

# Load Balancing Eager K-truss on GPU and CPU via Fine-Grained Parallelism

*PRESENTED BY*

<sup>1,2</sup>Mark Blanco ([markb1@cmu.edu](mailto:markb1@cmu.edu)), <sup>2</sup>Tze Meng Low, <sup>1</sup>Kyungjoo Kim

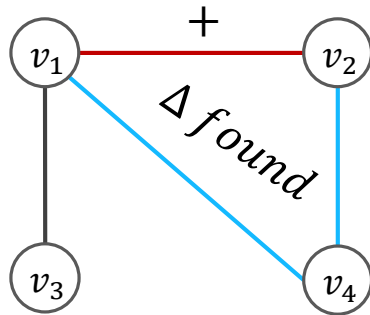
<sup>1</sup>*Center for Computing Research, Sandia National Laboratories*

<sup>2</sup>*Electrical & Computer Engineering, Carnegie Mellon University*

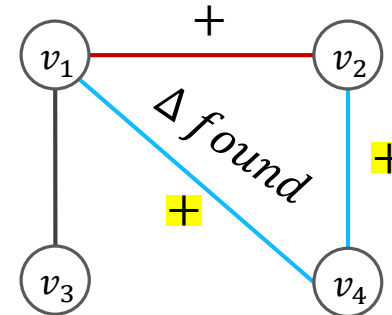
# Previously: Eager K-Truss



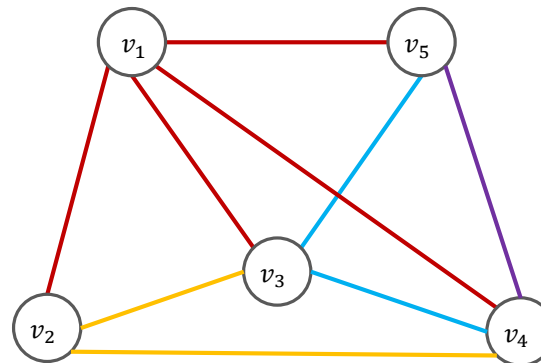
K-Truss



Eager K-Truss



Enumerate triangles in parallel over edges grouped by source vertex



$t_1, t_2, t_3, t_4$

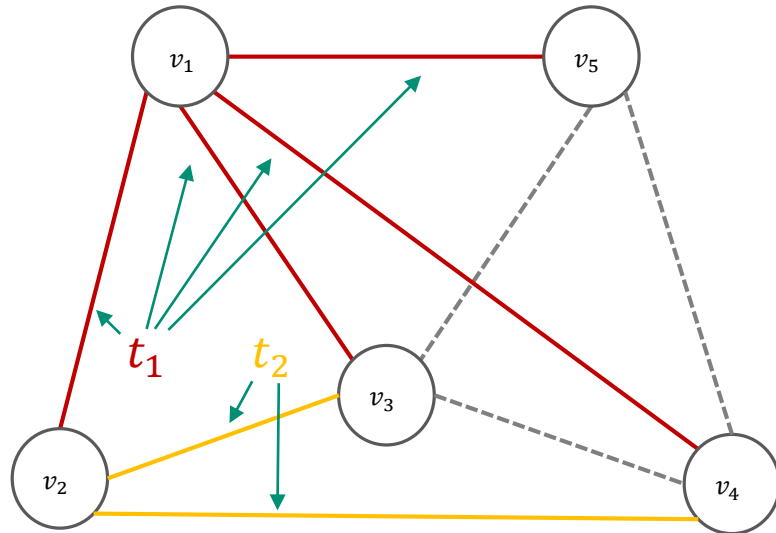
Load imbalance occurs between threads processing different outgoing neighborhoods!

# This work: Fine-grained Parallelism



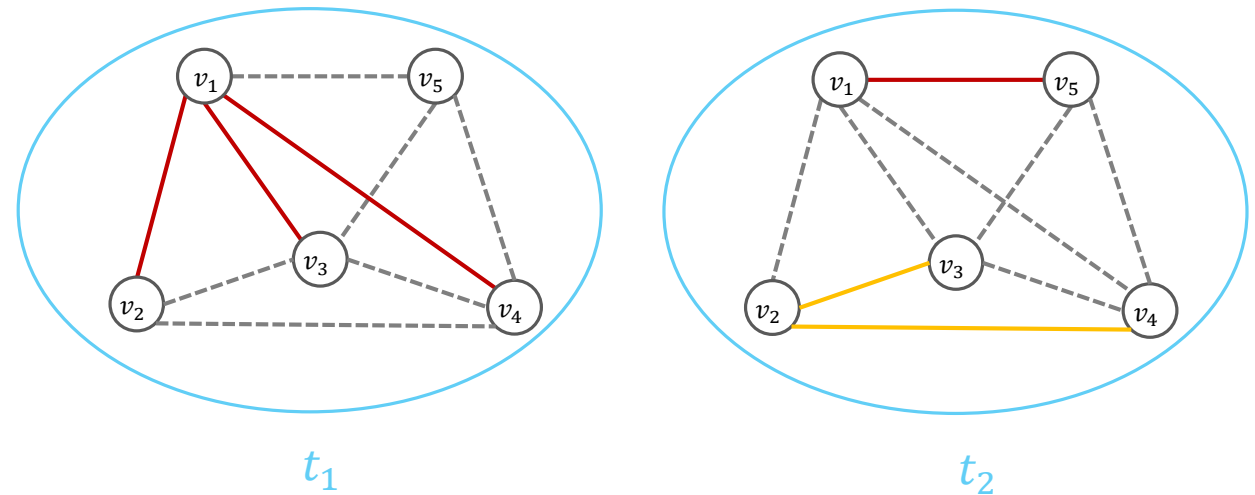
Idea: Define tasks by edges to create evenly-sized work for threads.

Coarse-Grained Tasks



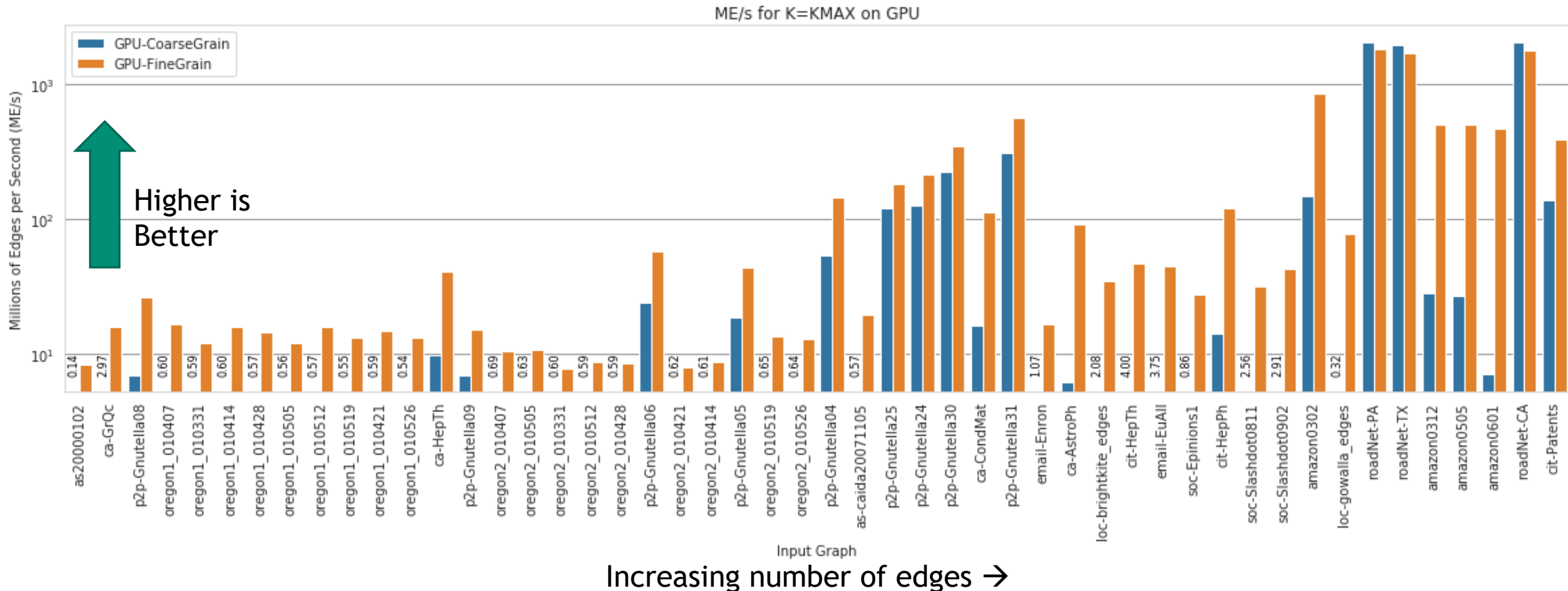
Tasks defined on edges  
grouped by source vertex

Fine-Grained Tasks



Tasks defined on  
groups of edges

# Results: Performance on Tesla V100 GPU with Kokkos



Fine-grained parallelism from edge-based tasks yields 16.9x - 9.9x speedup on GPU