Triangle-based analytics are important in many larger data-analytics based applications. Previously, a highly efficient linear algebra-based algorithm has been developed in Kokkos-Kernels. Primary contributions:

- We improve upon that work by developing an SpGEMM implementation that relies on a highly efficient, work-stealing, multithreaded runtime.
- We demonstrate that our implementation results in improving the runtime up to 5X to 12X on different architectures.

**Triangle Counting**

A triangle can be defined as a set of three mutually adjacent vertices in a graph.

**Triangle Counting Problem**

Given a graph $G = (V, E)$, the triangle counting problem is to find the number $(T)$ of all set of three vertices, $u, v, w \in V$, such that: $T = \{ u, v, w \mid (u, v), (v, w), (w, u) \in E \}$.

**Linear Algebra Formulations**

Two linear-algebra based formulations of triangle counting that are based on the adjacency matrix of the graph: $L$ and $U$ represent lower and upper parts.

- LU algorithm; $D = (L \cdot U) \rightarrow L$
- (Pro): Low operation count.
- (Con): Poor scalability.

- LL algorithm; $D = (L \cdot L) \rightarrow L$
- (Pro): Good scalability.
- (Con): More operations than LU.

**Parallelization strategy and the runtime are the main differences between KKTri-Cilk and KKTri.**

- Rows are ordered for avoiding computation:
  - LL; in decreasing degree,
  - LU; in increasing degree
- Balance number of non-zeroes within each partition.
- Each partition is spawned (in parallel) as a task.
- A task runs matrix matrix multiplication within a partition.

**Triangle Counting Scalability**

Letting $d_v$ be the degree of vertex $v$, the $4/3$-moment is defined as: $E[d_v^{4/3}] = 1/n \sum (d_v^{4/3})$.

**Experiments: Strong Scaling**

Following figures show strong scaling experiments for OpenMP and Cilk implementations of two algorithms.

- KKTri-Cilk scales the best in all three problems.
- uk-2005 achieves best rate: highly local computations.
- scale24 achieves worst rate: Poor cache usage.
- Friendster graph's distribution is in between (best scalability).

**Experiments: Relative Speedup**

Comparisons of KKTri-Cilk with TCM, a state-of-the-art graph library.

- KKTri achieves up to 7X speedup on graphs that have a good natural ordering such as wb-edu, uk-2005, and uk-2007. KKTri outperforms TCM in 23 of 27 cases.

**Experiments: Dataset and Peak Rate**

Times highlighted in green when KK-Cilk is the fastest.

- $^{10}$ bar is passed for the uk-2005 matrix and wb-edu graph.
- A high correlation (0.93) between the conductance and the rate.

**Conclusion**

- KKTri-Cilk surpasses $^{10}$ for the rate measure.
- KKTri-Cilk is faster on 63 of 78 instances.
- KKTri-Cilk is faster than state-of-the-art graph based implementation (up to 7X).
- We corroborate that the scalability of the triangle counting is bounded by $O(n)$ when the $4/3$-moment is bounded.
- We show correlation between the high rates achieved and the conductance of the graph.

**References**


