Hypergraphs in Chaos

JULIUS LISCHEID
Hypergraphs $\mathcal{H}(V, E)$ are generalised graphs where hyperedges $e \in E$ contain an arbitrary number of vertices $v \in V$

- In short, $E \subseteq \mathcal{P}(V)$
- Applications in recommender systems, image retrieval, data profiling, bioinformatics etc.
Graphs and Hypergraphs

- Hypergraphs can be represented as bipartite graphs
- MESH [4], the currently fastest distributed framework, builds on GraphX that builds on Spark that builds on JVM

Diagram:

- Vertices $v_1, v_2, v_3, v_4, v_5, v_6$

Stack:
- MESH (Hypergraph API)
- GraphX (Graph API)
- Spark (RDD API)
- JVM
Distributed (Hyper)Graph Processing Genealogy

< slower
≤ slower or equal

< PowerGraph
(C++) [2]

< GIRAPH
(JVM) [1]


≤

GraphX
(Spark on JVM) [3]

≤ HyperX
(Spark on JVM) [5]

≤

(GraphX on Spark on JVM) [4]

MESH: Minnesota Engine for Scalable Evolving Hypergraph Analysis

< CHAOS
(C++) [6]
PowerGraph vs. GraphX

PowerGraph \leq \text{GraphX} \quad \text{(Spark on JVM)} [3]

“[…] for graph algorithms, GraphX is over an order of magnitude faster than the base dataflow system [i.e. Spark] and is comparable to or faster than specialized graph processing systems [i.e. PowerGraph].”

Gonzalez et al., GraphX: Graph Processing in a Distributed Dataflow Framework [3]

<table>
<thead>
<tr>
<th>Graph</th>
<th>PowerG. GraphX</th>
<th>PowerL. Gemini</th>
<th>Speedup (×times)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>enwiki-2013</td>
<td>9.05</td>
<td>30.4</td>
<td>7.27, 0.484</td>
</tr>
<tr>
<td>twitter-2010</td>
<td>40.3</td>
<td>216</td>
<td>26.9, 3.02</td>
</tr>
<tr>
<td>uk-2007-05</td>
<td>64.9</td>
<td>416</td>
<td>58.9, 1.48</td>
</tr>
<tr>
<td>weibo-2013</td>
<td>117</td>
<td>-</td>
<td>100, 8.86</td>
</tr>
<tr>
<td>cluweb-12</td>
<td>-</td>
<td>-</td>
<td>31.1, n/a</td>
</tr>
<tr>
<td>CC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>enwiki-2013</td>
<td>4.61</td>
<td>16.5</td>
<td>5.02, 0.237</td>
</tr>
<tr>
<td>twitter-2010</td>
<td>29.1</td>
<td>104</td>
<td>22.0, 1.22</td>
</tr>
<tr>
<td>uk-2007-05</td>
<td>72.1</td>
<td>-</td>
<td>60.6, 1.76</td>
</tr>
<tr>
<td>weibo-2013</td>
<td>56.5</td>
<td>-</td>
<td>58.6, 2.62</td>
</tr>
<tr>
<td>cluweb-12</td>
<td>-</td>
<td>-</td>
<td>25.7, n/a</td>
</tr>
<tr>
<td>SSSP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>enwiki-2013</td>
<td>16.5</td>
<td>151</td>
<td>17.1, 0.514</td>
</tr>
<tr>
<td>twitter-2010</td>
<td>12.5</td>
<td>108</td>
<td>10.8, 1.15</td>
</tr>
<tr>
<td>uk-2007-05</td>
<td>117</td>
<td>-</td>
<td>143, 3.45</td>
</tr>
<tr>
<td>weibo-2013</td>
<td>63.2</td>
<td>-</td>
<td>60.6, 4.24</td>
</tr>
<tr>
<td>cluweb-12</td>
<td>-</td>
<td>-</td>
<td>56.9, n/a</td>
</tr>
<tr>
<td>GEOMEAN</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: 8-node runtime (in seconds) and improvement of Gemini over the best of other systems. “-” indicates failed execution.
Project Study

**CHAOS** (C++) [6] vs.

MESH: Minnesota Engine for Scalable Evolving Hypergraph Analysis (GraphX on Spark on JVM) [4]

- Implement hypergraph PageRank algorithm in Chaos
- Benchmark it against MESH
Status Quo
Questions?

PowerGraph (C++) \leq \text{HyperX (Spark on JVM)} \leq \text{GraphX on Spark (GraphX on Spark on JVM)} \leq \text{CHAOS (C++)}

MESH: Minnesota Engine for Scalable Evolving Hypergraph Analysis
References


