Monarch: Gaining Command on Geo-Distributed Graph Analytics

Anand Iyer∗, Aurojit Panda∗, Mosharaf Chowdhury▲, Aditya Akella∗, Scott Shenker∗, Ion Stoica∗

∗ UC Berkeley  ▲ NYU  ▲ University of Wisconsin  ▲ University of Michigan

HotCloud, July 09, 2018
Graph Analytics Popular

- Apache Giraph
- JanusGraph
- GraphX
- neo4j
- GraphLab
- TITAN
Graph Analytics Popular

Assume graph is aggregated to a single DC
Social Networks
Cellular Network Analytics
Financial Network Analytics
Generate data in a *geo-distributed* fashion
Generate data in a **geo-distributed** fashion

Can benefit from **timely** analysis
How do we perform efficient geodistributed graph analytics?
Geo-Distributed Analytics (GDA)

Slide courtesy: Clarinet authors
Geo-Distributed Analytics (GDA)

**Diagram:**
- **SELECT * ... FROM .. WHERE .. ;** Geo-distributed Analytics framework
- Query Optimizer
- Multi-stage parallelizable jobs
- Distributed Execution Layer
- Distributed Storage Layer
- One Logical Datacenter

**Requirements:**
- Clarinet [OSDI ’16]
- Requires WAN-aware optimization

**Notes:**
- Iridium [SIGCOMM 15]
- GeoDe [NSDI 15]

*Slide courtesy: Clarinet authors*
Geo-Distributed Analytics on Graphs

Can we use the same idea on graphs?

- GDA focuses on simple task placement/queries
  - Graph analytics iterative in nature
- Flexibility over data placement and join sites
  - Graph partitioning difficult
- Estimating intermediate data
  - Difficult in graph algorithms
Geo-Distributed Analytics on Graphs

Can we use the same idea on graphs?

- GDA focuses on simple task placement/queries
  - Graph analytics iterative in nature
- Flexibility over data placement and join sites
  - Graph partitioning difficult
- Estimating intermediate data

Key: Optimizing iterative graph-parallel processing
Graph Parallel Processing
Graph Parallel Processing

**Gather:** Accumulate information from neighborhood
Graph Parallel Processing

**Gather:** Accumulate information from neighborhood

**Apply:** Apply the accumulated value
Graph Parallel Processing

**Gather:** Accumulate information from neighborhood

**Apply:** Apply the accumulated value

**Scatter:** Update adjacent edges & vertices with new value
Our Proposal: Monarch
Our Proposal: Monarch

Sparsification
Our Proposal: Monarch

Sparsification

Execution Model
Our Proposal: Monarch

DC 1

DC 2

DC 3

DC 4

Sparsification

Execution Model

WAN Awareness
Graph Sparsification

- **Sparsification extensively studied in graph theory**
  - **Idea:** approximate the graph using a sparse, much smaller graph
  - Drop edges/vertices

- **Sparsify without accuracy loss**
  - Only worry about reducing cross-DC entities
  - Leverage graph-parallel model and algorithm properties
Graph Sparsification

- **Sparsification extensively studied in graph theory**
  - **Idea:** approximate the graph using a sparse, much smaller graph
  - Drop edges/vertices

- **Sparsify without accuracy loss**
  - Only worry about reducing cross-DC entities
  - Leverage graph-parallel model and algorithm properties
Geo-Distributed Graph Computation Model
Geo-Distributed Graph Computation Model

Bootstrap
Geo-Distributed Graph Computation Model

Bootstrap
Geo-Distributed Graph Computation Model

Bootstrap

Global Sync
Geo-Distributed Graph Computation Model

Bootstrap

Global Sync
Geo-Distributed Graph Computation Model

DC 1

DC 2

DC 3

DC 4

Bootstrap

Global Sync

iGAS
Geo-Distributed Graph Computation Model

Bootstrap

Global Sync

iGAS
Incremental GAS Model
Incremental GAS Model
Incremental GAS Model
Incremental GAS Model
Incremental GAS Model
Incremental GAS Model
Incremental GAS Model
Incremental GAS Model
Which graph algorithms can use the iGAS model? How much state needs to be kept at the entities for accuracy?
Geo-Distributed Graph Computation Model

Bootstrap

Global Sync

iGAS
Geo-Distributed Graph Computation Model

DC 1

DC 2

DC 3

DC 4

Bootstrap

Global Sync

iGAS
Geo-Distributed Graph Computation Model

DC 1 → DC 2
DC 3 → DC 4

Bootstrap

Global Sync

iGAS
Geo-Distributed Graph Computation Model

DC 1

DC 2

DC 3

DC 4

Bootstrap

Global Sync

Apply GDA techniques on task placement and data movement
Evaluation of Potential

- 16 node Apache Spark cluster across 4 regions
- Modified GraphX to incorporate the proposed model
Other Open Questions

- Convergence properties due to our modified execution model
- Better execution models at bootstrap stage
  - How would the global sync work?
- Multi-tenancy
  - Would it provide opportunities to leverage existing GDA techniques?
- Graph updates
  - What is an incremental model in this case?
Conclusion

- Several emerging applications produce graph data in a geo-distributed fashion
  - Can benefit from geo-distributed graph analytics.
- Our proposal Monarch:
  - Early attempt at bringing geo-distributed analytics to graph processing.
  - Initial results are encouraging.

http://www.cs.berkeley.edu/~api
api@cs.berkeley.edu