CIS 602-01: Scalable Data Analysis

Spatiotemporal Data & Topology

Dr. David Koop
Graphs as Data

### Nodes

<table>
<thead>
<tr>
<th>ID</th>
<th>Atom</th>
<th>Electrons</th>
<th>Protons</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>N</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>C</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>S</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>N</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

### Edges

<table>
<thead>
<tr>
<th>ID1</th>
<th>ID2</th>
<th>Bonds</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>
What is a Graph?

• In computing, a graph is an abstract data structure that represents set objects and their relationships as vertices and edges, and supports a number of graph-related operations

• Objects (nodes): \{ A, B, C, D \}

• Relationships (edges):
\{ (D, B), (D, A), (B, C), (B, A), (C, A) \}

• Operation: shortest path from D to A

[K. Salama, 2016]
Graphs with Properties

• Each vertex or edge may have properties associated with it
• May include identifiers or classes

Person
  name = 'Tom Hanks'
  born = 1956

Person
  name = 'Robert Zemeckis'
  born = 1951

ACTED_IN
  roles = ['Forrest']

DIRECTED

Movie
  title = 'Forrest Gump'
  released = 1994
Types of Graph Analytics

- Connectivity Analytics:
  - number of vertices/edges, in- and out-degrees of vertices
  - histogram of degrees can be useful in comparing graphs
- Path Analytics: cycles, reachability, shortest path, minimum spanning tree
- Community Analytics: clusters (cohesion and separation)
- Centrality Analytics: degree, vulnerability, PageRank
- Pattern Matching: subgraph isomorphism
  - can use properties
  - useful in fraud/threat detection, social network suggestions

[K. Salama, 2016]
Scalable Graph Analytics

- Nodes have graph subsets
- Vertices can communicate to vertices on other nodes if there is an edge between them
- Process vertices in parallel
- Pregel (Google)
  - Uses Bulk Synchronous Parallelism
  - Superstep does update and compute over all vertices and then synchronizes
- Apache Giraph
- GraphX (Spark)
Graph Databases and Graph Frameworks

• Graph Databases represent data as graphs
  - E.g. Neo4j, Titan, Apache Giraph

• Graph Frameworks allow graph operations to be computed efficiently but do not require data to be stored as graphs
  - E.g. Spark GraphX
Evaluating Different Graph Algorithms

- Select a variety of different algorithms:
  - PageRank
  - Breadth First Search
  - Collaborative Filtering
  - Triangle Counting

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Graph type</th>
<th>Vertex property</th>
<th>Edge access pattern</th>
<th>Message size (Bytes/edge)</th>
<th>Vertex active?</th>
</tr>
</thead>
<tbody>
<tr>
<td>PageRank</td>
<td>Directed, unweighted edges</td>
<td>Double (pagerank)</td>
<td>Streaming</td>
<td>Constant (8)</td>
<td>All iterations</td>
</tr>
<tr>
<td>Breadth First Search</td>
<td>Undirected, unweighted edges</td>
<td>Int (distance)</td>
<td>Random</td>
<td>Constant (4)</td>
<td>Some iterations</td>
</tr>
<tr>
<td>Collaborative Filtering</td>
<td>Bipartite graph; Undirected, weighted edges</td>
<td>Array of Doubles (pu or qv)</td>
<td>Streaming</td>
<td>Constant (8K)</td>
<td>All iterations</td>
</tr>
<tr>
<td>Triangle Counting</td>
<td>Directed, unweighted edges</td>
<td>Long (Ntriangles)</td>
<td>Streaming</td>
<td>Variable (0-10^6)</td>
<td>Non-iterative</td>
</tr>
</tbody>
</table>
Graph Frameworks

<table>
<thead>
<tr>
<th>Framework</th>
<th>Programming model</th>
<th>Multi node</th>
<th>Language usage</th>
<th>Graph Partitioning</th>
<th>Communication layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native</td>
<td>N/A</td>
<td>Yes</td>
<td>C/C++</td>
<td>N/A</td>
<td>MPI</td>
</tr>
<tr>
<td>GraphLab</td>
<td>Vertex</td>
<td>Yes</td>
<td>C++</td>
<td>1-D</td>
<td>Sockets</td>
</tr>
<tr>
<td>CombBLAS</td>
<td>Sparse matrix</td>
<td>Yes</td>
<td>C++</td>
<td>2-D</td>
<td>MPI</td>
</tr>
<tr>
<td>SociaLite</td>
<td>Datalog</td>
<td>Yes</td>
<td>Java</td>
<td>1-D</td>
<td>Sockets</td>
</tr>
<tr>
<td>Galois</td>
<td>Task-based</td>
<td>No</td>
<td>C/C++</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Giraph</td>
<td>Vertex</td>
<td>Yes</td>
<td>Java</td>
<td>1-D</td>
<td>Netty</td>
</tr>
</tbody>
</table>

PageRank can be expressed in matrix form as follows:

\[
\text{Algorithm 1:}
\]

```
begin
    for t = 0 to infinity
        n[r] = \sum_{s \in \text{neighbors}(r)} \frac{1}{d[s]} n[t] \text{(for the graph in Figure 2),}
end
```

As mentioned earlier, the pagerank algorithm can be expressed in a variety of frameworks. In a similar fashion, we also describe the execution for this algorithm in Alg. 1. The exact semantics of how the messages are packed and received, how much local data can be accessed, and how we store the incoming edge table, are given in Section 6.

Given by equation (1) performs one multiply-add operation per edge.

In this implementation, all join operations in the rule body are tail-nested table [30], effectively implementing a CSR format used in the native implementation and CombBLAS. Another version of PageRank is implemented from the perspective of a vertex that discovers its incoming out-degree list, allows for the edges to be stored as a single, contiguous row of columns, and improves the memory bandwidth utilization through hardware prefetching. Since each vertex has to access the pagerank values to be sent to the other nodes. These messages are then used to calculate the remote updates. More details on the optimization in pagerank are given in Section 6.
Results (Run Time)

We see the following key inferences: (1) Native code, as expected, delivers best performance as it is optimized for the underlying architecture. (2) Galois performs better than other frameworks, in particular, SociaLite, Giraph and CombBLAS. While Galois is a pure MPI program designed for the Intel micro-architecture, with particular optimizations for vectorization and memory bandwidth. The efficiencies are generally within 2-2.5X off the ideal results. Given the diversity of bottlenecks within the 4 algorithms, and also between single and multiple node implementations, we expect that it would be difficult for any one framework to excel at all scales in terms of performance and productivity.
Results (Run Time)

(c) Collaborative Filtering

(d) Triangle counting

[Satish et al., 2014]
Graphs are Central to Analytics

Raw Wikipedia → XML

Discussion Table → Editor Graph

Text Table → Term-Doc Graph

Hyperlinks → Community Detection

PageRank → User Community

Topic Model (LDA) → Community Topic

Word Topics

Top 20 Pages

[J. Gonzalez, 2014]
Separate Systems to Support Each View

Table View

- Hadoop
- Spark

Graph View

- Prege GraphLab
- Apache Giraph

Table

Row
Row
Row
Row

Dependency Graph

[J. Gonzalez, 2014]
PageRank on the Live-Journal Graph

GraphLab is \textit{60x faster} than Hadoop
GraphLab is \textit{16x faster} than Spark

[J. Gonzalez, 2014]
Moving between tables and graphs

XML → HDFS → HDFS → HDFS → HDFS

[J. Gonzalez, 2014]
GraphX: Composable views of the same data

Each view has its own operators that exploit the semantics of the view to achieve efficient execution

[J. Gonzalez, 2014]
GraphX Example Pipeline Runtimes

Raw Wikipedia → Hyperlinks → PageRank → Top 20 Pages

Spark Preprocess → Compute → Spark Post.

- Spark: 1492 seconds
- Giraph + Spark: 605 seconds
- GraphX: 342 seconds
- GraphLab + Spark: 605 seconds

[J. Gonzalez, 2014]
Projects

- Reviews Complete
- Disappointed in some projects and their scalability challenges
- Focus on projects
Topological Data Analysis

- G. Carlsson
- Data has shape, blurry view
- Very High-Level Description [Video]
Persistent Homology

- Video
Data Polygamy: The Many-Many Relationships among Urban Spatio-Temporal Data Sets

F. Chirigati, H. Doraiswamy, T. Damoulas, and J. Freire
When searching for relationships...

• With enough data, you'll find something:

• “A new study shows that drinking a glass of wine is just as good as spending an hour at the gym” [Fox News, 02/15].

• “A new study shows how sugar might fuel the growth of cancer” [Today, 01/16].

• “A new study shows late night snacking could damage the part of your brain that creates and stores memories” [Fox News, 05/16].

• People who go to saunas are more likely to know that Mike Stonebraker is not a character in "The Simpsons"

[C. Binnig et al., 2017]
Testing Data Polygamy on Random Data

<table>
<thead>
<tr>
<th># Records</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td># Attributes</td>
<td>11</td>
</tr>
<tr>
<td># Datasets</td>
<td>2</td>
</tr>
<tr>
<td>Extreme data prob.</td>
<td>20%</td>
</tr>
</tbody>
</table>

(a) Random input

(b) False discoveries

- Data Polygamy paper talks about the need to check potential relationships
- Issue may be the number of relationships to check

[C. Binnig et al., 2017]