Is Apache Spark an effective graph processing system?
Ancy Sarah Tom
Advised by: George Karypis
University of Minnesota, Twin Cities

• Graph processing systems – distributed systems used to process and analyze large graphs.

• Applications of graph processing include page rank computation, graph clustering, label propagation, graph coloring, etc.

• Characteristics of such applications include unstructured communication between vertices and neighbours and similar computation patterns.

• Apache Spark: In-memory cluster computing framework. Uses lineage to ensure resilience – RDDs. GraphX - a module over Spark to enable graph processing over a distributed data flow system.

• Question: Is Spark with its graph-processing extension (GraphX) well suited for problems dealing with irregular communication and computation?

Sources of performance gaps


• Immutability in Scala.

• Fault tolerance – excessive lineage growth leads to checkpointing.

• Potential underlying differences: C vs Java: 4x speedup.

Approach
Evaluate the performance gaps between GraphX and specialized implementations of graph applications.

Findings

Datasets: live journal (5m vertices, 49m edges)
System Specs:
8 nodes, 16 cores, 64GB mem

<table>
<thead>
<tr>
<th>Kernel (runtime)</th>
<th>Apache Spark</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>PageRank</td>
<td>9.67</td>
<td>0.096 (MPI)</td>
</tr>
<tr>
<td>Luby’s</td>
<td>10.48</td>
<td>0.21 (OpenMP)</td>
</tr>
<tr>
<td>Stochastic Block Model</td>
<td>TBD</td>
<td>TBD</td>
</tr>
</tbody>
</table>

Future Steps

• Implement more graph kernels to evaluate performance gaps and bottlenecks.

• Can we move parts of data structures to mutable data structures?

• Explore existing methods which involve interfacing with MPI based libraries to increase performance.

• Evaluate Spark Shuffle operations.