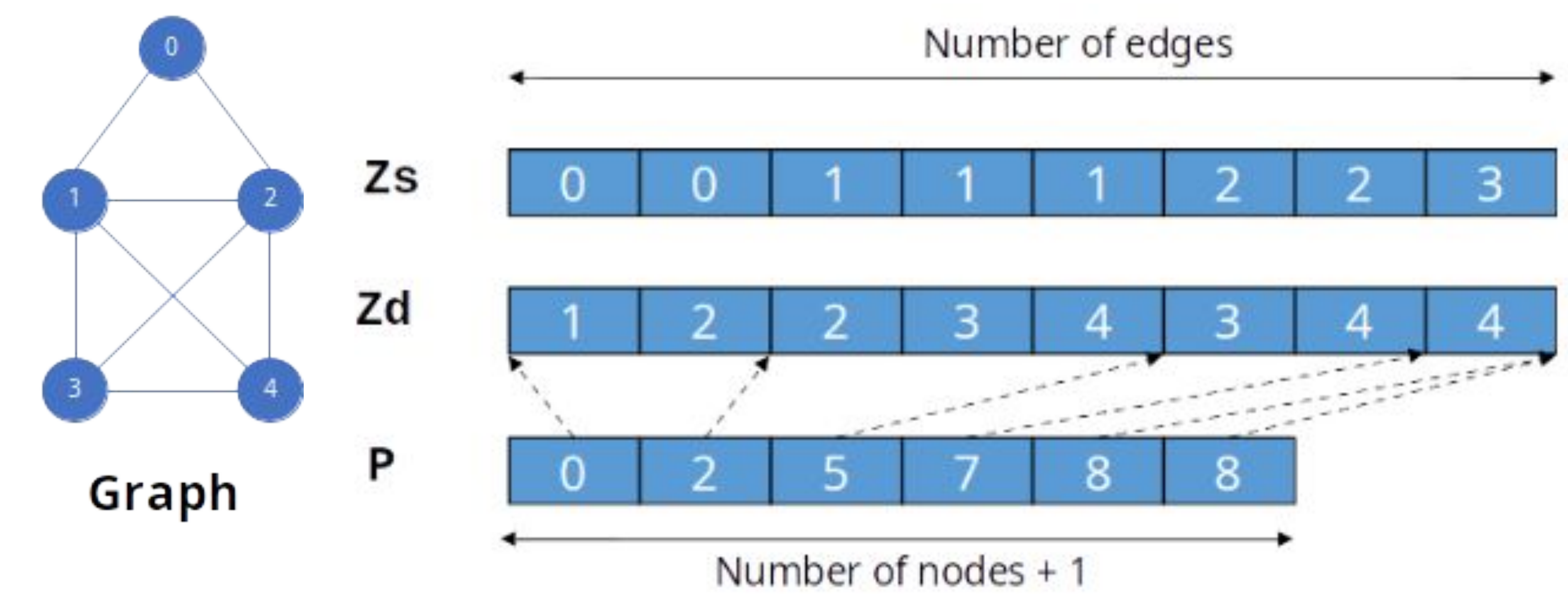
[1] ECE, [2] ISE, [3] CS, University of Illinois at Urbana-Champaign, Urbana, IL 61801

[4] Cognitive Computing & University Partnership, IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598

Introduction

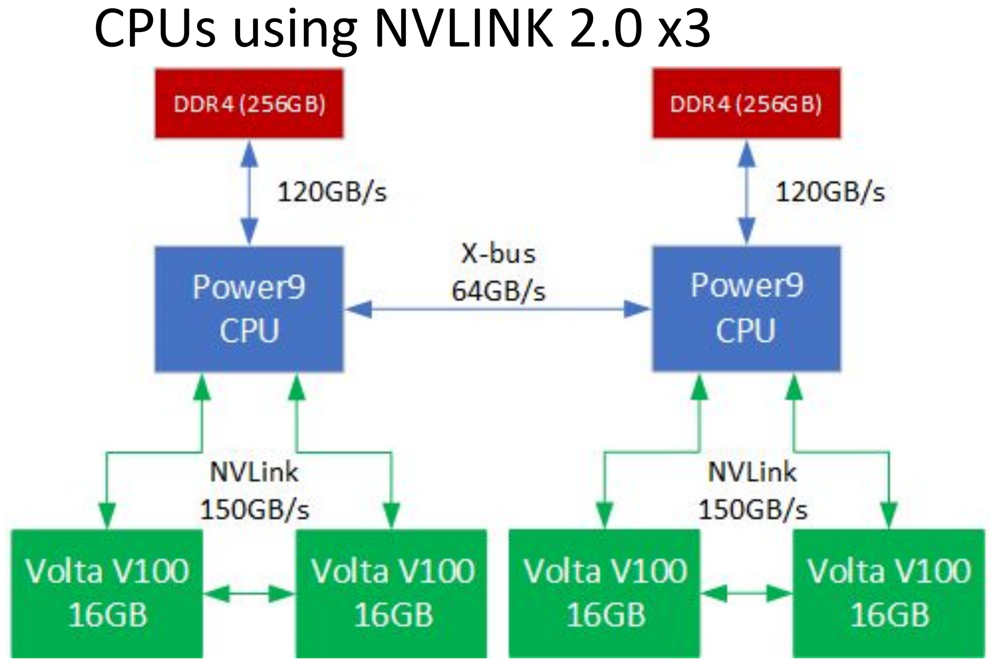
- Finding **Triangles** and **Trusses** in large graphs efficiently using both GPUs and CPUs on the IBM Newell Machine
- **Triangle Count (TC)**: # of cycles of length 3
- **k-Truss Decomposition (TD)**: Non-trivial subgraph with each edge connected to at least k-2 triangles [1, 2].
- Objective:
  - To **optimize** our previous work [4] for better resource (compute and bandwidth) utilization
  - Submission for Static Graph Challenge [3]

Data Layout



Newell System Architecture

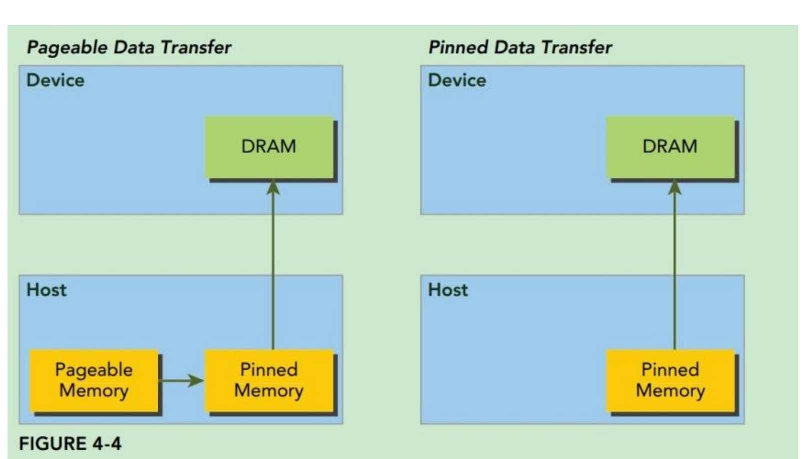
- 2 IBM Power9 CPUs each with
  - 10 Cores (80 threads)
  - 256GB of Memory
- 4 NVIDIA Tesla V100 (Volta) GPUs each with
  - 5120 Cuda cores
  - 16GB of HBM2 Memory
- GPUs connected to each other and CPUs using NVLINK 2.0 x3



Memory Management

Zero-copy Memory:

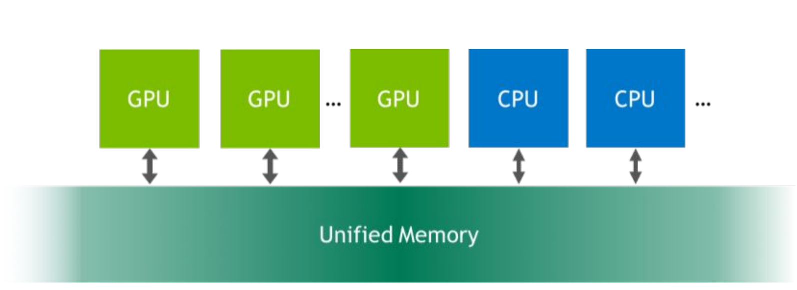
Allocated using `cudaHostAlloc` with `cudaHostAllocMapped` flag.



http://nanxiao.me/tag/cuda/

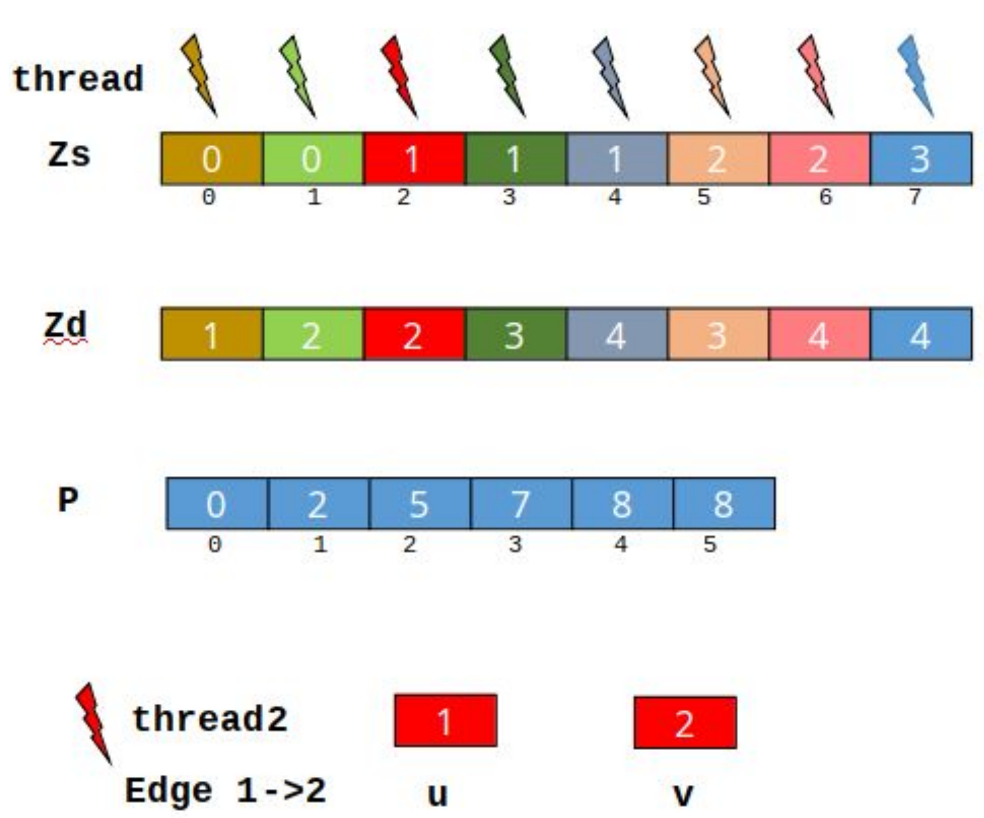
Unified Memory:

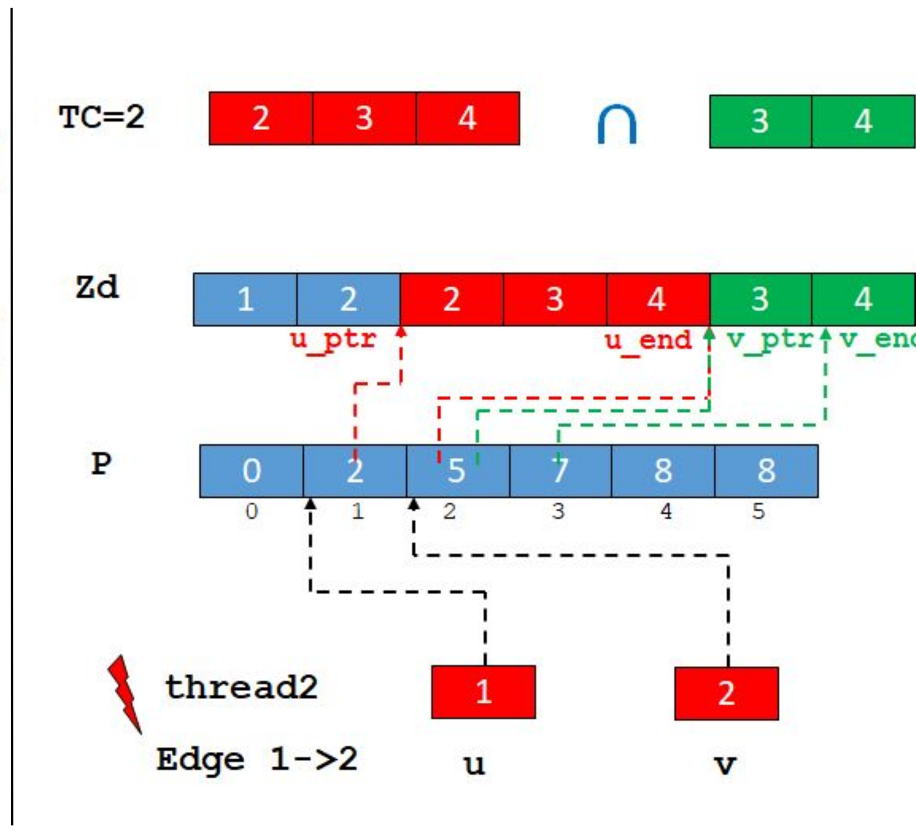
Allocated using `cudaMallocManaged`.



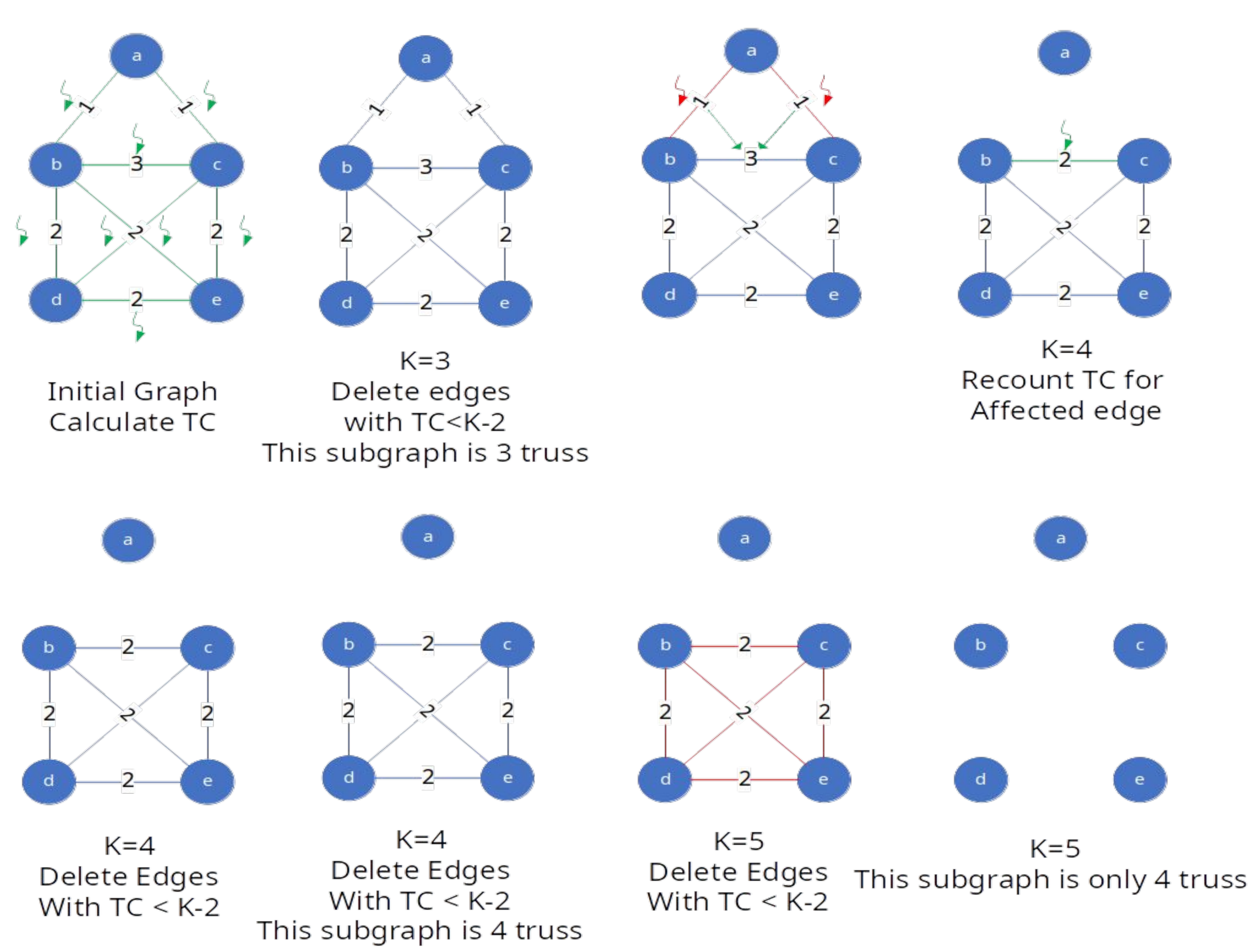
https://devblogs.nvidia.com/parallelforall/unified-memory-cuda-beginners/

Triangle Counting





Truss Decomposition



Optimization: Delete edges in **2** phases

- **Short Update**: Mark affected to-be-deleted edges with a sentinel value, and not count them for future triangle counts.
- **Long Update:** Once per k-value, aggregate all short updates and perform stream compaction to remove edges from graph.

Results

TABLE IV

ZERO-COPY AND UNIFIED MEMORY TRIANGLE COUNTING BENCHMARKS ON LARGE GRAPHS

Graph [1]	n	m	TC	Single GPU		Zero-copy (4 GPUs)		Unified (4 GPUs)	
				Time (s)	Edges/s	Time (s)	Edges/s	Time (s)	Edges/s
flickrEdges	105,938	2,316,948	107,987,357	0.011	215,880,798	0.213	10,867,996	0.026	88,016,563
cit-Patents	3,774,768	16,518,947	7,515,023	0.049	334,014,352	0.196	84,071,878	0.140	118,021,984
Kmer - Graph5	55,042,369	58,608,800	1,443	0.092	638,440,087	0.490	119,559,044	0.603	97,132,052
Network - Graph5	226,196,185	240,023,945	26	5.925	40,511,923	9.139	26,262,472	7.091	33,848,266
graph500-scale18-ef16	174,147	7,600,696	82,287,285	0.037	203,201,787	1.287	5,906,233	1.356	5,606,713
graph500-scale19-ef16	335,318	15,459,350	186,288,972	0.099	155,426,618	2.109	7,329,200	2.399	6,445,287
graph500-scale20-ef16	645,820	31,361,722	419,349,784	0.277	113,259,001	4.323	7,255,159	2,841.665	11,036
graph500-scale21-ef16	1,243,072	63,463,300	935,100,883	0.699	90,784,271	19.481	3,257,717	7,018.016	9,043
graph500-scale22-ef16	2,393,285	128,194,008	2,067,392,370	1.823	70,330,803	77.586	1,652,290	17,019.189	7,532
graph500-scale23-ef16	4,606,314	258,501,410	4,549,133,002	5.238	49,346,912	85.741	3,014,908	>18,000	-
graph500-scale24-ef16	8,860,450	520,523,686	9,936,161,560	14.853	35,045,368	367.430	1,416,659	>18,000	-
graph500-scale25-ef16	17,043,780	1,046,934,896	21,575,375,802	38.294	27,339,066	1,266.137	826,873	>18,000	-
Friendster	119,432,957	1,799,999,986	191,716	-	-	57.036	31,558,975		

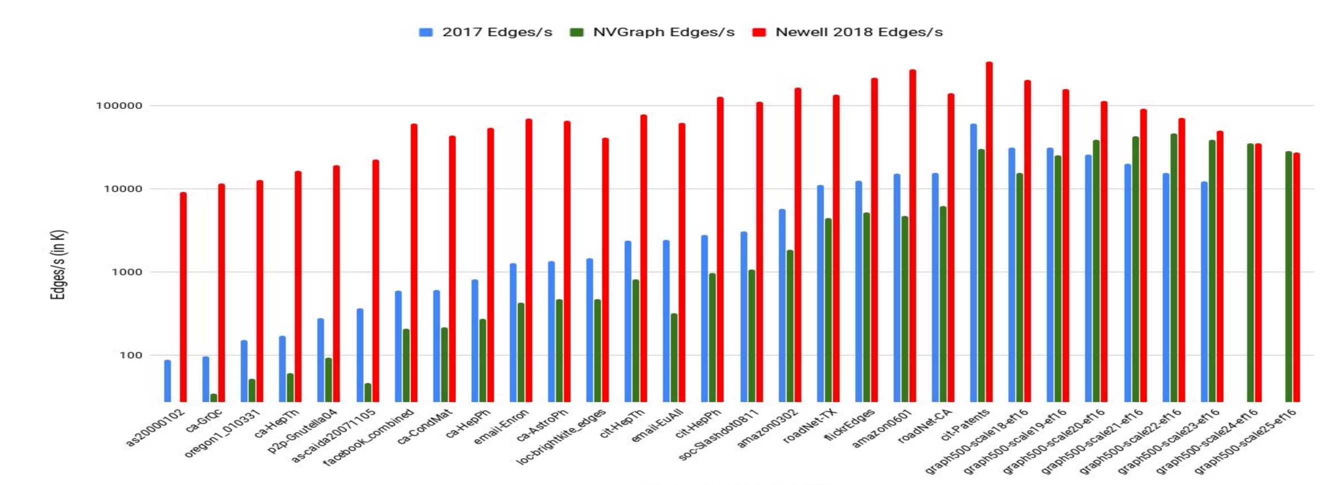
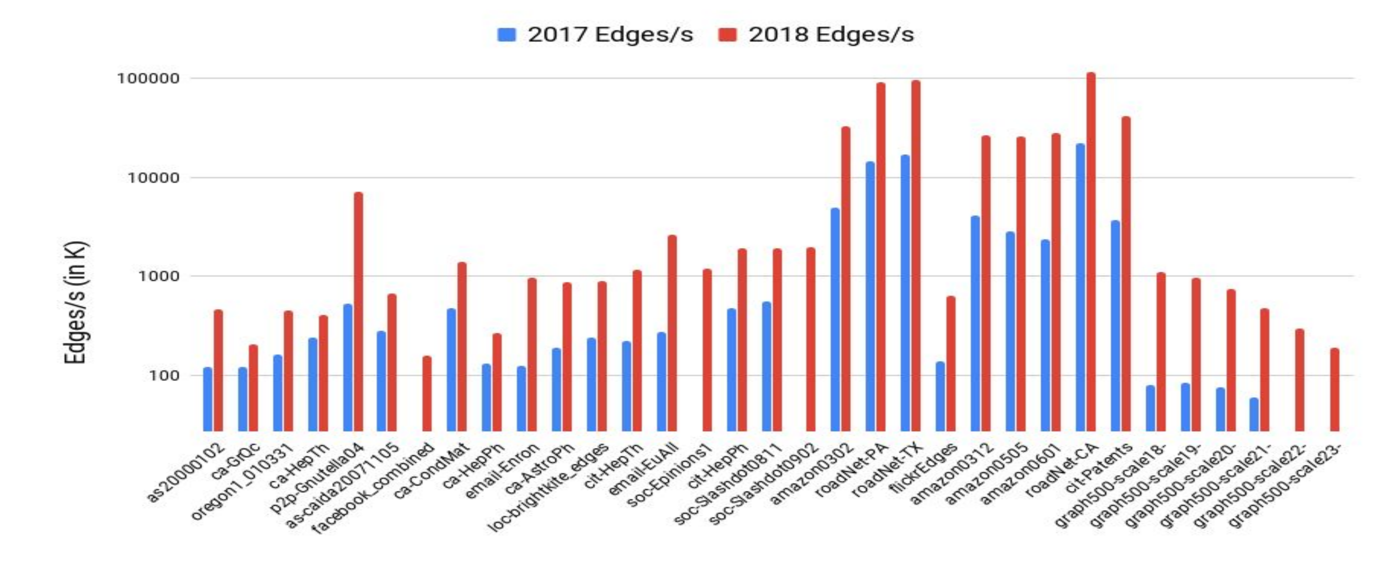


TABLE V

ZERO-COPY TRUSS DECOMPOSITION BENCHMARKS ON LARGE GRAPHS

Graph[1]	n	m	k_{max}	Single GPU		Zero-copy (4 GPUs)	
				Time (s)	Edges/s	Time (s)	Edges/s
flickrEdges	105,938	2,316,948	574	7.225	320,704	79.607	29,105
cit-Patents	3,774,768	16,518,947	36	0.790	20,899,139	4.917	3,359,763
Kmer - Graph5	55,042,369	58,608,800	3	0.497	117,814,922	2.757	21,261,171
graph500-scale18-ef16	174,147	7,600,696	159	6.896	1,102,217	61.353	123,885
graph500-scale19-ef16	335,318	15,459,350	213	15.946	969,474	192.019	80,509
graph500-scale20-ef16	645,820	31,361,722	284	41.741	751,343	683.304	45,897
graph500-scale21-ef16	1,243,072	63,463,300	373	133.110	476,772	2,868.196	22,127
graph500-scale22-ef16	2,393,285	128,194,008	485	432.494	296,406	9,709.680	13,203
graph500-scale23-ef16	4,606,314	258,501,410	625	1,377.733	187,628	>10,000	-



Conclusions

- Our optimizations provided excellent speedup of our previous submission [4] (TC: 47.6x avg, 117x max; TD: 5.7x avg, 13.52x max)
- Large graphs can be processed efficiently using Zero Copy memory (ie. 57 sec for TC on Friendster graph)
- Future research:Applying graph partitioning to improve multi-GPU performance

References

[1] J. Cohen. *Trusses: Cohesive subgraphs for social network analysis*. In *National Security Agency Technical Report*, page 16, 2008.
[2] J. Cohen. *Graph twiddling in a MapReduce world.* In *Computing in Science & Engineering*, 11(4):29-41, 2009
[3] S. Samsi, V. Gadepally, M. Hurley, M. Jones, E. Kao, S. Mohindra, P. Monticciolo, A. Reuther, S. Smith, W. Song, D. Staheli, and J. Kepner. *Static graph challenge: Subgraph isomorphism.* In *IEEE HPEC,* 2017.
[4] K. Date, K. Feng, R. Nagi, J. Xiong, N. S. Kim, and W. M. Hwu. *Collaborative (cpu + gpu) algorithms for triangle counting and truss decomposition on the minsky architecture.* In *IEEE HPEC*, page 1-7, 2017.