Experimental Analysis of Streaming Algorithms for Graph Partitioning

2019-07-04

Presented by: Anil Pacaci

Anil Pacaci, M. Tamer Özsu
Graph Partitioning
Graph Partitioning

Load Balance & Network I/O

Scale-out GPS

Results

Query & Analysis
Graph Partitioning

METIS:
~8 Hours
Twitter Graph

Load Balance & Network I/O

Scale-out GPS

Results

Query & Analysis
Streaming Algorithms

Experimental Analysis of Streaming Algorithms for Graph Partitioning

A. Pacaci & M. T. Özsu

2019-07-04 PAGE 5
### Streaming Partitioning Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cut</th>
<th>Stream</th>
<th>Graph</th>
<th>Cost</th>
<th>Prior Knowledge</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>FENNEL [Tsourakakis et al., 2014]</td>
<td>Edge</td>
<td>Vertex</td>
<td>-</td>
<td>EC</td>
<td>Adj. List</td>
<td>Greedy</td>
</tr>
<tr>
<td>RandomWalk [Tsourakakis et al., 2015]</td>
<td>Edge</td>
<td>Vertex</td>
<td>PP</td>
<td>EC</td>
<td>Adj. List</td>
<td>Greedy</td>
</tr>
<tr>
<td>Online&amp;On-demand [Filippidou &amp; Kotidis, 2015]</td>
<td>Edge</td>
<td>Edge</td>
<td>-</td>
<td>EC</td>
<td>Adj. List</td>
<td>CST</td>
</tr>
<tr>
<td>Ginger [Chen et al., 2015]</td>
<td>Hybrid</td>
<td>Edge</td>
<td>Power-law</td>
<td>RF</td>
<td>-</td>
<td>Greedy</td>
</tr>
<tr>
<td>PowerGraph [Gonzalez et al., 2014]</td>
<td>Vertex</td>
<td>Edge</td>
<td>Power-law</td>
<td>RF</td>
<td>Degree</td>
<td>Greedy</td>
</tr>
<tr>
<td>HDRF [Petroni et al., 2015]</td>
<td>Vertex</td>
<td>Edge</td>
<td>Power-law</td>
<td>RF</td>
<td>-</td>
<td>Greedy</td>
</tr>
<tr>
<td>Grid [Jain et al., 2013]</td>
<td>Vertex</td>
<td>Edge</td>
<td>Power-law</td>
<td>RF</td>
<td>-</td>
<td>Const.</td>
</tr>
<tr>
<td>DBH [Xie et al., 2014]</td>
<td>Vertex</td>
<td>Edge</td>
<td>Power-law</td>
<td>RF</td>
<td>Degree</td>
<td>Hash</td>
</tr>
<tr>
<td>Least IC [Bourse et al., 2014]</td>
<td>Vertex</td>
<td>Edge</td>
<td>Power-law</td>
<td>RF</td>
<td>-</td>
<td>Greedy</td>
</tr>
</tbody>
</table>

.....
## Streaming Partitioning Algorithms

<table>
<thead>
<tr>
<th>Method</th>
<th>Cut</th>
<th>Stream</th>
<th>Graph</th>
<th>Cost</th>
<th>Prior Knowledge</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>FENNEL</td>
<td>Edge</td>
<td>Vertex</td>
<td>-</td>
<td>EC</td>
<td>Adj. List</td>
<td>Greedy</td>
</tr>
<tr>
<td>LDG</td>
<td>Edge</td>
<td>Vertex</td>
<td>-</td>
<td>EC</td>
<td>Adj. List</td>
<td>Greedy</td>
</tr>
<tr>
<td>RandomWalk</td>
<td>Edge</td>
<td>Vertex</td>
<td>PP</td>
<td>EC</td>
<td>Adj. List</td>
<td>Greedy</td>
</tr>
<tr>
<td>RS-FENNEL</td>
<td>Edge</td>
<td>Vertex</td>
<td>-</td>
<td>EC</td>
<td>Adj. List</td>
<td>Greedy</td>
</tr>
<tr>
<td>RS-LDG</td>
<td>Edge</td>
<td>Vertex</td>
<td>-</td>
<td>EC</td>
<td>Adj. List</td>
<td>Greedy</td>
</tr>
<tr>
<td>Online&amp;On-demand</td>
<td>Edge</td>
<td>Edge</td>
<td>Power-law</td>
<td>RF</td>
<td>-</td>
<td>CST</td>
</tr>
<tr>
<td>Ginger</td>
<td>Vertex</td>
<td>Edge</td>
<td>Power-law</td>
<td>CV</td>
<td>D.-order</td>
<td>Elim. Tree</td>
</tr>
<tr>
<td>Sheep</td>
<td>Vertex</td>
<td>Edge</td>
<td>Power-law</td>
<td>RF</td>
<td>Degree</td>
<td>Greedy</td>
</tr>
<tr>
<td>PowerGraph</td>
<td>Vertex</td>
<td>Edge</td>
<td>Power-law</td>
<td>RF</td>
<td>Degree</td>
<td>Greedy</td>
</tr>
<tr>
<td>Grid</td>
<td>Vertex</td>
<td>Edge</td>
<td>Power-law</td>
<td>RF</td>
<td>Degree</td>
<td>Hash</td>
</tr>
<tr>
<td>DBH</td>
<td>Vertex</td>
<td>Edge</td>
<td>Power-law</td>
<td>RF</td>
<td>Degree</td>
<td>Hash</td>
</tr>
<tr>
<td>Least IC</td>
<td>Vertex</td>
<td>Edge</td>
<td>Power-law</td>
<td>RF</td>
<td>Degree</td>
<td>Hash</td>
</tr>
</tbody>
</table>

...and many more...
Streaming Algorithms

**Edge-cut**

**Vertex-cut**

Experimental Analysis of Streaming Algorithms for Graph Partitioning

A. Pacaci & M. T. Özsu

2019-07-04
Is SGP viable alternative to hash-partitioning?

- Most systems use hash
  - Simplicity vs performance
- Graph partitioning is expensive
  - High overhead of offline methods
Motivation

Is SGP viable alternative to hash-partitioning?

- Most systems use **hash**
  - Simplicity vs performance
- Graph partitioning is **expensive**
  - High overhead of offline methods

Is vertex-cut model superior to edge-cut model?

- Literature mostly focus on partitioning quality
  - Replication factor and partition size
- **Vertex-cuts** has lower replication factor
  - Better performance on graph processing?

(Bourse et. al., SIGKDD 2014), (Rahimian et. al., DAIS 2014)
Motivation

Goals

Impact of Partitioning Strategy
- Application Performance
- Resource Usage

Contributions

Comparison of Partitioning Models
- Edge-cut & Vertex-cut

Offline & Online Graph Workloads

Goals

Is cut-size a good proxy for network communication?

Does balanced partitioning lead to balanced computation?

What is the impact of partitioning quality on the workload performance?

How does the partitioning impact throughput vs latency trade-off?

What is the impact of the workload skew on online graph workloads?
METHODOLOGY
Streaming Algorithms

Edge-cut

- LDG
- Fennel

Vertex-cut

Baseline

- Hash
- METIS
Streaming Algorithms

Edge-cut
- LDG
- Fennel

Baseline
- Hash
- METIS

Vertex-cut
- HDRF
- Grid
- DBH
- Hash

Experimental Analysis of Streaming Algorithms for Graph Partitioning

A. Pacaci & M. T. Özsu

2019-07-04
Streaming Algorithms

Edge-cut
- LDG
- Fennel

Hybrid
- Hybrid-Ginger
- Hybrid-Hash

Vertex-cut
- HDRF
- Grid
- DBH

Baseline
- Hash
- METIS

Experimental Analysis of Streaming Algorithms for Graph Partitioning
A. Pacaci & M. T. Özsu
2019-07-04
## Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Type</th>
<th>Edges</th>
<th>Vertices</th>
<th>Degree Distribution</th>
<th>Avg. Degree</th>
<th>Max Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Twitter Graph</td>
<td>Social Network</td>
<td>1.46B</td>
<td>41M</td>
<td>Skewed</td>
<td>35</td>
<td>2.9M</td>
</tr>
<tr>
<td>UK2007 Web Graph</td>
<td>Hyperlink Graph</td>
<td>3.73B</td>
<td>105M</td>
<td>Power-law</td>
<td>35</td>
<td>975K</td>
</tr>
<tr>
<td>US Road Network</td>
<td>Road Network</td>
<td>58M</td>
<td>23M</td>
<td>Uniform</td>
<td>2.5</td>
<td>9</td>
</tr>
<tr>
<td>LDBC SNB</td>
<td>Synthetic Social Network</td>
<td>447M</td>
<td>3.6M</td>
<td>Power-law</td>
<td>124</td>
<td>3680</td>
</tr>
</tbody>
</table>

Workloads & Systems

Offline Graph Analytics

Iterative & Global Graph Access

Gather-Apply-Scatter Engine

Queries

Single-Source SP
Weakly CC
PageRank

Gather
Apply
Scatter

Systems

PowerLyra
Workloads & Systems

Offline Graph Analytics

Iterative & Global Graph Access

- Single-Source SP
- Weakly CC
- PageRank

Online Graph Queries

Traversals & Local Graph Access

- 1-hop Traversal
- 2-hop Traversal
- Single-pair Shortest Path

Queries

Systems

- PowerLyra
- JanusGraph
FINDINGS
Major Findings

Is SGP viable alternative to hash-partitioning?

Is vertex-cut model superior to edge-cut model?
**Major Findings**

Is SGP viable alternative to hash-partitioning?  **Yes**

Is vertex-cut model superior to edge-cut model?  **No**
Major Findings

Is cut-size a good proxy for network communication?

Does balanced partitioning lead to balanced computation?

What is the impact of partitioning quality on the workload performance?

How does the partitioning impact throughput vs latency trade-off?

What is the impact of the workload skew on online graph workloads?
Replication factor vs Network Overhead

- **Replication Factor** vs **Network Communication (GB)**
- *Edge-Cut Model*
- *Grouped by Cut Model*
- *Lower is Better*

Cut size converted to Replication Factor

PageRank on Twitter Graph
Replication Factor vs Network Overhead

Cut size converted to Replication Factor

**Edge-cut model:** Less I/O for same replication factor

Vertex-cut  |  Hybrid-cut  |  Edge-cut

Network Communication (GB)

Replication Factor

Edge-Cut Model
Hidden Danger: Load Imbalance

PageRank on UK2007 Web Graph
64 Machines

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Replication Factor</th>
<th>Execution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDRF (Vertex-cut)</td>
<td>3.45</td>
<td>31s</td>
</tr>
<tr>
<td>Fennel (Edge-cut)</td>
<td>2.12</td>
<td>58s</td>
</tr>
</tbody>
</table>

Load Imbalance: Smaller is better

All partitions have approximately the same number of edges
Hidden Danger: Load Imbalance

PageRank on UK2007 Web Graph
64 Machines

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Replication Factor</th>
<th>Execution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDRF</td>
<td>3.45</td>
<td>31s</td>
</tr>
<tr>
<td>Fennel</td>
<td>2.12</td>
<td>58s</td>
</tr>
</tbody>
</table>

Load Balance: Smaller is better

Replication factor is not an indicator of performance alone

All partitions have approximately the same number of edges
# Online Queries: Power of Hash Partitioning

1-hop Queries on LDBC SNB Graph

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Tput</th>
<th>Mean Latency</th>
<th>99th Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hash</td>
<td>6168</td>
<td>46</td>
<td>95</td>
</tr>
<tr>
<td>LDG</td>
<td>6240</td>
<td>47</td>
<td>155</td>
</tr>
<tr>
<td>FENNEL</td>
<td>6463</td>
<td>56</td>
<td>323</td>
</tr>
<tr>
<td>METIS (Offline)</td>
<td>7414</td>
<td>42</td>
<td>96</td>
</tr>
</tbody>
</table>

Query Latency (ms): Lower is better

Load Imbalance: Smaller is better

Experimental Analysis of Streaming Algorithms for Graph Partitioning

A. Pacaci & M. T. Özsu

2019-07-04
Online Queries: Power of Hash Partitioning

1-hop Queries on LDBC SNB Graph

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Tput</th>
<th>Mean</th>
<th>99th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>6168</td>
<td>46</td>
<td>95</td>
</tr>
<tr>
<td>LDG</td>
<td>6240</td>
<td>47</td>
<td>155</td>
</tr>
<tr>
<td>FENNEL</td>
<td>6463</td>
<td>56</td>
<td>323</td>
</tr>
<tr>
<td>METIS (Offline)</td>
<td>7414</td>
<td>42</td>
<td>96</td>
</tr>
</tbody>
</table>

Hash partitioning: lower latency

Query Latency (ms): Lower is better

Load Imbalance: Smaller the better
Takeaways

More results & analysis are in the paper!

Framework Available:
https://github.com/anilpacaci/streaming-graph-partitioning