An Out-of-Core GPU Accelerated Multi-Predicate Join Algorithm for Graph Processing

Haicheng Wu and Sudhakar Yalamanchili
School of Electrical and Computer Engineering, Georgia Institute of Technology

Motivation

- Treat graphs as relations
  - Empower domain users to use general declarative languages.
  - Joint work with LogicBlox Inc.

A LogiQL Rule

Multi-predicate Join

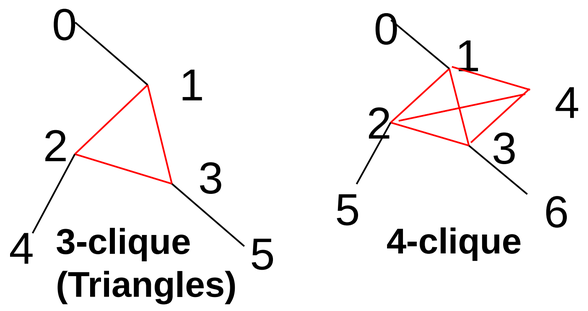
triangle(x,y,z)<-E(x,y),E(y,z),E(x,z), x<y<z.

4cl(x,y,z,w)<-E(x,y),E(x,z),E(x,w),E(y,z),E(y,w),E(z,w), x<y<z<w.

- Formulations as a multi-predicate join can
  - Reduce data movement between binary joins.
  - Reduce data reorganization (sorting or hash table construction).

Example: Clique Listing

- Key ingredients for many graph algorithms such as
  - Triangule clustering
  - Cohesive subgraph
- Extensive attention from
  - Graph theory
  - Database
  - Network analysis



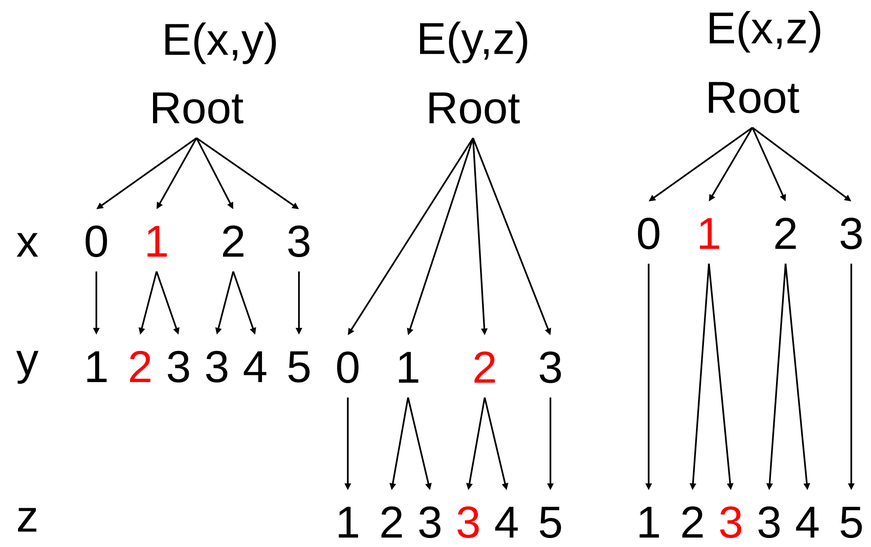
Core Algorithm: LeapFrog TrieJoin (LFTJ) [1]

- A general multi-predicate join algorithm.
- Worst case optimal.

LFTJ on 3 Tries to find triangles

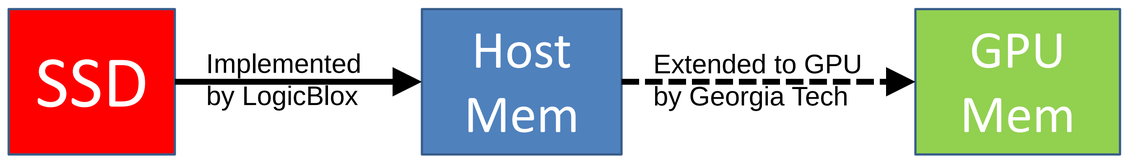
1. Find one intersection in the current layer

2. Down to the next layer

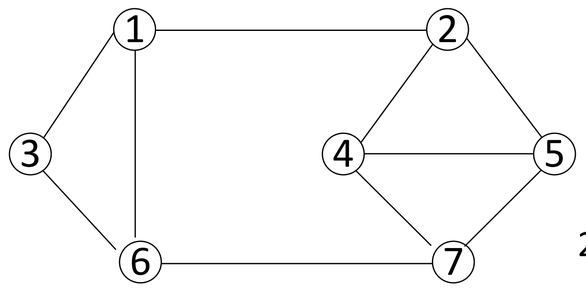


Operates similar to Depth First Search (DFS)

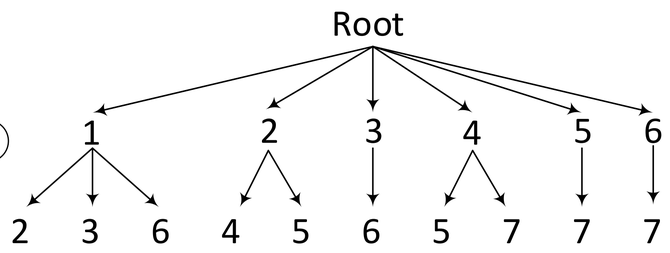
Out-of-Core Management: Boxing LFTJ [2]



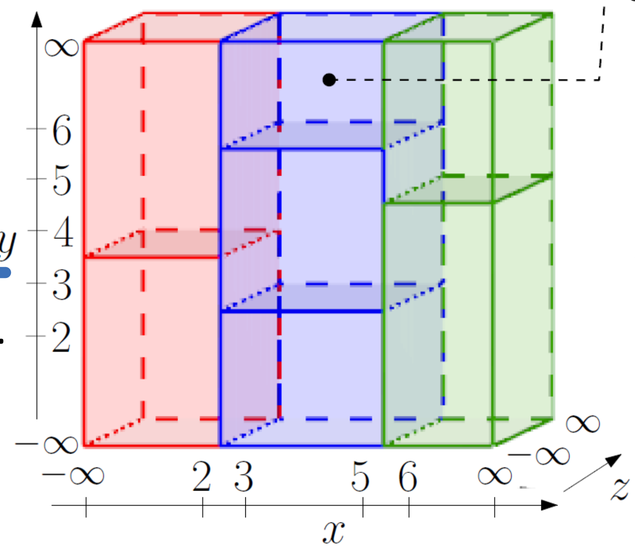
- The partition algorithm from the Logicblox Runtime.
- When the graph is out-of-core, the data space is partitioned into boxes that fit in memory and executed with in-memory LFTJ.
- The boxing algorithm is worst case optimal.
- The boxed LFTJ has the same complexity as in-memory LFTJ.
- The boxed LFTJ matches the state-of-art specialized algorithm when finding triangles.



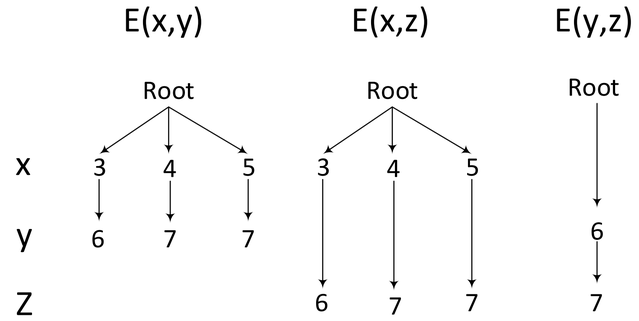
(a) Original Graph



(b) Corresponding Trie



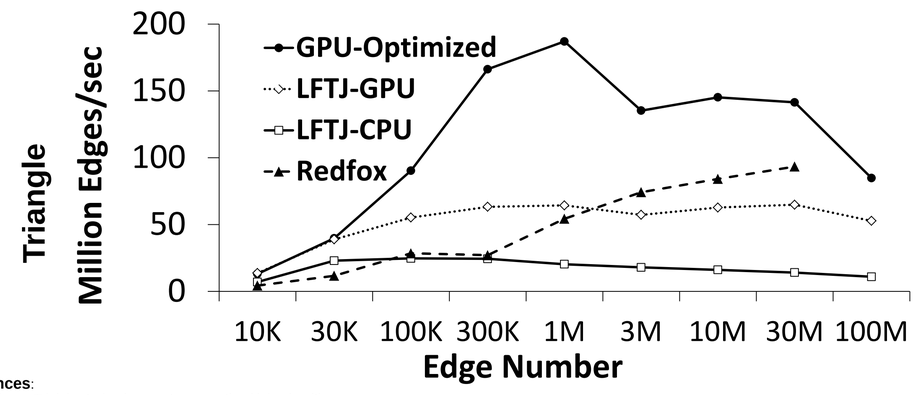
(c) Box Example



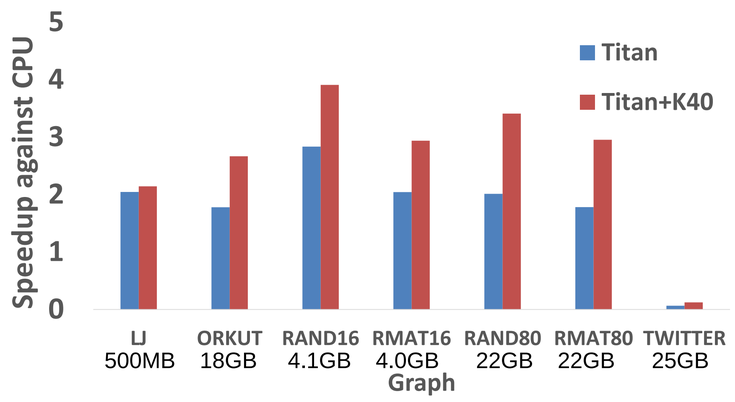
(d) Boxed Trie Slices

Performance: GPU-Optimized LFTJ [3]

- Change depth first to breadth first to exploit more parallelism.
- Optimized for load balance and memory access patterns.
- Evaluated over randomly generated graphs.
- Much faster than original LFTJ and binary join (Red Fox).
- Throughput is smaller than PCIe bandwidth.



Execute Large Graphs in GPUs ([1]+[2]+[3])



- Evaluate large real or synthesized graphs.
- Baseline is CPU boxed LFTJ.
- GPU is usually more efficient.
- SSD and PCIe are not the bottleneck.

References:
1.Veldhuizen. Triejoin: A simple, worst-case optimal join algorithm.
2.Zinn. General-purpose join algorithms for listing triangles in large graphs.
3.Wu. Multipredicate join algorithms for accelerating relational graph processing on GPUs.

Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University · Georgia Tech · intel · PRINCETON UNIVERSITY · UC Berkeley · UNIVERSITY of WASHINGTON