Building a Java Recommender System in 15 Minutes with Graph Technologies

Hassan Chafi, Director, Research & Advanced Development, Oracle

Zhe Wu, Ph.D. Architect
Oracle Spatial and Graph Product Team
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.
Announcing Oracle Database 12c Release 2 on Oracle Cloud

• Available now
  – Exadata Express Cloud Service

• Coming soon
  – Database Cloud Services
  – Exadata Cloud Machine

Oracle is presenting features for Oracle Database 12c Release 2 on Oracle Cloud. We will announce availability of the On-Prem release sometime after Open World.
Agenda

• What is graph?
• Description of common graph use cases
• Overview of Big Data Spatial and Graph Property Graph
• Build recommender system with graph technologies
  • Content-based filtering
  • Personalized PageRank
  • Collaborative Filtering
• Summary
Overview of Graph

• What is a graph?
  – A set of vertices and edges (with optional properties)
  – A graph is simply **linked data**

• Why do we care?
  – Graphs are everywhere
    • Road networks, power grids, biological networks
    • Social networks/Social Web (Facebook, Linkedin, Twitter, Baidu, Google+,…)
    • Knowledge graphs (RDF, OWL)
  – Graphs are intuitive and flexible
    • Easy to navigate, easy to form a path, natural to visualize
    • Do not require a predefined schema
Benefits of the Graph Data Model

• Dealing with unstructured data
  – Flexible schema, arbitrary KV pairs

• Integrating various data sources
  – Flexible schema, can gradually connect data

• Expressing complicated patterns in intuitive ways
  – Graph query engine and language

• Enables advanced topological analytics
  – Graph Analytics
Big Data and Graph Analysis

• The Big Data era is here:
  – Volume, Velocity, Variety, Veracity

• Just storing and managing this data is not sufficient
  – Typically Big Data is low value per byte

• Need to get useful information out of the huge data sets
  – Through applying data analysis

Methodologies:
• Classic OLAP
• Statistical analysis
• Machine learning
• Graph analysis (valence)
Enhancing Data Analytics with Graphs

• Graph analysis can enhance the quality of data analytics

• Graph representation enables to discover hidden information about the data
  – Multi-hop relationship between data entities

• This can be used to further improve predictive model
Graph Analysis Use Case Summary

**Reachability**
Quickly identify multi-hop relations between (a set of) vertices under various constraints based on how they are connected.

- e.g. security breach trace

**Anomaly Detection**
Analyze the relationships between data entities to detect subsets of data that are different from others.

- e.g. fraud detection

**Centrality Analysis**
Analyze the topology of the network, in addition to data values, in order to identify data entities that are more important than others.

- e.g. influencer identification

**Link Prediction**
Inspect similarities between data entities using the global graph structure, and predict potential future links.

- e.g. product recommendation
Graph Analytics: Path and Reachability

**Question:**
In a large data set, given an entity
(1) what are other entities that are connected to the given entity?
(2) how are these entities connected to each other (i.e. via which paths)?

**Approach:**
Using a graph representation, traverse the graph starting from a given vertex.
Business Use Cases

• Supply Chain Management
  – Which products are affected if one supplier experiences issues?
  – Have I increased supply-chain risk after this acquisition (now too much dependency on one supplier?)

• Cyber-security and cyber-threat analysis
  – Which systems might be compromised once infiltration source located?

• Border Protection
  – Identify persons of interest based on $N^{th}$ degree interaction with a suspect

• National Health
  – Identify people who might have been exposed to Zika Virus (took flights together or were in same location) given confirmed infected patient
Graph Analytics: Influencer Identification

**Question:**
Given a graph data, identify vertices that are relatively more important than others (from topology)

**Approach:**
There are various algorithms to compute this:
Graph Analytics: What is Centrality?

- Centrality is a measure of relative importance of vertices in a graph.
- Graph theory defines many different centralities:
  - Betweenness Centrality
  - Closeness Centrality
  - Eigenvector Centrality
  - Pagerank
  - HITS
  - ...

Each algorithm suggests different definition of importance.

(images from Wikipedia)
Use Case: Customer Churn Analysis

• Customer Churn Analysis for a Mobile Network Company
• Data Set
  – Call relationship between customers
• Need to identify
  – Important users among their customers
    ... or central users in the network
  – Goal: keep customers in their network
• How?
  – By computing centrality on the call graph
Graph Analytics: Anomaly Detection

**Question:**
Given a large dataset, identify entities that look different than others especially in their relationships

**Two Possible Approaches:**
(1) Define an anomaly pattern, find all the instances of the pattern in the graph

(2) Given nodes in the same category, find nodes that stand out
Use Case: Anomaly Detection in Medical Records

• Another example for potential fraud detection
  – Public domain dataset
  – Medical providers and their operations

• Question
  – Is there any medical providers that are suspicious
  ➔ medical providers that perform different operations than their fellows
  (e.g. eye doctors doing plastic surgery)

• Approach
  – Create graph between doctors and operations
  – Apply personalized pagerank
  – Identify doctors that are different from their fellows
Question:
Given customer-item purchase (rating) records
For a given customer, recommend items
that the customer may like

Approaches:
A technique called collaborative filtering,
-- Identify users who bought similar items to
  the given user
→ Identify items purchased by those users
→ Recommend those items
Other techniques include using Personalized PageRank,
Content-based Filtering, and Clustering
Use Case: Customer Segmentation with Clustering

• User Clustering
  – Users that have similar taste signature can be grouped together
  – i.e., people who have similar tastes as you
Support Graph Data Types...

- Add graph analytics to applications, tools, and information technology platforms
- Deliver a scalable, secure, and high performing product
- Simplify development with integrated graph analysis, APIs and services

...On all enterprise platforms

- Oracle Database
- Apache Hadoop (Cloudera, Hortonworks)
- Oracle NoSQL Database
- Oracle Big Data Appliance
- Oracle Cloud
Property Graph: A Generic 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
  – 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.

• Blueprints Java APIs
• Implementations
  • Oracle, Neo4j, DataStax, InfiniteGraph, Dex, Sail, MongoDB ...

• A property graph can be modeled as an RDF Graph

https://github.com/tinkerpop/blueprints/wiki/Property-Graph-Model
Property Graph: Big Data Spatial and Graph
Data management + In-memory Graph Analytics

• Massively-Scalable Graph Database
  – Scales securely to **trillions** edges
  – Multiple back-ends: NoSQL, Hbase ... more planned

• Memory-based Parallel Graph Analyst (PGX)
  – 39 built-in memory-based graph analysis algorithms
  – Property Graph Query Language (PGQL)
  – Smart filtering of large graphs

• Flexible interfaces
  – Java: Tinkerpop: Blueprints, Gremlin, Rexster
  – Groovy, Python
  – Apache Lucene and SolrCloud
Feature Overview of Oracle Big Data Spatial and Graph
Architecture of Property Graph Support

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

Access Layer
- Blueprints & Lucene/SolrCloud

Spatial and Graph
- Oracle RDBMS

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

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

Java APIs
- REST/Web Service
- Java, Groovy, Python, ...

Java APIs/JDBC/SQL/PLSQL

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.
Multiple Property Graph Data Format Support

- GML, GraphML, GraphSON
- Oracle-defined Property Graph flat files
  - Vertex file, Edge file
  - Support basic data types + Date with Timezone + Serializable objects
  - Allow multiple data types to be associated with one key
  - UTF8 based
  - Example
    - 1,name,1,Barack%20Obama,`
    - 1,age,2,,53,
    - 1,likes,1,scrabble,`
    - 2,likes,5,,2009-01-20T00:00:00.000-05:00
    - 2,occupation,1,44th%20president%20of%20United%20States%20of%20America,`
Graph Construction: Convert from CSV to Flat Files

- **Key Java APIs:**
  - OraclePropertyGraphUtils.convertCSV2OPV (E)
  - ColumnToAttrMapping array

```java
String inputCSV = "/path/mygraph-vertices.csv";
String outputOPV = "/path/mygraph.opv";
ColumnToAttrMapping[] ctams = new ColumnToAttrMapping[4];
ctams[0]                    = ColumnToAttrMapping.getInstance("VID",  Long.class);
ctams[1]                    = ColumnToAttrMapping.getInstance("name", String.class);
ctams[2]                    = ColumnToAttrMapping.getInstance("score",  Double.class);
ctams[3]                    = ColumnToAttrMapping.getInstance("age",   Integer.class);
String vidColumn            = "VID";
FileInputStream isCSV = new FileInputStream(inputCSV);
FileOutputStream osOPV = new FileOutputStream(new File(outputOPV));
// Convert Vertices
OraclePropertyGraphUtilsBase.convertCSV2OPV(isCSV, vidColumn, 0, ctams, 1, 0, osOPV, null);
```

**Example CSV file**

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Score</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John</td>
<td>4.2</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>Mary</td>
<td>4.3</td>
<td>32</td>
</tr>
<tr>
<td>3</td>
<td>&quot;Skywalker, Anakin&quot;</td>
<td>5.0</td>
<td>46</td>
</tr>
<tr>
<td>4</td>
<td>&quot;Darth Vader&quot;</td>
<td>5.0</td>
<td>46</td>
</tr>
<tr>
<td>5</td>
<td>&quot;Skywalker, Luke&quot;</td>
<td>5.0</td>
<td>53</td>
</tr>
</tbody>
</table>

**Example output .opv file**

```
1,name,1,John,,
1,score,4,,4.2,  
1,age,2,,30,     
2,name,1,Mary,,
2,score,4,,4.3,  
2,age,2,,32,     
```
Graph Construction: Convert from Relational to Flat Files

• Key Java APIs:
  – OraclePropertyGraphUtils.convertRDBMSTable2OPV (E)
  – ColumnToAttrMapping

String opv = "./EmployeeTab.opv"; OutputStream opvOS = new FileOutputStream(opv);
ColumnToAttrMapping[] ctams = new ColumnToAttrMapping[3];
  // map column "hasName" to attribute "name" of type
String ctams[0] = ColumnToAttrMapping.getInstance("hasName", "name", String.class);
  // map column "hasAge" to attribute "age" of type Integer
ctams[1] = ColumnToAttrMapping.getInstance("hasAge", "age", Integer.class);
  // map column "hasSalary" to attribute "salary" of type Double
ctams[2] = ColumnToAttrMapping.getInstance("hasSalary", "salary", Double.class);
OraclePropertyGraphUtils.convertRDBMSTable2OPV(conn, "EmployeeTab", "empID",
  1000l, ctams, 8, opvOS, (DataConverterListener) null);

<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>
</tbody>
</table>
...

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,
Data Access (APIs)

• Core Java API implementation on top of a Property Graph schema
  – Leverage compression, partitioning, security, parallel execution provided by Oracle Database
• Blueprints 2.3.0, Gremlin 2.3.0, Rexster 2.3.0
• Groovy shell for accessing property graph data
• REST APIs (through Rexster integration)
Example of Using the Property Graph REST Interface

• Installation
  – Get the following three jar files under
    $ORACLE_HOME/md/property_graph/dal/webapp
    Rexster*.jar, jersey*.jar, grizz*.jar
  – Configure rexster-opg.xml
  – ./rexster-opg.sh --start -c rexster-opg.xml

• Usage examples
  setenv pg_graph  http://dbhost.us.oracle.com:8182/graphs/connections
  curl -X POST "${pg_graph}/vertices/1001"
  curl -X POST "${pg_graph}/vertices/1002"
  curl -X POST "${pg_graph}/edges/2000?_outV=1001&_label=admires&_inV=1002"
  curl -X DELETE "${pg_graph}/edges/2000"
Example Python Interface

• Installation
  – property_graph/pyopg/README

• Usage
  – cd ${ORACLE_HOME}/md/property_graph/pyopg
    ./pyopg.sh
  ipython notebook

```python
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(16,12));
community_frame["size"].plot(kind="bar", title="Communities and Sizes")
ax.set_xticklabels(community_frame.index);
```
Text Search through Apache Lucene/Solr Cloud

- Integration with Apache Lucene/Solr
- 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*"
Powerful Text Search Capabilities by Solr

• A very rich set of capabilities and query syntaxes
  – Query: chief AND officer
  – Query: "chief officer"~1 (one position away, within 1 edit distance)
  – Query: title:[boat TO boulder] Matches boat, boil, book, boulder...
  – Query: created:[2012-02-01T00:00.0Z TO 2012-08-02T00:00.0Z]
  – Query: administrator~ Matches: administratore, administrater, administratior,...
  – Query: title:(cool graph)^2.5 description:(cool graph) supports customized boost
  – GeoSpatial
  – Faceted Search
  – Hierarchical matching,
  – More Like This, ...
Build Recommender System with Property Graph (Part I)
Building a Recommender System
-- with Oracle Big Data Spatial and Graph Property Graph

• Environment
  – Oracle Big Data Lite VM 4.5.0+
  – Oracle Big Data Spatial and Graph v1.2.0+
  – SolrCloud 4.10.x

• A “user-item” property graph
  – Vertices (items, descriptions, and users)
  – Edges (linking users and items)

Recommendation: you may also like
Building a Recommender System
-- with Oracle Big Data Spatial and Graph Property Graph

• BDSG offers multiple approaches and they can be *mixed* together

**Content-based filtering**
• Match item description
• Match user profile
• Relevancy ranking

**Collaborative filtering**
• People liked similar items in the past will like similar items in the future

**Personalized Page Ranking**
• Randomly navigate from a user to a product, then back to a user, ...
• Randomly jump to starting point(s)
Demo/HoL: Content-based Recommendation with Property Graph and SolrCloud
Property Graph Query Language (PGQL)

Querying graph with patterns

– SQL-like syntax but with graph pattern description and property access

Example: Find friends of friends of Paul that are in between 20 and 30 years old

```
SELECT friendOfFriend.name, friendOfFriend.age
FROM 'SocialNetworkGraph'
WHERE
  (:Person WITH name = 'Paul') -[:likes]-> (friend:Person),
  (friend) -[:likes]-> (friendOfFriend:Person),
  friendOfFriend.age >= 20 AND friendOfFriend.age <= 30
ORDER BY friend.name
```

• SQL Like syntax format:
  SELECT, WHERE, FROM, ORDER BY

• Graph pattern description:
  (...) -> (...)

• Label and property access:
  (:Person WITH name = 'Paul')
PGQL: Pattern Matching

- Graph query ➔ Given a pattern, find all the matching instances in the data graph.
- BDSG provides fast graph query features with an intuitive query language.

```
SELECT v3.name, v3.age
FROM 'myGraph'
WHERE
  (v1:Person WITH name = 'Amber') -[:friendOf]-> (v2:Person) -[:knows]-> (v3:Person)
```

**Query:** Find all people who are known to friends of ‘Amber’.
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
Build Recommender System with Property Graph (Part II)
Personalized Page Rank-based Recommender System
Random walk with restart

For brevity, each edge on the left consists of two directed edges: “userA purchased productB”, and “productB purchasedBy userA”.

Reference: https://blogs.oracle.com/bigdataspatialgraph/entry/intuitive_explanation_of_personalized_page
Key API for Personalized Page Rank

- API: `ppr=analyst.personalizedPagerank` (pgxGraph, vertexSet, 0.0001 /*max error*/, 0.85 /*damping factor*/, 1000 );
- Result: `ppr.getTopKValues()`

```java
it=ppr.getTopKValues(9).iterator(); while (it.hasNext()) {
  entry=it.next(); vid=entry.getKey().getId();
  System.out.format("ppr=%.4f  vertex=%s\n", entry.getValue(), opg.getVertex(vid));
}
```

```
ppr=0.2496  vertex=Vertex ID 1  {name:str:John, age:int:10}
ppr=0.1758  vertex=Vertex ID 11 {type:str:Prod, desc:str:Kindle Fire}
ppr=0.1758  vertex=Vertex ID 10 {type:str:Prod, desc:str:iPhone5, released:dat:Sat Jan 21 00:02:00 EST 2012}
```
Demo/HoL: Personalized Page Rank-based Recommender System
Recommendation with C.F.

• Matrix factorization

\[
W \times H \approx V
\]

• Graph intuition

A recursive graph algorithm solves taste signature of both customers and items

A customer’s *taste signature* is defined by what he/she likes

An item’s *taste signature* is (recursively) defined by who likes it
Demo/HoL: Collaborative Filtering Recommender System
What’s New: Property Graph Features
BDSG 1.2.0 (with BDA 4.5.0)

Faster, powerful and more scalable

- PGQL Graph Query (Pattern Matching)
- Distributed In-memory Graph Analysis
- Fast Centrality Computation with MS-BFS
- Filtered Sub-graph Reading
- Graph Build and Change API

- Label Support
- Recommendation Based-on Collaborative Filtering
- Security Enhancement
- Default Web Server
- Over One Dozen New SNA Algorithms
What’s New: Property Graph Features
BDSG 2.0 (with BDA 4.6.0)

Faster, easier to use

- PGQL 1.0 Support
- Node.js client support
- Integration of Apache Spark with the in-memory analyst
- Vertex label support by treating value of a designated property as label

- Solr Cloud 5.2.x support
  - (Hortonworks certification)
- Improved CSV to Flat Files converter and relational to Flat Files converter
- Improved text index concurrency
- More New SNA Algorithms
- Support for more data types: long, char, byte, short, Spatial data wrapper
Planned: Property Graph on Oracle Database
Oracle Spatial and Graph: Database 12.2

- In-Memory Analyst (PGX) ships with Oracle Spatial and Graph
- Integration with Oracle R, Oracle Data Miner, Oracle Business Intelligence
- Spatial support in property graphs
- Oracle Text and Apache Lucene/Solr Cloud integration
- In-database graph analytics
  - Sparsification, shortest path, page rank, triangle counting, WCC, sub graph generation...
- SQL-based graph navigation and graph query
- Property graph view
- RDBMS benefits: Parallel load, parallel query, B-tree index, compression (columnar, ACO, etc), Label security and authentication.
Integrated Cloud
Applications & Platform Services
Appendix and Additional Resources
Graph at OOW 2016 View this list at [http://bit.ly/FODSpatialGraph](http://bit.ly/FODSpatialGraph)

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Title</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday, Sept 19</td>
<td>Building a Java Recommender System in 15 Minutes with Graph Technologies [TUT5289]</td>
<td>Parc 55 - Cyril Magnin II/III</td>
</tr>
<tr>
<td>8:30 a.m. - 10:30 a.m.</td>
<td>Graph and Link Analysis: Discovering Network Relationships in Big Data [CON6445]</td>
<td>Park Central - Olympic</td>
</tr>
<tr>
<td>Tuesday, Sept 20</td>
<td>Bring Graph Analysis to Relational and Hadoop Data [THT7821]</td>
<td>Moscone South Exhibition Hall - Big Data Theater</td>
</tr>
<tr>
<td>1:00 p.m. – 1:20 p.m.</td>
<td>Meet the Experts: Oracle’s Big Data Management System [MTE7224]</td>
<td>Moscone South - 306</td>
</tr>
<tr>
<td>6:15 p.m. - 7:00 p.m.</td>
<td>Meet the Experts: Spatial and Graph Technologies for Database, Big Data, and the Cloud</td>
<td>Moscone South - 306</td>
</tr>
<tr>
<td>Thursday, Sept 22</td>
<td>Build Applications on Spark, Hadoop, and NoSQL with Oracle Big Data Spatial and Graph [CON6585]</td>
<td>Park Central - Olympic</td>
</tr>
</tbody>
</table>
### Spatial and Graph Demos

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Title</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday - Wednesday</td>
<td>Oracle’s Spatial Technologies for Database, Big Data, and the Cloud</td>
<td>Moscone South Exhibition Hall – 101 Oracle Database DEMOgrounds</td>
</tr>
<tr>
<td>Monday - Wednesday</td>
<td>Oracle’s Graph Database and Analytics for Database, Big Data, and the Cloud</td>
<td>Moscone South Exhibition Hall– 101 Oracle Database DEMOgrounds</td>
</tr>
<tr>
<td>Monday - Wednesday</td>
<td>Discover and Analyze Relationships with Oracle Big Data Spatial and Graph</td>
<td>Moscone South Exhibition Hall Big Data Showcase (SBD-016)</td>
</tr>
</tbody>
</table>

### Partners

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Title</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday – Wednesday</td>
<td>Tom Sawyer Software</td>
<td>Moscone South Exhibition Hall - Booth #436</td>
</tr>
<tr>
<td>Monday – Wednesday</td>
<td>DataNinja (DocomoInnovations)</td>
<td>Moscone South Exhibition Hall - Booth #1741</td>
</tr>
<tr>
<td>Monday – Wednesday</td>
<td>interRel Consulting</td>
<td>Moscone South Exhibition Hall - Booth #3208</td>
</tr>
</tbody>
</table>
The Spatial & Graph SIG User Community

The SIG connects and exchanges knowledge via online communities and at conferences and events.

• Chat & Coffee at OOW
  Join us & chat with other S&G users about your experiences! Bring your own coffee 😊
  Wednesday at 8:00 am
  Meet in front of Oracle Bookstore – Moscone South Lobby

• Join us online
  tinyurl.com/oraclespatialcommunity

LinkedIn, Google+ & IOUG SIG groups

@oraspatailssig | oraclespatialsig@gmail.com
BIWA SUMMIT 2017
WITH SPATIAL SUMMIT + YESSION SUMMIT
THE Big Data + Analytics + Spatial + Cloud + IoT + Everything Cool User Conference
January 31 - February 2, 2017

Call for Speakers Now Open!
Visit www.biwasummit.org

REGISTER SOON
The Spatial & Graph SIG User Community

The SIG connects and exchanges knowledge via online communities and at conferences and events

- **Chat & Coffee at OOW**
  Join us & chat with other S&G users about your experiences! Bring your own coffee 😊
  Wednesday at 8:00 am
  Meet in front of Oracle Bookstore – Moscone South Lobby

- **Join us online**
  tinyurl.com/oraclespatialcommunity

LinkedIn, Google+ & IOUG SIG groups

@oraspatsig | oraclespatialsig@gmail.com
Learn More About Oracle’s Spatial and Graph Technologies

- [blogs.oracle.com ➔ oraclespatial ➔ oracle_maps_blog ➔ bigdataspatialgraph](http://blogs.oracle.com/oraclespatial/oracle_maps_blog/bigdataspatialgraph)
More Resources
Resources on Big Data Spatial and Graph

• Oracle Big Data Spatial and Graph on Oracle.com: www.oracle.com/database/big-data-spatial-and-graph

• 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: www.oracle.com/technetwork/database/bigdata-appliance/oracle-bigdatalite-2104726.html

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

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

• @OracleBigData, @SpatialHannes, @JeanIhm
Appendix:

Feature Overview of Oracle Spatial and Graph
(skipped features common on both Big Data platform and Oracle Database)
Key Features: Optimized Data Management for Oracle Database

- **Fast, Parallel Data Loader**
  - External Table
  - SQL*Loader
  - JDBC
  - Support merge operations

- **Quick Graph Data Change Identification**
  - Uses SCN and flashback
  - Similar functions are **not available** in either Oracle NoSQL Database or Apache HBase

- **User-friendly Management UI with Enterprise Manager**
SQL-based Query for Property Graph

- Customized SQL query against `<graph>VT$` and `<graph>GE$`
- Example: multiple step navigation with filters

```sql
with G as (select svid, dvid
    from connectionsGE$
    where el in ('collaborates', 'admires')  -- filters: only interested in 'collaborates' and 'admires'
)
select count(distinct g4.dvid)
from G g1, G g2, G g3, G g4
where g1.svid = 1                                                    -- 3 hops away
    and g1.dvid = g2.svid
    and g2.dvid = g3.svid
    and g3.dvid = g4.dvid
;
```
SQL-based Query for Property Graph (2)

- Customized SQL query against `<graph>`VT$ and `<graph>`GE$
- Example: from “vertical” to “horizontal” with **Pivot**

with G as (  
    select vid, k, v  
    from connectionsVT$)  
select * from G  
pivot (  
    min(v) for k in  
    ('company', 'occupation',  
    'show', 'religion')  
)  
;
SQL-based Query for Property Graph (3)

• Customized SQL query against <graph>VT$ and <graph>GE$

• Join VT$ and/or GE$ with other data types/table(s)/view(s)
  • Spatial
  • XML
  • JSON
  • RDF
  • Relational
  • Big Data (HDFS/Hive/HBase/NoSQL/…) via Big Data SQL

• **Customization allowed**
  • Create additional B-Tree indices
  • Turn on advanced compression
  • Enable database in-memory
SQL-based Query for Property Graph (4)

- Customization example: Enable Database In-Memory
  
  ```sql
  alter table connectionsVT$ inmemory;
  ```

```
  select v.owner, v.segment_name name, v.populate_status status
  from v$im_segments v;
```

<table>
<thead>
<tr>
<th>OWNER</th>
<th>NAME</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCOTT</td>
<td>CONNECTIONSVT$</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>SCOTT</td>
<td>CONNECTIONSVT$</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>SCOTT</td>
<td>CONNECTIONSVT$</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>SCOTT</td>
<td>CONNECTIONSVT$</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>SCOTT</td>
<td>CONNECTIONSVT$</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>SCOTT</td>
<td>CONNECTIONSVT$</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>SCOTT</td>
<td>CONNECTIONSVT$</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>SCOTT</td>
<td>CONNECTIONSVT$</td>
<td>COMPLETED</td>
</tr>
</tbody>
</table>

Script Output: All Rows Fetched: 8 in 0.039 seconds
Key Features: SQL-based Query and Analytics for Oracle Database

• SQL-Based Analytics for **both PG graph and RDF Graph**
  – Page ranking
  – Shortest path
  – Graph sparsification
  – Clustering
  – Triangle counting
  – Sub graph identification
  – Take advantage of **parallel execution, skew handling, window function**, etc. in Oracle Database
Key Features: Parallel Property Graph Data Loader/Scanner

- Provides a simple Java API to perform parallel graph data loading
  - Splitter threads
    - split vertex/edge flat files into multiple chunks
  - Loader threads
    - load each chunk into database using separate database connection
  - Use Java pipes to connect Splitter and Loader threads
    - Splitter: PipedOutputStream
    - Loader: PipedInputStream

- Example

  ```java
  opgdl = OraclePropertyGraphDataLoader.getInstance();
vfile = "~/home/data/pg-bda-nosql/demo/connections.opv";
efile = "~/home/data/pg-bda-nosql/demo/connections.ope";
opgdl.loadData(opg, vfile, efile, 32, 1000, true, "PDML=T,PDDL=T,NO_DUP=T,");
  ```
Key Features: Flexible Graph Analytics Deployment
- Shared memory mode

• Embedded
  – Java application embeds the in memory analytics component
  – Same address space
  – Efficient but requires sufficient RAM on the client side

• Remote
  – Graph analytics deployed in a web container
    • WebLogic Server, Tomcat, Jetty
    • SSL/HTTP Basic Authentication
Key Features: Flexible Graph Analytics Deployment (2)

Client Machine

- Embedded Analytics
- Data Access Layer

Embedded Analytics

Oracle Database/Real Application Clusters (RAC)
Key Features: Flexible Graph Analytics Deployment (3)
Key Features: Flexible Graph Analytics Deployment (4)
Enterprise Security for Property Graph Data

- **Property Graph provides**
  - Graph level security through GRANT/REVOKE privileges
  - Fine grained security through integration with Oracle Label Security

- **Oracle Label Security** - mandatory access control
  - **Labels** assigned to both users and graph data (vertices/edges)
  - Data labels determine the *sensitivity* of the rows or the rights a person must posses in order to read or write the data.
  - User labels indicate their *access rights* to the data records.

---

**An example security label assignment for edges**

- Security label: Top Secret
- Security label: Secret
- Security label: Unclassified
Features: Change Tracking of Graph Elements

• Integration with Oracle Flashback Technology

```java
public interface OracleChangeTracker {
    public Long getCurrentSCN() throws SQLException;
    public Iterator<VertexChange> getVertexChanges(Long startSCN, Long endSCN);
    public Iterator<VertexChange> getVertexChanges(Long startSCN, Long endSCN, int queryDOP);
    public Iterator<EdgeChange> getEdgeChanges(Long startSCN, Long endSCN);
    public Iterator<EdgeChange> getEdgeChanges(Long startSCN, Long endSCN, int queryDOP);
}
```

• Need to adjust UNDO_RETENTION
Sub-graph Extraction using SQL View

• Example: Create a sub-graph view with only edges having label “friendOf”

create view my_sub_graphVT$
as
select *
from my_graph VT$
;

create view my_sub_graphGE$
as
select *
from my_graphGE$
where el !='friendOf'
;

create view my_sub_graphVT$
as
select *
from my_graph VT$
;

create view my_sub_graphGE$
as
select *
from my_graphGE$
where el !='friendOf'
;
Property Graph View on RDF

• Motivation
  – Create a view on RDF so that property graph analytics can be applied

• Key method: `createPropertyGraphViewOnRDF`
  
  ```java
  predsForVertexAttrs = new PGUtils.RDFPredicate[4];
  predsForVertexAttrs[0] = PGUtils.RDFPredicate.getInstance("http://oracle.com/nsg/interactionCount", "interactionCount");
  predsForVertexAttrs[1] = PGUtils.RDFPredicate.getInstance("http://www.w3.org/2000/01/rdf-schema#label", "name");
  
predsForEdges = new PGUtils.RDFPredicate[2];
  predsForEdges[0] = PGUtils.RDFPredicate.getInstance("http://oracle.com/nsg/refsPerson", "refsPerson");
  
  PGUtils.createPropertyGraphViewOnRDF(oracle.getConnection(), "pgsocial", "social", false, predsForVertexAttrs, predsForEdges);
  ```

RDF Predicate URIs ➔ Property graph vertex attributes
RDF Predicate URIs ➔ Property graph edges

RDF Model name ➔ PG view name
Geo-Spatial Data Support

• Motivation
  – Location (GeoSpatial) data is important to entities (vertices) and relationships (edges)
  – Need to store, index and query spatial data together with property graph data

• Key flow:

  Add spatial data to the graph
  – v.setProperty("geoloc", new SimpleSpatialDataWrapper("POINT(-121.839,37.6373)"))
  – e = oraclePropertyGraph.addEdge(200l, vStart, vEnd, "call")
  – e.setProperty("originated_from", new SimpleSpatialDataWrapper("POLYGON((-76.57418668270113 38.91891450597657, -76.57484114170074 38.91758725401061, -76.57661139965057 38.91881851059802, -76.57418668270113 38.91891450597657)")))

  Create Spatial index with Oracle Spatial and Graph
  Perform GeoSpatial Search and filter with a spatial "window of interest"