

# Combining Graph and Machine Learning Technology using R

Hassan Chafi  
Oracle Labs

Mark Hornick  
Oracle Advanced Analytics

February 2, 2017

**BIWA SUMMIT 2017  
WITH SPATIAL SUMMIT**

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



**ORACLE**

# Safe Harbor Statement

The following is intended to outline our research activities and 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.

# Contents

- Graph Analysis and Machine Learning
  - Graph Analysis and Applications
  - Combining Graph Analysis with Machine Learning
- OAA.Graph
  - ORE/ORAAH and PGX
  - Integration
- Demo

# Graph Analysis And Machine Learning

# Big Data and Data Analysis

- The Big Data era is here
  - Volume
  - Velocity
  - Variety
- However, just storing and managing this data is not sufficient
  - Typically Big Data is low value per byte
- ➔ We want to get useful information out of the huge data sets

## Methodologies:

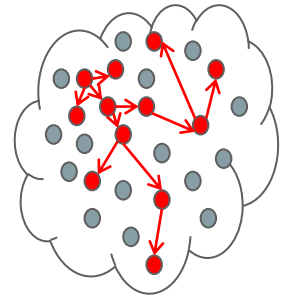
- Classic OLAP
- Statistical analysis
- Machine learning
- Graph analysis



# Graph Analysis

- A methodology in data analysis
- Represent your data as a graph
  - Data entities become nodes
  - Relationships become edges
- Analyze fine-grained relationships through the graph
  - Navigate multi-hop relationships quickly
  - Without computing expensive joins repeatedly

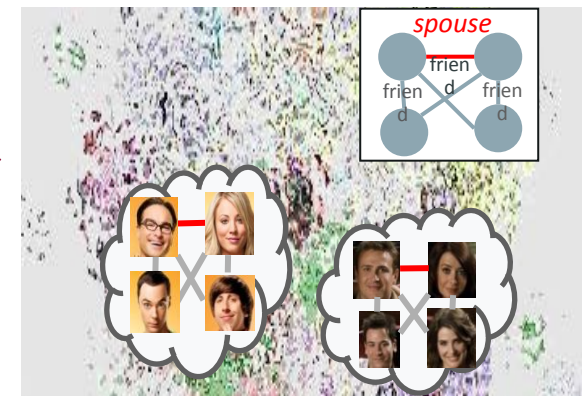
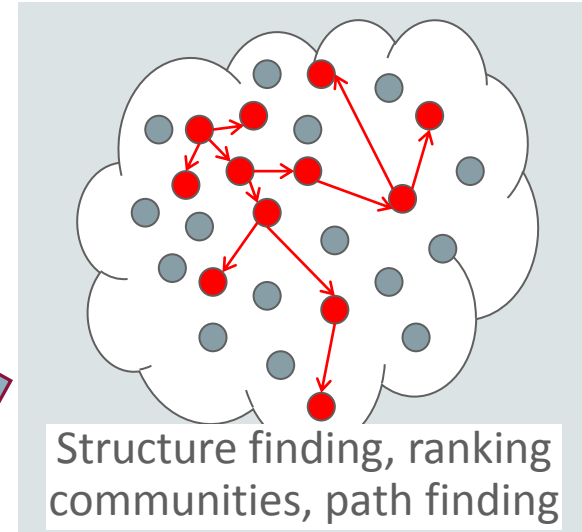
VIDEO_SALES_ORDERS		SALES_ORDER_LINE_ITEMS			VIDEO_PRODUCTS	
SALES_ID	CUST_NAME	SALES_ID	LINE_ID	PROD_ID	PROD_ID	PROD_DESC
10	SMITH	10	1	1000	1000	TOY STORY
20	JONES	10	2	3000	2000	TRUE LIES
30	TURNER	20	1	4000	3000	POPCORN
40	ADAMS	20	2	3000	4000	STARGATE
		20	3	2000		
		30	1	1000		
		30	2	1000		
		40	1	4000		



# Graph Analysis

Inter-relationships between data and networks are growing in importance

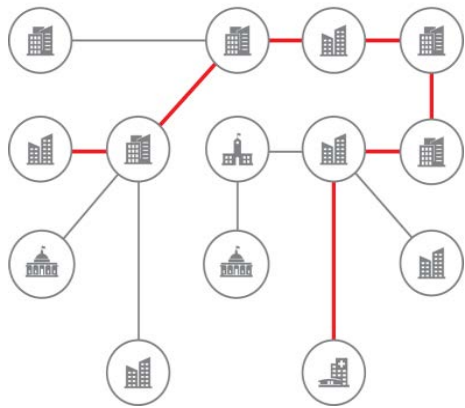
- Graphs are everywhere
  - Facebook (friends of friends), Twitter, LinkedIn, etc.
    - Most data has inter-relationships that contain insights
- Two major types of graph algorithms
  - Computational Graph Analytics: Analysis of entire Graph
    - Influencer ID, community detect, pattern machine, recommendations
  - Graph Pattern Matching
    - Queries that find sub-graphs fitting relationship patterns



# Graph Analysis Examples

## Reachability Analysis

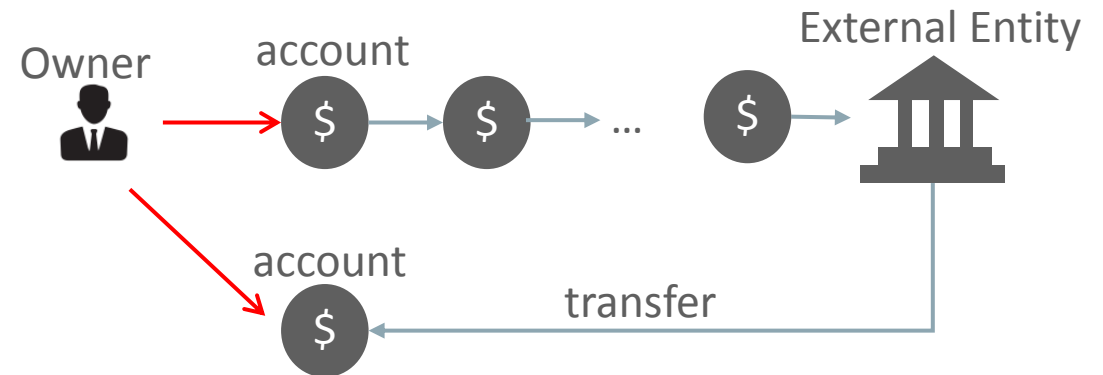
Find out how data entities are connected with each other via multiple hops



- Example Application

- Money laundering pattern detection in bank applications
- Identify a chain of wire transfers, including an external entity, between two accounts of a single owner

➔ Graph pattern matching with cycle detection





# Graph Analysis Examples

## Centrality Analysis

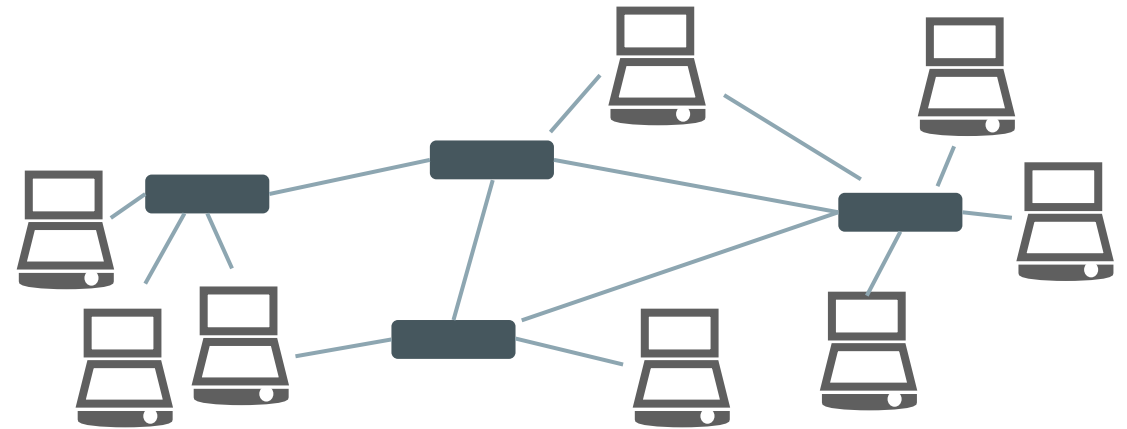
Identifying important entities from connections between data entities



- Example Application

- Computer network vulnerability analysis
- Identify network components whose failure would cause the largest damage

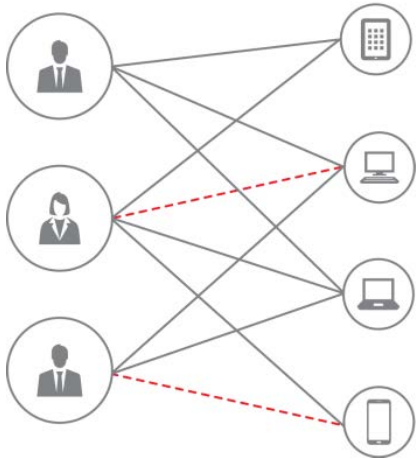
➔ *Betweenness Centrality* Computation



# Graph Analysis Examples

## Link Prediction

Identifying pairs of entities that are likely to have connections in future, due to their closeness or similarity



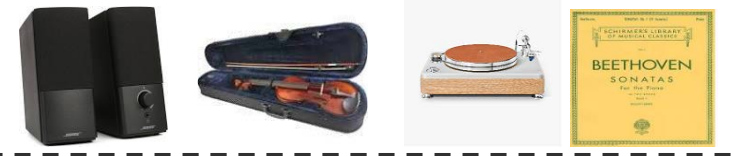
## • Example Application

- Product recommendation for retail
  - Given an item, identify *close* items from user-item or item-feature graph and recommend those items
  - Given a user, identify *close* users who purchased similar items and recommend items popular among those
- ➔ Matrix (Graph) Factorization, Personalized Pagerank, ...

Items similar to this:



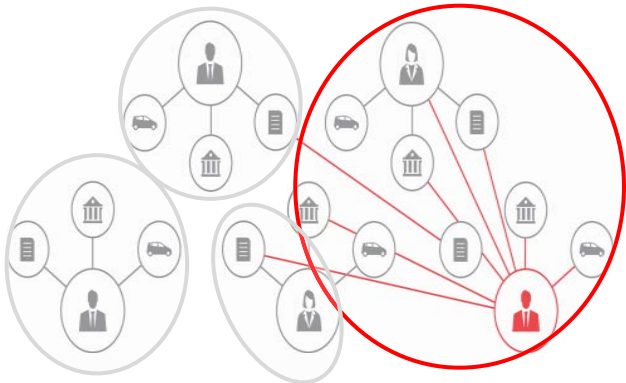
Other people also liked:



# Graph Analysis Examples

## Community Analysis

Identify grouping of data entities from their interconnection structure



- Example Application

- Classification of data entities based on their relationship
- E.g. classify students from the same department by the courses that they take

➔ Label propagation, Relax Map, ...

## Classification Result – For students of ‘Math/CS department’

- Courses taken by community A

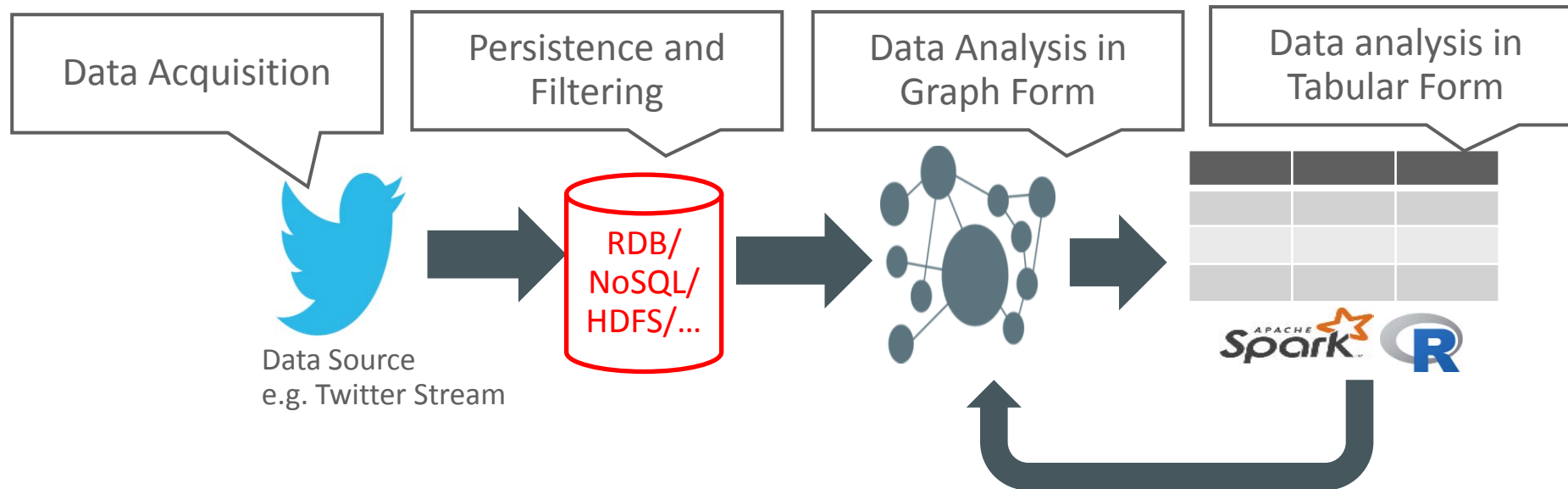
course_department	course_title	count
Dept of Math/CS	Data Structures	114
Dept of Math/CS	Computer Organization	104
Dept of Math/CS	Principles of Programming II	102
Dept of Math/CS	Database Design I	91
Dept of Math/CS	Operating Systems	89

- Courses taken by community B

course_department	course_title	count
Dept of Math/CS	Ordinary Differential Equation	72
Dept of Math/CS	Set Theory	71
Dept of Math/CS	Probability and Statistics	65
Dept of Math/CS	Linear Algebra	55
Dept of Math/CS	Modern Algebra I	53

# Graph Analysis and Other Data Analyses

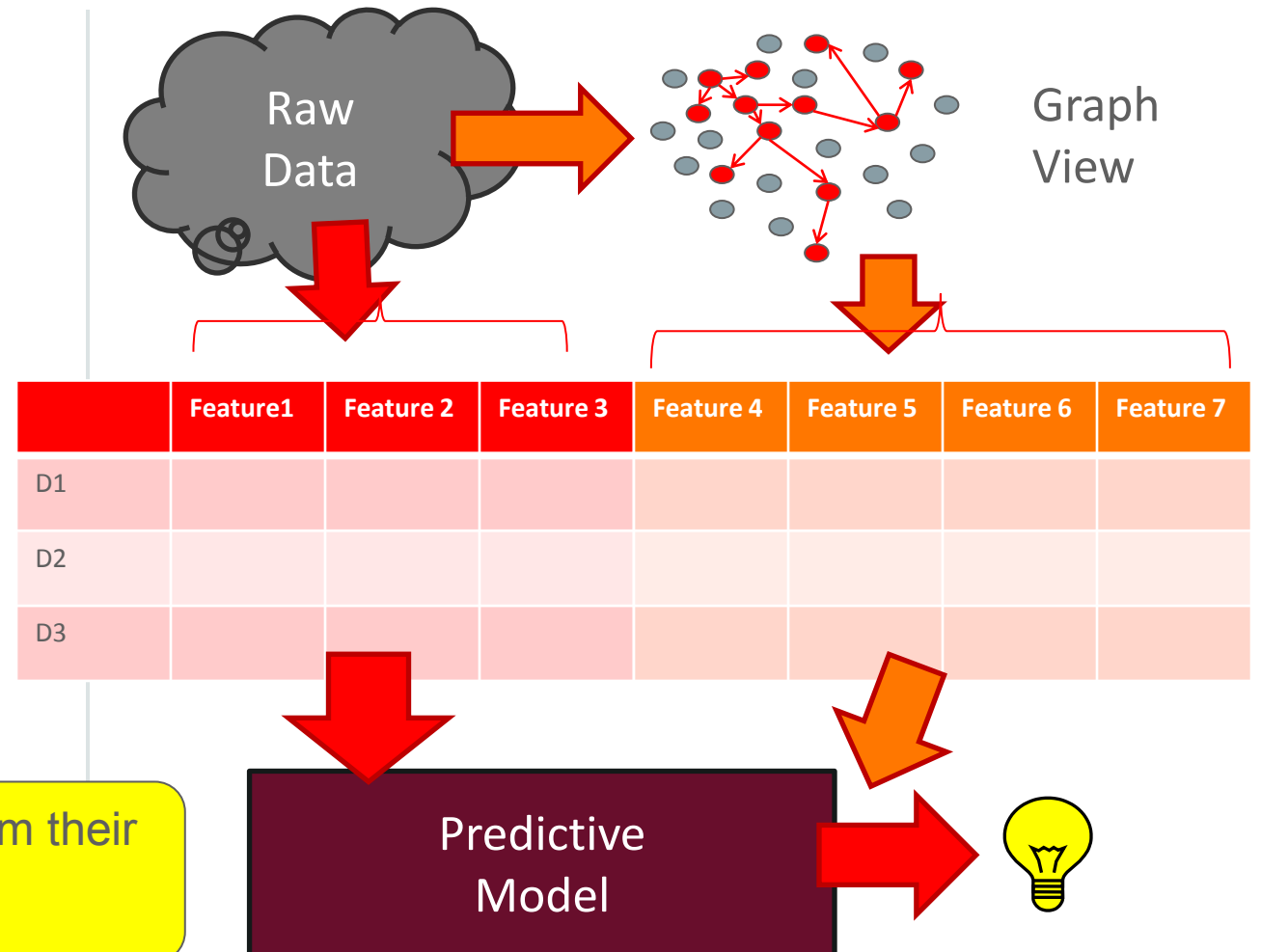
- Naturally, graph analysis pairs well with other data analyses
  - Traditional analysis steps favors tabular data representation
  - Graph analysis can occur as a separate data processing step



# Graph Analysis and Machine Learning

- Graph analysis can augment Machine Learning
  - Typical machine learning techniques create/train models based on observed features
  - Graph analysis can provide additional *strong* signals
  - That make predictions more accurate

e.g. Can you identify groups of close customers from their call graph in order to predict customer churn?



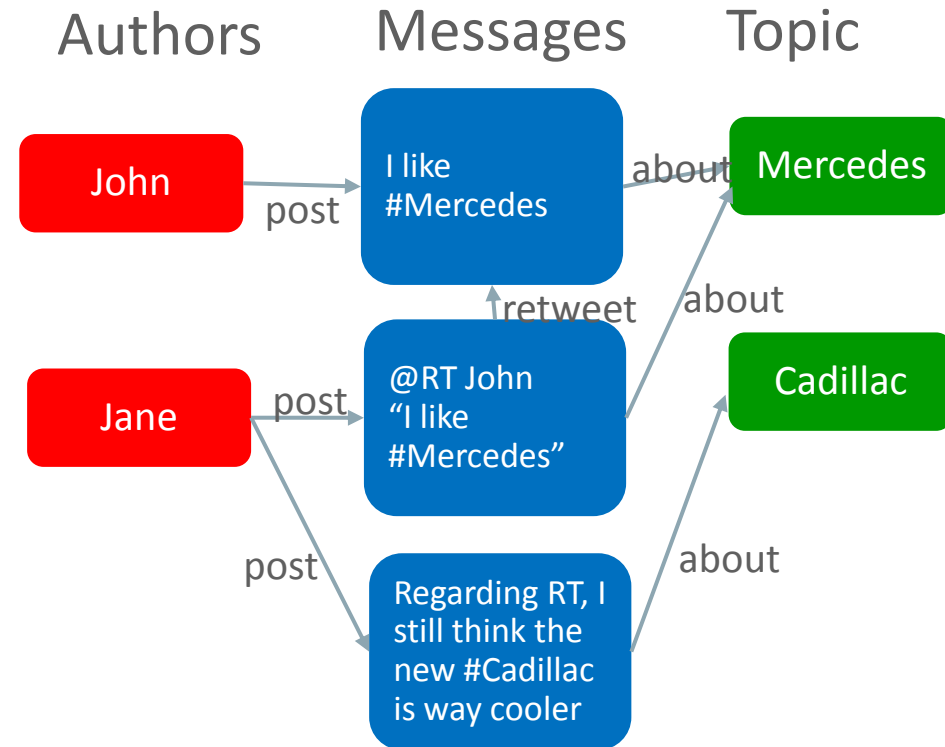
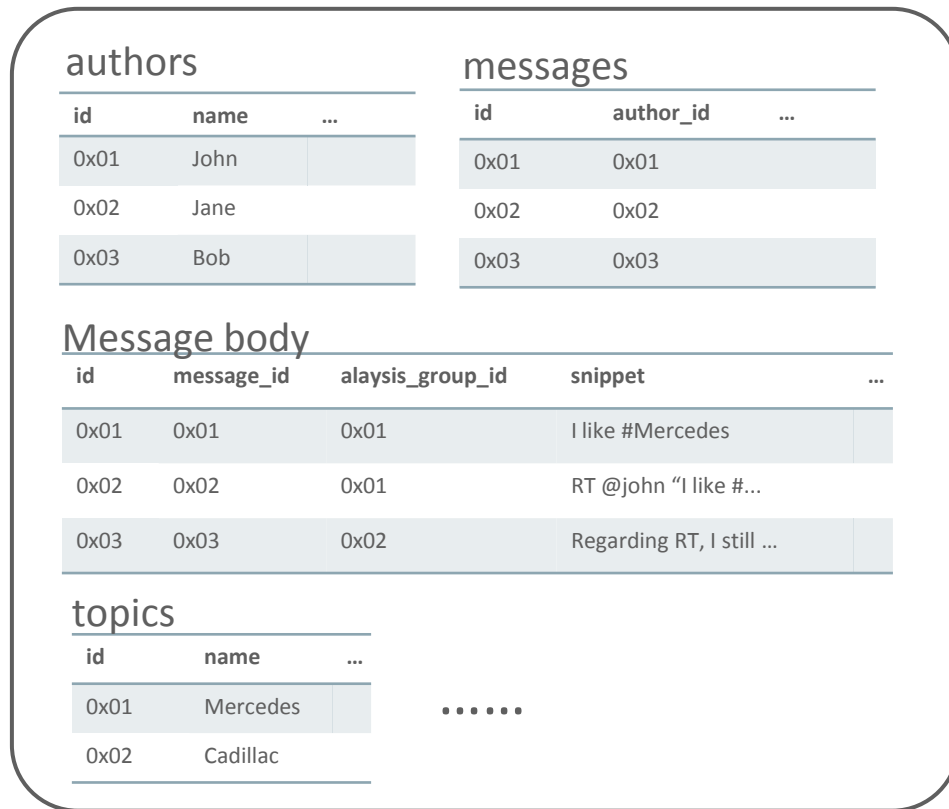
# Example – SNS Stream analysis

- Twitter streams can often be manipulated to achieve some goal
  - *Social and viral marketing (or alternative fact based news)*
  - True view on trends can be polluted by these streams
  - How can we eliminate such *noise*?

The image shows a Twitter profile for 'Stylish Rentals' (@StylishRentals) on the left, with a bio that reads: 'Looking for a vacation rental anywhere in the world? We tweet the best condo, oceanfront and rental homes available (trends.com/stylishrentals)'. The profile statistics show 12.3K tweets, 56.3K following, 53.9K followers, and 116 likes. A dark grey callout box at the bottom of the profile says: 'These are bots to make other accounts look more important'. To the right, a screenshot of a Twitter stream shows several tweets. A yellow callout box at the top of the stream says: 'Different accounts re-tweet messages (mostly) from certain accounts'. Red arrows point from this box to several tweets in the stream, which have a small 'SR' logo in their avatars. The tweets include advertisements for 'Waller Jewelry', 'Smash Fund', 'Ummerson Fitness', and '1,000 REAL Human Followers'.

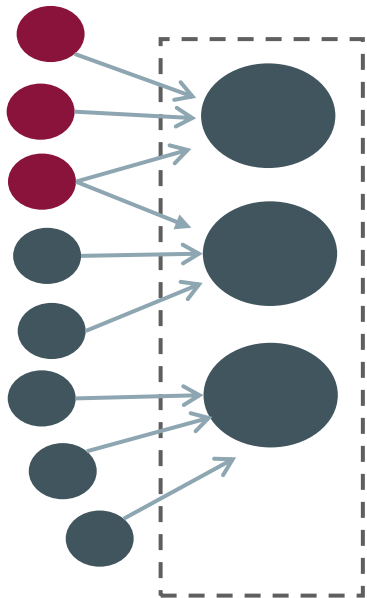
# Data Processing Steps: Creating Graph

- Create graph representation from tables data set

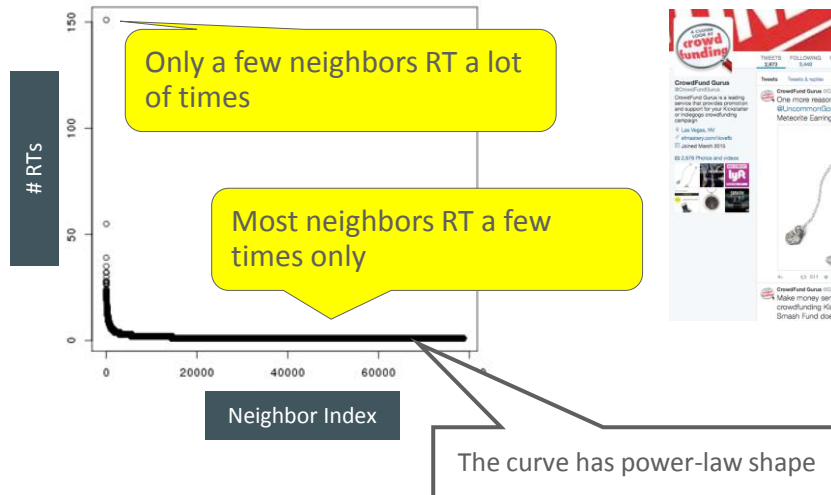


# Data Processing Steps: Analyzing Graph

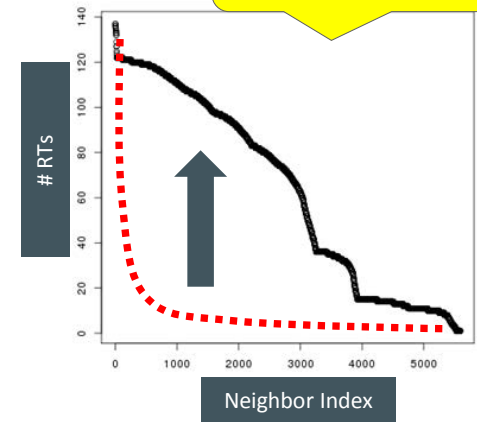
- Analyze Graph
  - Extract Retweet (RT) between accounts
  - Focus only on Total RT counts between accounts



A natural pattern



An unnatural pattern

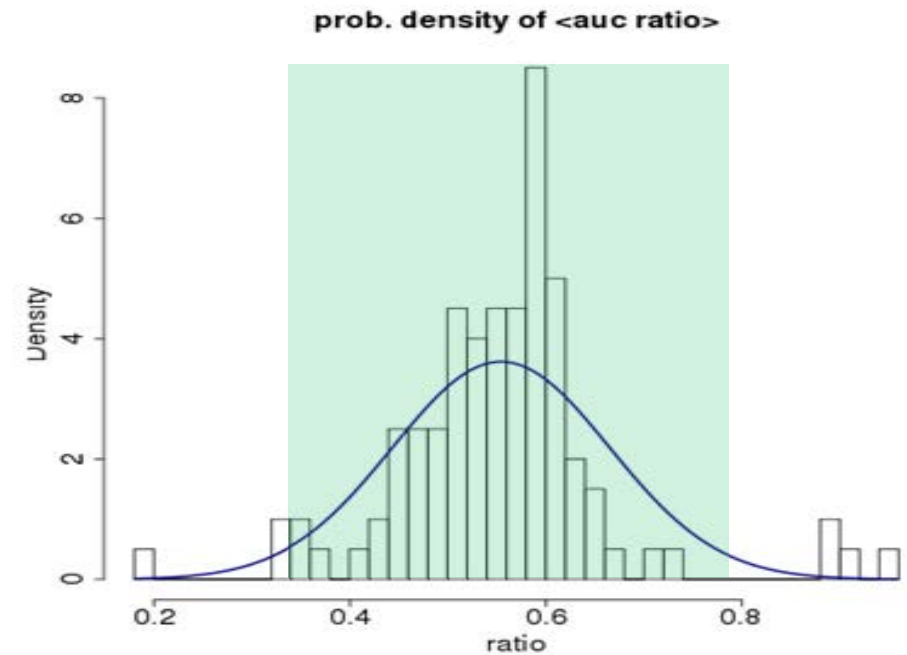
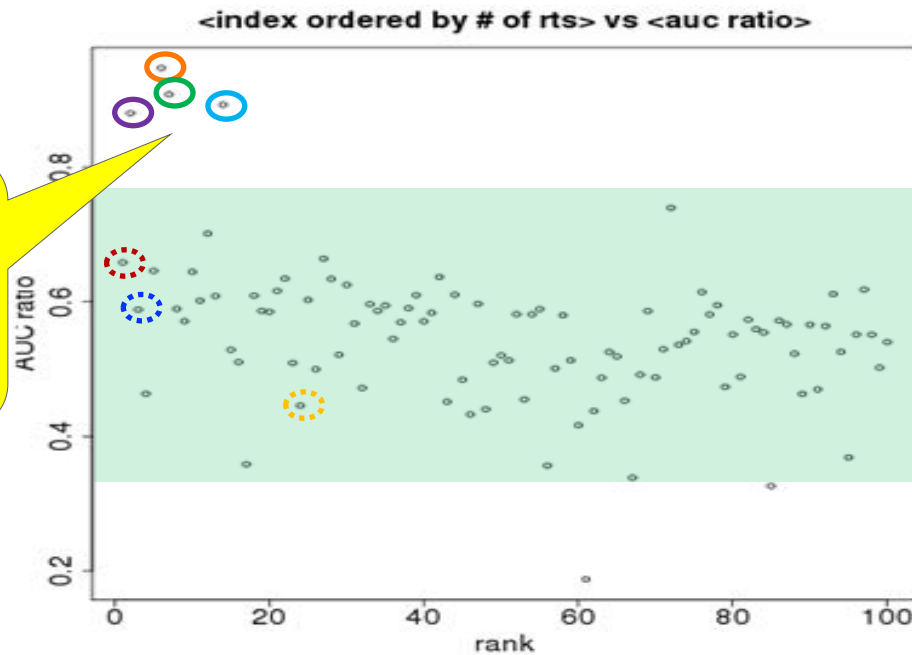




# Data Processing Steps: Statistical Analysis

- Compute objective function for top-100 RT'ed accounts
- Identify anomalies from simple statistical methods

Certain accounts have obviously unnatural deviation



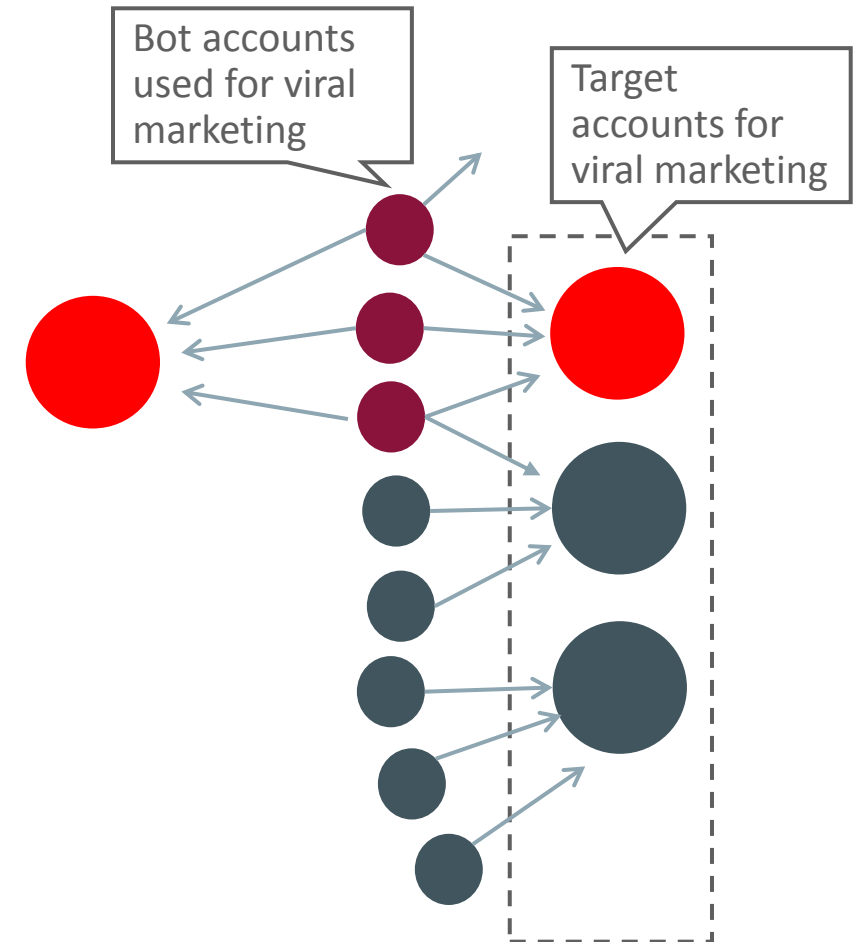
- crowdFundGurus
- carCrashesTV
- LoftyFollows
- StylishRentals

- realDonalTrump
- HillaryClinton
- BernieSanders
- $[\mu - 2\sigma, \mu + 2\sigma]$



# Data Processing Steps: Analyzing Graph

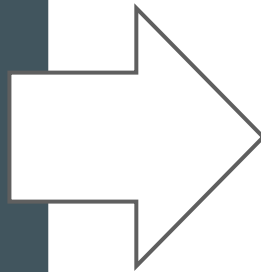
- Resume from previous step's results
- Already identified targets (among top RT'ed accounts) and bots
- Analyze graph even further to identify more target accounts



# Analysis Result

## Data Set

- Data acquisition period: 2016-Feb (1 week)
- Number of topics: 675
- Number of messages: 2.6 million
- Number of accounts: 788,360



- # RT Bots: 3,092
- # RT Beneficiaries: 5
- # Removed messages: 551,177

→ Significant changes in important accounts and trends

# What is PGX (part of BDSG)?

OTN Technology Preview

- PGX (Parallel Graph Analytics)
  - An in-memory graph analysis engine
  - Originated from Oracle Labs
  - Provides fast, parallel graph analysis
    - Built-in Algorithm Packages
    - Graph Query (Pattern-Matching)
    - Custom Algorithm Compilation (Advanced Use case)
  - Integrated with Oracle Product(s)
    - Oracle Big Data Spatial and Graph (with BDA)
    - Property Graph Support at RDBMS 12.2c (Planned)
  - 35+ graph algorithms
  - Exceeds open source tool capabilities

Welcome Sungpack  
Account Sign Out Help Country Communities I am a... I want to...  
Products Solutions Downloads Store Support

Oracle Technology Network > Oracle Labs > Parallel Graph Analytics > Overview

Parallel Graph Analytics  
Programming Languages and Runtimes

Overview Downloads Documentation Community Learn More

### Welcome to Parallel Graph Analytics (PGX)

**What is PGX?**  
PGX is a fast, parallel, in-memory graph analytic framework. Using PGX, the user can load graphs into main-memory, run various graph algorithms on them very efficiently, and export them back into the file system.

**What can I do with PGX?**

- Loading graphs into memory: PGX is an in-memory graph analytic frame

## Oracle Big Data and Spatial and Graph

Welcome Sungpack  
Account Sign Out Help Country Communities I am a... I want to... Search

Products Solutions Downloads Store Support Training Partner

Oracle Technology Network > Database > Database Technology Index > Big Data Spatial and Graph > Overview

Database 12c  
Database In-Memory  
Multitenant  
Options  
Application Development  
Big Data Appliance  
Data Warehousing & Big Data  
Database Appliance  
Database Cloud  
Exadata Database Machine  
High Availability  
Manageability  
Migrations  
Security

Overview Downloads Documentation Community Learn More

## Oracle Big Data Spatial and Graph

Spatial and graph analytic services and data models that support Big Data workloads on Apache Hadoop and NoSQL database technologies.

# PGX Graph Algorithms

- Ranking
  - Pagerank (+ variants)
  - Vertex Betweenness Centrality (including approximations)
  - Closeness Centrality
  - Eigenvector Centrality
  - Degree Centrality
  - Hyperlink-Induced Topic Search (HITS)
- Path Finding
  - Dijkstra (+ variants)
  - Bellman Ford (+ variants)
  - Hop Distance (+ variants)
  - Fattest path
- Partitioning
  - Weakly and Strongly Connected Components
  - Conductance and Modularity
  - Community Detection
- Recommendation
  - Twitter's whom-to-follow
  - Matrix Factorization
- Other
  - Breadth First Search with filter
  - Triangle Counting
  - Degree Distribution
  - K-core
  - Adamic Adar

# PGX Performance (Algorithm Computation)

- Comparisons against existing graph engines

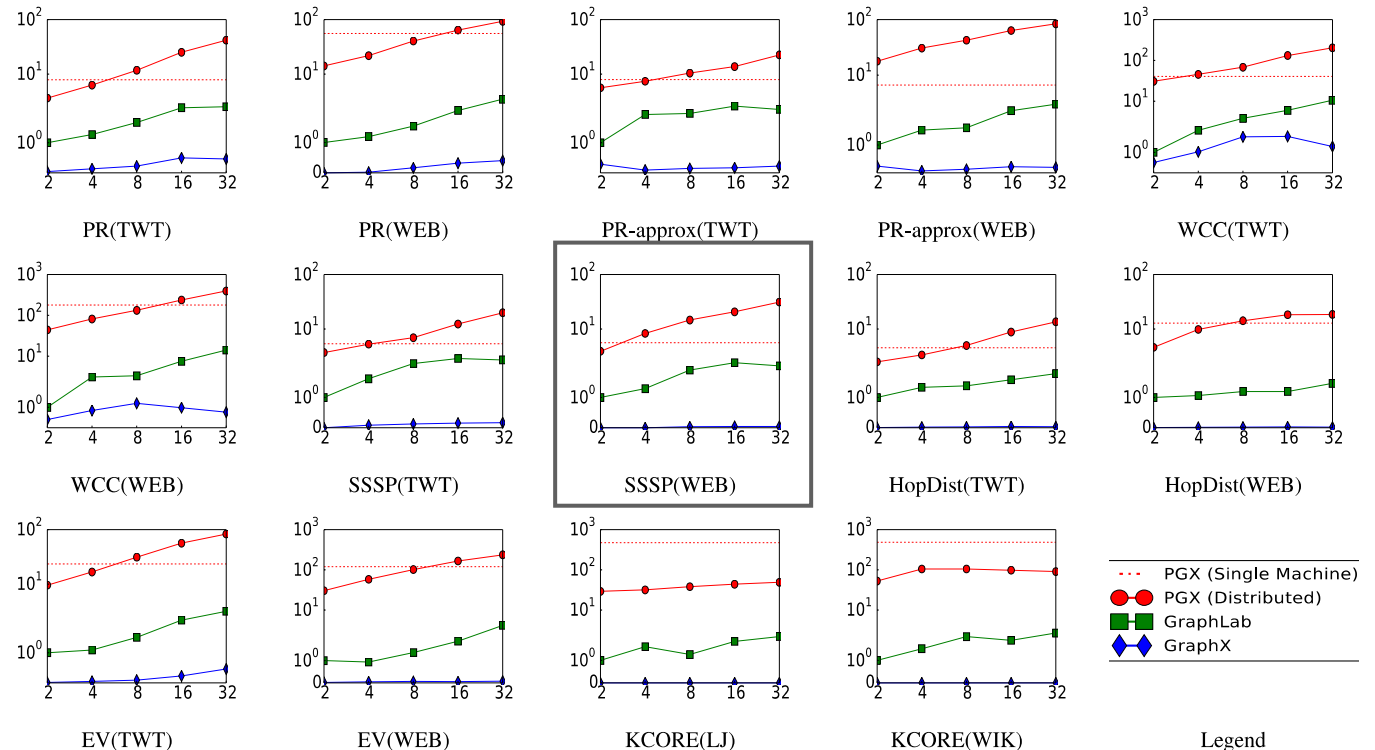
- GraphX (Spark)
- GraphLab (Dato)

- With seven popular algorithms

- Pagerank (exact and approx), Weakly Connected Components, Single-Source Shortest Path, Hop-Distance (BFS), Eigen Vector, K-Core

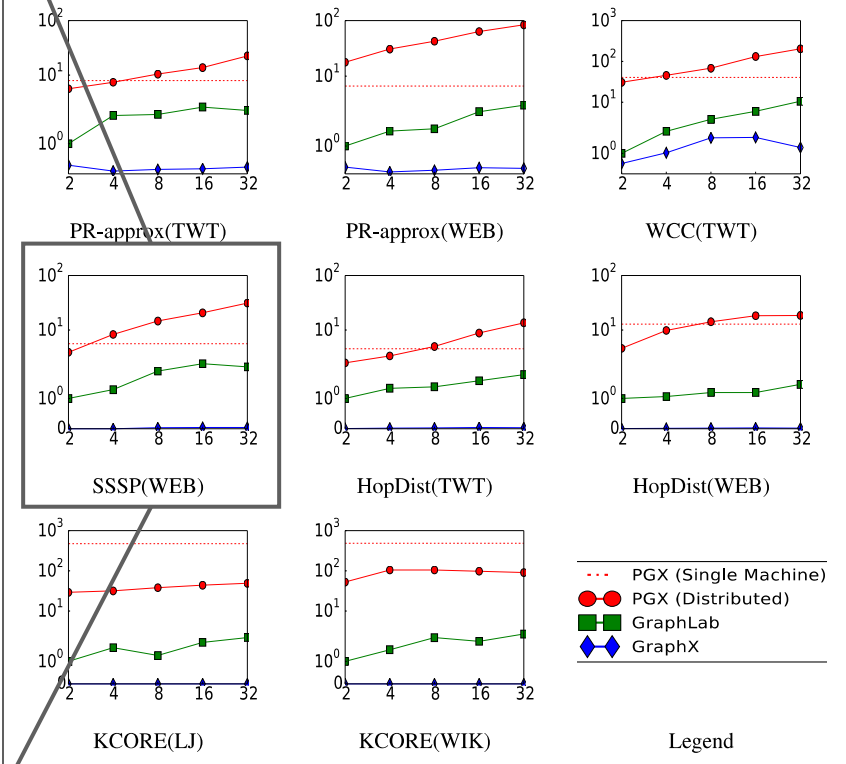
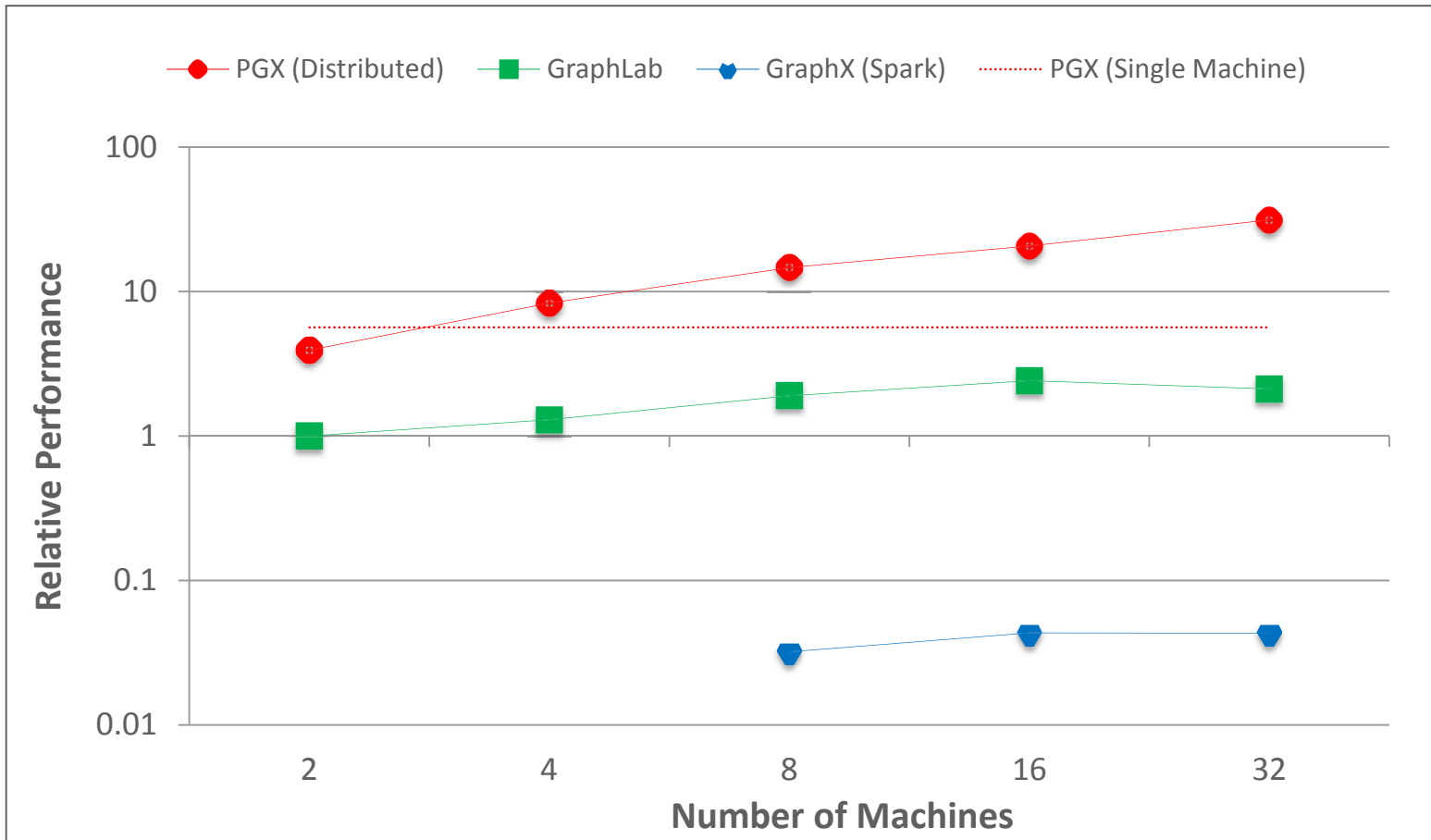
- On Two Graph instances

- Twitter Graph (TWT): 41 million nodes, 1.4 Billion edges
- Web Graph (WEB): 77 millions nodes, 2.9 Billion edges



Hardware: Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz - 256 RAM  
 Network: Mellanox Infiniband (56Gbps)

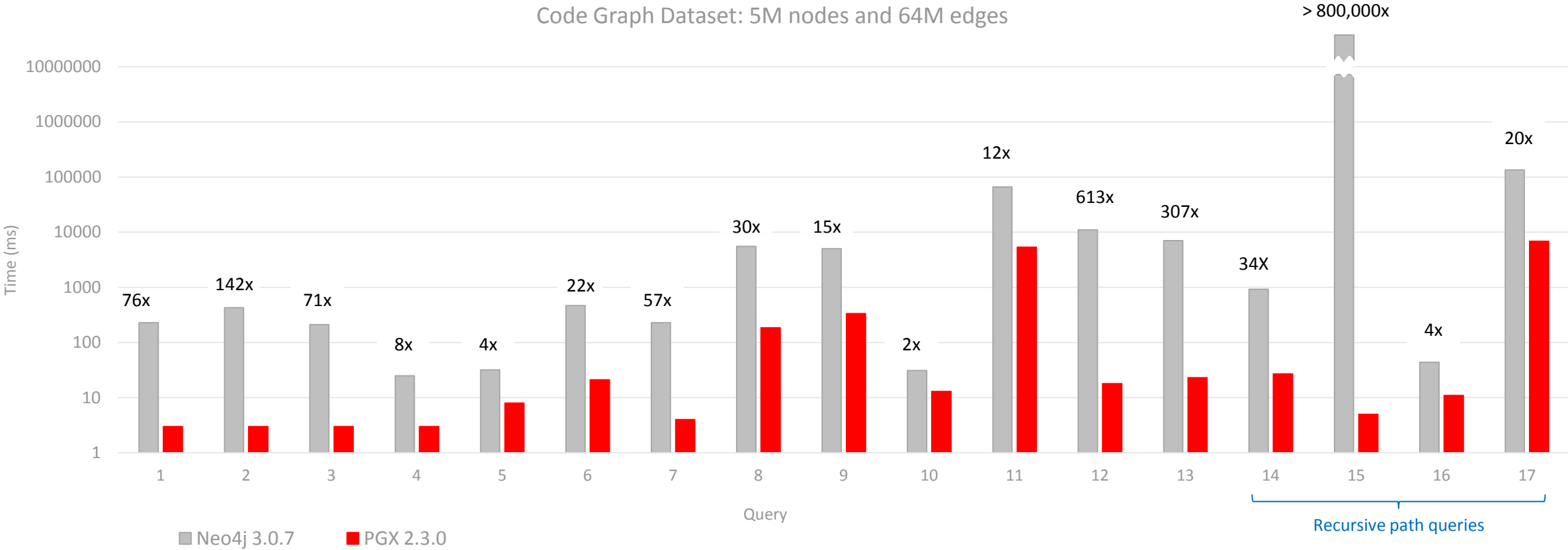
# PGX Performance (Algorithm Computation)



Hardware: Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz - 256 RAM  
 Network: Mellanox Infiniband (56Gbps)



# PGX Performance (Query) vs. Neo4j



Numbers shown are "hot" numbers:  
Neo4j is no longer accessing disk.

Hardware: 88 x Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz - 256 RAM  
 Warmup: ignore the first two runs, measure the third run  
 Index usage: no special indexes (use defaults)





# PGX (Single Node) Performance on SPARC

## SPARC M7 up to 1.5x faster per core than x86

Graph Algorithm	Workload Size	4-chip X86 E5 v3	4-chip SPARC T7-4	SPARC per chip Advantage	SPARC per core Advantage
SSSP Bellman-Ford	448M vertices, 17.2B edges	39.2s	14.7s	2.7x	1.5x
	233M vertices, 8.6B edges	21.3s	8.5s	2.5x	1.4x
PageRank	448M vertices, 17.2B edges	136.7s	62.6s	2.2x	1.2x
	233M vertices, 8.6B edges	72.1s	27.6s	2.6x	1.5x

- Graph computations accelerated by SPARC's memory bandwidth
  - Bellman-Ford/SSSP (single-source shortest path) – optimal route or connection
  - PageRank - measuring website importance

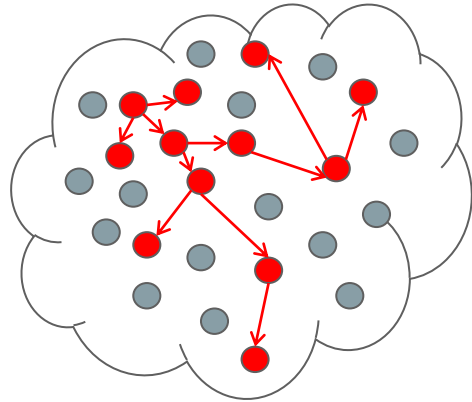
# OAAgraph

**An R interface integrating PGX and ORE/ORAAH  
for Machine Learning**

# Why an R interface to Graph?

- Single, unified interface across complementary technologies
  - Work with R data.frames and convenient functions across ML and graph
  - Results returned as R data.frames allows further processing in R env
- R users take advantage of multiple, powerful technologies
  - Highly scalable PGX engine on both Oracle Database and Hadoop
  - Integrated with **Oracle R Enterprise**, part of Oracle Database Advanced Analytics option
  - Integrated with **Oracle R Advanced Analytics for Hadoop**, part of Oracle Big Data Connectors

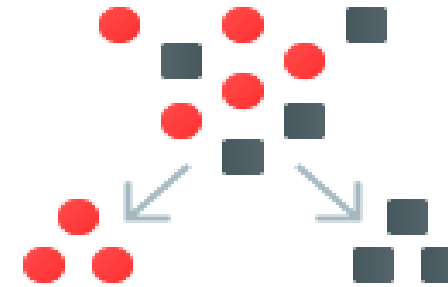
# Graph Analytics



*Compute graph metric(s)*

Add to  
structured data

# Machine Learning



*Build predictive model  
using graph metric*

*Explore graph or compute  
new metrics using ML result*

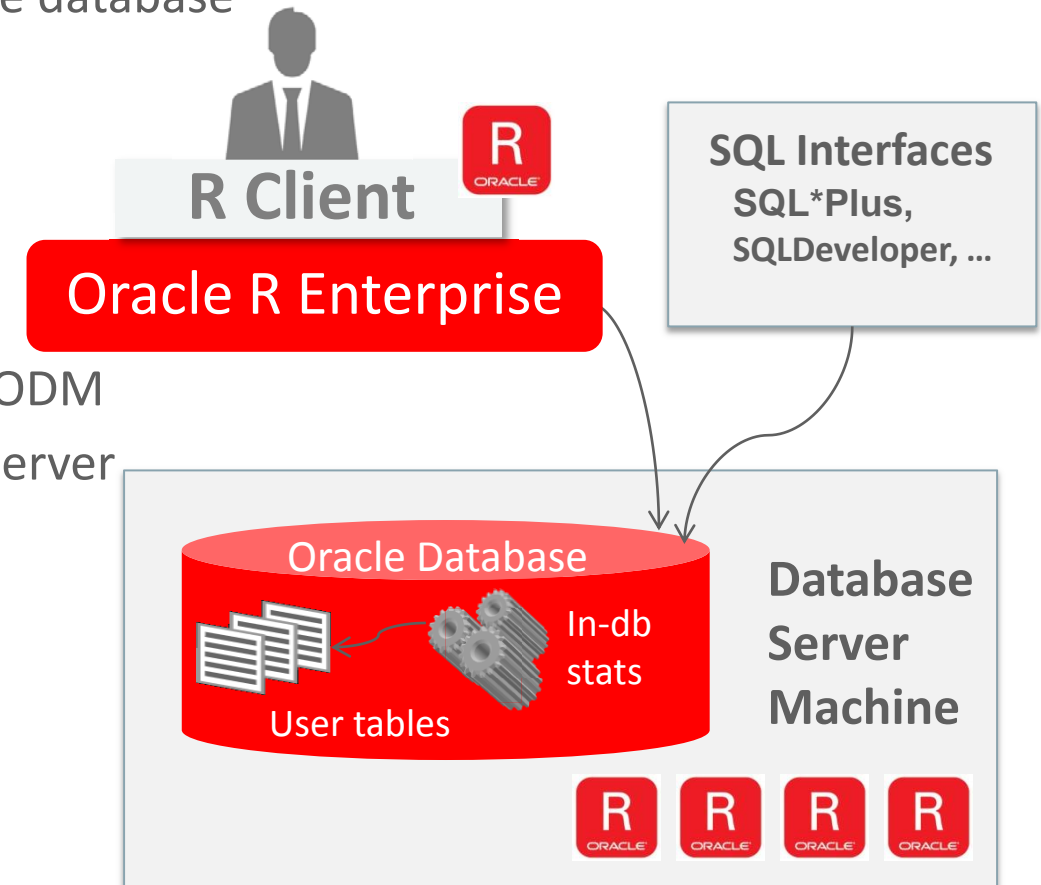
Add to graph

*Build model(s) and  
score or classify data*

# Oracle R Enterprise



- Use Oracle Database as a high performance compute environment
- Transparency layer
  - Leverage proxy objects (ore.frames) - data remains in the database
  - Overload R functions that translate functionality to SQL
  - Use standard R syntax to manipulate database data
- Parallel, distributed algorithms
  - Scalability and performance
  - Exposes in-database machine learning algorithms from ODM
  - Additional R-based algorithms executing and database server
- Embedded R execution
  - Store and invoke R scripts in Oracle Database
  - Data-parallel, task-parallel, and non-parallel execution
  - Use open source CRAN packages



# OAA / Oracle R Enterprise 1.5

## Machine Learning algorithms in-Database

*...plus open source R packages for algorithms in combination with embedded R data- and task-parallel execution*

### Classification

- Decision Tree
- Logistic Regression
- Naïve Bayes
- Support Vector Machine
- RandomForest

### Clustering

- Hierarchical k-Means
- Orthogonal Partitioning Clustering

### Market Basket Analysis

- Apriori – Association Rules

### Regression

- Linear Model
- Generalized Linear Model
- Multi-Layer Neural Networks
- Stepwise Linear Regression
- Support Vector Machine

### Attribute Importance

- Minimum Description Length

### Feature Extraction

- Nonnegative Matrix Factorization
- Principal Component Analysis
- Singular Value Decomposition

### Anomaly Detection

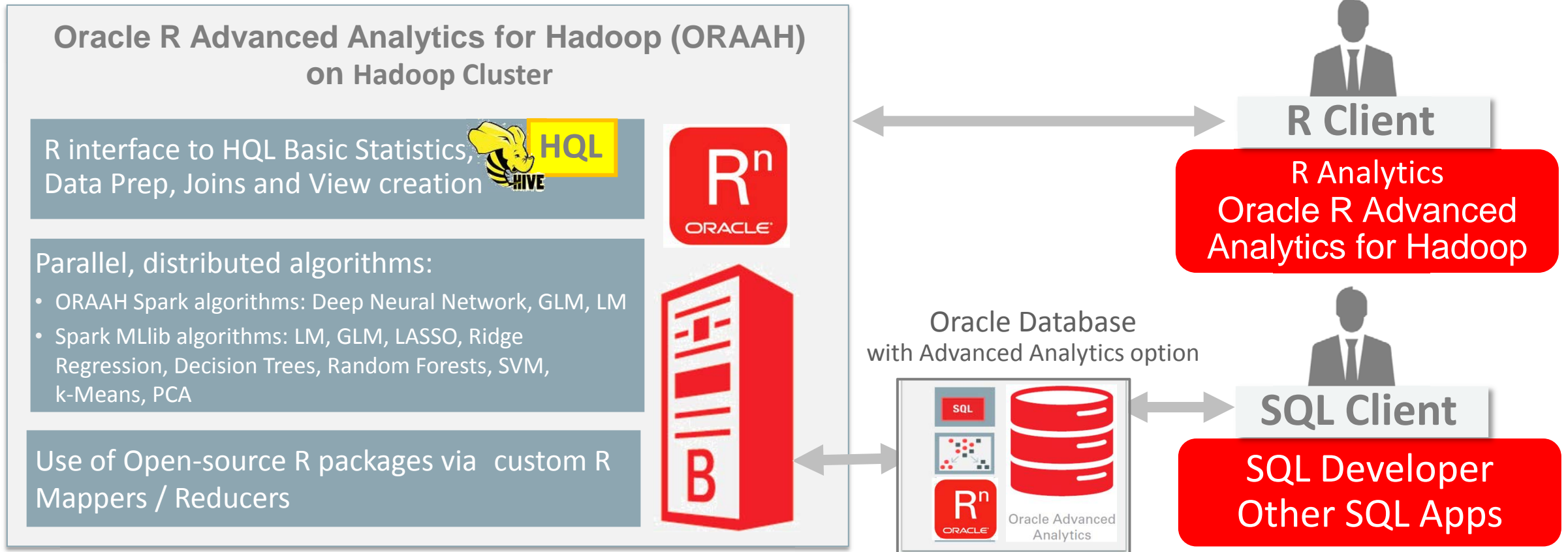
- 1 Class Support Vector Machine

### Time Series

- Single Exponential Smoothing
- Double Exponential Smoothing

# Oracle R Advanced Analytics for Hadoop

Using Hadoop/Hive/Spark Integration, plus R Engine and Open-Source R Packages






# Oracle R Advanced Analytics for Hadoop 2.7.0

## Machine Learning algorithms

### Classification

GLM ORAAH 

Logistic Regression ORAAH 

Logistic Regression 


Random Forests 


Decision Trees 

Support Vector Machines 

### Clustering


Hierarchical k-Means 

Hierarchical k-Means 

Gaussian Mixture Models 

### Regression

MLP Neural Networks ORAAH 

LASSO 


Ridge Regression 

Support Vector Machines 

Random Forest 


Linear Regression 

### Basic Statistics

Correlation/Covariance 


### Feature Extraction


Non-negative Matrix Factorization 

Collaborative Filtering (LMF) 

Singular Value Decomposition 

### Attribute Importance

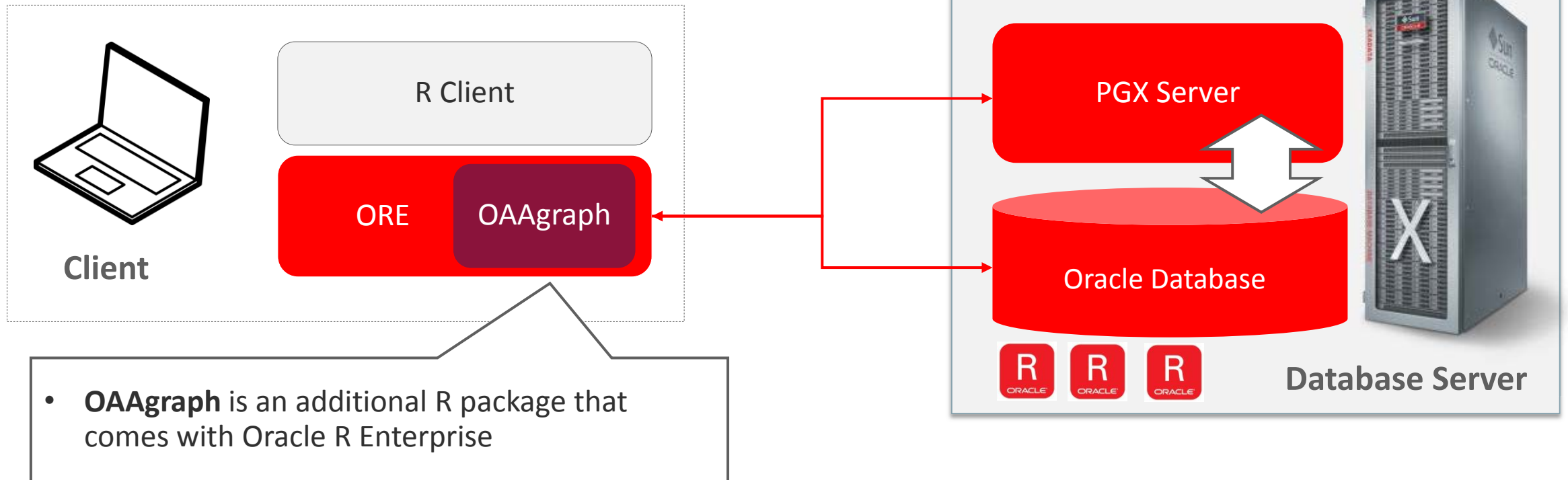
Principal Components Analysis 

Principal Components Analysis 



# OAAgraph Architecture with Oracle Database

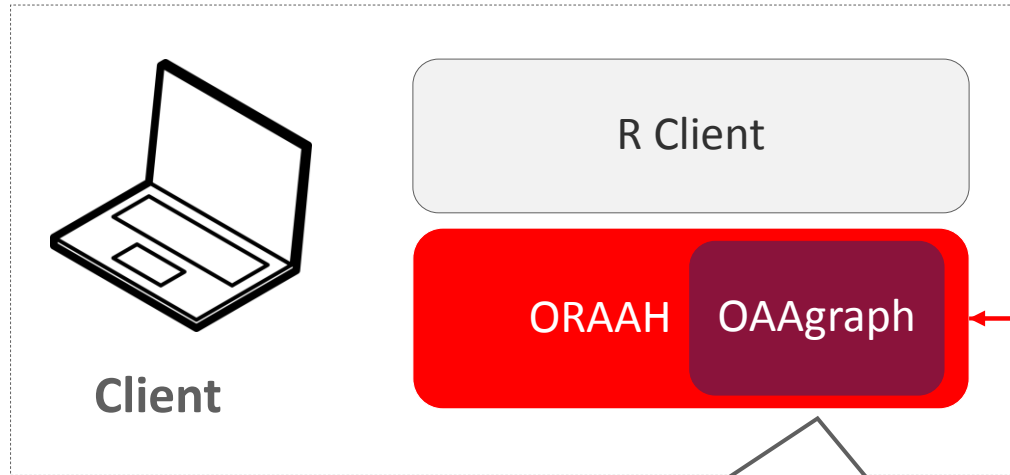
- **OAAgraph** gives remote control of PGX server
- PGX loads graph from database (ore.frames)



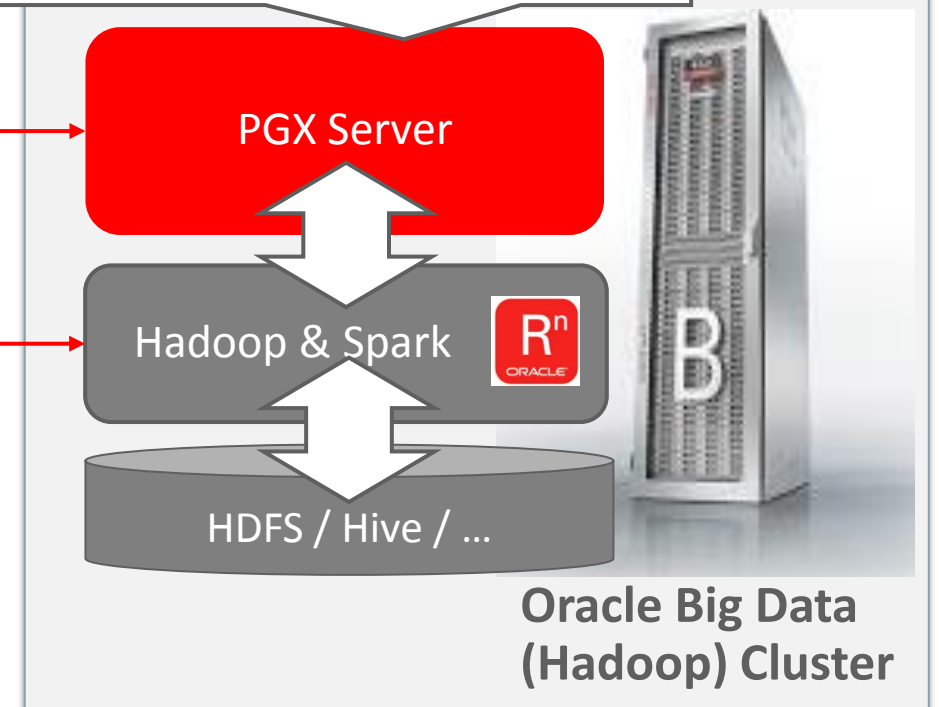
- **OAAgraph** is an additional R package that comes with Oracle R Enterprise

# OAAgraph Architecture with Spark/Hadoop

- **OAAgraph** gives remote control of PGX server
- PGX loads graph via SPARK data frames

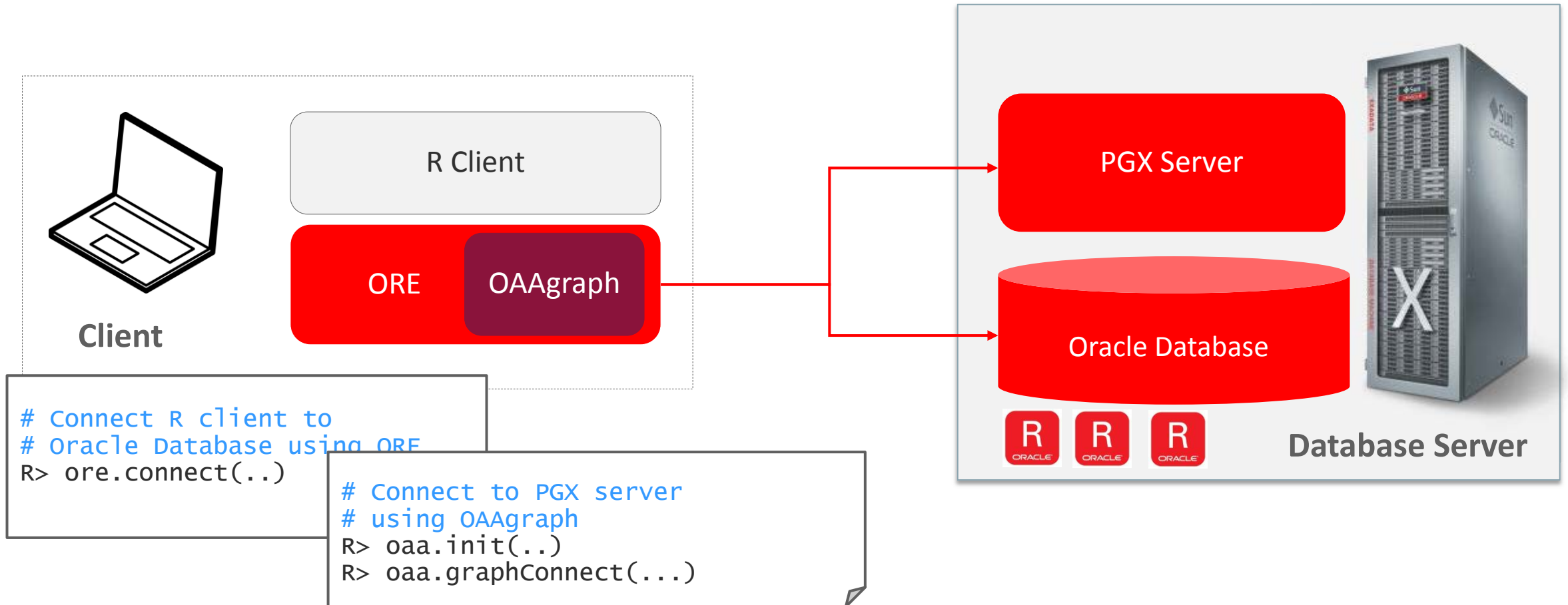


- **OAAgraph** is also available with Oracle R Advanced Analytics for Hadoop



# Execution Overview (ORE)

- Initialization and Connection



# Execution Overview (ORE)

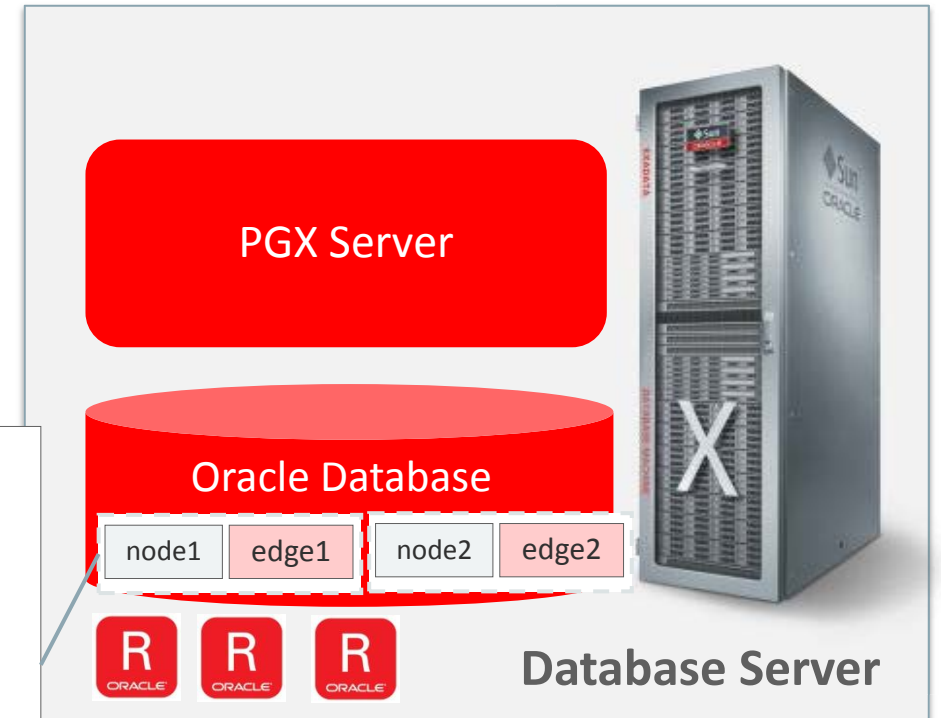
- Data Source
  - Graph data represented as two tables
    - Nodes and Edges
  - Multiple graphs stored in database
    - Using user-specified, unique table names

Node Table

Node ID	Node Prop 1 (name)	Node Prop 2 (age)	...
1238	John	39	...
1299	Paul	41	...
4818	...	...	...

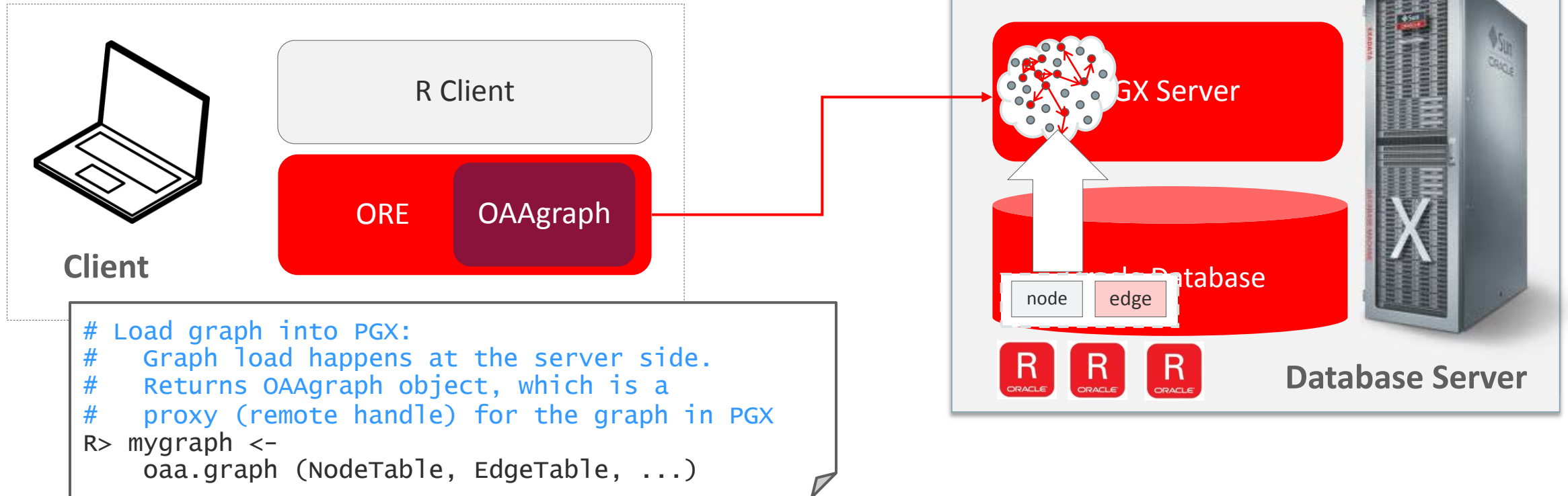
Edge Table

From Node	To Node	Edge Prop 1 (relation)	...
1238	1299	Likes	...
1299	4818	FriendOf	...
1299	6637	FriendOf	...



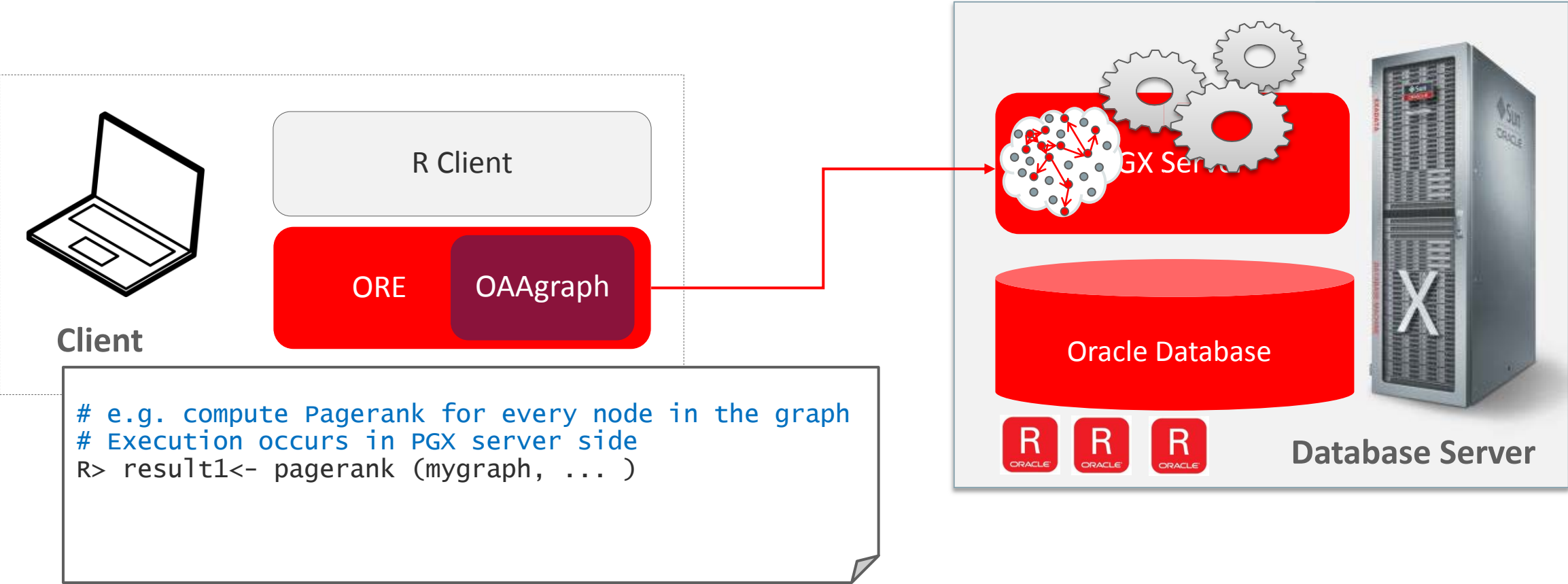
# Execution Overview (ORE)

- Loading Graph



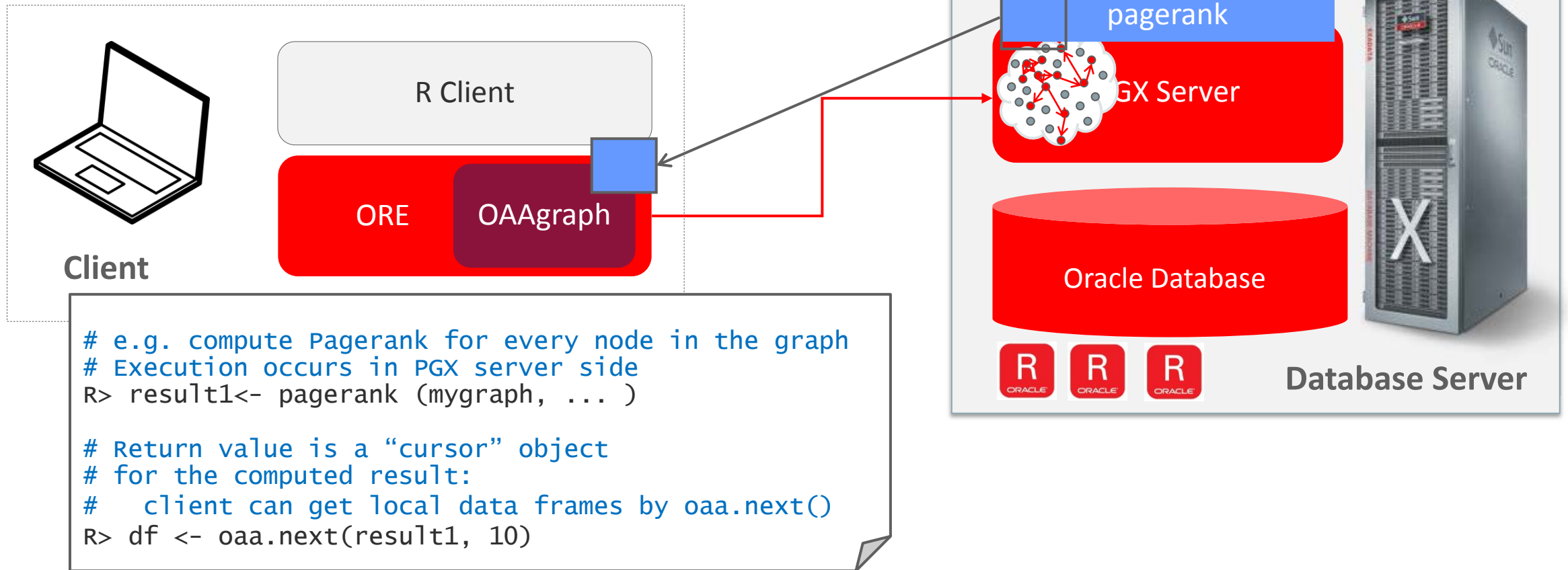
# Execution Overview (ORE)

- Running Graph Algorithm



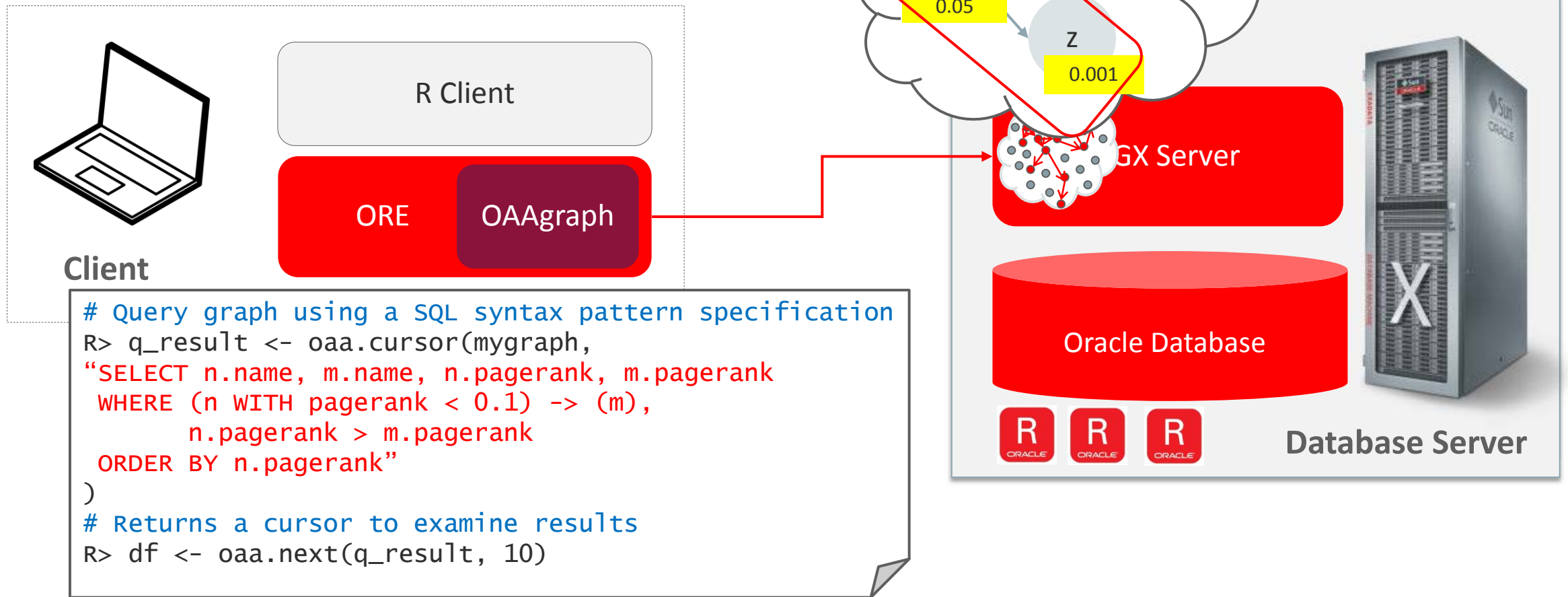
# Execution Overview (ORE)

- Iterating remote values with cursor



# Execution Overview (ORE)

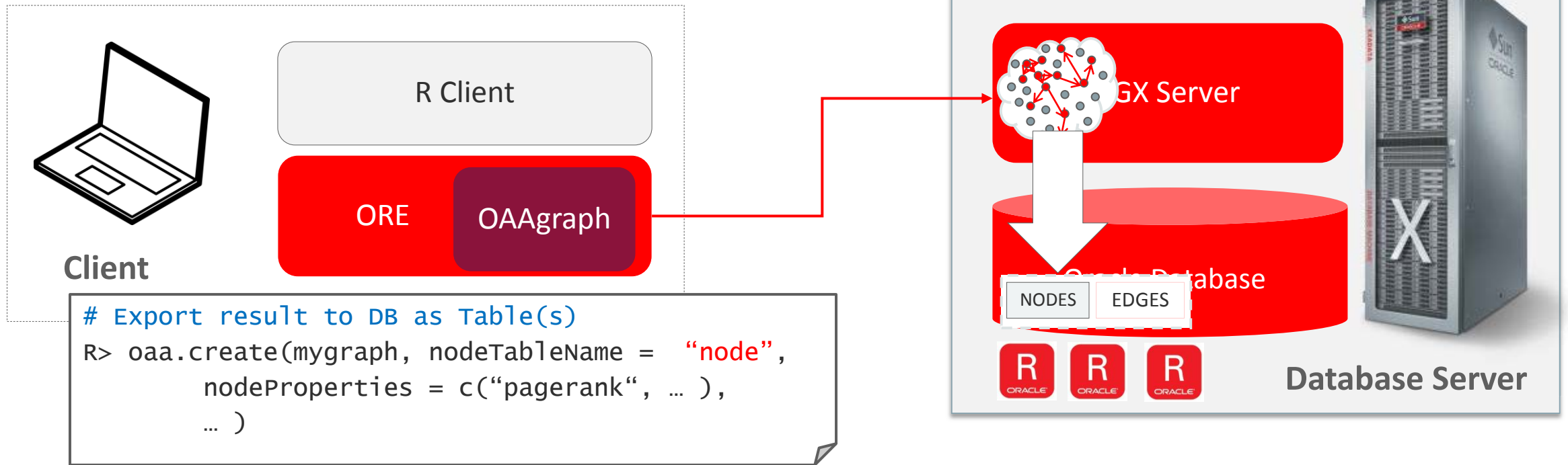
- Querying the graph





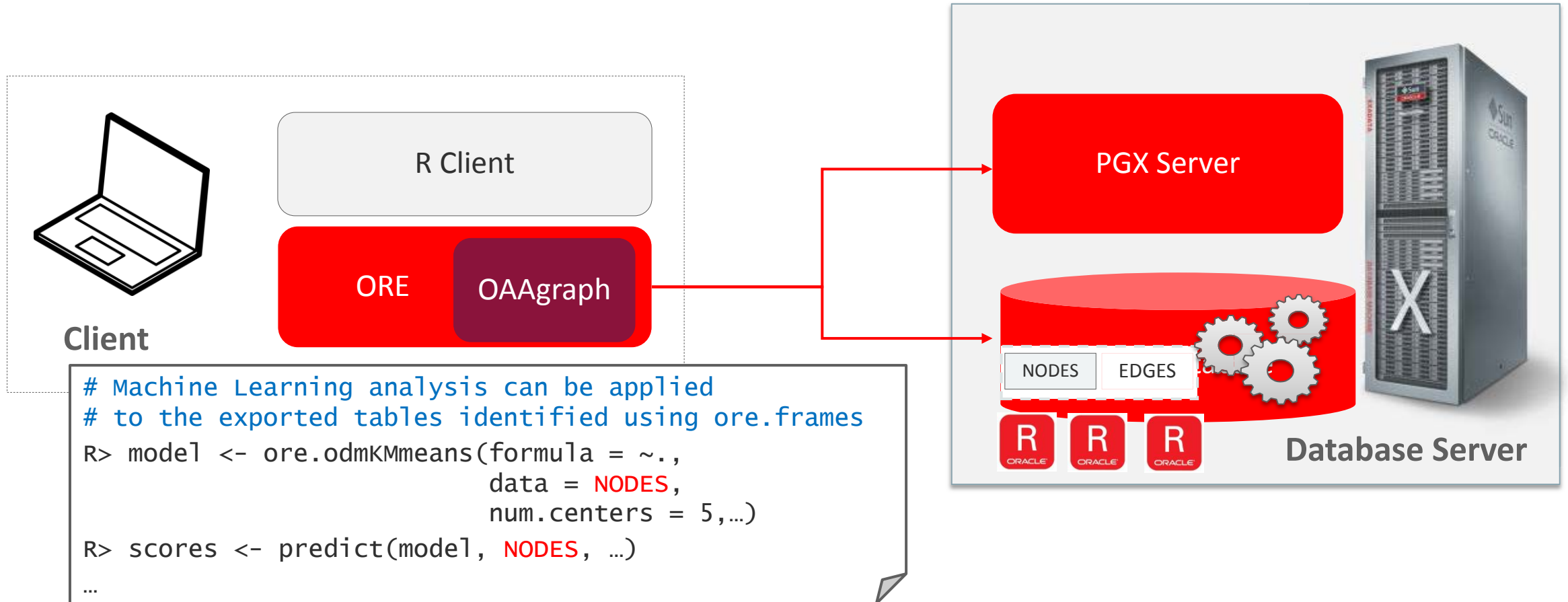
# Execution Overview (ORE)

- Exporting the result to DB



# Execution Overview (ORE)

- Continuing analysis with ORE Machine Learning



# Demo

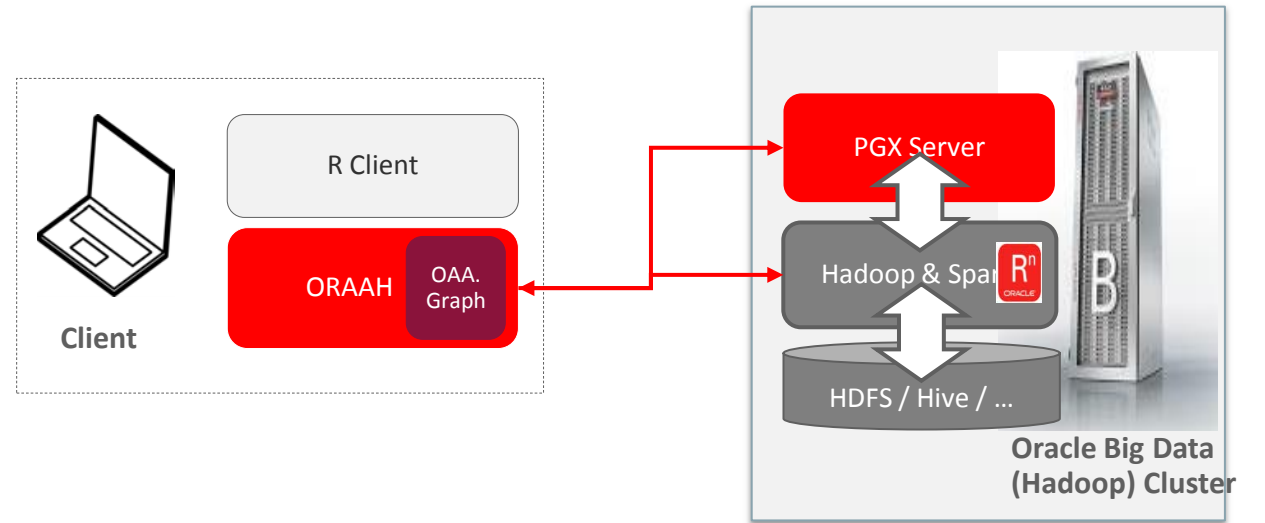
# Demo

- Environment

- OAAgraph with ORAAH
- PGX + SPARK + HDFS

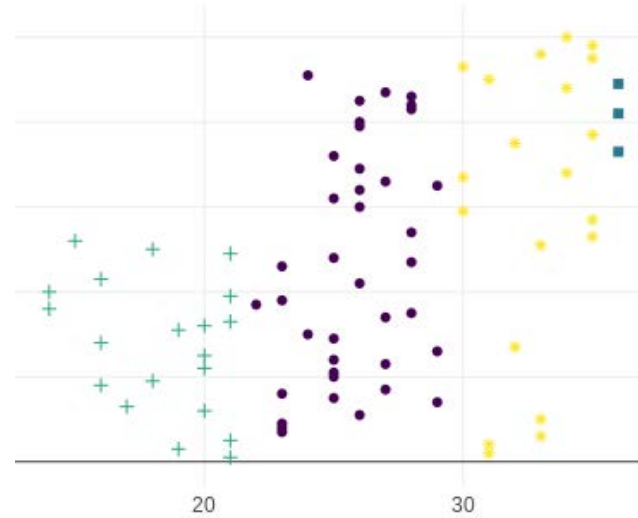
- Dataset

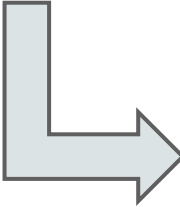
- Persons : name, age, zip, ...
- Calls: phone calls person-to-person

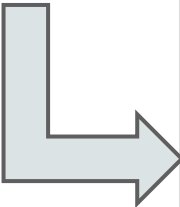


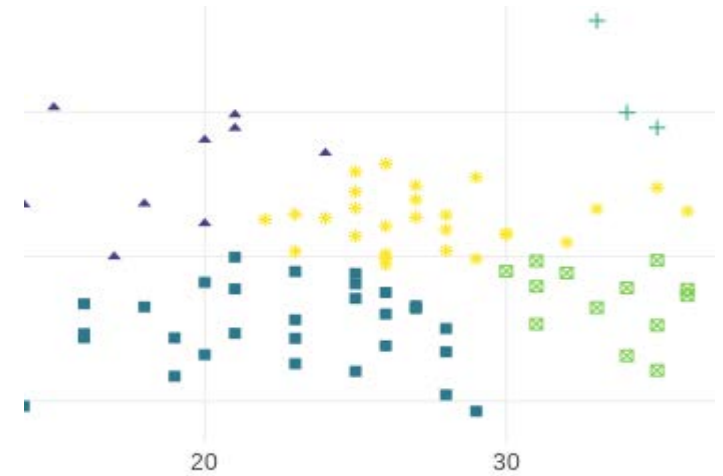
# Demo Scenario

- Load *persons* data into ORAAH
- Check the data set
- Cluster *persons* by their age with K-means



- 
- Load *calls* data into ORAAH
  - Create an OAAgraph object with *persons* and *calls*
  - Compute Pagerank and check results

- 
- Export results back to ORAAH
  - Cluster *persons* by their age AND pagerank values (with K-means)



# Summary

- Powerful, scalable graph analytics enabled from R
- Cross-pollinate graph analytics and machine learning

# **Hardware and Software Engineered to Work Together**

ORACLE®



# RDBMS queries (1)

- Q1 `SELECT n WHERE (n:macro WITH name = 'ksqget') ORDER BY n`
- Q2 `SELECT n WHERE (n:macro WITH name =~ 'EVTDV\$') ORDER BY n`
- Q3 `SELECT f.name WHERE (n:macro WITH name = 'ksqget') <-[c:expands]- (f:source_file) ORDER BY f.name LIMIT 10`
- Q4 `SELECT c.use_file_id, c.use_end_line, c.name_file_id, c.use_start_line, c.name_start_line, c.name_start_column  
WHERE (n:macro WITH name = 'ksqget') <-[c:expands]- (f:source_file WITH name = 'rdbms/src/server/vos/ksfd.c')  
ORDER BY c.use_file_id, c.use_end_line, c.name_file_id, c.use_start_line, c.name_start_line, c.name_start_column  
LIMIT 10`
- Q5 `SELECT n1.id(), n2.id(), n3.id() WHERE (n1:field) -[:isa_type]-> (n2:function_type) -[:has_param_type]->  
(n3:struct) -[:contains]-> (n1) ORDER BY n1, n2, n3 LIMIT 20`
- Q6 `SELECT f.name, c.use_start_line  
WHERE  
(n:macro WITH name = 'ksqget') <-[c:expands]- (f:source_file) -[e:expands]-> (m WITH name = 'KSQO_GLOBAL'),  
c.use_start_line <= e.use_start_line AND e.use_start_line <= c.use_start_line + 2  
ORDER BY f.name, c.use_start_line  
LIMIT 20`
- Q7 `SELECT f.name WHERE (n:macro WITH name = 'KSQO_GLOBAL') <-[c:expands]- (f:source_file) -[e:expands]-> (n),  
c.use_start_line = e.use_start_line ORDER BY f.name LIMIT 20`
- Q8 `SELECT n1 WHERE (n1:function) -[e:calls]-> (n2:function)  
ORDER BY e.use_start_line DESC, n1.name ASC, n2.name ASC LIMIT 10`

# RDBMS queries (2)

Q9 `SELECT COUNT(*), MIN(e.use_end_line), MAX(e.use_start_line), AVG(e.name_start_line), SUM(e.name_start_column)  
WHERE (:function) -[e:calls]-> (:function), e.name_start_line != -1`

Q10 `SELECT f.name, o.name, r.label(), COUNT(*)  
WHERE  
 (f:source_file WITH name = 'rdbms/src/generic/psm/kgfk.c') -[:file_contains|contains]-> () -  
 [r:calls|reads|writes]-> () <-[:file_contains|contains]- (o:source_file)  
GROUP BY  
 f, o, r.label()  
ORDER BY f.name, o.name, r.label()  
LIMIT 20`

Q11 `SELECT f.name, o.name, r.label(), COUNT(*)  
WHERE  
 (f:source_file) -[:file_contains|contains]-> () -[r:calls|reads|writes]-> () <-[:file_contains|contains]-  
 (o:source_file)  
GROUP BY  
 f, o, r.label()  
ORDER BY f.name, o.name, r.label()  
LIMIT 20`

Q12 `SELECT n.id(), n.outDegree()  
WHERE (n:function)  
ORDER BY n.outDegree() DESC  
LIMIT 10`

# RDBMS queries (3)

```
Q13 SELECT n.id() WHERE (n WITH name = 'ksqget' OR name =~ 'EVTDV\$') ORDER BY n
```

```
Q14 PATH contains := () -[:file_contains|dir_contains]-> ()
SELECT m
WHERE
  (d:directory WITH name = 'rdbms/src/client') -/:contains*/-> (f),
  (f) -[:file_contains]-> (m:macro WITH name =~ 'EVTDV\$')
```

```
Q15 PATH includes := () -[:includes]-> ()
SELECT f.name
WHERE
  (f) -/:includes*/-> (h:source_file WITH name = 'rdbms/include/kge.h')
ORDER BY f.name
LIMIT 20
```

```
Q16 PATH contains := () -[:file_contains|contains]-> ()
SELECT f.name, o.name, r.label(), COUNT(*)
WHERE
  (f:source_file WITH name = 'rdbms/src/generic/psm/kgfk.c') -/:contains*/-> (x),
  (o:source_file) -/:contains*/-> (y),
  (x) -[:calls|reads|writes]-> (y)
GROUP BY
  f, o, r.label()
ORDER BY o.name
LIMIT 20
```

# RDBMS queries (4)

```
Q17 PATH contains := () -[:file_contains|contains]-> ()
SELECT f.name, o.name, r.label(), COUNT(*)
WHERE
  (f:source_file) -/:contains*/-> (x),
  (o:source_file) -/:contains*/-> (y),
  (x) -[r:calls|reads|writes]-> (y)
GROUP BY
  f, o, r.label()
ORDER BY o.name
LIMIT 20
```