Graph Databases and Analytics: How to Use Them

Melli Annamalai and Jean Ihm

Graph Technologies
Oracle

Copyright © 2019 Oracle and/or its affiliates.
Safe Harbor

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, timing, and pricing of any features or functionality described for Oracle’s products may change and remains at the sole discretion of Oracle Corporation.

Statements in this presentation relating to Oracle’s future plans, expectations, beliefs, intentions and prospects are “forward-looking statements” and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle’s Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading “Risk Factors.” These filings are available on the SEC’s website or on Oracle’s website at http://www.oracle.com/investor. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.
What is a Graph?

A collection of points (vertices) and lines between those points (edges)
What is a Graph?

A collection of points (vertices) and lines between those points (edges)
A More Complex Graph:
Find popular nodes in a social network

Identify influencers
Graph Analytics

- Community detection and influencer analysis
  - Churn risk analysis/targeted marketing, HR Turnover analysis
- Clustering
  - Product recommendation
- Anomaly detection
  - Identify fraud
- Path analysis and reachability
  - Manage Bill of Materials, Outage analysis in utilities networks, vulnerability analysis in IP networks
- Pattern matching
  - Tax fraud detection, data extraction
Gartner: top 10 data and analytics technology trends for 2019

Trend #5: Graph

According to Gartner, graph analytics is a set of analytic techniques that help enterprises explore the relationships between entities of interest such as transactions, processes and staff.

The application of graph processing and graph database management systems will grow at 100% annually through 2022.
Gartner MQ - Data Management Solutions for Analytics

Figure 1. Magic Quadrant for Data Management Solutions for Analytics

Copyright © 2019 Oracle and/or its affiliates
Graph Data Models

**Property Graph Model**
- Path Analytics
- Graph Analytics
- Detect patterns and anomalies

**RDF Graph Model**
- Data federation
- Knowledge representation
- Semantic Web

**Use Cases**
- Financial
- Retail, Marketing
- Social Media
- Smart Manufacturing

**Industry Domain**
- Life Sciences
- Health Care
- Publishing
- Finance

**Shipping for 3+ years**
Customer Success Stories
Graph Technology Applications

- Oracle’s graph technology has been applied to solve many real problems for internal/external customers.
Paysafe: Money Transfer Fraud Detection

Multiple paths going to the same destination

Limited number of source and destination vertices
Banco De Galicia: Which Nodes are Transferring money to other Banks?
Circular Payments and Tax Fraud

- Company in other EU member country
- Importer "fake" company
- Buffer Reseller
- Exporter company
- Company in other EU member country

Not paying VAT
Paying VAT
Getting VAT refund

Copyright © 2019 Oracle and/or its affiliates.
Find nodes that are in most shortest paths between other nodes

Identify nodes that can cause vulnerabilities in a grid
Manufacturing: What is the Impact of Changing this Part?

A car has 30,000 parts
Southern European Police Force

Analyzing suspicious travel patterns
Detecting potential threats early

Data sources
Integrated Operations Management System (SIGO, built by Accenture)
Advance Passenger Information System (APIS, data from Non-Schengen Airport and Ports)

Combining legal requests and co-travelers
eg. determining „hot flights“ with at least one known criminal on board

Looking at passenger relationships
eg. Betweenness centrality to determine recurring travelers
Property Graphs
Property Graph Product Overview

- Store, manage, query and analyze graphs

- Highly scalable in-memory analytics
  - 10s of billions of edges and vertices

- 50+ pre-built graph analysis algorithms
  - Detecting components and communities
    - Tarjan’s, Kosaraju’s, Weakly connected components, label propagation, etc.
  - Ranking and walking
    - Pagerank, personalized pagerank, betweenness centrality, etc.
  - Evaluating community structures
    - Conductance, modularity, triangle counting, Adamic-Adar, etc.
  - Path-finding
    - Path distance, Dijkstra’s, Bellman-Ford’s, etc.
Property Graph Product Overview

- **PGQL**: Powerful graph query language
  
  SQL-like language for specifying graph patterns
  Working on graph additions to the SQL standard
  
- **Java API** for analytics
- **Rich user interface**
  
  Notebook
  Shell UI
  Graph Visualization
  
- **Enterprise capabilities** – built on Oracle infrastructure
  
  Manageability, fine-grained security, high availability, integration, and more

PGQL example:

```sql
PATH any_edge as ()-[[]]-()
SELECT n, m MATCH(n) -/: any_edge/ -> m
WHERE n.name='Lucy' and m.name='Jean'
```
Product:
Oracle Database Spatial and Graph

- In-memory parallel graph analytics server (PGX)
  - Load graph into memory for analysis
  - Automate graph refresh

- In-database parallel graph traversal
  - Run PGQL queries (converted to SQL) in the database

Client libraries
- Java API to develop applications
- Command-line submission of graph queries
- Graph visualization tool
- APIs to update graph store
Using Graphs on a Big Data Platform

Product: Oracle Big Data Spatial and Graph

Runs on the Hadoop platform

- In-memory parallel graph analytics server (PGX)
  - Load graph into memory for analysis
  - Automate graph refresh

- Client libraries
  - Java API to develop applications
  - Command-line submission of graph queries
  - Graph visualization tool
  - APIs to update graph store
## From Tables to a Graph

### Organizations

<table>
<thead>
<tr>
<th>org_id</th>
<th>name</th>
<th>type</th>
<th>religion</th>
<th>genre</th>
<th>country</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBS</td>
<td></td>
<td>TV Network</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amazon</td>
<td></td>
<td>Company</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### People

<table>
<thead>
<tr>
<th>person_id</th>
<th>name</th>
<th>company</th>
<th>occupation</th>
<th>country</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Pope Francis</td>
<td>Pope</td>
<td></td>
<td>Vatican</td>
</tr>
<tr>
<td>1</td>
<td>Barack Obama</td>
<td>44th Pres.</td>
<td></td>
<td>USA</td>
</tr>
<tr>
<td>7</td>
<td>Keegan-Michael Key</td>
<td>Actor</td>
<td></td>
<td>USA</td>
</tr>
</tbody>
</table>

### Relationships

<table>
<thead>
<tr>
<th>relation_id</th>
<th>from_person</th>
<th>from_type</th>
<th>to_person</th>
<th>to_type</th>
<th>relation_type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>person</td>
<td></td>
<td>person</td>
<td></td>
<td>admires</td>
</tr>
<tr>
<td>1</td>
<td>person</td>
<td></td>
<td>person</td>
<td></td>
<td>collaborates</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>person</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td>person</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Example Relationships

- **Name:** Barack Obama  
  **Occupation:** Former President  
  **Relationship:** admires  
  **Relation Type:** 1

- **Name:** Pope Francis  
  **Occupation:** Pope  
  **Relationship:** collaborates with  
  **Relation Type:** 7

- **Name:** Keegan-Michael Key  
  **Occupation:** Actor  
  **Relationship:**  
  **Relation Type:**
Query Graphs
PGQL Graph Query Language

- Graph pattern matching
  - (person) –[:collaborates] -> (person)

- Basic patterns and reachability patterns
  - Can we reach from A to B with an arbitrary number of hops?

- Familiarity for SQL users
  - Similar language constructs and syntax
  - SELECT ... WHERE ....GROUP BY ... ORDER BY
  - “Result set” (table) as output
PGQL Examples

SELECT e
MATCH ()-[:e]->()
PGQL

```
SELECT e
MATCH (n)-[e]->(m)
WHERE n.name='Barack Obama'
```
PGQL

PATH any_edge as ()-[()]()
SELECT n, m MATCH(n) -/: any_edge*/ ->(m)
WHERE n.name = 'Barack Obama'
PGQL

```
SELECT label(e), count(*)
MATCH () -[e]-> ()
GROUP BY label(e)
```

<table>
<thead>
<tr>
<th>label(e)</th>
<th>count(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>admires</td>
<td>28</td>
</tr>
<tr>
<td>leads</td>
<td>9</td>
</tr>
<tr>
<td>feuds</td>
<td>45</td>
</tr>
<tr>
<td>collaborates</td>
<td>82</td>
</tr>
</tbody>
</table>
PGQL

SELECT n, e, m
MATCH (n)-[e:collaborates]->(m)
Analyze Graphs
Analyzing the Marvel Graph

g = session.readGraphWithProperties("config.json")

analyst.pagerank(g)

analyst.vertexBetweennessCentrality(g)

g.publish(VertexProperty.ALL, EdgeProperty.ALL)
Displaying Query Results with Pagerank
PGQL Graph Query
1 SELECT n, n0, n1, e0, e1, n.betweenness, n0.betweenness, n1.betweenness
2 MATCH (n)-[e0]-(n0)-[e1]- (n1)
3 WHERE ID(n) = 'SPIDER-MAN/PETER PAR' AND n1.betweenness > 0 AND n0.betweenness > 0
4 ORDER BY n.betweenness DESC, n0.betweenness ASC, n1.betweenness ASC LIMIT 500
Compute Shortest Path

Find shortest path between “WARR” and “MEREE”

```
w = g.createEdgeProperty(PropertyType.DOUBLE, "weight")
w.fill((Double) 1.0)
src = g.getVertex("WARR")
dst = g.getVertex("MEREE")
path = analyst.shortestPathDijkstra(g, src, dst, w)

// Creating boolean properties to highlight vertices and edges in the path easier
in_path = g.createEdgeProperty(PropertyType.BOOLEAN, "in_path")
for (PgxEdge e: path.edges) {
    in_path.set(e, true);
}

v_in_path = g.createVertexProperty(PropertyType.BOOLEAN, "v_in_path")
for (PgxVertex v: path.vertices) {
    v_in_path.set(v, true);
}

// Shortest Path
SELECT n, e, m, e2, m2, e3, m3, n.pagerank, m.pagerank, m2.pagerank, m3.pagerank, e.in_path, e2.in_path, e3.in_path,
n.v_in_path, m.v_in_path, m2.v_in_path, m3.v_in_path
MATCH (n)-[e]-(m), (n)-[e2]-(m2), (m2)-[e3]-(m3)
WHERE e.in_path AND e2 != e AND e3 != e AND e2 != e3 AND NOT m2.v_in_path AND OUT_DEGREE(m2) < 10 AND NOT m3.v_in_path
ORDER BY n.pagerank ASC, m2.pagerank ASC LIMIT 500
```
Compute Shortest Path

Find shortest path between “WARR” and “MEREE”
Compute Shortest Path

Find shortest path between “WARR” and “MEREE”
Performance and Scale
LDBC benchmark

**Scale factor 1**
Number of vertices: 3,181,724
Number of edges: 17,256,038
Memory footprint: 1.3GB

**Scale factor 3**
Number of vertices: 9,281,922
Number of edges: 52,695,735
Memory footprint: 4.0GB

**Scale factor 10**
Number of vertices: 29,987,835
Number of edges: 176,623,445
Memory footprint: 13.3 GB

- **Oracle**
- **Postgres SQL**
- **Sparksee**
# Property Graph Sizing Recommendations

<table>
<thead>
<tr>
<th>Graph Size</th>
<th>Recommended Physical Memory to be Dedicated</th>
<th>Recommended Number of CPU Processors</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 to 100M edges</td>
<td>Up to 14 GB RAM</td>
<td>2 to 4 processors, and up to 16 processors for more compute-intensive workloads</td>
</tr>
<tr>
<td>100M to 1B edges</td>
<td>14 GB to 100 GB RAM</td>
<td>4 to 12 processors, and up to 16 to 32 processors for more compute-intensive workloads</td>
</tr>
<tr>
<td>Over 1B edges</td>
<td>Over 100 GB RAM</td>
<td>12 to 32 processors, or more for especially compute-intensive workloads</td>
</tr>
</tbody>
</table>
Future Plans
Graph Cloud Service
Graph Cloud Service

Automated, end-to-end analytic service
Fully managed

- Create graphs
  - Automated modeler creates graphs from database tables and file formats
- Explore connections
  - Using visualization tools, notebooks, and query languages
- Analyze relationships
  - With pre-built analytics, visualization tools, and query languages
Why use Graph Cloud?

Quick to Deploy
Graph Studio, a new paradigm for data scientists and analysts to build graph applications

Quick to Get Started
Industry-specific analytic workflows

Simplified Graph Application Development
Automatic graph model creation from relational tables

Easy to Share Results
Publish and share analysis in notebooks or REST endpoints
Developer and Data Scientist Friendly
Graph Studio

Graph Notebook

Apache Zeppelin-compatible interpreters

User Interface built with Oracle JET
Consistent look-and-feel

Powerful and interactive visualization
Custom graph visualization
Extensions to the SQL Standard
SQL/PGQ

- SQL extensions to query property graphs
  - Our team is working with ISO and ANSI committees
  - Target: Next version of SQL (~2020/21)

**Create a Property Graph using SQL Data Definition**

```sql
CREATE PROPERTY GRAPH myGraph
  VERTEX TABLES (Person, Message)
  EDGE TABLES (Created SOURCE Person DESTINATION Message,
                Commented SOURCE Person DESTINATION Message)
```

**Query a Property Graph in SQL**

```sql
SELECT GT.creationDate, GT.content
FROM myGraph GRAPH_TABLE (
  MATCH
    (Creator IS Person WHERE Creator.email = :email1)
    -[ IS Created ]->
    (M IS Message)
    <-[ IS Commented ]-
    (Commenter IS Person WHERE Commenter.email = :email2)
  WHERE ALL_DIFFERENT (Creator, Commenter)
ONE ROW PER MATCH
COLUMNS (M.creationDate, M.content)
) AS GT
```
SQL/PGQ: Implementation Plan

Transition graph support to align with proposed SQL/PGQ standard

External graph analytics engine not required

Query and analysis within database

Performance trade-off
- Large data: use existing SQL engine
- In-memory columnar graph index for performance

External engine still useful

For alternative deployment models
Big Data audience: Spark, Hadoop, HBase …
Scalable distributed in-memory execution
What’s Ahead

Tuesday

11:15-12:00  New Tools to Fight Against Financial Crime [CON6222]
Moscone South – Room 204

12:30-1:15  Using Graph Analysis and Fraud Detection in the Fintech Industry (Paysafe customer session)
Moscone South – Room 152C

12:30-1:15  Blazing-Fast Distributed Graph Query Processing: 100x as Fast as Spark
[DEV3712] Moscone South – Room 307

3:15-4:00  Introducing Oracle Graph Cloud: Automating Graph Analysis [TRN4754]
Moscone South – Room 159B

Wednesday

10:00-10:45  Graph Database and Analytics: How To Use Them [TRN 4755]
Moscone South – Room 152C

1:30-2:15  Traversing and Querying Graphs with PGQL and Gremlin with Oracle Spatial and
Graph [DEV4084] Moscone South – Room 202

Meet the Experts

1:30-2:20  Graph Database and Analysis

2:30-3:20  Graph Cloud Service: Automating Graph Analysis
Lounge C, Code One Groundbreakers Hub, Moscone South Level 1
### Demogrounds

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Title</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday 10:00 am – 4:00 pm</td>
<td>Spatial and Graph: Database, Analytics and Cloud</td>
<td>Moscone South Exhibit Hall (‘The Exchange’)</td>
</tr>
<tr>
<td>Tuesday 10:30 am – 5:30 pm</td>
<td>• Oracle Demogrounds &gt; Data Management area &gt; Kiosk # ODB-017</td>
<td></td>
</tr>
<tr>
<td>Wednesday 10:00 am – 4:30 pm</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The Spatial & Graph SIG User Community
Now part of BIWA User Group

We are a vibrant community of customers and partners that connects and exchanges knowledge online, and at conferences and events.

Meet us at OpenWorld! Monday-Wednesday Moscone West, Level 3, User Group area at the BIWA/Analytics Community table

Join us online
tinyurl.com/oraclespatialcommunity

LinkedIn
@oraspatialsig
oraclespatialsig@gmail.com
analyticsanddatsummit.org
Seeking customer use cases and technology sessions
Dedicated Spatial & Graph track with 20+ sessions
Thank You

Melli Annamalai
(Melliyal.Annamalai@oracle.com)
Help us make the content even better. Please complete the session survey in the Mobile App.