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

Java
Your
Next
(Cloud)

JavaOne
ORACLE

Safe Harbor Statement

Announcing Oracle Database 12c Release 2 on Oracle Cloud

- Available now
  - Exadata Express Cloud Service
- Coming soon
  - Database Cloud Services
  - Exadata Cloud Machine

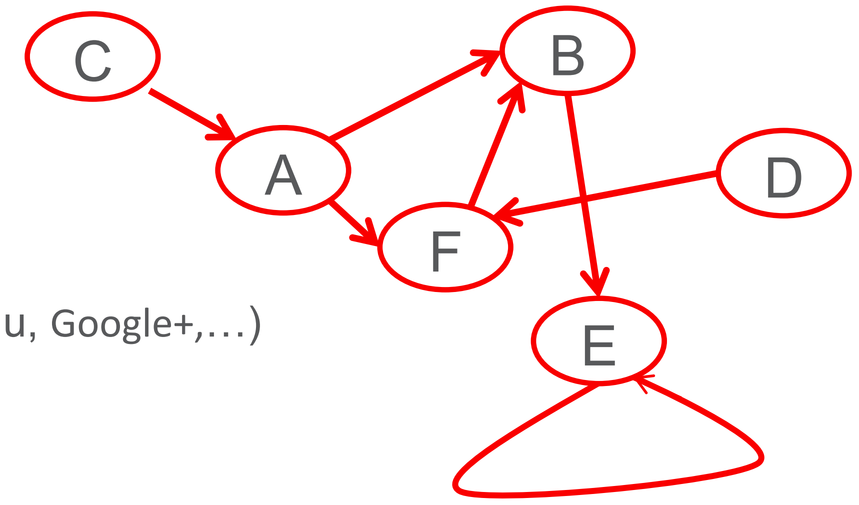
Oracle is presenting features for Oracle Database 12c Release 2 on Oracle Cloud. We will announce availability of the On-Prem release sometime after Open World.

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

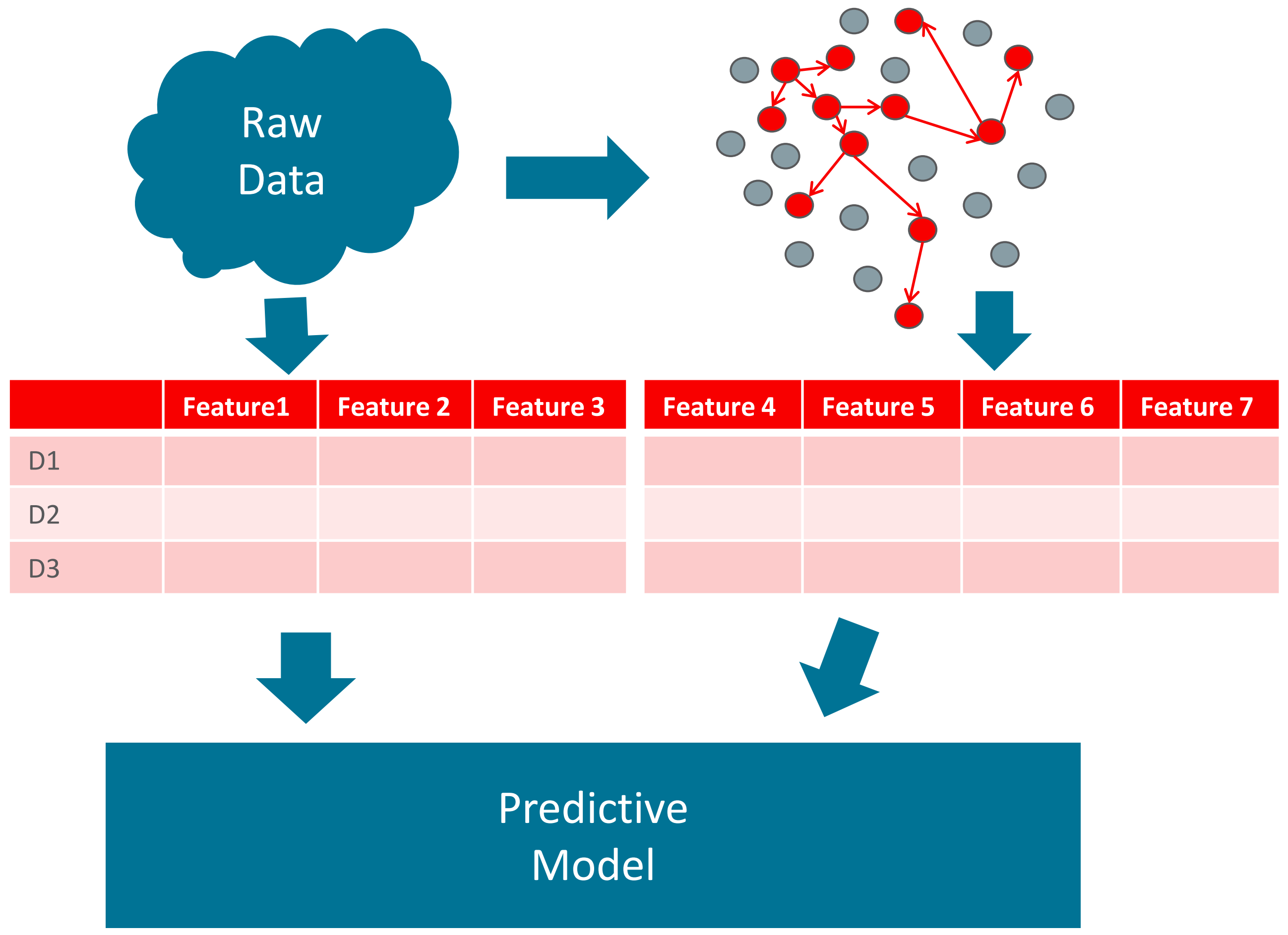
- The Big Data era is here:
  - Volume, Velocity, Variety, Veracity
- Just storing and managing this data is not sufficient
  - Typically Big Data is low value per byte
- Need to get useful information out of the huge data sets
  - Through applying data analysis

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



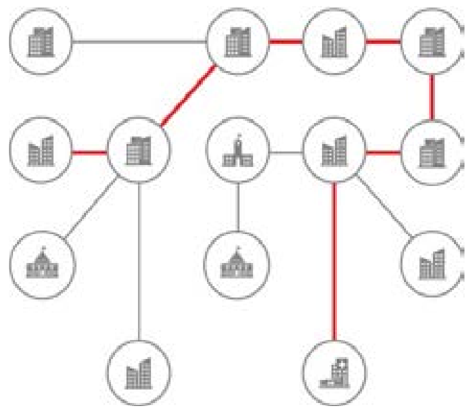
	Feature1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Feature 7
D1							
D2							
D3							

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

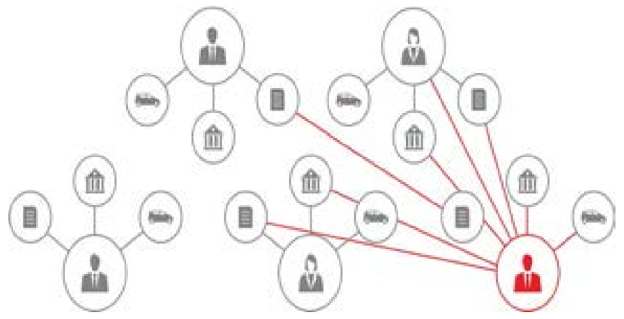
e.g. security breach trace



Anomaly Detection

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

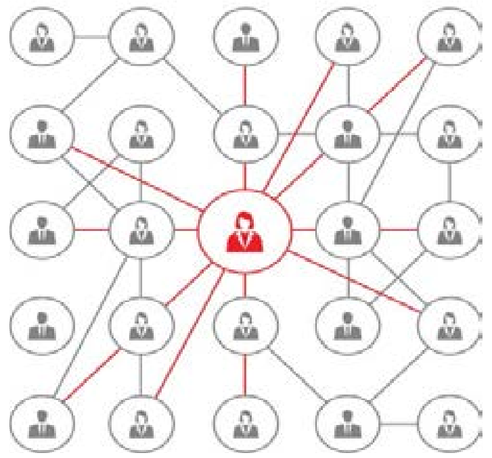
e.g. fraud detection



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.

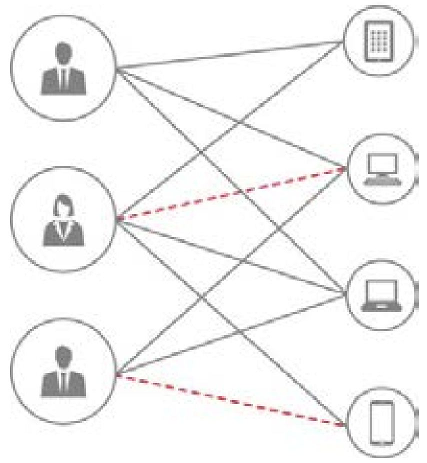
e.g. influencer identification



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

<u>Question</u>:

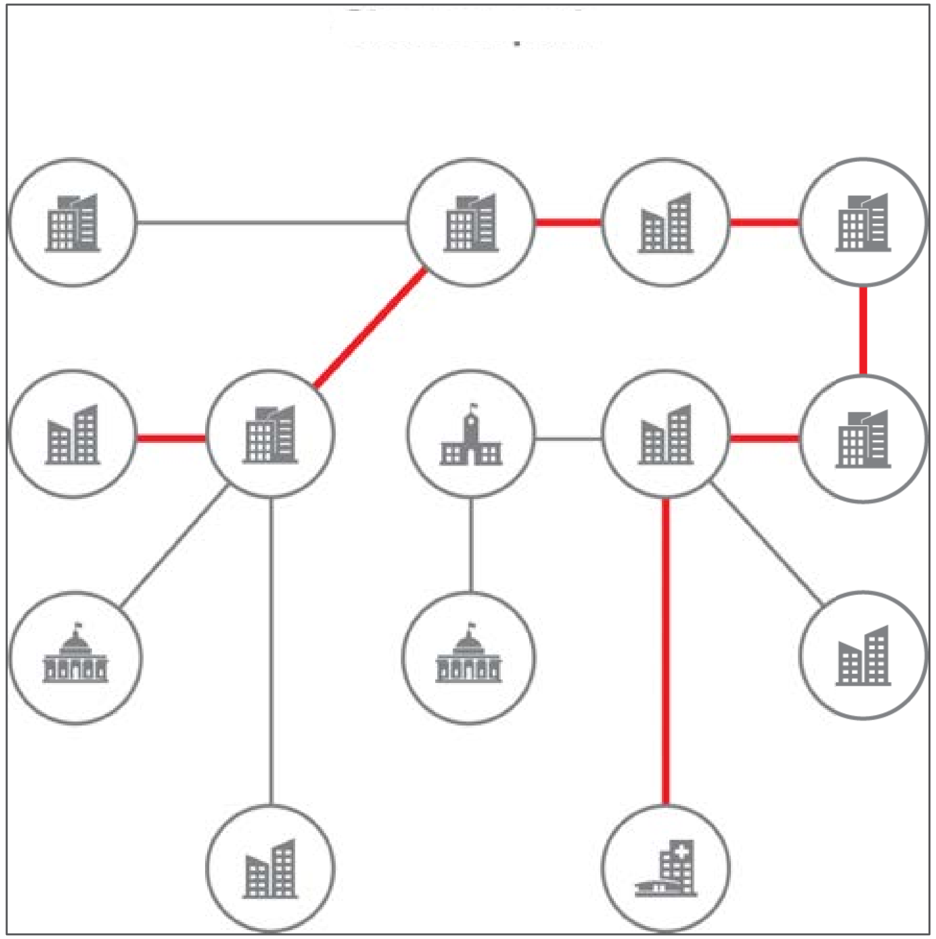
In a large data set, given an entity

(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)?

<u>Approach</u>:

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:

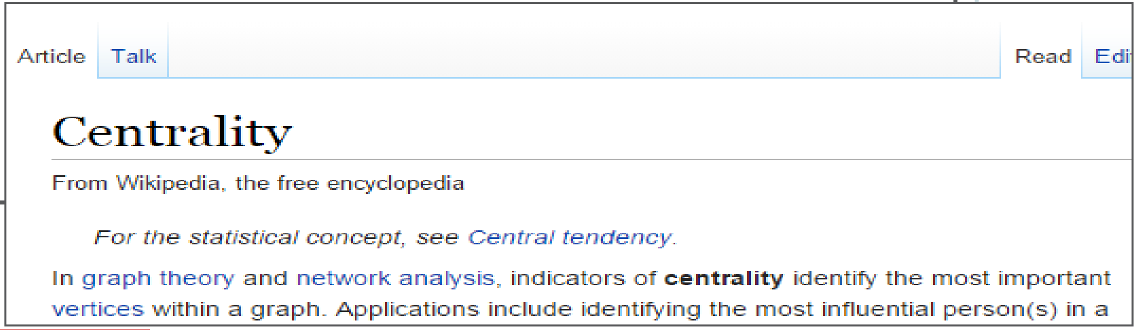
Given a graph data,

identify vertices that are relatively

more important than others (from topology)

Approach:

There are various algorithms to compute this:



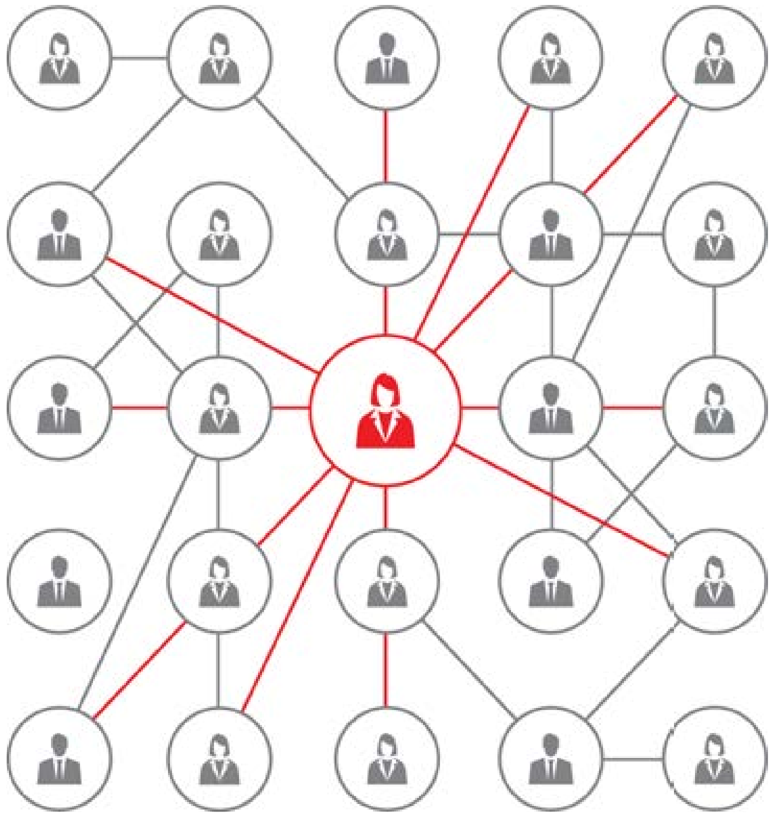
Article Talk Read Edi

Centrality

From Wikipedia, the free encyclopedia

For the statistical concept, see Central tendency.

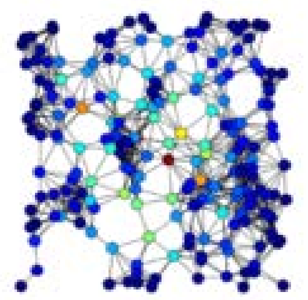
In graph theory and network analysis, indicators of **centrality** identify the most important vertices within a graph. Applications include identifying the most influential person(s) in a

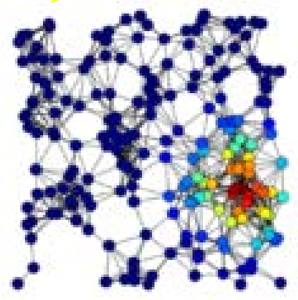


Graph Analytics : What is Centrality?

- 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*


Betweenness centrality

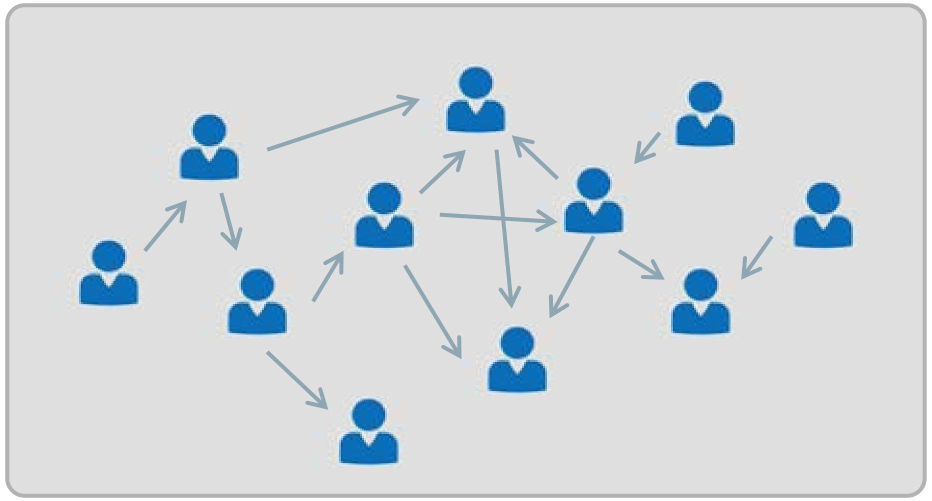

Eigenvector centrality

(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:

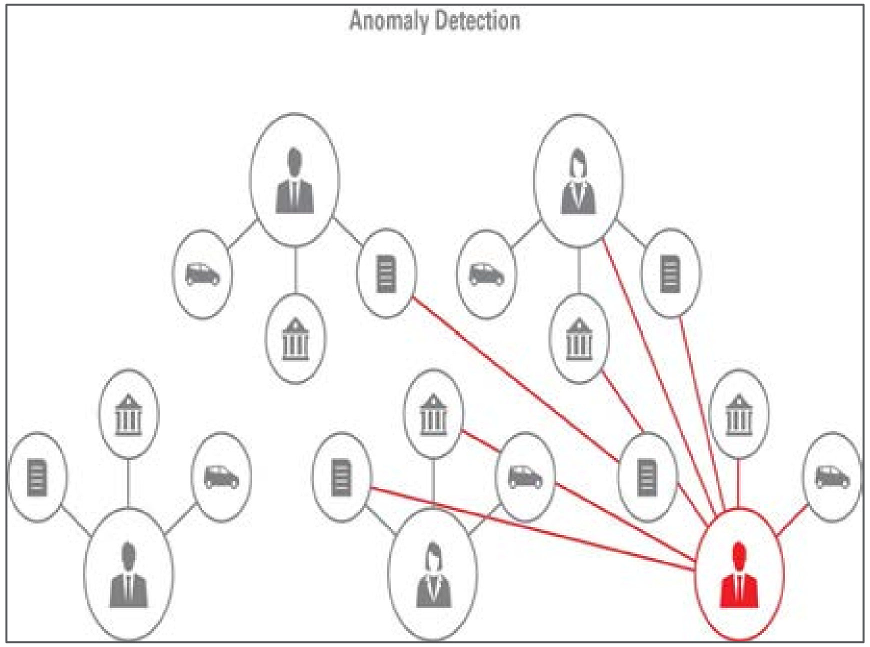
Given a large dataset,

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



ORACLE

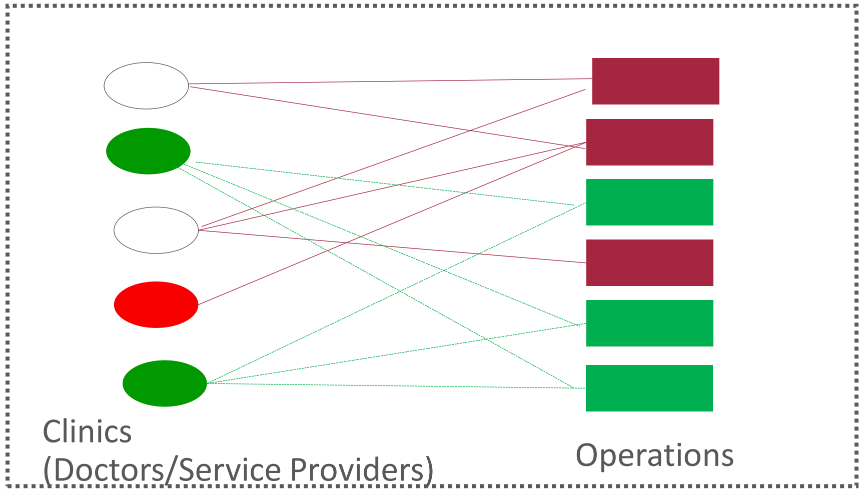
Use Case: Anomaly Detection in Medical Records

- 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



Graph Analytics: Recommendation

Question:

Given customer-item purchase (rating) records

For a given customer, recommend items that the customer may like

Approaches:

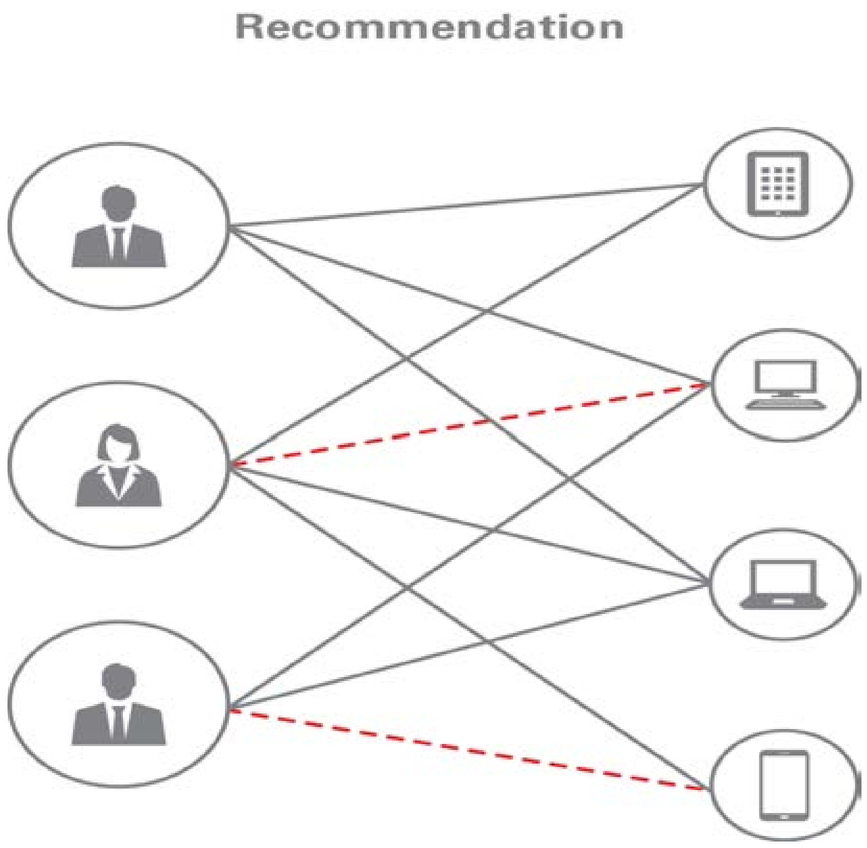
A technique called *collaborative filtering*,

-- Identify users who bought similar items to the given user

➔Identify items purchased by those users

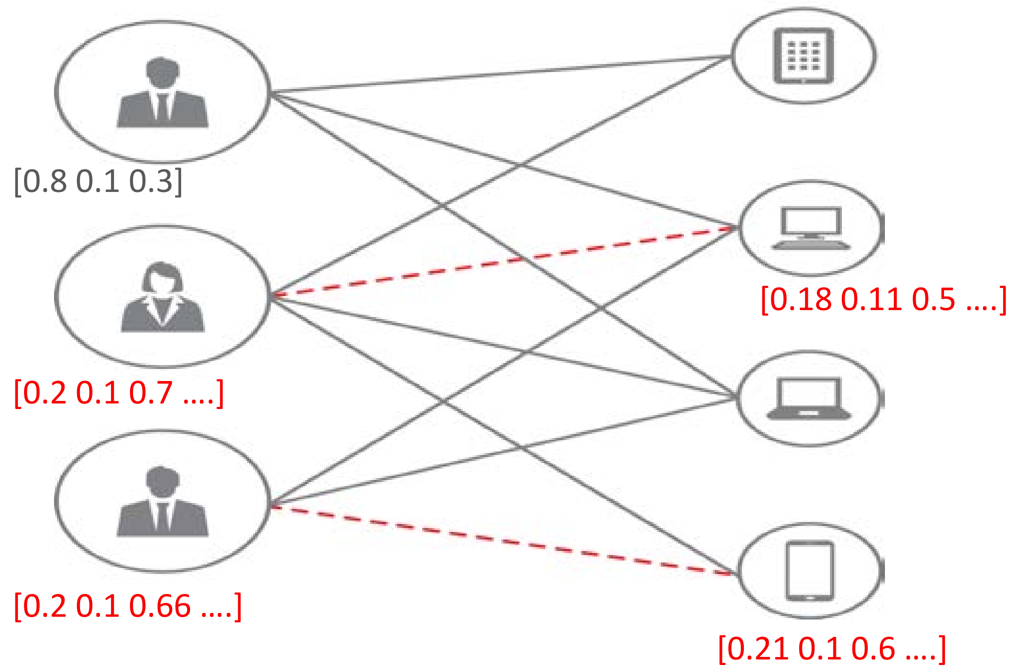
➔Recommend those items

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



Oracle’s Graph Database Strategy

Support Graph Data Types…

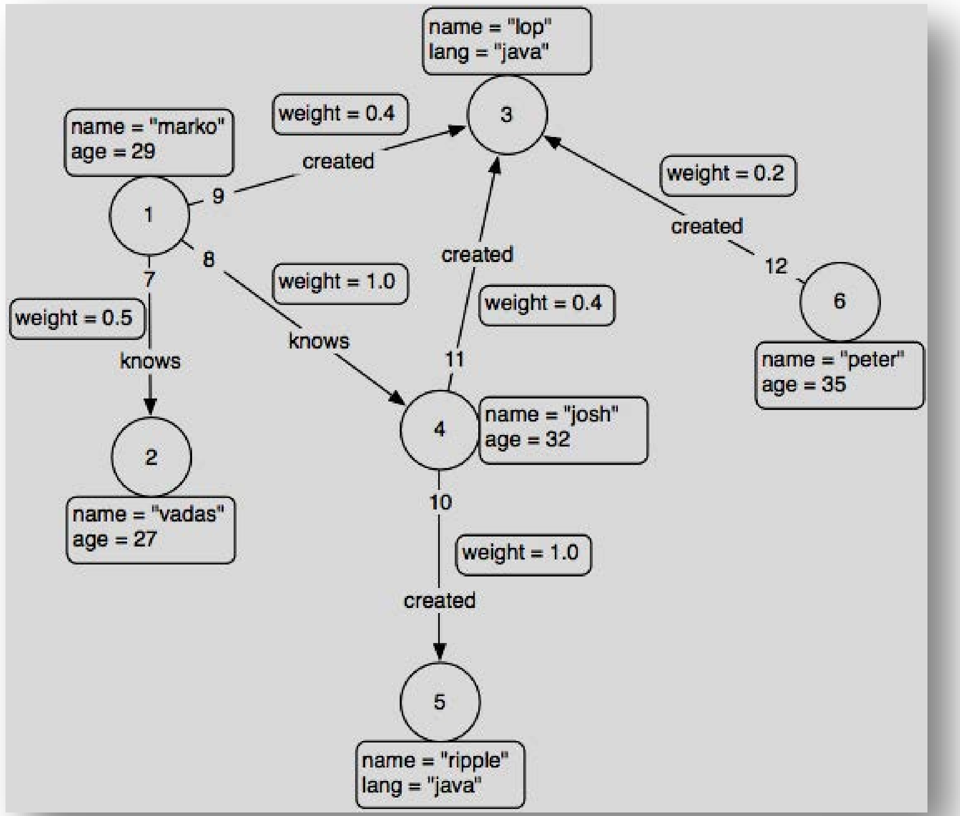
- Add graph analytics to applications, tools, and information technology platforms
- Deliver a scalable, secure, and high performing product
- Simplify development with integrated graph analysis, APIs and services

…On all enterprise platforms

- Oracle Database
- Apache Hadoop (Cloudera, Hortonworks)
- Oracle NoSQL Database
- Oracle Big Data Appliance
- Oracle Cloud

ORACLE

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

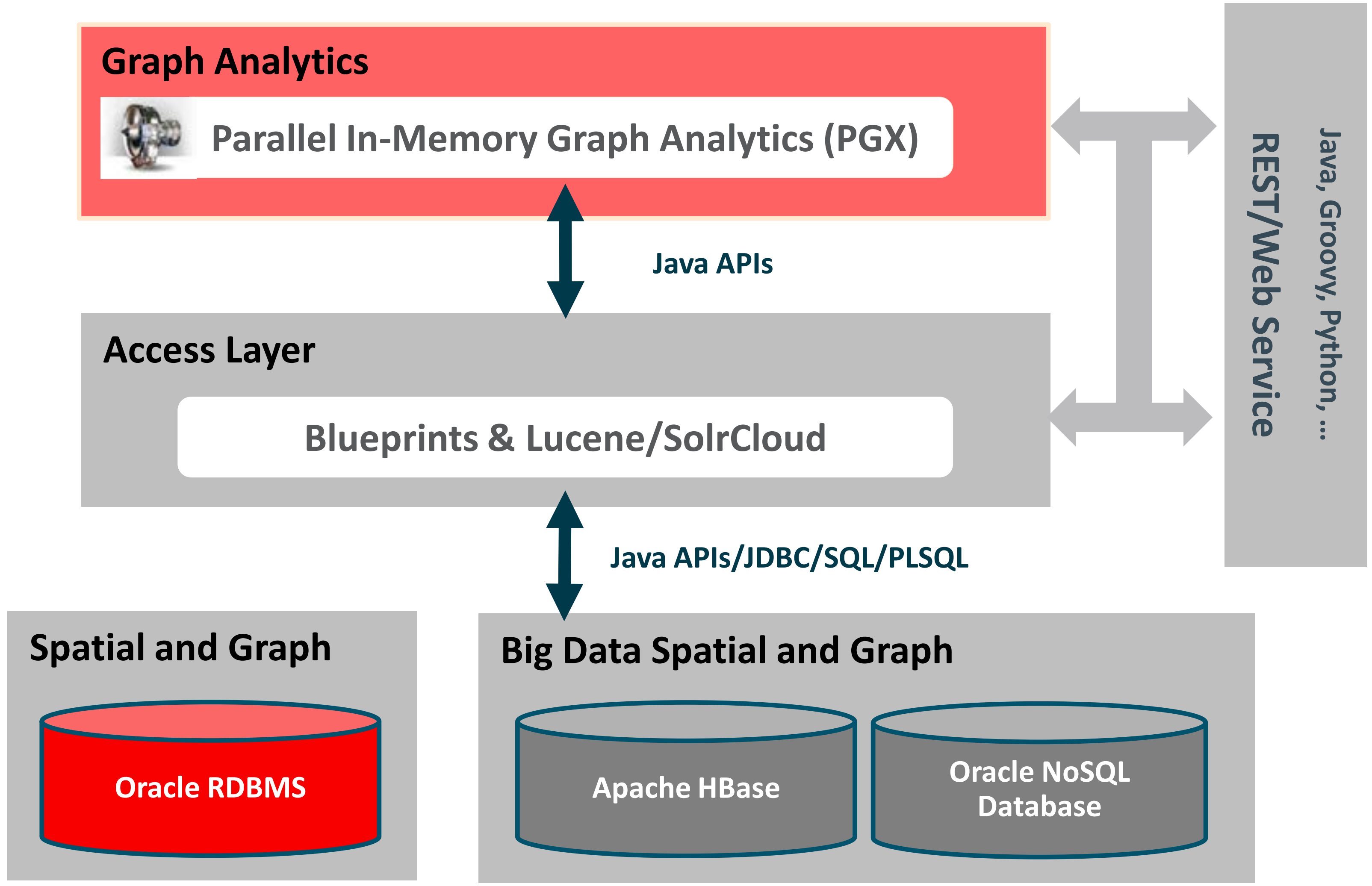
https://github.com/tinkerpop/blueprints/wiki/Property-Graph-Model

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



Property graph formats supported

- GraphML
- GML
- Graph-SON
- Flat Files
- CSV
- Relational Data Sources

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

```
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";      isCSV = new FileInputStream(inputCSV);
osOPV = new FileOutputStream(new File(outputOPV));
   // Convert Vertices
OraclePropertyGraphUtilsBase.convertCSV2OPV(isCSV, vidColumn, 0, ctams, 1, 0, osOPV, null);
```

Example CSV file

```
1,John,4.2,30
2,Mary,4.3,32
3,"Skywalker, Anakin",5.0,46
4,"Darth Vader",5.0,46
5,"Skywalker, Luke",5.0,53
```

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);
```

EmployeeTab

EMPID	hasName	hasAge	hasSalary
101	Jean	20	120.0
102	Mary	21	50.0
…	…	…	…

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”
```

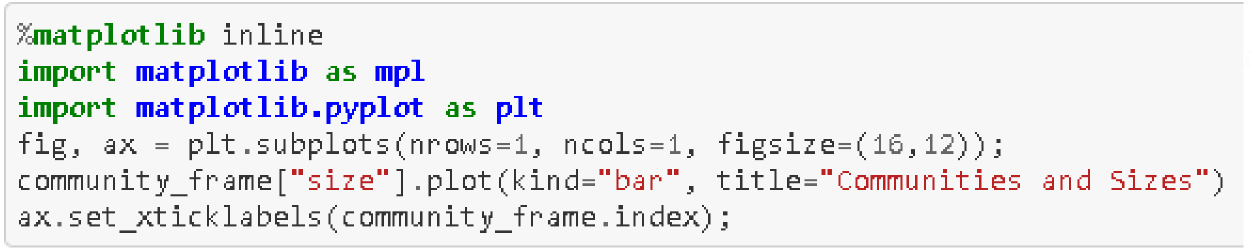
ORACLE

Example Python Interface

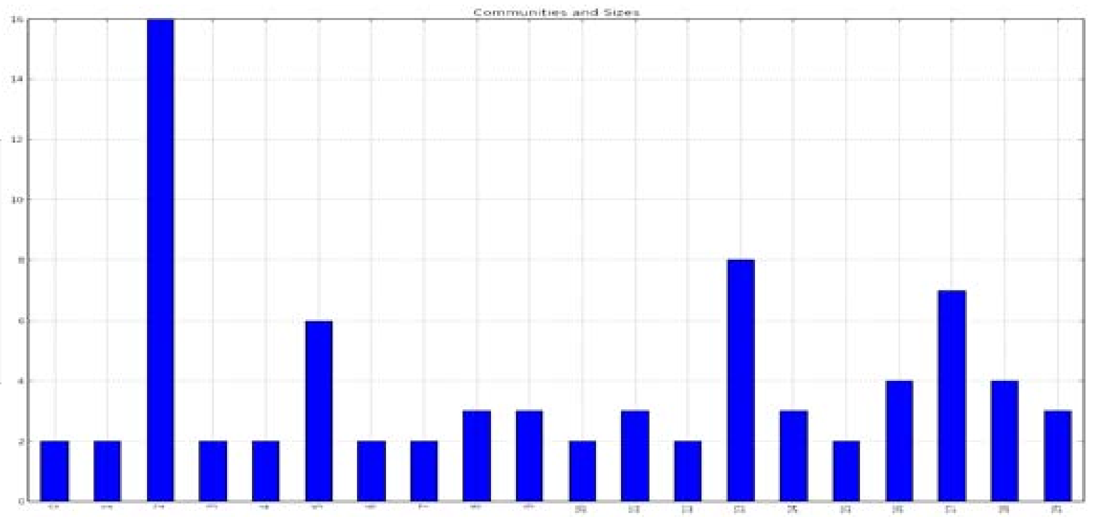
- Installation
  - property_graph/pyopg/README
- Usage
  - cd ${ORACLE_HOME}/md/property_graph/pyopg

  ./pyopg.sh

  ipython notebook

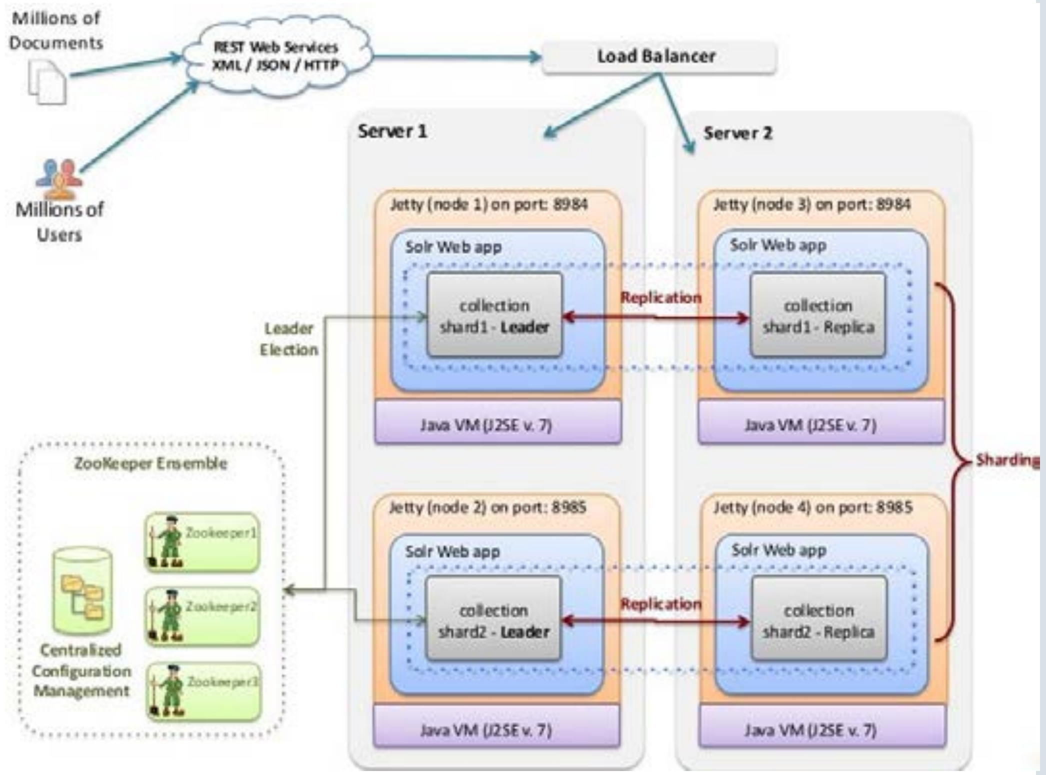


```
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(16,12));
community_frame["size"].plot(kind="bar", title="Communities and Sizes")
ax.set_xticklabels(community_frame.index);
```



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*”



Millions of Documents
REST Web Services XML / JSON / HTTP
Load Balancer
Millions of Users
Server 1
Server 2
Jetty (node 1) on port: 8984
Solr Web app
collection shard1 - Leader
Java VM (J2SE v. 7)
Jetty (node 3) on port: 8984
Solr Web app
collection shard1 - Replica
Java VM (J2SE v. 7)
Replication
Leader Election
Sharding
ZooKeeper Ensemble
Centralized Configuration Management
Zookeeper1
Zookeeper2
Zookeeper3
Jetty (node 2) on port: 8985
Solr Web app
collection shard2 - Leader
Java VM (J2SE v. 7)
Jetty (node 4) on port: 8985
Solr Web app
collection shard2 - Replica
Java VM (J2SE v. 7)
Replication

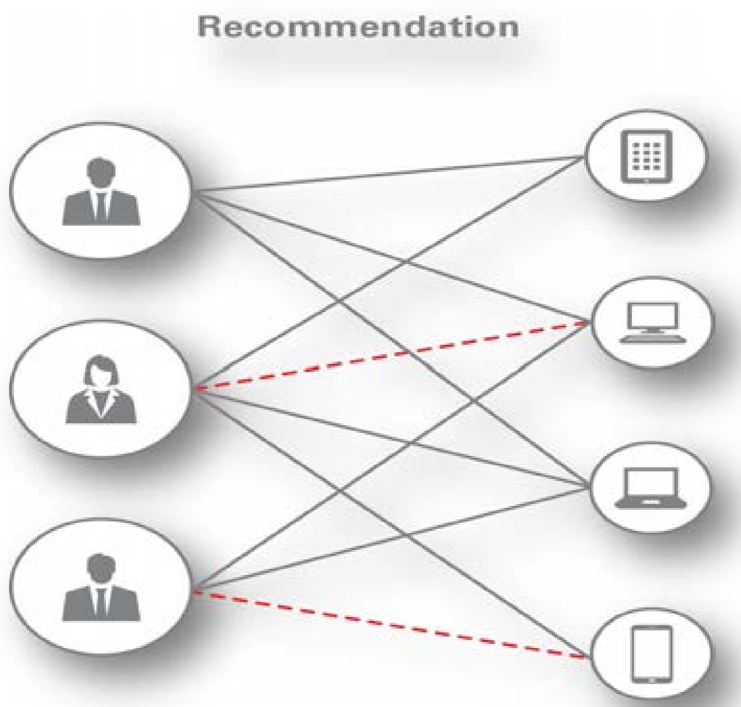
Powerful Text Search Capabilities by Solr

- 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: price:[12.99 TO 14.99] Matches 12.99, 13.000009, 14.99, etc.
  - Query: created:[2012-02-01T00:00.0Z TO 2012-08-02T00:00.0Z]
  - Query: administrator~ Matches: adminstrator, 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)



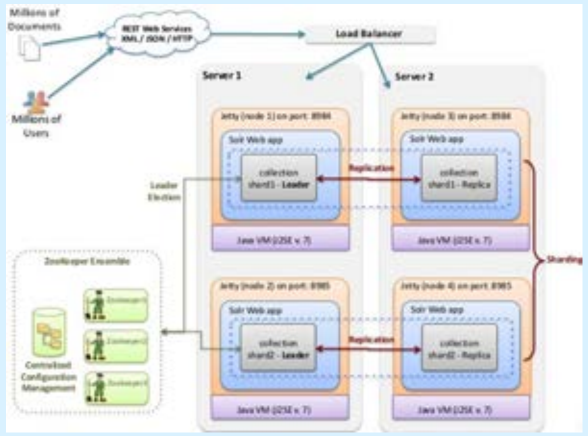
Recommendation: **you may also like**

Building a Recommender System
-- with Oracle Big Data Spatial and Graph Property Graph

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

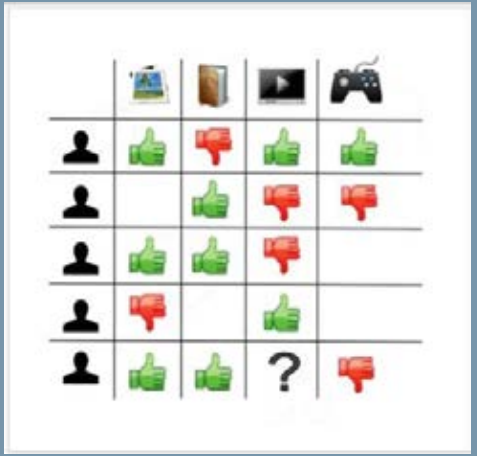
Content-based filtering

- Match item description
- Match user profile
- Relevancy ranking


Millions of Documents
REST Web Services XML / JSON / HTTP
Load Balancer
Millions of Users
Server 1
Server 2
Jetty (node 1) on port: 8984
Jetty (node 3) on port: 8984
Solr Web app
collection shard1 - Leader
Replication
collection shard1 - Replica
Leader Election
Java VM (J2SE v. 7)
Sharding
ZooKeeper Ensemble
Jetty (node 2) on port: 8985
Jetty (node 4) on port: 8985
collection shard2 - Leader
collection shard2 - Replica
Centralized Configuration Management

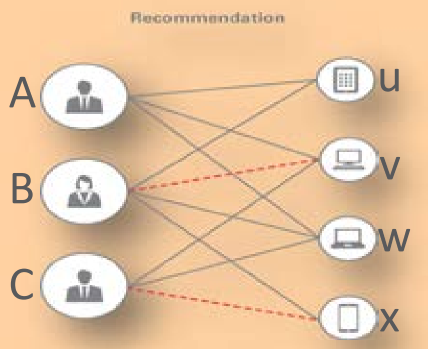
Collaborative filtering

- 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)



- A → u
- u → B
- B → w
- w → C

…

Demo/HoL: Content-based Recommendation with Property Graph and SolrCloud

Property Graph Query Language (PGQL)

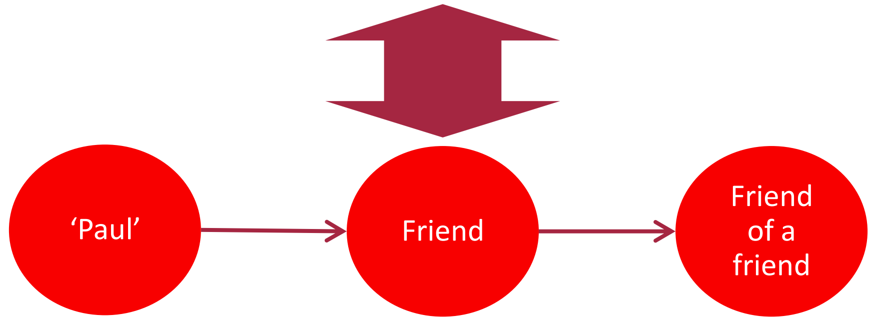
Querying graph with patterns

– 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')

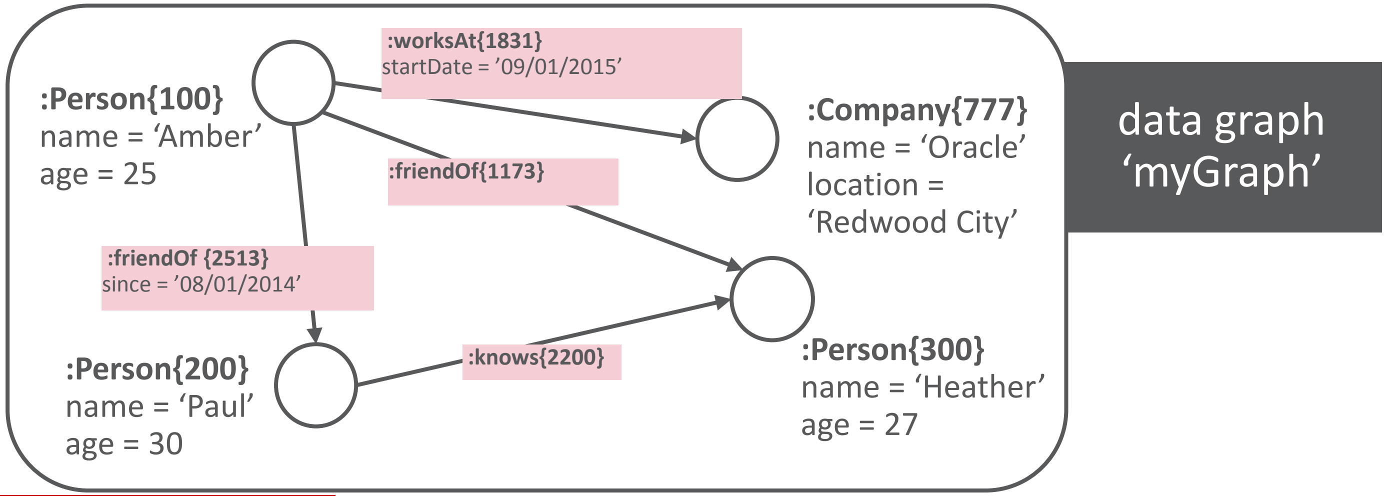


PGQL: Pattern Matching

- 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



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
- weighted Pagerank

ORACLE

Social Network Analysis Algorithms (2)

Pathfinding

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

Recommendation

- salsa
- personalizedSalsa
- whomToFollow

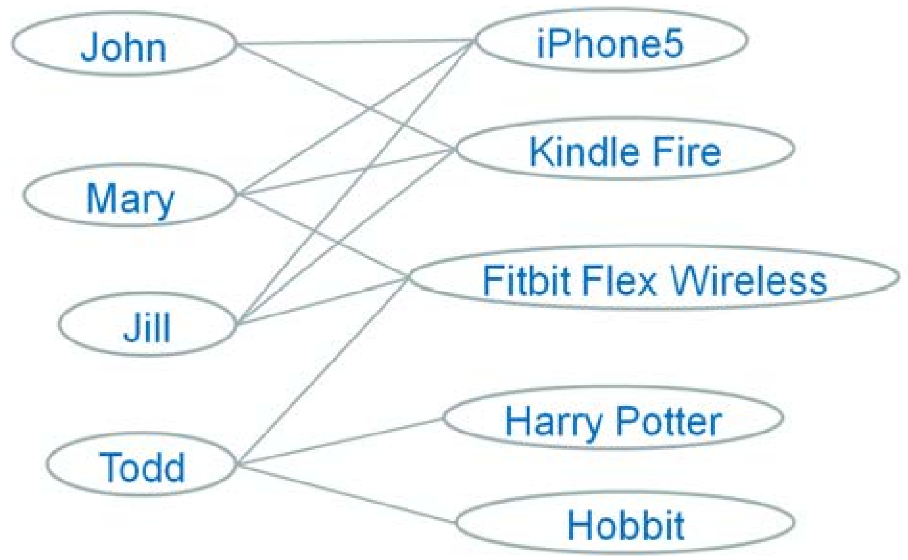
Classic - Connected Components

- sccKosaraju
- sccTarjan
- wcc

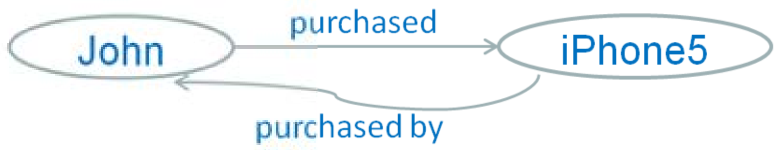
Build Recommender System with Property Graph (Part II)

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()

```
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));}
```

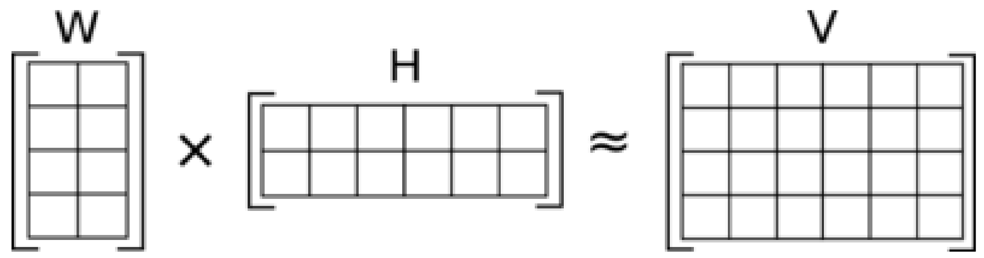
```
ppr=0.2496  vertex=Vertex ID 1    {name:str:John, age:int:10}
ppr=0.1758  vertex=Vertex ID 11 {type:str:Prod, desc:str:Kindle Fire}
ppr=0.1758  vertex=Vertex ID 10 {type:str:Prod, desc:str:iPhone5, released:dat:Sat Jan 21 00:02:00 EST 2012}
```

Demo/HoL: Personalized Page Rank-based Recommender System

Recommendation with C.F.

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

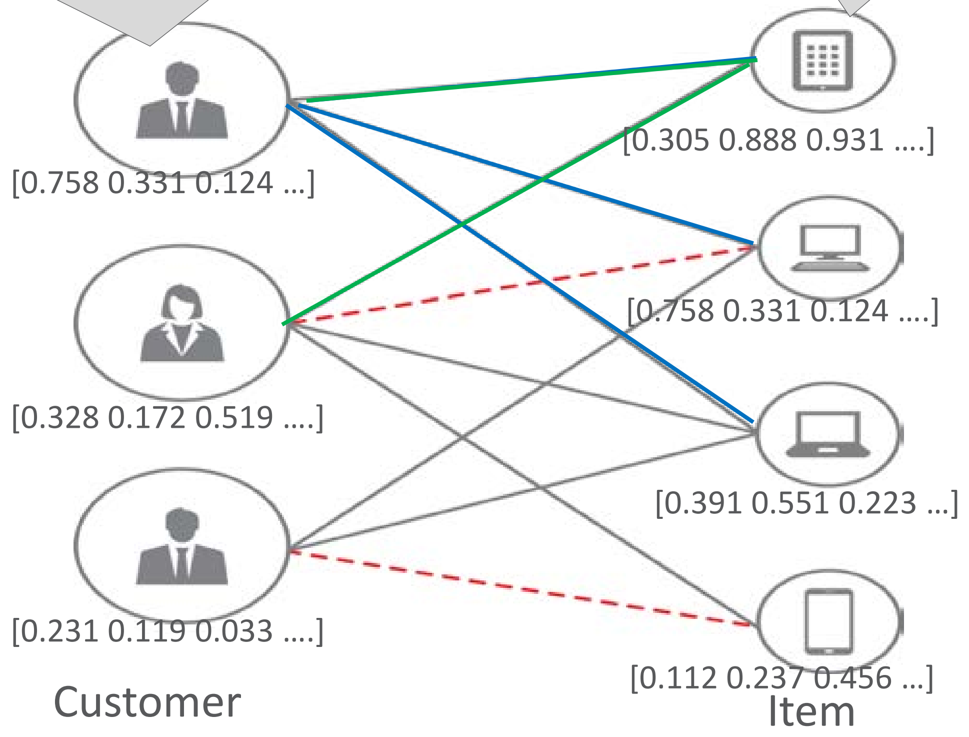
- Matrix factorization



- Graph intuition

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

ORACLE

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

ORACLE

Planned: Property Graph on Oracle Database

Oracle Spatial and Graph: Database 12.2

NEW IN 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.

ORACLE

Integrated Cloud

Applications & Platform Services

ORACLE

Appendix and Additional Resources

Graph at OOW 2016 View this list at http://bit.ly/FODSpatialGraph

Graph Sessions

Date/Time	Title	Location
Monday, Sept 19		
8:30 a.m. - 10:30 a.m.	Building a Java Recommender System in 15 Minutes with Graph Technologies [TUT5289] JavaOne BIG DATA	Parc 55 - Cyril Magnin II/III
4:15 p.m. - 5:00 p.m.	Graph and Link Analysis: Discovering Network Relationships in Big Data [CON6445] BIG DATA	Park Central - Olympic
Tuesday, Sept 20		
1:00 p.m. – 1:20 p.m.	Bring Graph Analysis to Relational and Hadoop Data [THT7821] BIG DATA	Moscone South Exhibition Hall - Big Data Theater
6:15 p.m. - 7:00 p.m.	Meet the Experts: Oracle's Big Data Management System [MTE7224] BIG DATA	Moscone South - 306
7:15 p.m. - 8:00 p.m.	Meet the Experts: Spatial and Graph Technologies for Database, Big Data, and the Cloud 12c BIG DATA	Moscone South - 306
Thursday, Sept 22		
10:45 a.m. - 11:30 a.m	Build Applications on Spark, Hadoop, and NoSQL with Oracle Big Data Spatial and Graph [CON6585] BIG DATA	Park Central - Olympic

ORACLE

Spatial and Graph at OOW 2016 *View this list at* http://bit.ly/FODSpatialGraph

Spatial and Graph Demos

Date/Time	Title	Location
Monday - Wednesday	Oracle's Spatial Technologies for Database, Big Data, and the Cloud 12c BIG DATA	Moscone South Exhibition Hall – 101 Oracle Database DEMOgrounds
Monday - Wednesday	Oracle's Graph Database and Analytics for Database, Big Data, and the Cloud 12c BIG DATA	Moscone South Exhibition Hall– 101 Oracle Database DEMOgrounds
Monday - Wednesday	Discover and Analyze Relationships with Oracle Big Data Spatial and Graph BIG DATA	Moscone South Exhibition Hall Big Data Showcase (SBD-016)

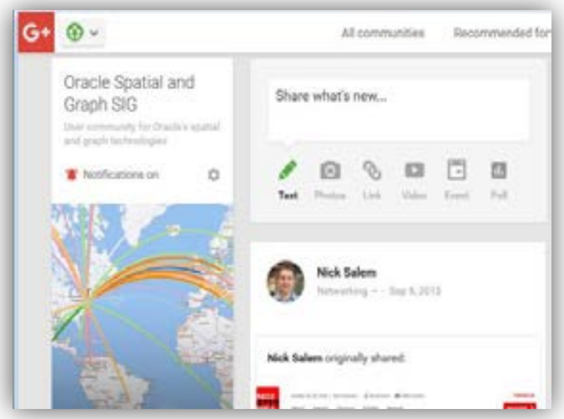
Partners

Date/Time	Title	Location
Monday – Wednesday	Tom Sawyer Software BIG DATA	Moscone South Exhibition Hall - Booth #436
Monday – Wednesday	DataNinja (DocomoInnovations)	Moscone South Exhibition Hall - Booth #1741
Monday – Wednesday	interRel Consulting	Moscone South Exhibition Hall - Booth #3208

ORACLE

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

  @oraspatialsig | oraclespatialsig@gmail.com

BIWA SUMMIT 2017 WITH SPATIAL SUMMIT + YESSQL SUMMIT

THE Big Data + Analytics + Spatial + Cloud + IoT + Everything Cool User Conference
January 31 - February 2, 2017

REGISTER SOON

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

BIWA SIG

ORACLE Spatial & Graph SIG

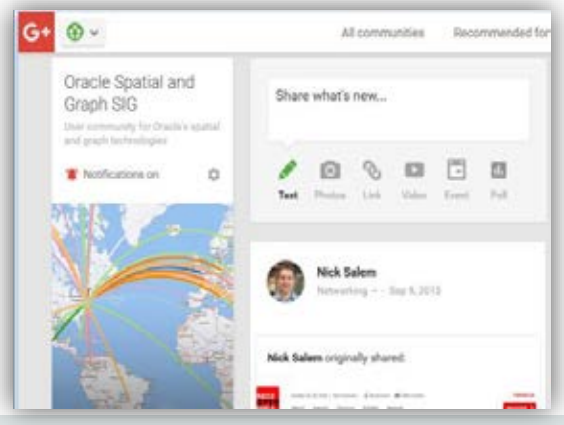
IOUG independent oracle users group

NoCOUG NORTHERN CALIFORNIA ORACLE USERS GROUP

The Spatial & Graph SIG User Community

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

Oracle Spatial & Graph SIG



Oracle Spatial and Graph SIG

Share what's new...

Nick Salem

Nick Salem originally shared:

- **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

  @oraspatialsig | oraclespatialsig@gmail.com

Learn More About Oracle's Spatial and Graph Technologies

- www.oracle.com/technetwork/database/options/spatialandgraph
  www.oracle.com/database/big-data-spatial-and-graph
- blogs.oracle.com ➔oraclespatial ➔ oracle_maps_blog ➔bigdataspatialgraph

ORACLE
Search
Products Solutions Downloads Store Support Training Partners About OTN
Overview
Oracle
Spatial and Graph
Visit Oracle at
ODTUG Kscope14
REGISTER TODAY!
Get the Latest Oracle Database 12c Tutorials
Access Now

The industry's most advanced database for fast, scalable, reliable and secure spatial and massive graph applications

Oracle Big Data Spatial and Graph

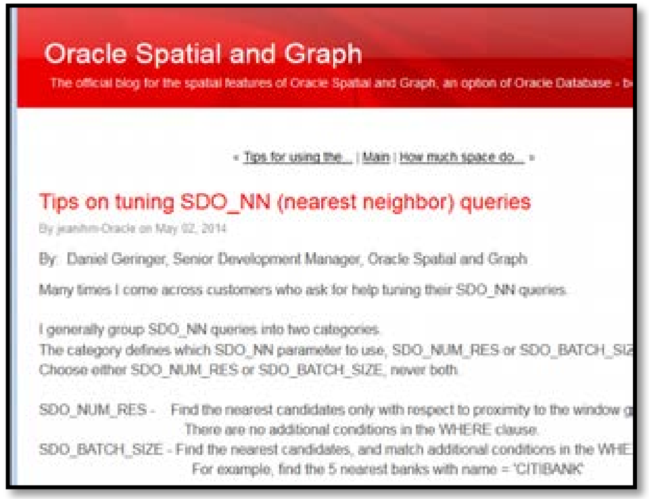
Rich Spatial and Graph Analytics for Hadoop

Discover relationships and connections among customers, organizations, and assets, and enrich your big data with location. Handle the most challenging graph, spatial and raster processing workloads on Apache Hadoop and NoSQL database technologies.

Uncover location relationships and patterns in big data to better understand customers, retail performance, and insurance risk.

Data sheet (PDF)

Spatial features (PDF)



Oracle Spatial and Graph

The official blog for the spatial features of Oracle Spatial and Graph, an option of Oracle Database - b

« Tips for using the... | Main | How much space do... »

Tips on tuning SDO_NN (nearest neighbor) queries

By jeanihm-Oracle on May 02, 2014

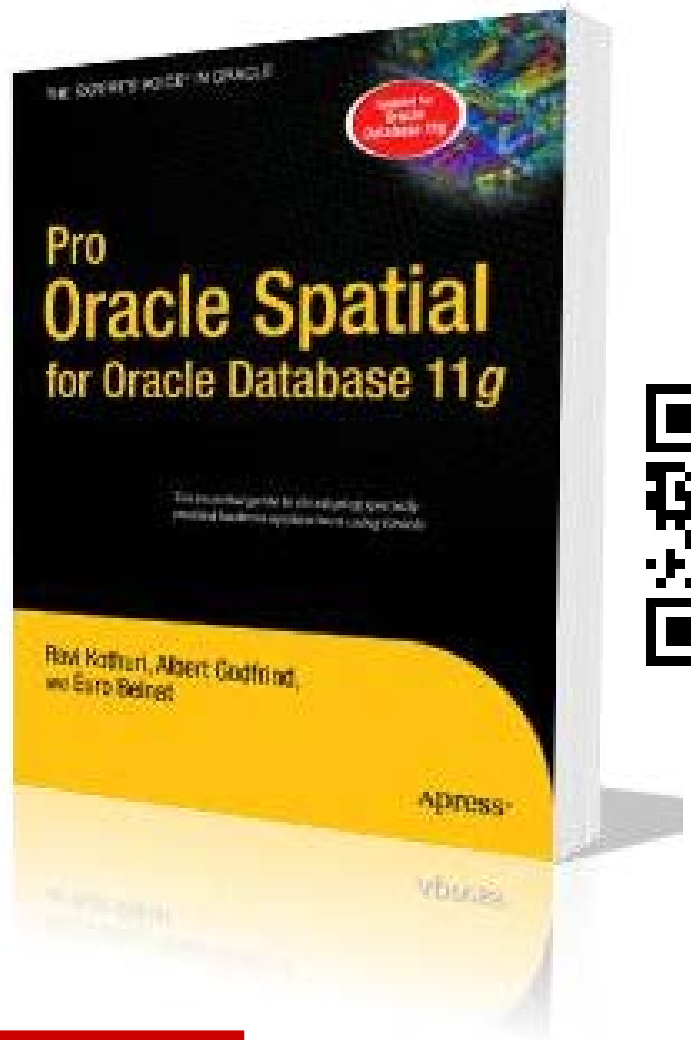
By: Daniel Geringer, Senior Development Manager, Oracle Spatial and Graph

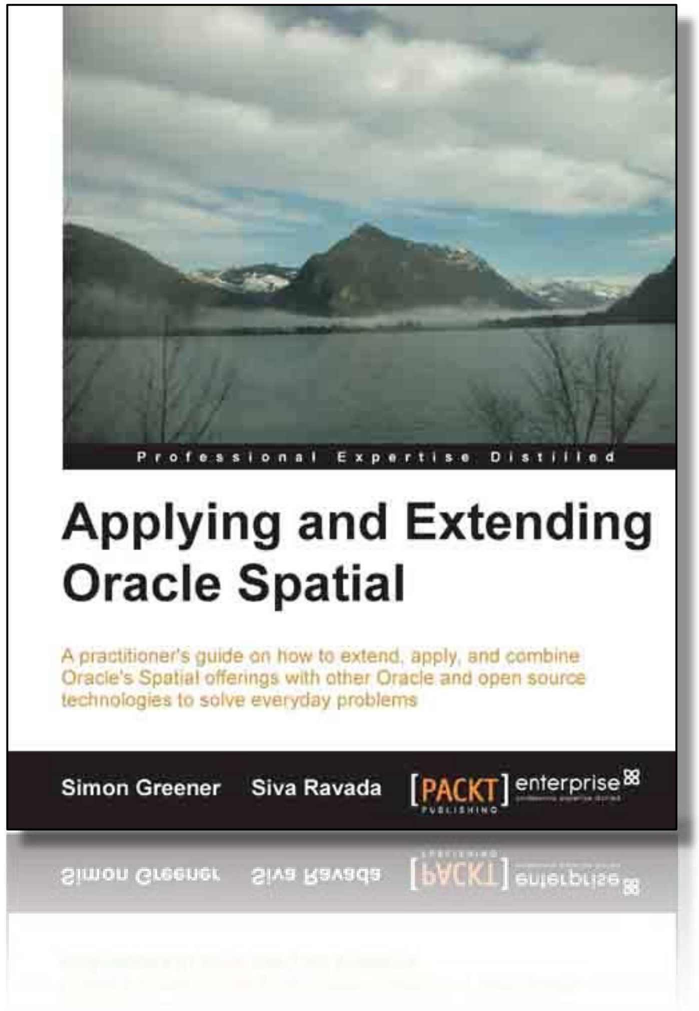
Many times I come across customers who ask for help tuning their SDO_NN queries.

I generally group SDO_NN queries into two categories.
The category defines which SDO_NN parameter to use, SDO_NUM_RES or SDO_BATCH_SIZ
Choose either SDO_NUM_RES or SDO_BATCH_SIZE, never both.

SDO_NUM_RES - Find the nearest candidates only with respect to proximity to the window g
There are no additional conditions in the WHERE clause.
SDO_BATCH_SIZE - Find the nearest candidates, and match additional conditions in the WHE
For example, find the 5 nearest banks with name = 'CITIBANK'

More Resources


THE EXPERT'S VOICE® IN ORACLE
Pro
Oracle Spatial
for Oracle Database 11g
Ravi Kothuri, Albert Godfrind, and Euro Beinat
Apress®


Professional Expertise Distilled
Applying and Extending Oracle Spatial
A practitioner's guide on how to extend, apply, and combine Oracle's Spatial offerings with other Oracle and open source technologies to solve everyday problems
Simon Greener
Siva Ravada
[PACKT] enterprise

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

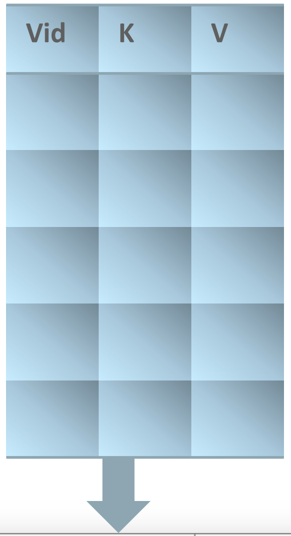
- Customized SQL query against <graph>VT$ and <graph>GE$
- Example: multiple step navigation with filters

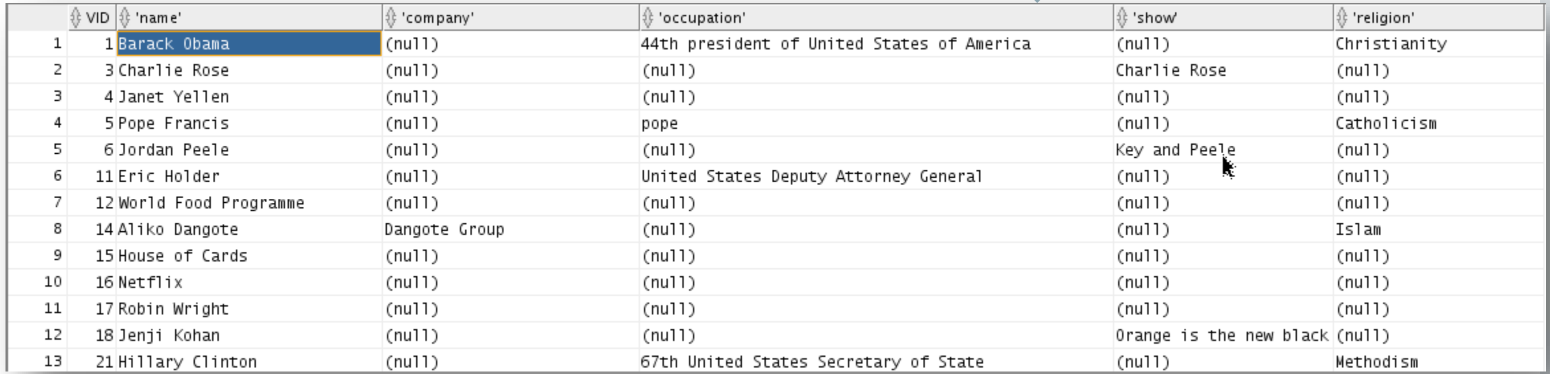
```
with G as (select svid, dvid
            from connectionsGE$
           where el in ('collaborates', 'admires')          -- filters: only interested in 'collaborates' and 'admires'
           )
select count(distinct g4.dvid)
 from G g1, G g2, G g3, G g4
where g1.svid = 1                                            --  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')
    )
   ;
```





	VID	'name'	'company'	'occupation'	'show'	'religion'
1	1	Barack Obama	(null)	44th president of United States of America	(null)	Christianity
2	3	Charlie Rose	(null)	(null)	Charlie Rose	(null)
3	4	Janet Yellen	(null)	(null)	(null)	(null)
4	5	Pope Francis	(null)	pope	(null)	Catholicism
5	6	Jordan Peele	(null)	(null)	Key and Peele	(null)
6	11	Eric Holder	(null)	United States Deputy Attorney General	(null)	(null)
7	12	World Food Programme	(null)	(null)	(null)	(null)
8	14	Aliko Dangote	Dangote Group	(null)	(null)	Islam
9	15	House of Cards	(null)	(null)	(null)	(null)
10	16	Netflix	(null)	(null)	(null)	(null)
11	17	Robin Wright	(null)	(null)	(null)	(null)
12	18	Jenji Kohan	(null)	(null)	Orange is the new black	(null)
13	21	Hillary Clinton	(null)	67th United States Secretary of State	(null)	Methodism

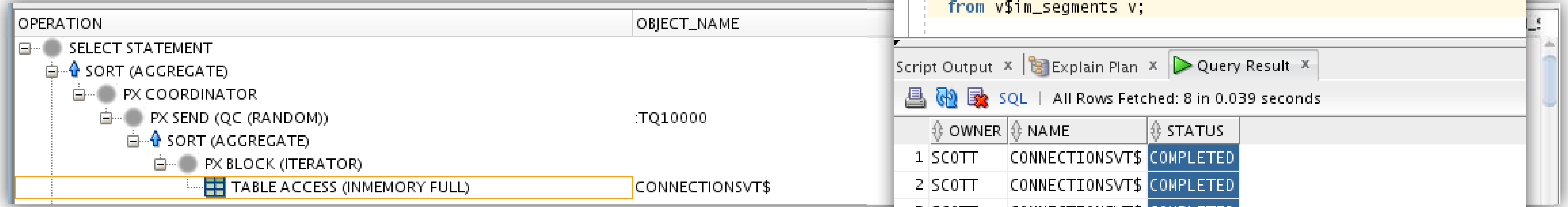
SQL-based Query for Property Graph (3)

- 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

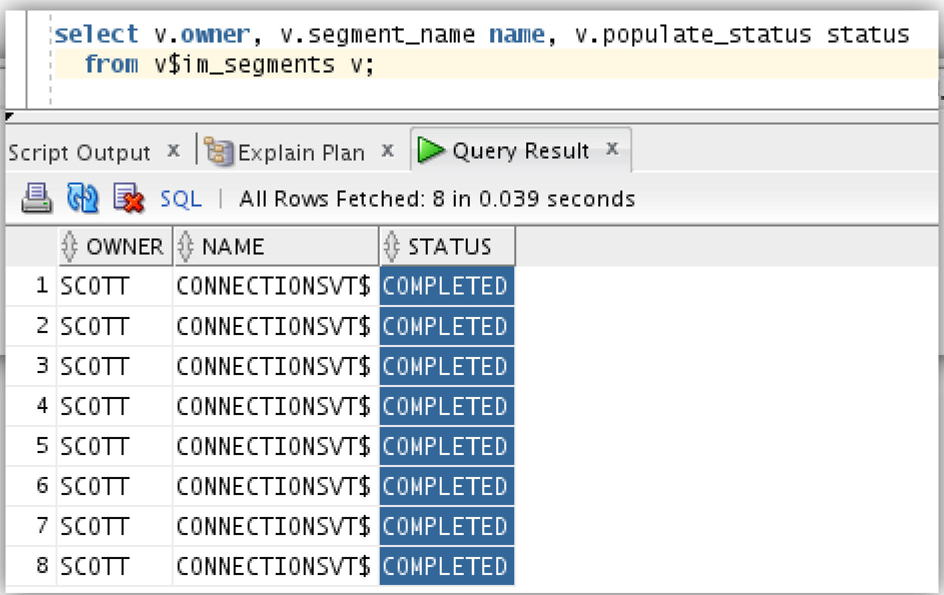
SQL-based Query for Property Graph (4)

- Customization example: Enable Database In-Memory

alter table connectionsVT$ inmemory;



OPERATION	OBJECT_NAME
SELECT STATEMENT	
SORT (AGGREGATE)	
PX COORDINATOR	
PX SEND (QC (RANDOM))	:TQ10000
SORT (AGGREGATE)	
PX BLOCK (ITERATOR)	
TABLE ACCESS (INMEMORY FULL)	CONNECTIONSVT$



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

All Rows Fetched: 8 in 0.039 seconds

	OWNER	NAME	STATUS
1	SCOTT	CONNECTIONSVT$	COMPLETED
2	SCOTT	CONNECTIONSVT$	COMPLETED
3	SCOTT	CONNECTIONSVT$	COMPLETED
4	SCOTT	CONNECTIONSVT$	COMPLETED
5	SCOTT	CONNECTIONSVT$	COMPLETED
6	SCOTT	CONNECTIONSVT$	COMPLETED
7	SCOTT	CONNECTIONSVT$	COMPLETED
8	SCOTT	CONNECTIONSVT$	COMPLETED

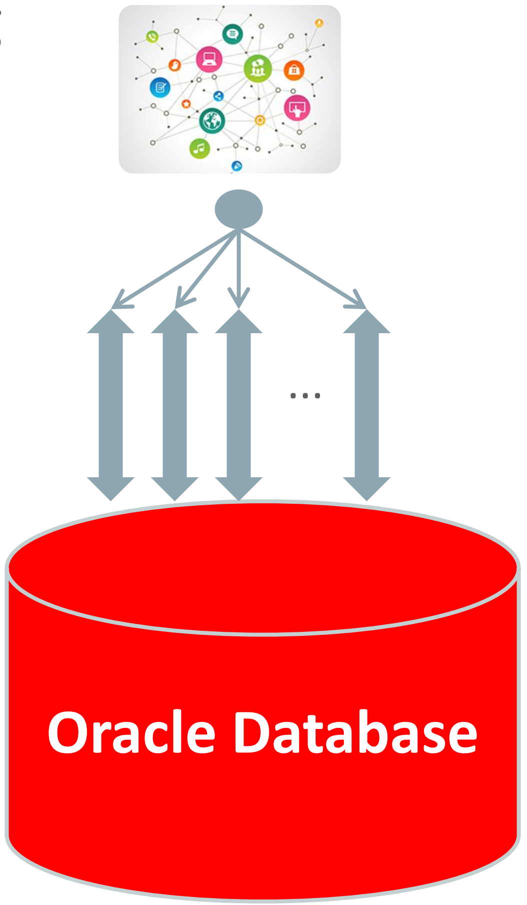
Key Features: SQL-based Query and Analytics for Oracle Database

- 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

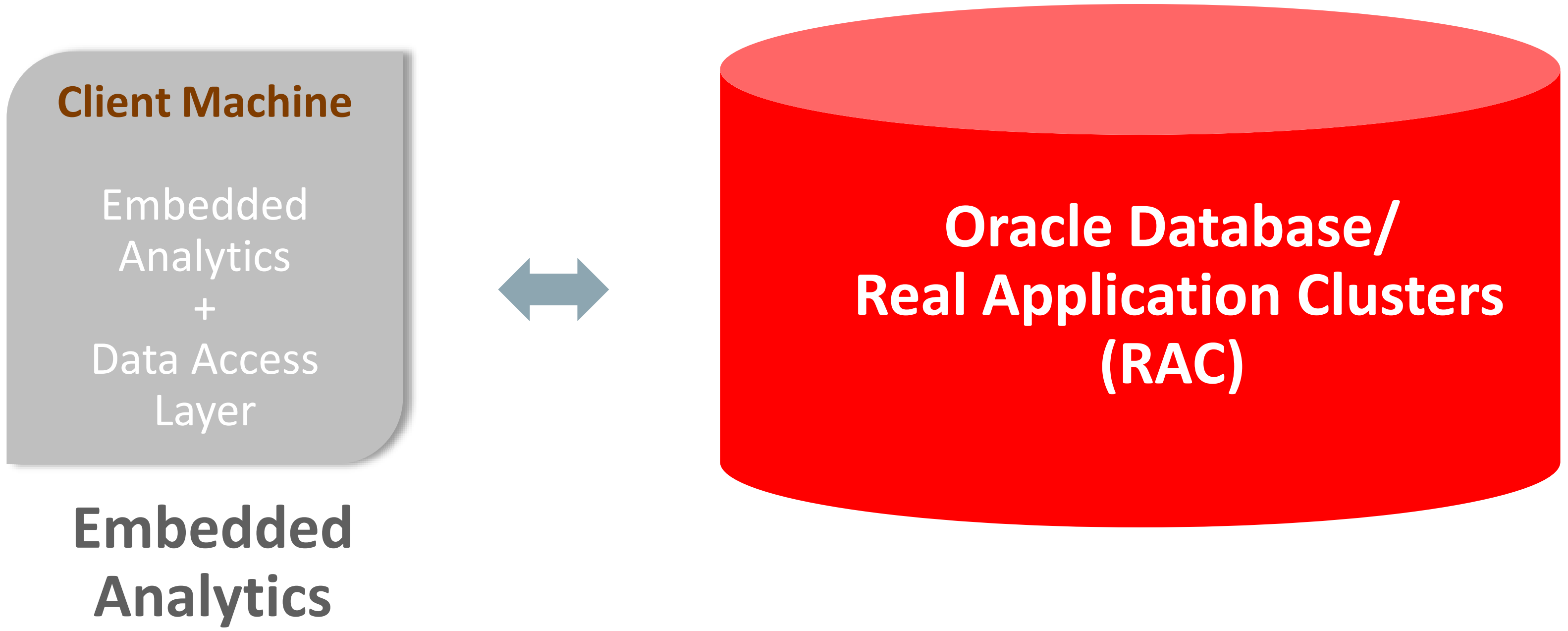
```
opgdl = OraclePropertyGraphDataLoader.getInstance();
vfile = "/home/data/pg-bda-nosql/demo/connections.opv";
efile = "/home/data/pg-bda-nosql/demo/connections.ope";
opgdl.loadData(opg, vfile, efile, 32, 1000, true, "PDML=T,PDDL=T,NO_DUP=T,");
```



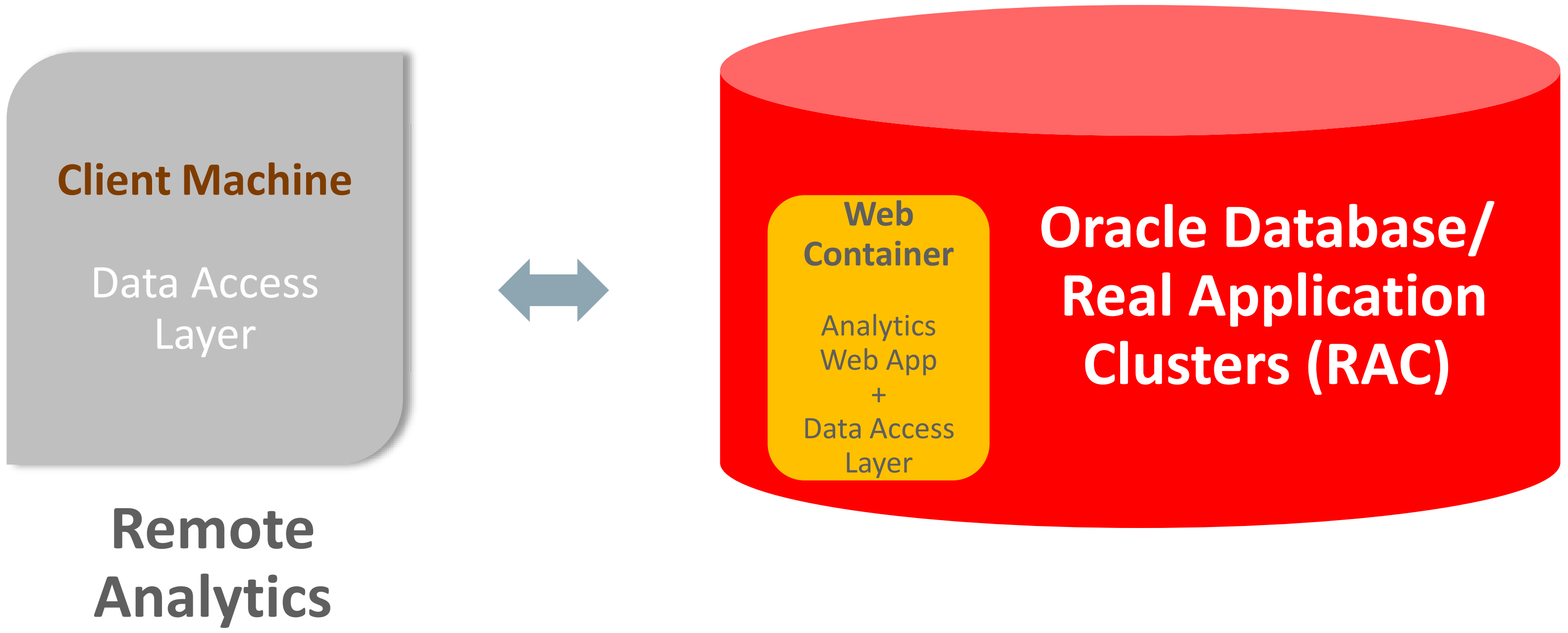
Key Features: Flexible Graph Analytics Deployment
- Shared memory mode

- 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

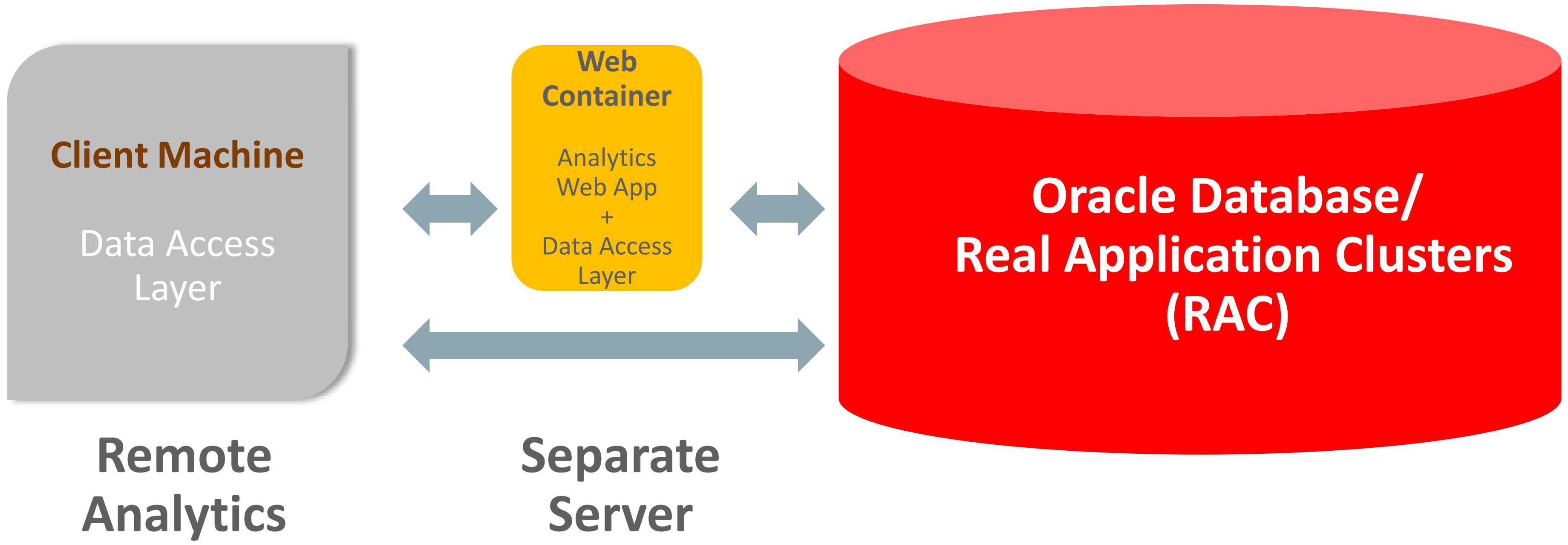
Key Features: Flexible Graph Analytics Deployment (2)


Client Machine
Embedded Analytics + Data Access Layer
Embedded Analytics
Oracle Database/ Real Application Clusters (RAC)

Key Features: Flexible Graph Analytics Deployment (3)


Client Machine
Data Access Layer
Remote Analytics
Web Container
Analytics Web App + Data Access Layer
Oracle Database/ Real Application Clusters (RAC)

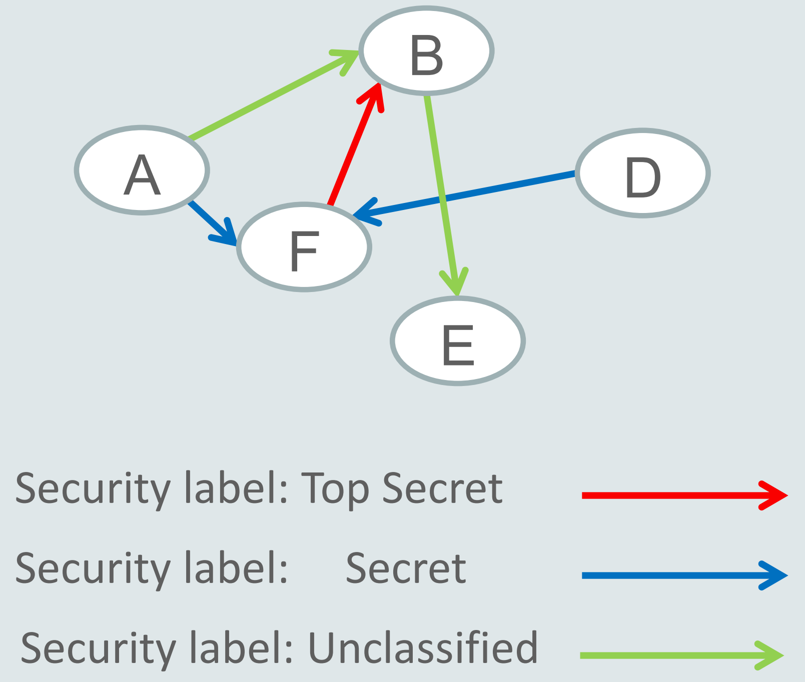
Key Features: Flexible Graph Analytics Deployment (4)


Client Machine
Data Access Layer
Web Container
Analytics Web App + Data Access Layer
Oracle Database/ Real Application Clusters (RAC)
Remote Analytics
Separate Server

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



Features: Change Tracking of Graph Elements

- Integration with Oracle Flashback Technology

  public interface OracleChangeTracker

  {

  – public Long getCurrentSCN() throws SQLException;
  – public Iterator<VertexChange> getVertexChanges(Long longStartSCN, Long longEndSCN)
  – public Iterator<VertexChange> getVertexChanges(Long longStartSCN, Long longEndSCN, int iQueryDOP)
  – public Iterator<EdgeChange> getEdgeChanges(Long longStartSCN, Long longEndSCN)
  – public Iterator<EdgeChange> getEdgeChanges(Long longStartSCN, Long longEndSCN, int iQueryDOP)

  }

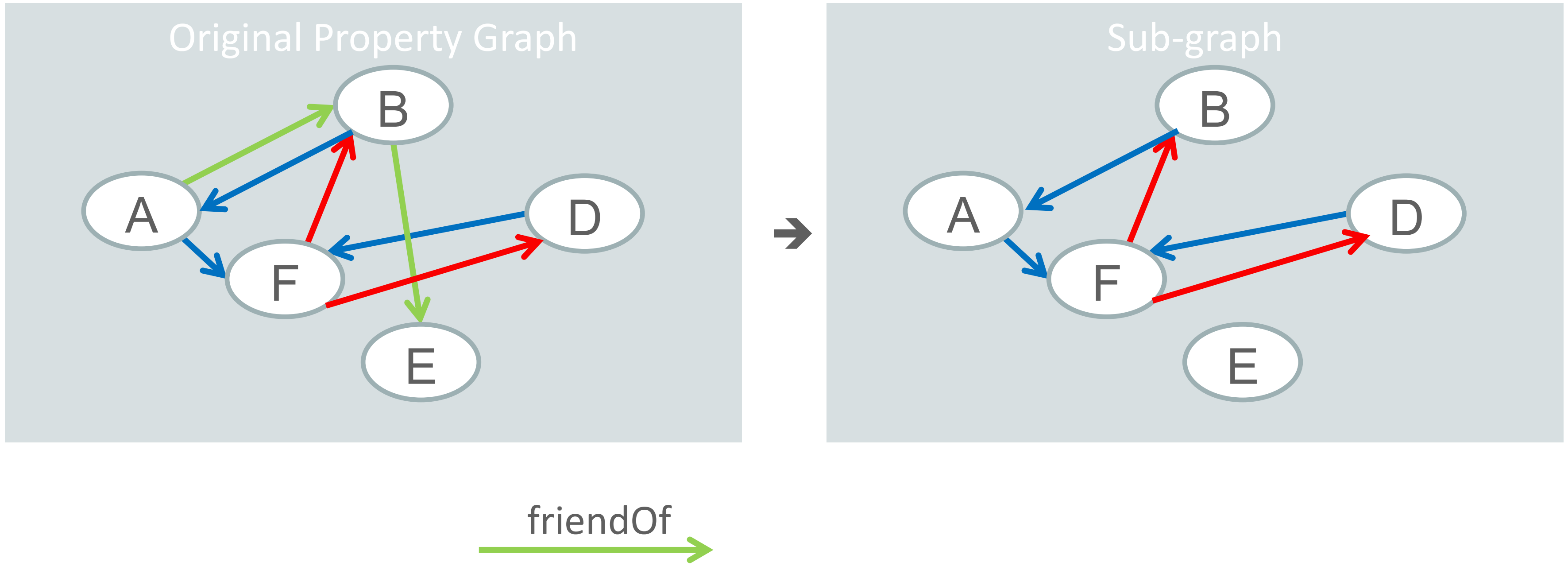
- Need to adjust UNDO_RETENTION

Sub-graph Extraction using SQL View

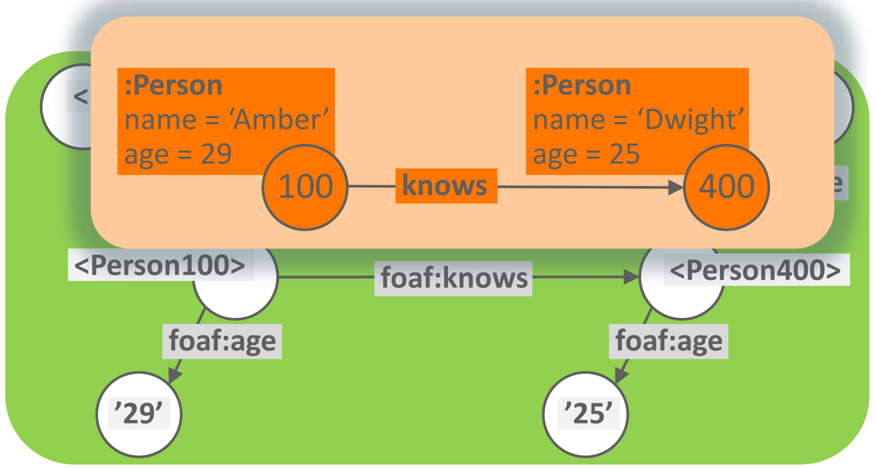
- Example: Create a sub-graph view with only edges having label “friendOf”

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

create view my_sub_graphGE$
as
   select *
     from my_graphGE$
    where el !=‘friendOf’
;
```



Property Graph View on RDF



- Motivation
  - Create a view on RDF so that property graph analytics can be applied
- Key method: **createPropertyGraphViewOnRDF**

```
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");
predsForVertexAttrs[2] = PGUtils.RDFPredicate.getInstance("http://oracle.com/nsg/followerCount", "followerCount");
predsForVertexAttrs[3] = PGUtils.RDFPredicate.getInstance("http://oracle.com/nsg/countryName", "countryName");
```

RDF Predicate URIs ➔ Property graph vertex attributes

```
predsForEdges = new PGUtils.RDFPredicate[2];
predsForEdges[0] = PGUtils.RDFPredicate.getInstance("http://oracle.com/nsg/refsPerson", "refsPerson");
predsForEdges[1] = PGUtils.RDFPredicate.getInstance("http://oracle.com/nsg/refsOrganization", "refsOrganization");
```

RDF Predicate URIs ➔ Property graph edges

```
PGUtils.createPropertyGraphViewOnRDF(oracle.getConnection(), "pgsocial", "social", false, predsForVertexAttrs,
                    predsForEdges);
```

PG view name

RDF Model name

ORACLE

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"