Graph and Link Analysis:

Discovering Network Relationships in Big Data

Xavier Lopez, Ph.D.
Senior Director, Product Management

Zhe Wu, Ph.D.
Architect, Development

September 19, 2016
Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Program Agenda with Highlight

1. Graph Data Management and Analysis: Usage & Use Cases
2. Oracle Big Data Spatial and Graph
3. In Memory Analyst (PGX)
4. What’s New
5. Demos
Relational Model vs. Property Graph Model

• Relational Model

• Graph Model

Courtesy: Tom Sawyer 2016
The Property Graph Data Model

- A set of vertices (or nodes)
  - each vertex has a unique identifier.
  - each vertex has a set of in/out edges.
  - each vertex has a collection of **key-value** properties.

- A set of edges (or links)
  - each edge has a unique identifier.
  - each edge has a head/tail vertex.
  - each edge has a label denoting type of relationship between two vertices.
  - each edge has a collection of **key-value** properties.

https://github.com/tinkerpop/blueprints/wiki/Property-Graph-Model
How graph analysis enhances business intelligence

• Answers from **Tabular Aggregation**
  – Who spends the most?
  – Who buys the highest margin goods?
  – Who is most consistently a top contributor?

• Answers from **Graph Connectivity**
  – Who’s most influential?
  – Which supplier do I depend on the most?
  – What is the right product mix for millennials?

---

**Tabular questions:**
Well-suited to SQL-like tools

**Graph questions:**
We need something different!
How is graph analysis important to business?

- What **patterns** are there in fraudulent behavior?
- Which supplier am I **most dependent upon**?
- Who is the most **influential customer**?
- Do my products appeal to certain **communities**?
- What targeted products or services do I **recommend** to customers?
Graph Use Case Scenarios

• Fraud detection
  – Find parties in insurance data who are on both sides of multiple claims, who live near each other

• Internet of Things
  – Manage graph of interconnected devices and predict the effect of an disruptions across network

• Cyber Security
  – Find entry points and affected machines

• Border Control
  – Analyze flight histories of a suspicious passenger. Indentify his co-travelers, co-traveler’s co-travelers, ...
Graph Analysis in Business

**Product Recommendation**
Recommend the most similar item purchased by similar people

**Influencer Identification**
Find out people that are central in the given network – e.g., influencer marketing

**Community Detection**
Identify group of people that are close to each other – e.g., target group marketing

**Graph Pattern Matching**
Find out all the sets of entities that match to the given pattern – e.g., fraud detection
Program Agenda with Highlight

1. Graph Data Management and Analysis
2. Oracle Big Data Spatial and Graph: Architecture & Features
3. In-memory Analyst (PGX)
4. What’s New
5. Demos
Oracle Big Data Spatial and Graph Property Graph Architecture

Graph Analytics
- Parallel In-Memory Graph Analytics (PGX)

Access Layer
- Apache Blueprints & Lucene/SolrCloud

Oracle Big Data Spatial and Graph
- Apache HBase
- Oracle NoSQL Database

Java APIs

REST/Web Service

Java, Groovy, Python, ...

Property graph formats supported
- GraphML
- GML
- Graph-SON
- Flat Files
- CSV
- Relational Data Sources
Property Graph Workflow

• Graph Data Management
  – Transform and load relational data (or files) to a graph schema

• Analysis and Exploration (in-memory analysis engine)
  – Data scientists try different ideas (algorithms) on the data
  – Flexible, interactive, iterative, small-scale (sampled), …. 

• Production
  – Operational queries and reporting
Graph Construction: Convert from Relational to Flat Files

- Two Key Java APIs:
  - OraclePropertyGraphUtils.convertRDBMSTable2OPV (E)
  - ColumnToAttrMapping

- Key Steps:
  - Column Mapping
  - Data Type Definition
  - Conversion

Example output .opv file

```
1101,name,1, Jean,, 1101,age,2,, 20, 1101,salary,4,, 120.0,
1102,name,1, Mary,, 1102,age,2,, 21, 1102,salary,4,, 50.0,
...
...
...
...
```

EmployeeTab

<table>
<thead>
<tr>
<th>EMPID</th>
<th>hasName</th>
<th>hasAge</th>
<th>hasSalary</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Jean</td>
<td>20</td>
<td>120.0</td>
</tr>
<tr>
<td>102</td>
<td>Mary</td>
<td>21</td>
<td>50.0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Data Access (APIs)

• Blueprints 2.3.0, Gremlin 2.3.0, Rexster 2.3.0
• Groovy shell for accessing property graph data
• REST APIs (through Rexster integration)
• PGQL (Property Graph Query Language)
Text Search through Apache Lucene/SolrCloud

- Integration with Apache Lucene & SolrCloud
- Support manual and auto indexing of Graph elements
  - Manual index:
    - oraclePropertyGraph.createIndex("my_index", Vertex.class);
    - indexVertices = oraclePropertyGraph.getIndex("my_index", Vertex.class);
    - indexVertices.put("key", "value", myVertex);
  - Auto Index
    - oraclePropertyGraph.createKeyIndex("name", Edge.class);
    - oraclePropertyGraph.getEdges("name", "*hello*world");
  - Enables queries to use syntax like "*oracle* or *graph*"
Support for Cytoscape Open Source Visualization
Program Agenda with Highlight

1. Graph Data Management and Analysis
2. Oracle Big Data Spatial and Graph: Architecture & Features
3. In-memory Analyst (PGX)
4. What’s New
5. Demos
Parallel In-Memory Graph Analyst

• An in-memory, parallel framework for fast graph analytics
  – Read a graph from NoSQL or HBase
  – Handles analytic workloads while the data access layer handles transactional workloads
  – Supports multiple users/graphs
  – Dozens of graph analysis functions
Social Network Analysis Algorithms (1)

- **Structure Evaluation**
  - Conductance
  - countTriangles
  - inDegreeDistribution
  - outDegreeDistribution
  - partitionConductance
  - partitionModularity
  - sparsify
  - K-Core computes

- **Community Detection**
  - communitiesLabelPropagation

- **Ranking**
  - closenessCentralityUnitLength
  - degreeCentrality
  - eigenvectorCentrality
  - Hyperlink-Induced Topic Search (HITS)
  - inDegreeCentrality
  - nodeBetweennessCentrality
  - outDegreeCentrality
  - pagerank
  - personalizedPagerank
  - randomWalkWithRestart
  - approximatePagerank
  - weightedPagerank
Social Network Analysis Algorithms (2)

Pathfinding

- fattestPath
- shortestPathBellmanFord
- shortestPathBellmanFordReverse
- shortestPathDijkstra
- shortestPathDijkstraBidirectional
- shortestPathFilteredDijkstra
- shortestPathFilteredDijkstraBidirectional
- shortestPathHopDist
- shortestPathHopDistReverse

Recommendation

- salsa
- personalizedSalsa
- whomToFollow

Classic - Connected Components

- sccKosaraju
- sccTarjan
- wcc
“No Coding” Graph Analysis

Degree Centrality

```
heroInfluence = analyst.inDegreeCentrality()
```

Page Rank

```
heroPR = analyst.pageRank().topK(15)
```

Betweenness Centrality

```
b = analyst.betweenness().topK(15)
```

Community Detection

```
comic_coms = analyst.communities()
```
Computational Analytics: Built-in Package

Rich set of built-in parallel graph algorithms

Detecting Components and Communities
- Tarjan’s, Kosaraju’s, Weakly Connected Components, Label Propagation (w/ variants), Soman and Narang’s Specification

Evaluating Community Structures
- Conductance, Modularity, Clustering Coefficient (Triangle Counting), Adamic-Adar

Ranking and Walking
- Pagerank, Personalized Pagerank, Betweenness Centrality (w/ variants), Closeness Centrality, Degree Centrality, Eigenvector Centrality, HITS, Random walking and sampling (w/ variants)

Path-Finding
- Hop-Distance (BFS), Dijkstra’s, Bi-directional Dijkstra’s, Bellman-Ford’s

Other Classics
- Vertex Cover
- Minimum Spanning-Tree (Prim’s)

... and parallel graph mutation operations

- Create Undirected Graph
- Simplify Graph
- Create Bipartite Graph
- Sort-By-Degree (Renumbering)
- Filter-Expression
- The original graph
- Filtered Subgraph
- Left Set: “a,b,e”
Program Agenda with Highlight

1. Graph Data Management and Analysis
2. Oracle Big Data Spatial and Graph: Architecture & Features
3. In-memory Analyst (PGX)
4. What’s New
5. Demos
What’s New: Property Graph Features
Big Data Spatial and Graph 2.0

Faster, more powerful and scalable

- Integration with Apache Spark
- PGQL: Declarative Graph Query Language
- Distributed In-memory Graph Analysis
- Hortonworks 2.4; Apache Solr 5.2x
- Conversion of CSV & Relational data to Graph
- Vertex Label Support
- Node.js Client Support
- Many new SNA Algorithms
- Data type support: long, char, byte, short, spatial
- and many more…
Oracle Differentiators -- Graph

• **Complete, Supported, Graph Solution:**
  – **Storage**: NoSQL, Hbase, RDBMS back-ends
  – **Data Access**: Blueprints, Java, Property Graph Query Language (PGQL)
  – **Rich Graph Analytics**: 40 pre-built, in-memory graph algorithms

• **Scalable:**
  – Analyze 20-30 billion edge graph in memory on single BDA node
  – Persist extremely large graphs on disk

• **Security**: Secure NoSQL, Kerberos CDH

• **10-50x Faster** than graph analysis competitors
Program Agenda with Highlight

1. Graph Data Management and Analysis
2. Oracle Big Data Spatial and Graph: Architecture & Features
3. In-memory Analyst (PGX)
4. What’s New
5. Demos
Resources on Big Data Spatial and Graph

• Oracle Big Data Spatial and Graph on Oracle.com:  

• OTN product page (white papers, software downloads, documentation, tutorials):  
  www.oracle.com/technetwork/database/database-technologies/bigdata-spatialandgraph

• Oracle Big Data Lite Virtual Machine - a free sandbox to get started:  

• Hands On Lab for Big Data Spatial:  tinyurl.com/BDSG-HOL

• Blog – examples, tips & tricks:  blogs.oracle.com/bigdataspatialgraph

• @OracleBigData, @SpatialHannes, @JeanIhm