Big Graph Analytics Engine

Yinglong Xia
6/23/2016
Introduction
Introduction

Recent Growth

### Revenue

<table>
<thead>
<tr>
<th>Segment</th>
<th>2015</th>
<th>2014</th>
<th>YoY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carrier Business</td>
<td>232,307</td>
<td>191,381</td>
<td>21.4%</td>
</tr>
<tr>
<td>Enterprise Business</td>
<td>27,609</td>
<td>19,201</td>
<td>43.8%</td>
</tr>
<tr>
<td>Consumer Business</td>
<td>129,128</td>
<td>74,688</td>
<td>72.9%</td>
</tr>
<tr>
<td>Others</td>
<td>5,965</td>
<td>2,927</td>
<td>103.8%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>395,009</strong></td>
<td><strong>288,197</strong></td>
<td><strong>37.1%</strong></td>
</tr>
</tbody>
</table>

### Net Profits

<table>
<thead>
<tr>
<th>Segment</th>
<th>2015</th>
<th>2014</th>
<th>YoY</th>
</tr>
</thead>
<tbody>
<tr>
<td>China</td>
<td>167,690</td>
<td>108,674</td>
<td>54.3%</td>
</tr>
<tr>
<td>EMEA</td>
<td>128,016</td>
<td>100,674</td>
<td>27.2%</td>
</tr>
<tr>
<td>Asia Pacific</td>
<td>50,527</td>
<td>42,409</td>
<td>19.1%</td>
</tr>
<tr>
<td>Americas</td>
<td>38,976</td>
<td>30,844</td>
<td>26.4%</td>
</tr>
<tr>
<td>Others</td>
<td>9,800</td>
<td>5,596</td>
<td>75.1%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>395,009</strong></td>
<td><strong>288,197</strong></td>
<td><strong>37.1%</strong></td>
</tr>
</tbody>
</table>

### Cash flow from operating activities

- **CNY 395,009 million**
- **CNY 36,910 million**
- **CNY 49,315 million**

Collaboration

Global Research Institute & Labs

- Technical organizations
- Standards
- Industrial Partners
- Universities
- Open Source
Graph Analytics for Smart Big Data
Graph in ONOS

Figure 2: Prototype 1 Architecture

Figure 3: Prototype 2 Architecture

HotSDN’2014
Topology Impact on Information Propagation

![Topology Diagram](image)

![Graph of Log (Infected Ratio) vs Time Ticks](image)

- K-EdgeAdd
- CompDeg
- CompEigs
- CompDelete
- Rand
- CompPage

**Initial Node**

- Node 2
- Node 6
- Node 7
- Node 8
- Node 10
- Node 11
- Node 12

**Graph**

- X-axis: Time Ticks
- Y-axis: Log (Infected Ratio)
Explore the Variety in Graph Analytics
Challenges

- Very large scale graphs for analysis
  - 10B~1000B in terms of the number of vertices
  - a few hundreds of properties, static and dynamic
  - distributed communication introduces additional overhead

- Irregularity in graph data access
  - Low data locality results in high disk/communication IO overhead
  - Data access patterns are diverse among graph analysis algorithms

- Near real-time requirement
  - Incorporate with incremental graph updates
  - Approximate query & analysis should be considered

- Efficiency and productivity to balance
# Graph Platform for Smart Big Data

<table>
<thead>
<tr>
<th>Infrastructure</th>
<th>Single Machine</th>
<th>Cluster</th>
<th>GPU Server</th>
<th>Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Management</td>
<td>Structure Management</td>
<td>Property Management</td>
<td>Metadata Management</td>
<td>Permission Control</td>
</tr>
<tr>
<td>Graph engines</td>
<td>Streaming Graph</td>
<td>Graphical Model</td>
<td>Hyper Graph</td>
<td></td>
</tr>
<tr>
<td>Analytics</td>
<td>Community</td>
<td>Bayes Net</td>
<td>Anomaly detection</td>
<td>Ego Feature</td>
</tr>
<tr>
<td></td>
<td>Centrality</td>
<td>Label propagation</td>
<td>Matching</td>
<td>Max Flow</td>
</tr>
<tr>
<td>Visualization</td>
<td>Dynamic Graph Vis</td>
<td>Property Vis</td>
<td>Large Graph Vis</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Incremental Update</td>
<td>Dynamic Graph Vis</td>
<td>Property Vis</td>
<td>Large Graph Vis</td>
</tr>
<tr>
<td></td>
<td>Basic Engine</td>
<td>Graphical Model</td>
<td>Hyper Graph</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Streaming Graph</td>
<td>Graphical Model</td>
<td>Hyper Graph</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Incremental Update</td>
<td>Dynamic Graph Vis</td>
<td>Property Vis</td>
<td>Large Graph Vis</td>
</tr>
<tr>
<td></td>
<td>Basic Engine</td>
<td>Graphical Model</td>
<td>Hyper Graph</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Streaming Graph</td>
<td>Graphical Model</td>
<td>Hyper Graph</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Incremental Update</td>
<td>Dynamic Graph Vis</td>
<td>Property Vis</td>
<td>Large Graph Vis</td>
</tr>
<tr>
<td></td>
<td>Basic Engine</td>
<td>Graphical Model</td>
<td>Hyper Graph</td>
<td></td>
</tr>
</tbody>
</table>
Unified Graph Data Access Patterns

Observation on PSW data access patterns inspires highly efficient sharding representation.
Construct Edge-set Flows

1 2 3 4 5 6
1 0.3 0.4 0.3 0.6
2 1.4
3 0.2 0.8 0.9 1.2
4 0.5 0.6 0.2 1.1
5 0.8 1.9
6

row permutation

1 2 3 4 5 6
1 0.2 0.8 0.9 1.2
2 0.5 0.6 0.2 1.1
3 0.3 0.4
4 0.3 0.6
5 1.4
6 0.8 1.9

column permutation

1 2 3 4 5 6
1 0.2 0.8 0.9 1.2
2 0.5 0.6 0.2 1.1
3 0.3 0.4
4 0.3 0.6
5 1.4
6 0.8 1.9

Physical edge-sets

row permutation

column permutation

Flow direction
Preliminary Experiments - Preproc.

Create the data in our format.
Preliminary Experiments - Comp.

Decent speedup achieved w/ or w/o loading time
Preliminary Experiments

In-memory PageRank Total Time v.s. GraphChi

PageRank Total Time

In-memory PageRank Total Speed v.s. GraphChi
Conclusion

- Many big data problems involve links among a lot of entities, naturally represented as a graph
- Property graph is highly expressive
- Industry is looking for graph/graphical model engines for complex network analysis, streaming graph, probabilistic graphical models, and RDF graph computing
- Efficiency is the key in many industry graph analysis systems, especially when the data volume is big
- Eventually, the graph engine should serve for AI Business systems