KALEIDOSCOPE: Graph Analytics on Evolving Graphs

Steffen Maass, Taesoo Kim

Georgia Institute of Technology

April 23, 2018
About me

- 4th year PhD Student at Georgia Tech
- Advisor: Taesoo Kim
- Research Area: Systems
  - Operating Systems, Heterogeneous Systems, and Graph Processing
- Thesis work: Processing of Evolving Graphs
KALEIDOSCOPE - Overview

- Problem: Low locality and high memory overhead for processing evolving graphs
- These problems hinder adoption of systems for evolving graphs, fallback: batch processing (large latency)
- We use a *tiled* representation of the evolving graph that mitigates memory overheads while allowing for higher processing performance
- KALEIDOSCOPE can help with the execution of more complicated algorithms on larger graphs in less time.
Specific problem: Insertion performance

- Inserting edges into Stinger in batches
- Observe performance of insertions over edges present in graph

⇒ Insertion performance collapses with > 1M edges
Specific problem: Memory overhead

- Inserting edges into Stinger in batches
- Observe impact on execution time of evolving PageRank

⇒ Execution time grows super-linearly due to insertion overhead
Goals for Kaleidoscope

Core design idea: Use *tiled* graph representation

Allows for:

- Lower memory overhead by using localized identifiers
- Asynchronous graph compaction
- Improvements in locality with space-filling curves

Also enables a straight-forward multi-core strategy by load-balancing tiles across processors
Conclusion

**Kaleidoscope** aims at addressing **three** problems when processing evolving graphs:
- Large memory footprint
- Synchronous compaction
- Low locality

**Kaleidoscope** proposes a tiled data structure to address these problems:
- Provides localized graphs
- Enables asynchronous compaction
- Exploits locality-optimizing space-filling curves
**Kaleidoscope** aims at addressing **three** problems when processing evolving graphs:
- Large memory footprint
- Synchronous compaction
- Low locality

**Kaleidoscope** proposes a tiled data structure to address these problems:
- Provides localized graphs
- Enables asynchronous compaction
- Exploits locality-optimizing space-filling curves

Thanks!