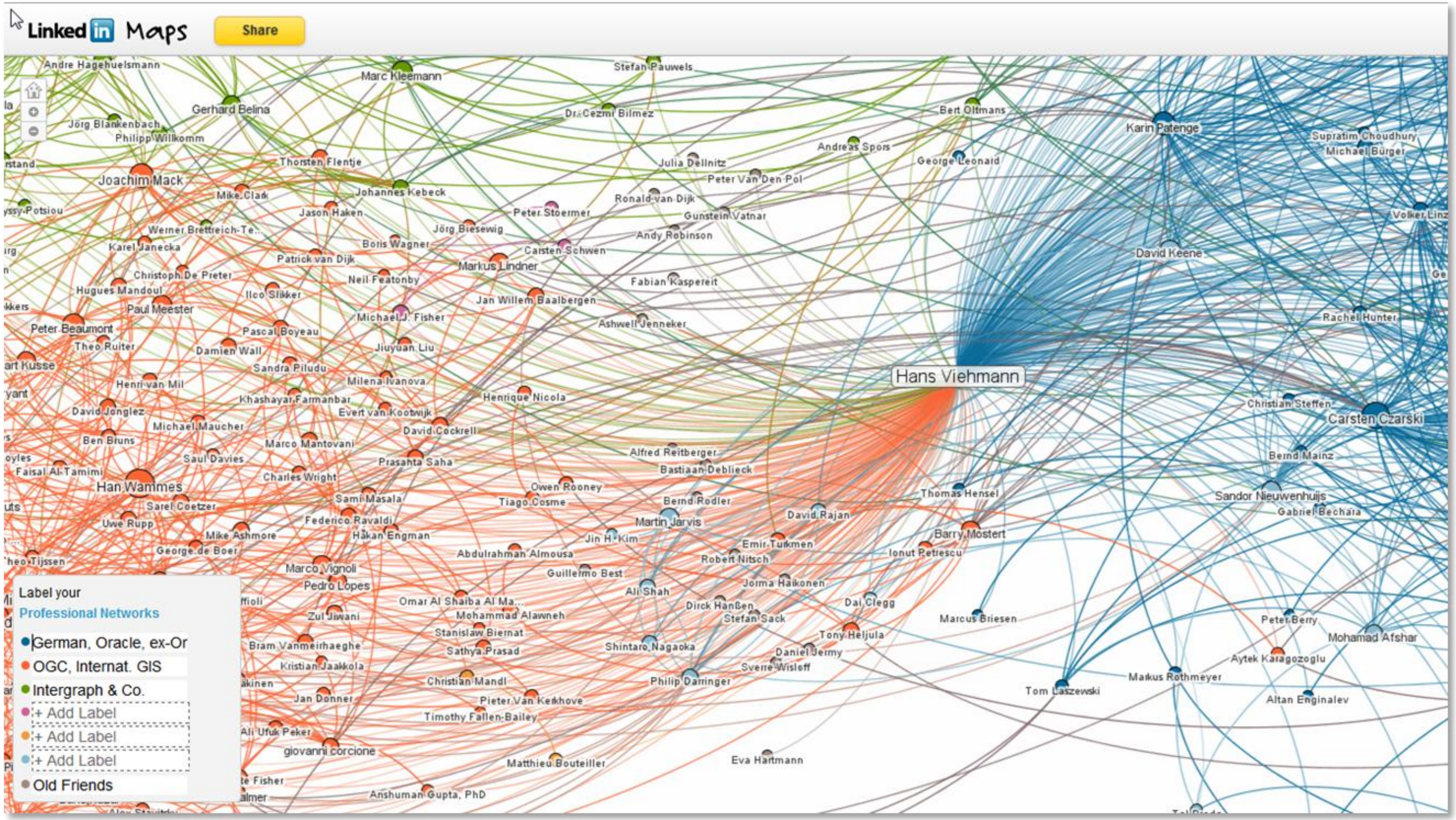
# Using Graph Analytics on Oracle NoSQL to Analyze and Segment Customers

**OUG Ireland 2016**

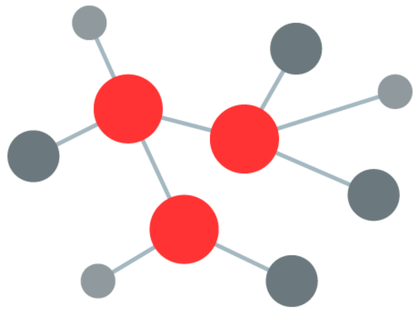
Hans Viehmann
Product Manager EMEA
March 3, 2016

## Safe Harbor Statement


LinkedIn Maps
Share
Hans Viehmann
Label your
Professional Networks
● German, Oracle, ex-Or
● OGC, Internat. GIS
● Intergraph & Co.
● + Add Label
● + Add Label
● + Add Label
● Old Friends
Andre Hagehuelsmann
Marc Kleemann
Stefan Pauwels
Gerhard Belina
Dr. Cezmi Bilmez
Bert Oltmans
Karin Patenge
Jörg Blankenbach
Philipp Willkomm
Supratim Choudhury
Michael Bürger
Andreas Spors
George Leonard
Thorsten Flentje
Julia Dellnitz
Joachim Mack
Peter Van Den Pol
Johannes Kebeck
Mike Clark
Ronald van Dijk
Jason Haken
Peter Stoermer
Gunstein Vatnar
Volker Linz
Werner Brettreich-Te...
Jörg Biesewig
Andy Robinson
Karel Janecka
Boris Wagner
Carsten Schwen
David Keene
Patrick van Dijk
Markus Lindner
Christoph De Preter
Neil Featonby
Fabian Kaspereit
Hugues Mandoul
Ilco Slikker
Jan Willem Baalbergen
Paul Meester
Michael J. Fisher
Ashwell Jenneker
Rachel Hunter
Peter Beaumont
Pascal Boyeau
Theo Ruiter
Damien Wall
Jiuyuan Liu
Sandra Piludu
Henri van Mil
Milena Ivanova
Henrique Nicola
Khashayar Farmanbar
David Jonglez
Evert van Kootwijk
Christian Steffen
Michael Maucher
David Cockrell
Carsten Czarski
Ben Bruns
Marco Mantovani
Alfred Reitberger
Bernd Mainz
Saul Davies
Prasanta Saha
Bastiaan Deblieck
Faisal Al-Tamimi
Charles Wright
Han Wammes
Owen Rooney
Thomas Hensel
Sandor Nieuwenhuijs
Sami Masala
Tiago Cosme
Bernd Rodler
Sarel Coetzer
Gabriel Bechara
David Rajan
Federico Ravaldi
Martin Jarvis
Uwe Rupp
Barry Mostert
Håkan Engman
Jin H. Kim
Mike Ashmore
Emir Turkmen
George de Boer
Abdulrahman Almousa
Robert Nitsch
Ionut Petrescu
Theo Tijssen
Marco Vignoli
Guillermo Best
Jorma Haikonen
Pedro Lopes
Ali Shah
Omar Al Shaiba Al Ma...
Dirck Hanßen
Dai Clegg
Zul Jiwani
Mohammad Alawneh
Stefan Sack
Marcus Briesen
Peter Berry
Stanislaw Biernat
Tony Heljula
Mohamad Afshar
Bram Vanmeirhaeghe
Sathya Prasad
Shintaro Nagaoka
Daniel Jermy
Aytek Karagozoglu
Kristian Jaakkola
Sverre Wisloff
Christian Mandl
Philip Darringer
Markus Rothmeyer
Tom Laszewski
Jan Donner
Pieter Van Kerkhove
Altan Enginalev
Timothy Fallen-Bailey
Ali Ufuk Peker
giovanni corcione
Matthieu Bouteiller
Eva Hartmann
Anshuman Gupta, PhD

## Graph and Spatial Analysis – It is about relationships

- Are things in the same location? Who is the nearest? What tax zone is this in? Where can we deliver in 35 minutes? What is in my sales territory? Is this built in a flood zone?

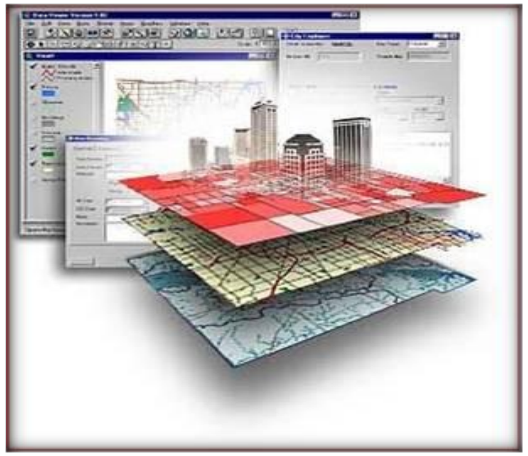


- Which supplier am I most dependent upon? Who is the most influential customer? Do my products appeal to certain communities? What patterns are there in fraudulent behavior?

## Oracle Big Data Spatial and Graph

### Spatial Analysis:

- Location Data Enrichment
- Proximity and containment analysis, Clustering
- Spatial data preparation (Vector, Raster)
- Interactive visualization



### Property Graph for Analysis of:

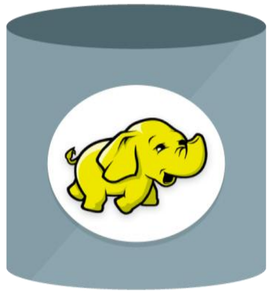
- Social Media Relationships
- eCommerce Targeted Marketing
- Cyber-Security, Fraud Detection
- IoT, Industrial Engineering

ORACLE

# Strategic Vision: The platform

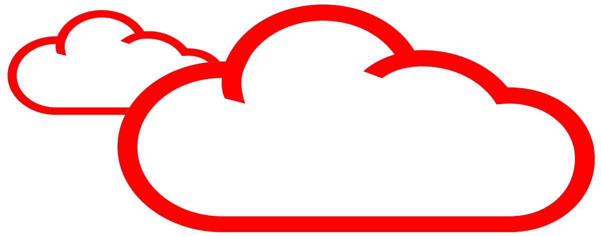
**Database and Big Data Platform support, both On-premise and in the Cloud**

Oracle Big Data Spatial and Graph



Oracle Database Spatial and Graph

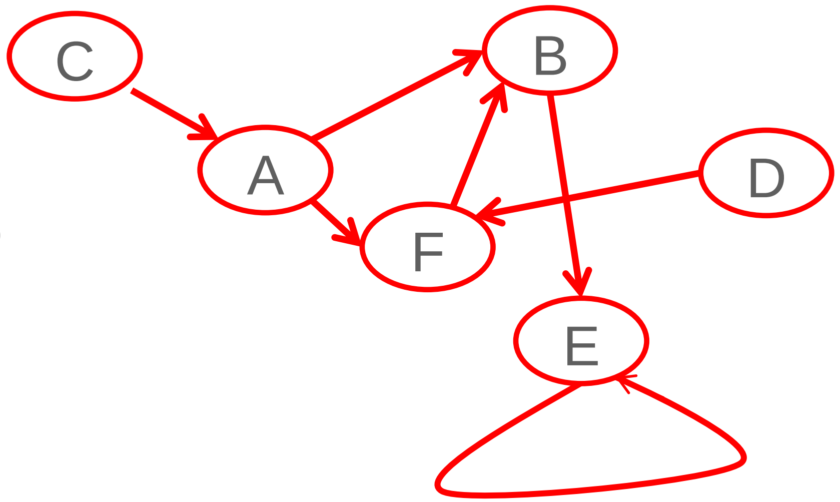
Spatial and Graph in Cloud Offerings



Big Data:
Single Model Data Store

Database 12c:
Polyglot (Multi-model) Data Store

## Graph Data Model



What is a graph?

- A set of links and nodes (and optionally attributes)
- A graph is simply **linked data**

Why do we care?

- Graphs are everywhere
  - Social networks/Social Web (Facebook, Linkedin, Twitter, Baidu, Google+,…)
  - Cyber networks, power grids, protein interaction graphs
  - Knowledge graphs (IBM Watson, Apple SIRI, Google Knowledge Graph)
- Graphs are intuitive and flexible
  - Easy to navigate, easy to form a path, natural to visualize
  - Do not require a predefined schema

# Background: Three Types of Graph Data Models

Social Network Analysis

**Property Graph Model**
- Graph Data Management
- Social Network Analysis
- Entity analytics

- National Intelligence
- Public Safety
- Social Media search
- Marketing - Sentiment

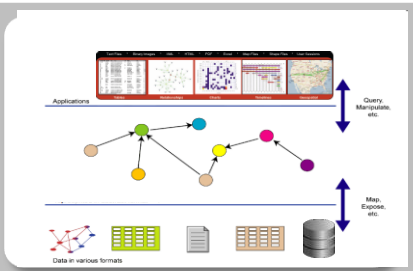
Spatial Network Analysis

**Network Data Model**
- Network path analysis
- Transportation modeling

- Logistics
- Transportation
- Utilities
- Telcoms

Linked Data / Metadata Layer

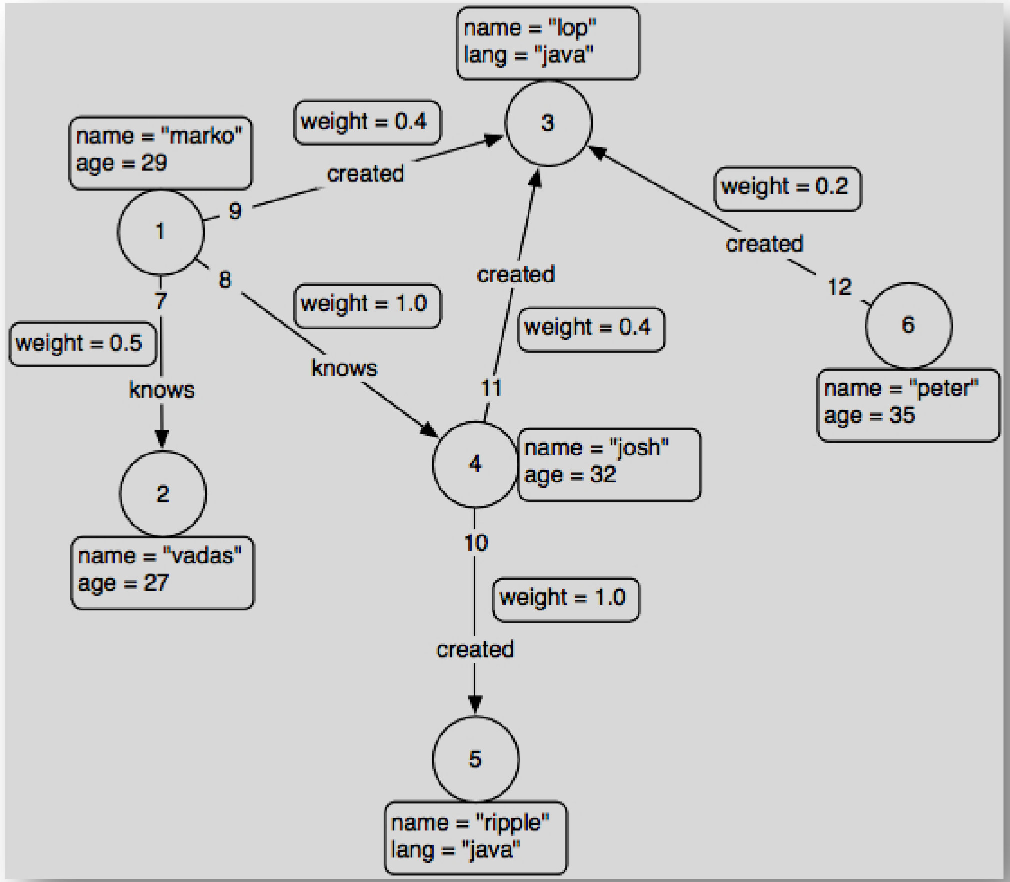
**RDF Data Model**
- Data federation
- Knowledge representation
- Semantic Web



- Life Sciences
- Health Care
- Publishing
- Finance

**Use Case** **Graph Model** **Industry Domain**

# The Property Graph Data Model

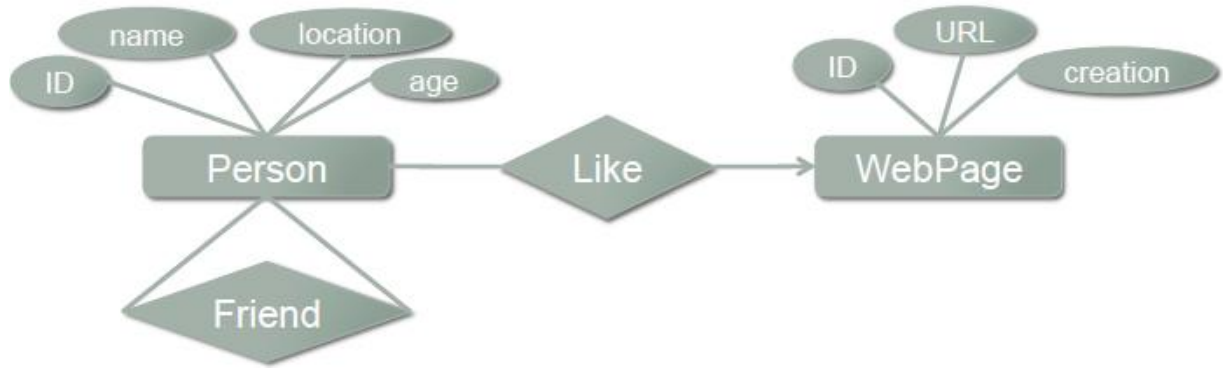


https://github.com/tinkerpop/blueprints/wiki/Property-Graph-Model

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

# Graph Analysis Examples

- Attribute searching (Get people with a given name)
- Node/edge adjacency (Get people that like a given Web page)
- Fixed-length paths (Get the friends of the friends of a given person)
- Reach-ability (Is there a “friend” connection between two people?)
- Pattern matching (Get the common friends between two people)
- Aggregates (Get the number of friends of a given person)



# Common Graph Analysis Use Cases

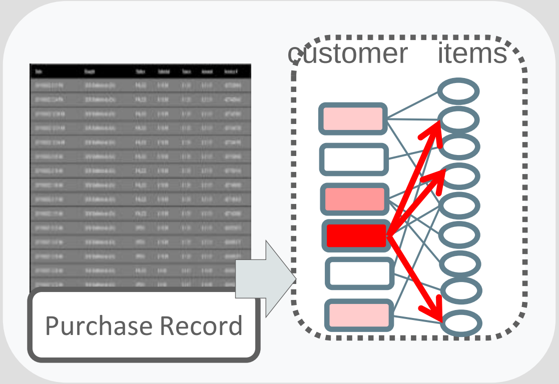
Recommend the most *similar* item purchased by *similar* people

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

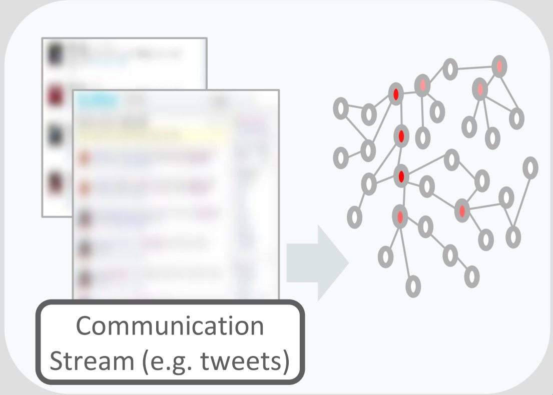
Identify group of people that are close to each other – e.g. target group marketing

Find out all the sets of entities that match to the given pattern – e.g. fraud detection

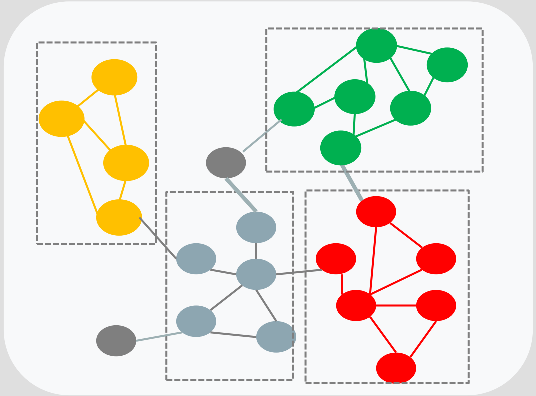
## Product Recommendation



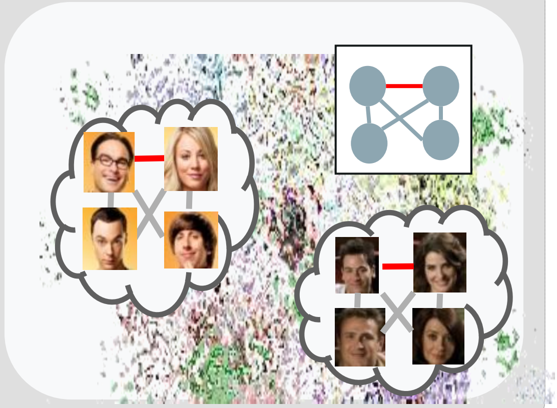
## Influencer Identification



## Community Detection



## Graph Pattern Matching

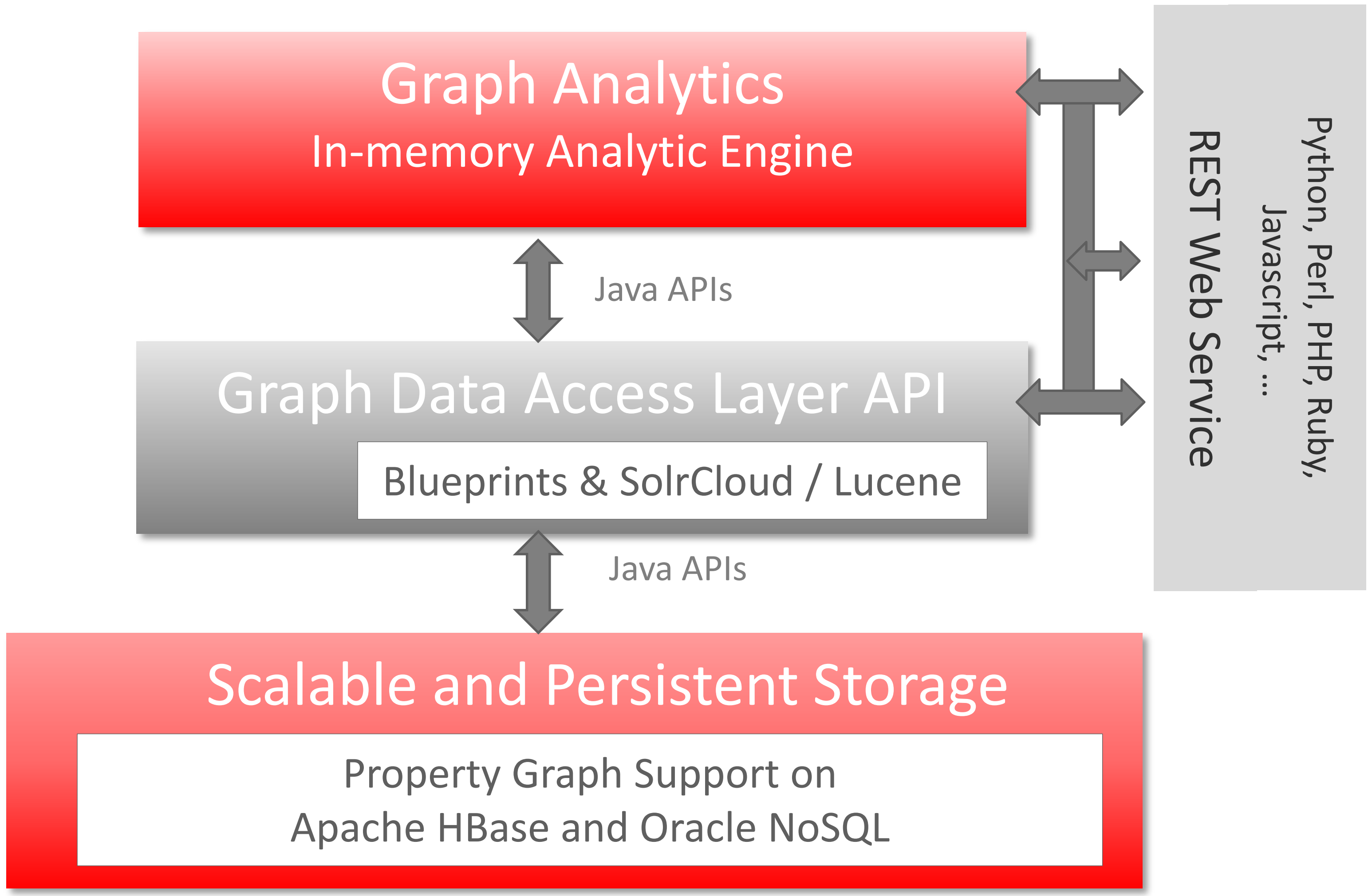


# Oracle Big Data Spatial and Graph – Graph Features

**Property Graph Database on Cloudera Distribution for Hadoop (CDH)**

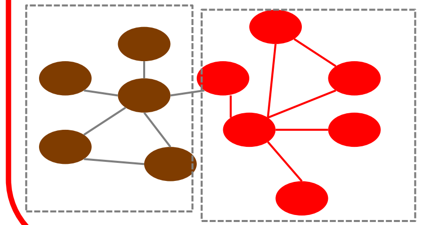
- Highly scalable graph database and **parallel**, **in-memory** analytics engine
- Implemented on Apache HBase and Oracle NoSQL Database
- Rich developer APIs
  - Blueprints, REST, Java graph plus support for Groovy, Python, PHP, Perl, Ruby, and JavaScript
- Fast, scalable suite of social network analysis functions
  - Ranking, centrality, recommender, community detection, path finding…
  - Targeted to address main industry requirements
- Data Management
  - Bulk load
  - Console to execute Java and Gremlin APIs

# Oracle Big Data Graph Architecture


Graph Analytics
In-memory Analytic Engine
Java APIs
Graph Data Access Layer API
Blueprints & SolrCloud / Lucene
Java APIs
Scalable and Persistent Storage
Property Graph Support on
Apache HBase and Oracle NoSQL
REST Web Service
Python, Perl, PHP, Ruby, Javascript, …

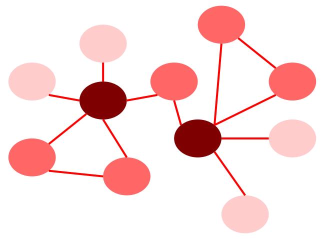
## 35 Graph Functions

### Detecting Components and Communities



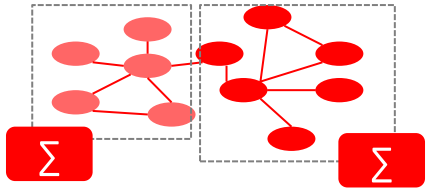
Tarjan's, Kosaraju's,
Weakly Connected Components, Label Propagation (w/ variants), Soman and Narang's

### Ranking and Walking



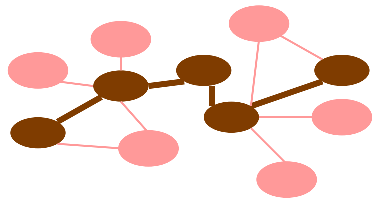
Pagerank, Personalized Pagerank,
Betweenness Centrality (w/ variants),
Closeness Centrality, Degree Centrality,
Eigenvector Centrality, HITS,
Random walking and sampling (w/ variants)

### Evaluating Community Structures



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

### Path-Finding



Hop-Distance (BFS)
Dijkstra's,
Bi-directional Dijkstra's
Bellman-Ford's

### Link Prediction

SALSA
(Twitter's Who-to-follow)

### Other Classics

Vertex Cover
Minimum Spanning-Tree(Prim's)

## Who’s the most important Marvel Hero?

## There Are Lots of Answers to That

- Answers from **Aggregation**
  - Who has the most appearances?
  - Who has the most series?
  - Who has the most movie appearances, toys, etc?

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

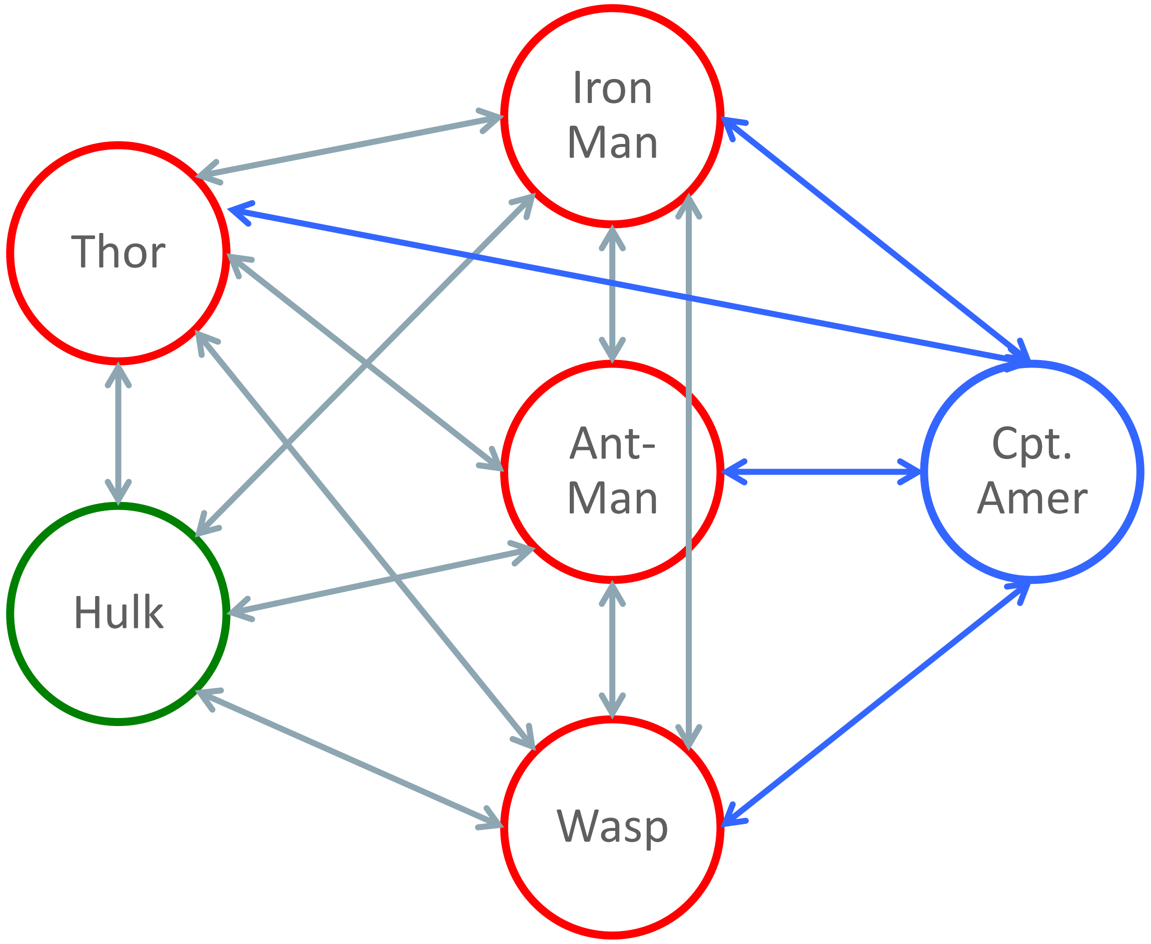
- Answers from **Connectivity**
  - Who’s most central to teams?
  - Who has the strongest cross-overs?
  - How important is the hero’s community?

**Graph questions**:
We need something different!

ORACLE

## So How Do Super Heroes Become a Graph?

- Each hero is a **vertex**
- Each joint appearance is an **edge**
- The original Avengers are **fully connected**



- Hulk leaves and Captain America joins
- No longer fully connected
- This graph gets **complex**

## Graph Analysis Can Be Complicated

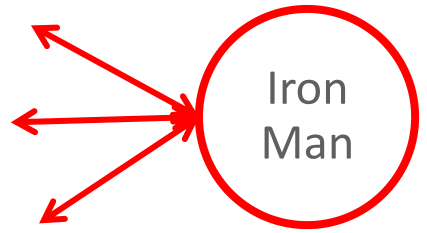
- Find the size of the 2-hop network of superheroes (Gremlin+Python)

```
sum([v.query() \
    .direction(blueprints.Direction.OUT).count() \
    for v in OPGIterator(v0.query() \
    .direction(blueprints.Direction.OUT) \
    .vertices().iterator())])
```

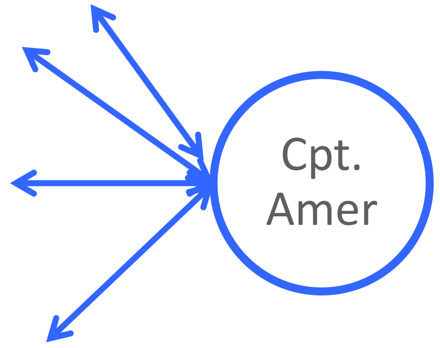
- Computing "importance" is more nebulous **and** harder to code

# Importance as **Degree Centrality**

- The more **edges** a vertex has, the higher its **degree**



- The greater the degree, the more important the vertex is
- This is one way to look at importance

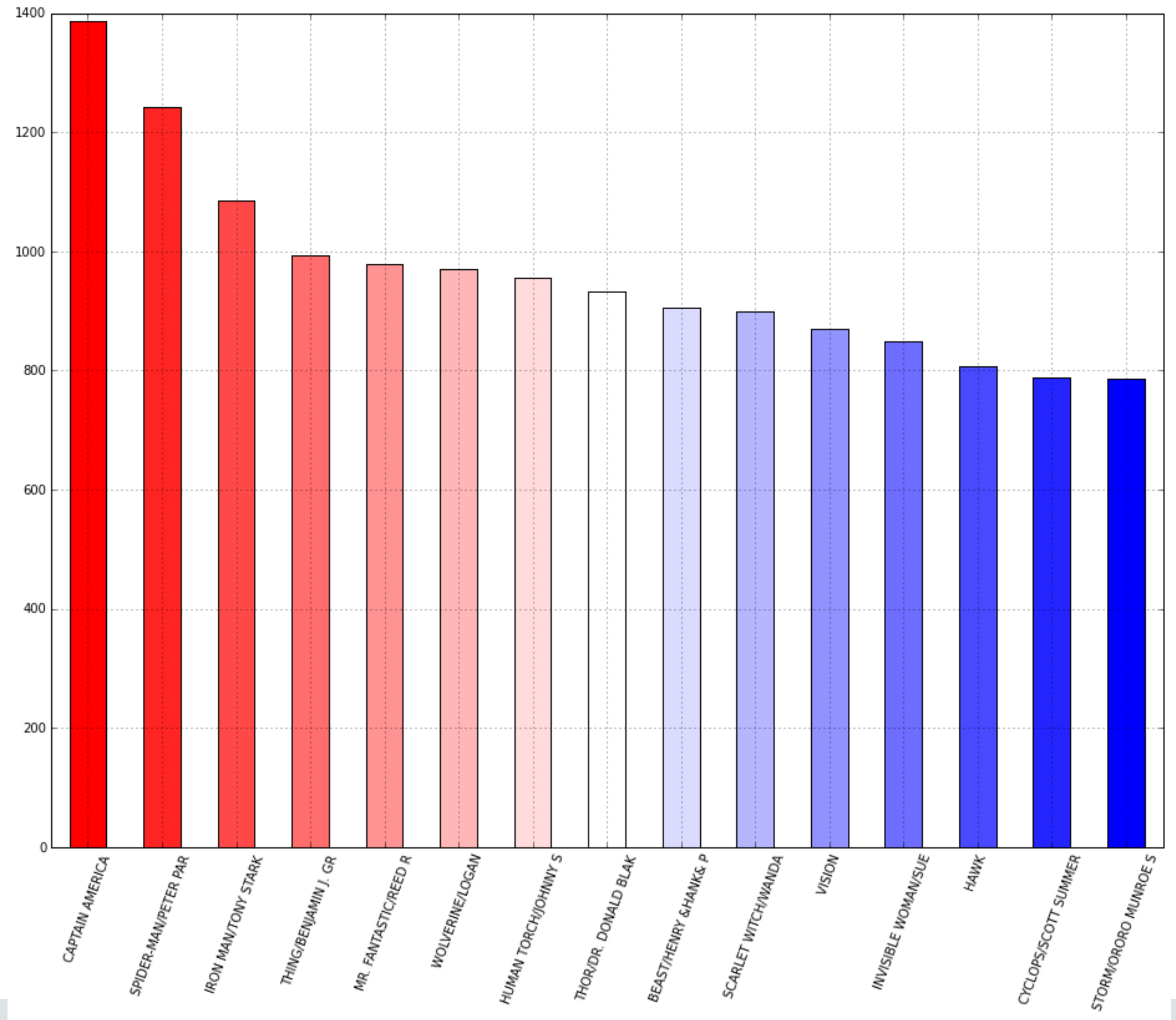


- Is your most connected customer most important?

# Degree Centrality in Big Data Graph

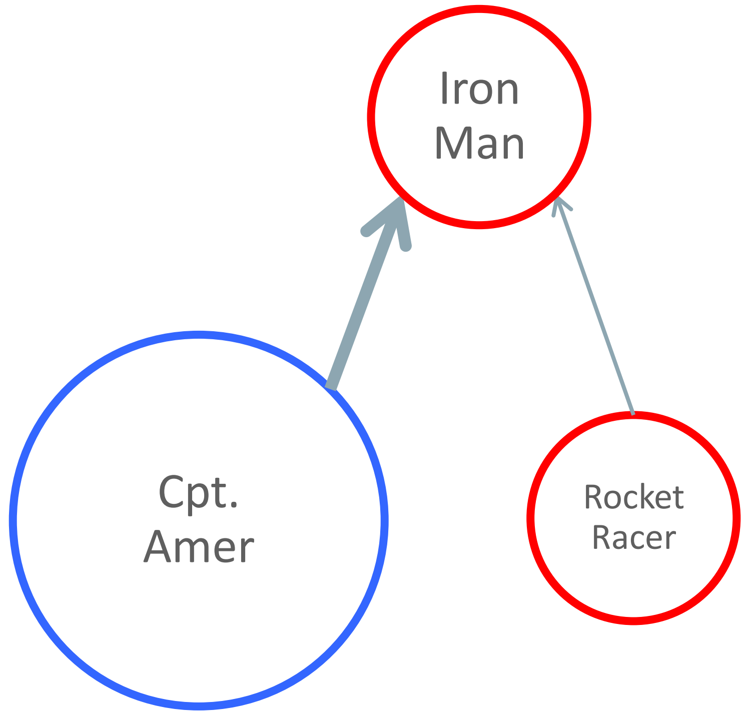
## Code

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



# Importance as **Page Rank**

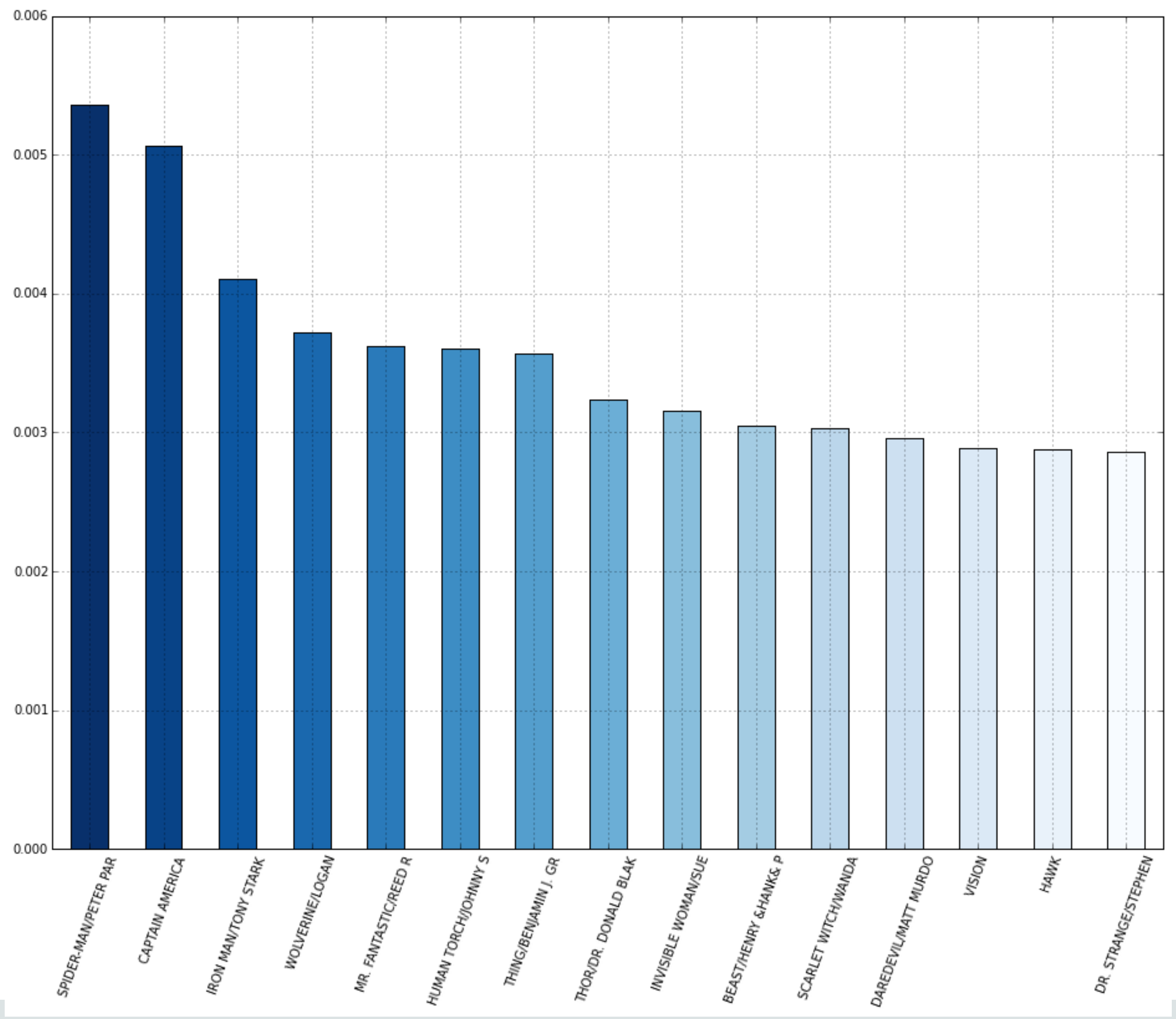
- Importance can flow **through** a graph
- A node connected to by important nodes is **also** important
- This is importance as a measure of
  - Trust
  - Prominence
- Thinking about customers in a graph requires multiple definitions of importance



# Page Rank in Big Data Graph

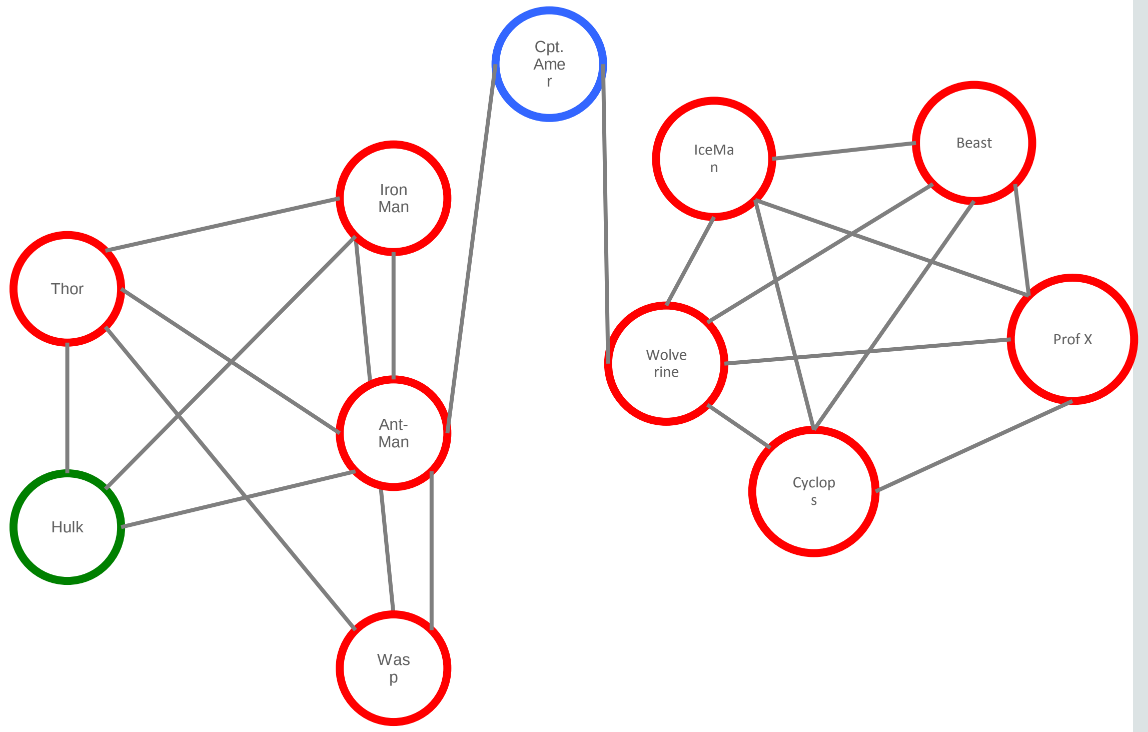
## Code

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



# Importance as “Between-ness”

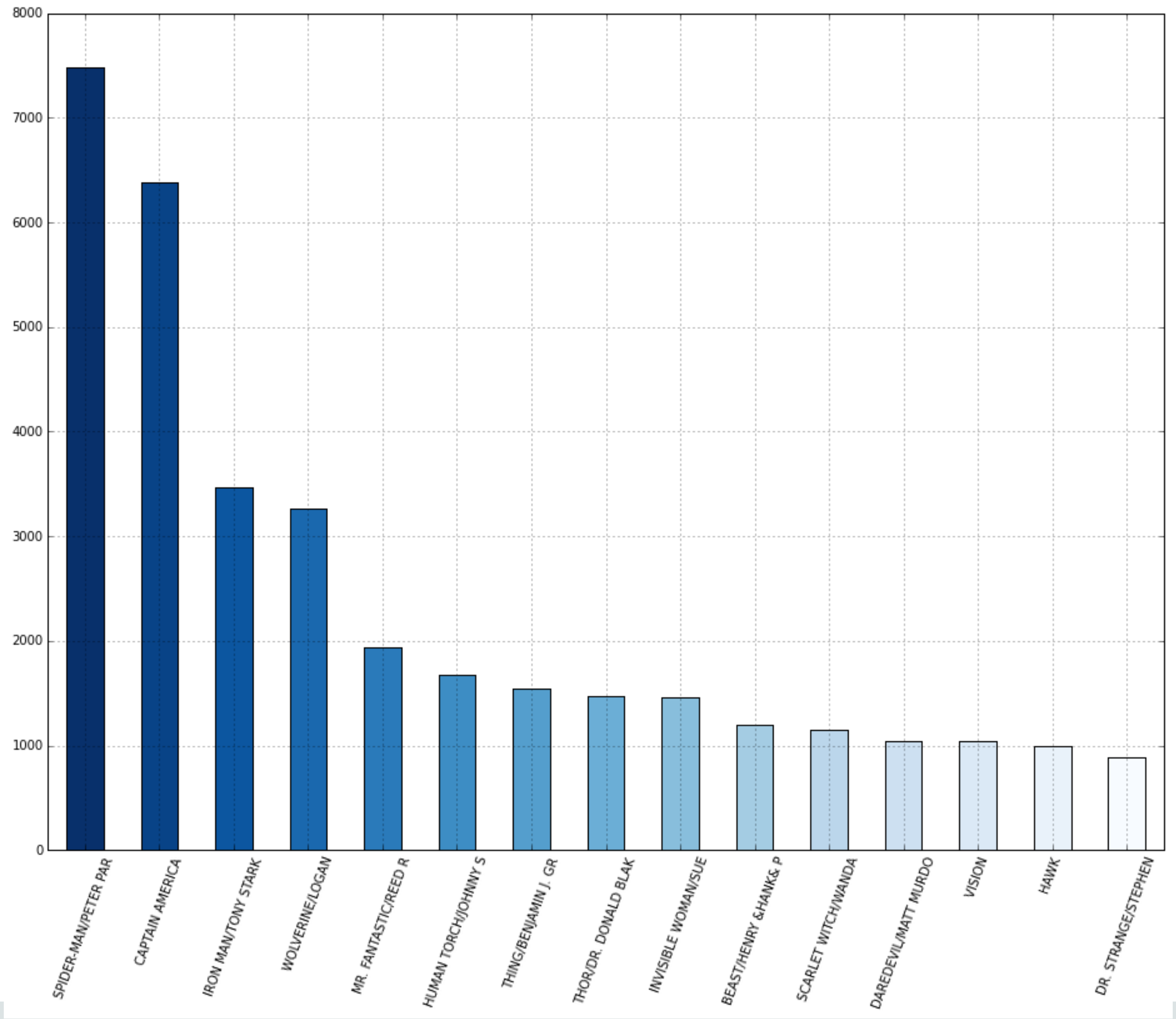
- Importance can be how often you’re on the critical path
- **Betweenness** is the number of shortest paths a node is part of
- E.g. The superhero on all the teams
- E.g. The player in all the scoring sequences



## Betweenness Centrality in Big Data Graph

### Code

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

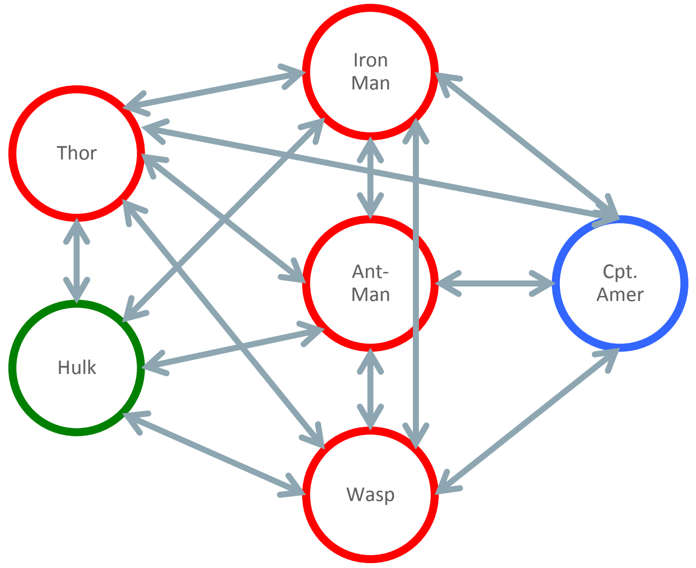


# Communities Matter Too

MARVEL COMICS
X-MEN
PLUS: MAVERICK AND THE XAVIER FILES
$1.25 US
11 AUG
30TH ANNIVERSARY THE AMAZING SPIDER-MAN 1962-1992

## Communities Are Just Special Subgraphs

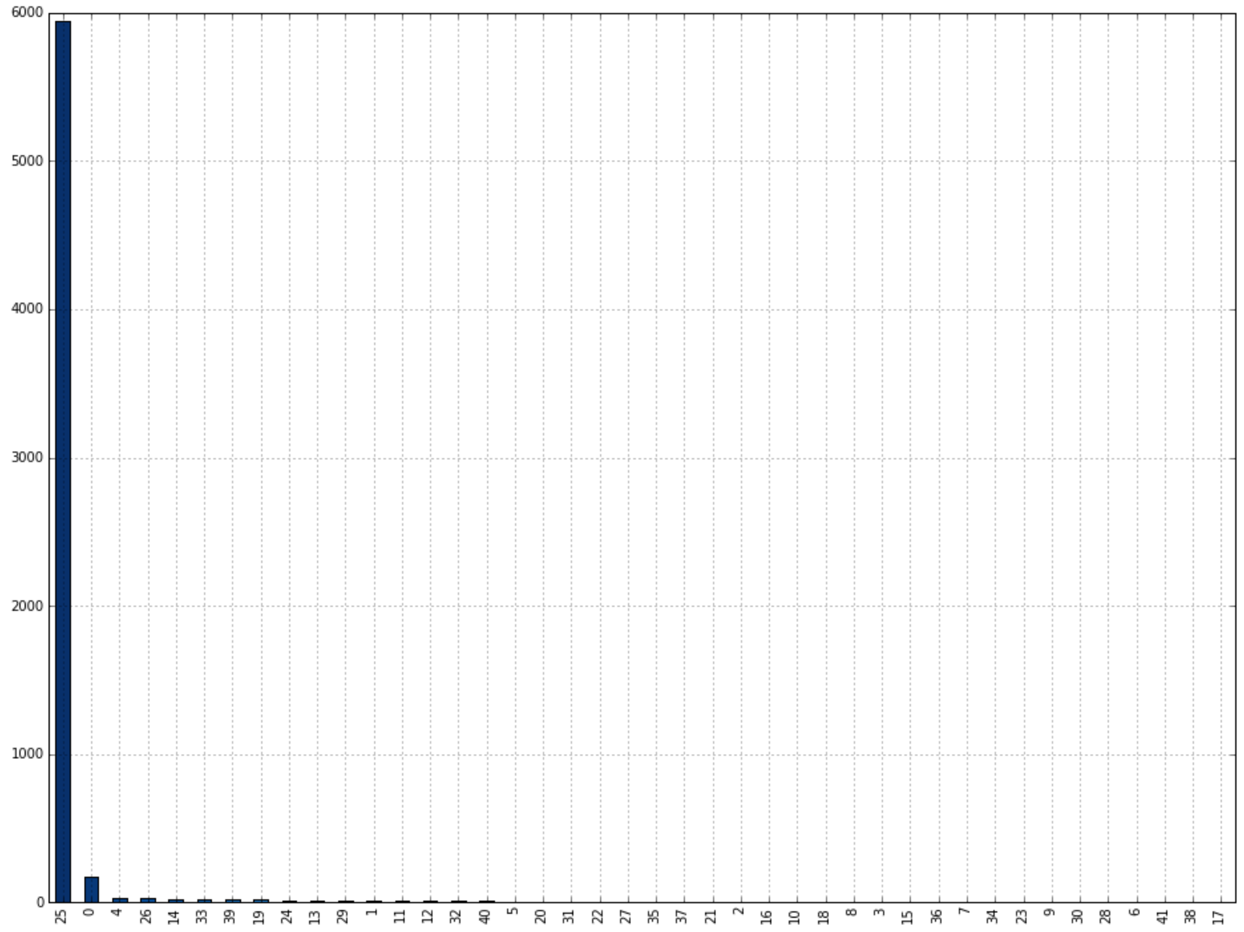
- Community
  - A subgraph in which
    - Nodes are **more** connected to each other
- Communities provoke interesting questions
  - How many?, How large?
  - How do they relate to each other?
- Analyses can be performed on a community
  - Who’s the most valuable customer in a community?



# Community Detection in Big Data Graph

## Code

```
comic_coms = analyst.communities()
```



## Coming back to the initial question ...

**Social Network Analysis for Churn Risk Prediction in the Telecom Industry**

## Social Network Analysis (SNA)

**Well-established approach for churn detection, cross-sell campaigns, ...**

- Underlying SNA mechanisms
  - Word of mouth – customers trust other customers
  - Communities – word of mouth spreads in communities, especially through leaders
- Using SNA for customer acquisition, retention and value growth by
  - Promoting positive messages
  - Countering negative messages
- Technical approach
  - Detect communities, identify leaders using graph analysis/graph mutation
  - Churn modeling based on data mining (OAA) predictive analytics

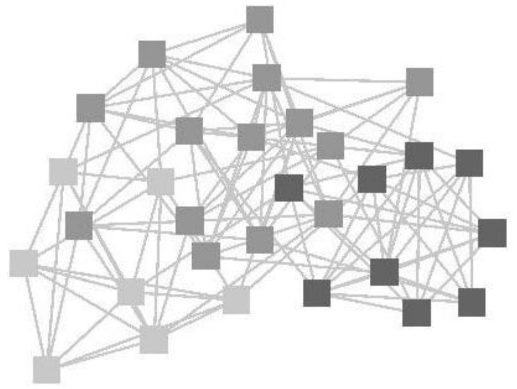
## Creating the social graph

**Telecom subscriber use case**

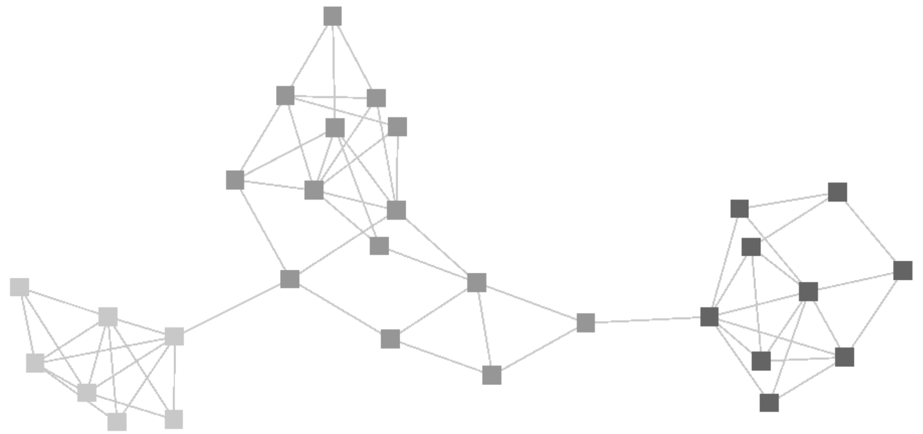
- Each subscriber is a vertex in the graph
- Interactions between subscribers are represented by edges
  - Taking into account both on-net and off-net
- Based on call data records for voice, SMS, MMS
  - Usually combining all interactions in a property representing the strength of the edge
- Graph clustering using well-known graph algorithms
  - Step 1 – Graph sparsification
  - Step 2 – Weakly Connected Components identification

# Sparsifying a graph

- Retaining edges within clusters, filtering edges across clusters
- Retaining all vertices



214 Edges

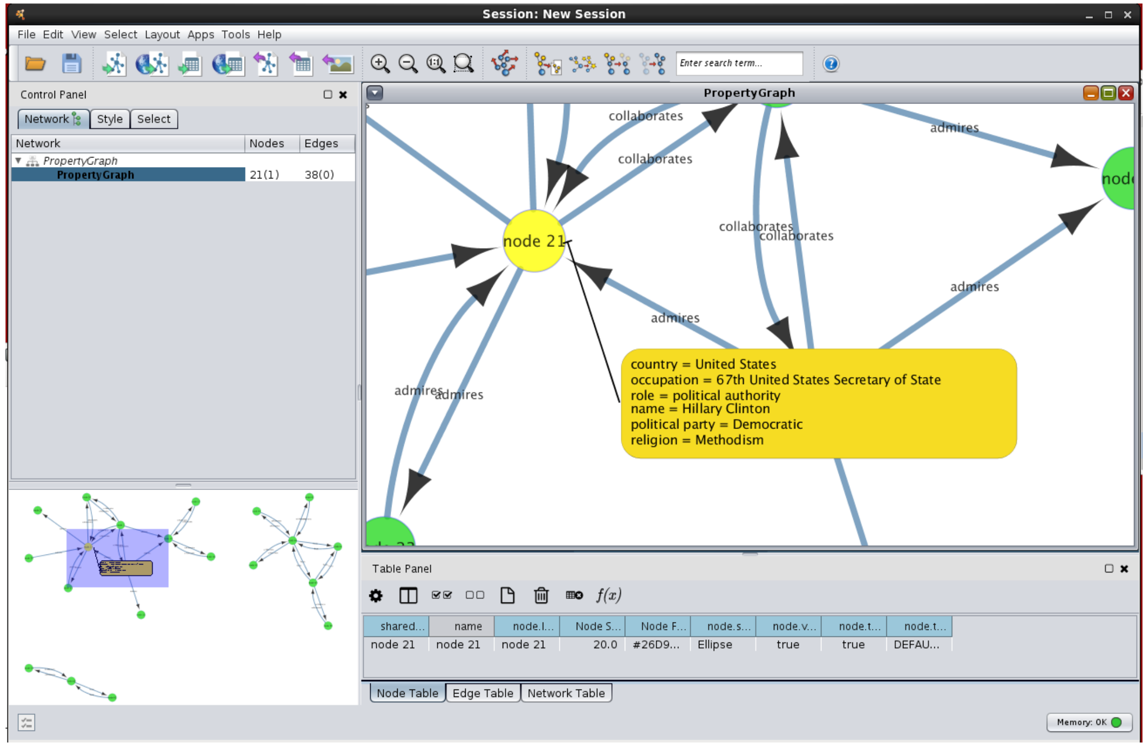


64 Edges

## Identifying leaders and further processing

- Determine subscribers with high centrality, cf. Marvel Hero analysis
  - Eigenvector centrality
  - Degree centrality
  - Pagerank
- Access mutated (output) property graph as relational data
  - Use Big Data SQL to access Hive table projected on property graph key/value pairs
- Churn modeling (Oracle Advanced Analytics – Data Mining predictive analytics)
  - Training, testing, and scoring

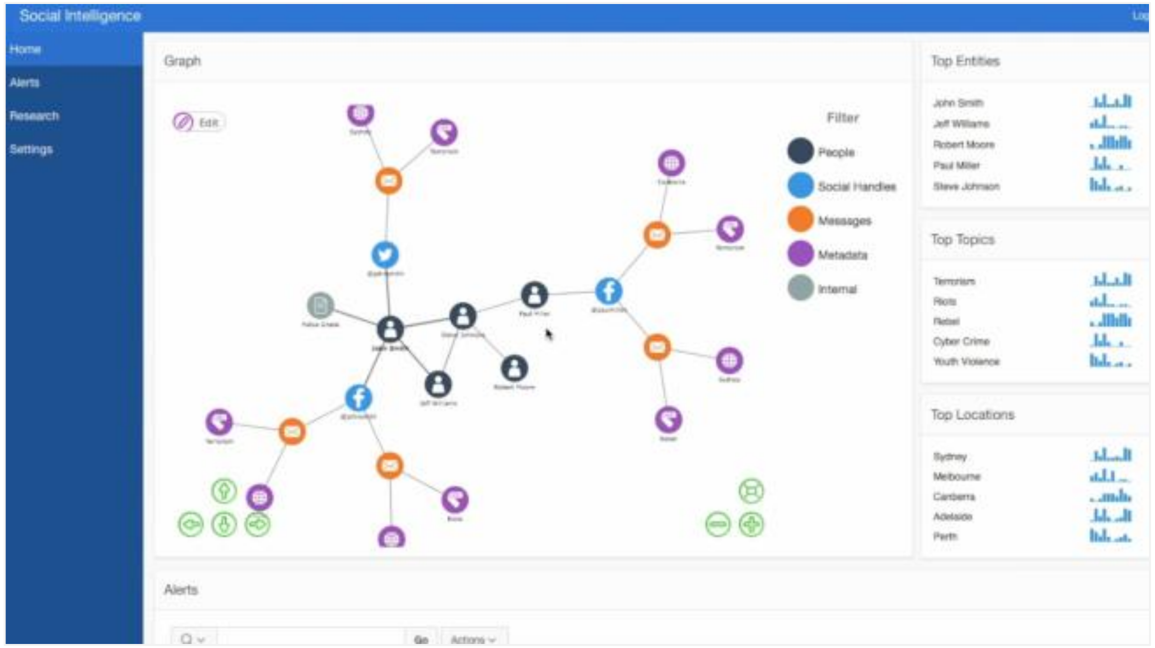
# Graph visualization – Cytoscape, Vis.js, ...



| Network | Nodes | Edges |
|---|---|---|
| PropertyGraph | | |
| PropertyGraph | 21(1) | 38(0) |

country = United States
occupation = 67th United States Secretary of State
role = political authority
name = Hillary Clinton
political party = Democratic
religion = Methodism

| shared... | name | node.l... | Node S... | Node F... | node.s... | node.v... | node.t... | node.t... |
|---|---|---|---|---|---|---|---|---|
| node 21 | node 21 | node 21 | 20.0 | #26D9... | Ellipse | true | true | DEFAU... |

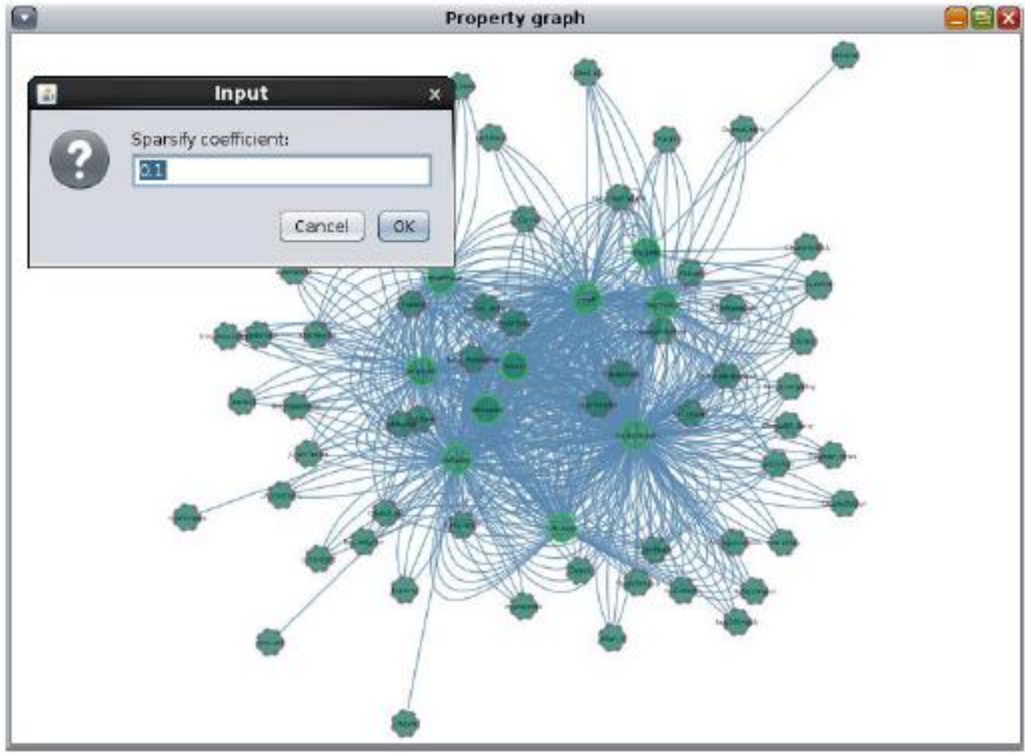


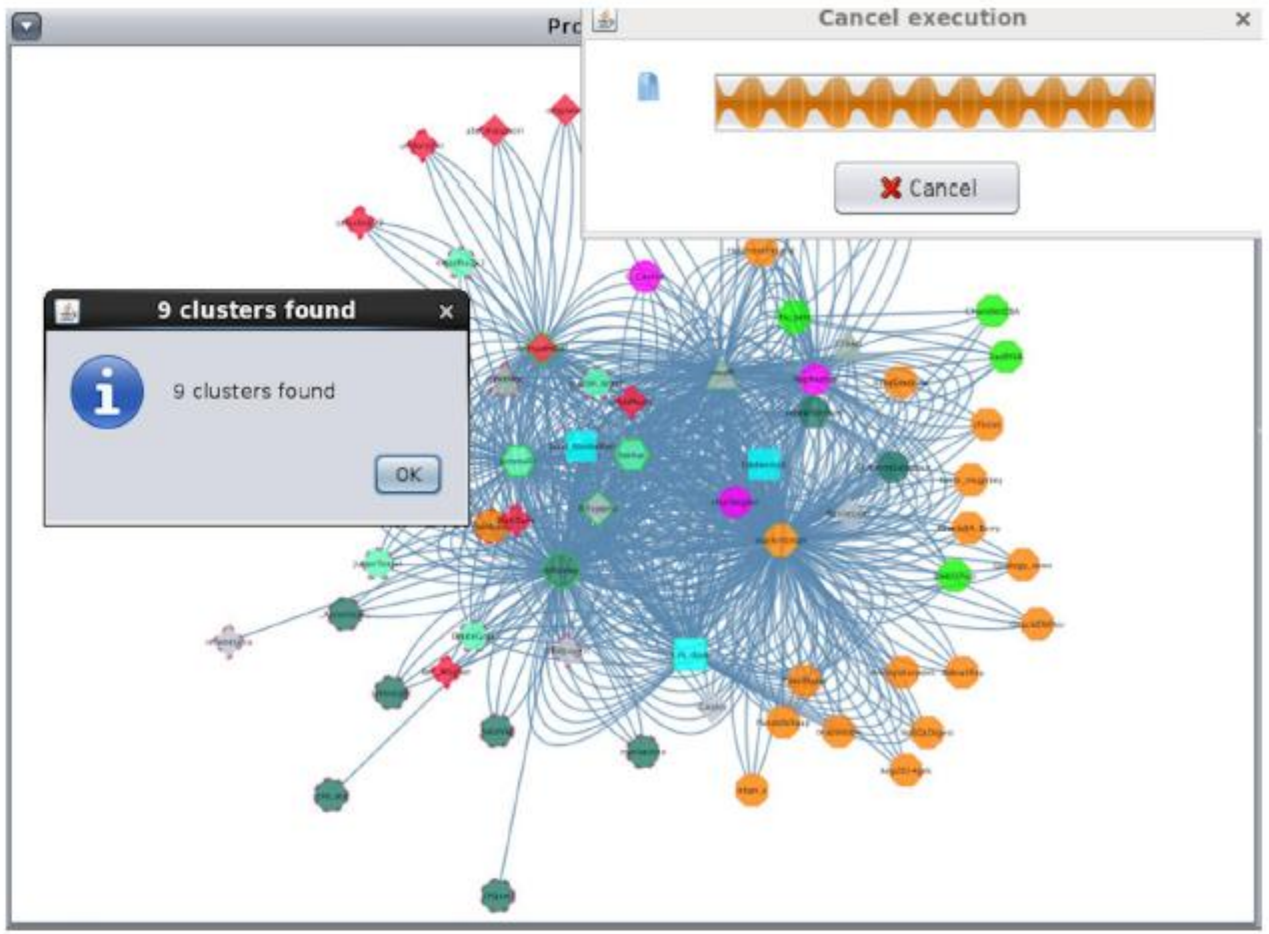
| Top Entities |
|---|
| John Smith |
| Jeff Williams |
| Robert Moore |
| Paul Miller |
| Steve Johnson |

| Top Topics |
|---|
| Terrorism |
| Riots |
| Rebel |
| Cyber Crime |
| Youth Violence |

| Top Locations |
|---|
| Sydney |
| Melbourne |
| Canberra |
| Adelaide |
| Perth |

# Mark Rittman's Twitter Analysis – stay tuned for more





## Summary

### Graph capabilities in Oracle Big Data Spatial and Graph

- Graph databases are powerful tools, complementing relational databases
  - Seamless integration possible
- Graph analytics offer new insight, especially into customer behaviour
- Oracle Big Data Spatial and Graph offers
  - Comprehensive analytics through various APIs
  - Scaleable, parallel in-memory processing
  - Secure and scaleable graph storage on Hadoop using Oracle NoSQL or HBase
- Runs on commodity hardware or BDA, both on-premise or in the Cloud
  - Part of Big Data as a Service offering

## I still owe you an answer ...

**Earth 616**

| **Measure** | **Top Hero** |
|---|---|
| Degree | Captain America |
| Page Rank | Spider-Man |
| Betweenness Centrality | Spider-Man |
| # Series | Spider-Man (by 3) |
| # Issues | Spider-Man |

ORACLE

## More information

- Oracle Big Data Spatial and Graph OTN Homepage
  - http://www.oracle.com/technetwork/database/database-technologies/bigdata-spatialandgraph/overview/index.html
- Oracle Big Data Spatial and Graph Data Sheet
  - http://download.oracle.com/otndocs/products/bigdata-spatialandgraph/bdsg-data-sheet.pdf
- Oracle Big Data Spatial and Graph Product Blog
  - https://blogs.oracle.com/bigdataspatialgraph/
- ... or try it out yourself using the latest „BigDataLite“ VM on OTN
  - v4.4 available for download here, samples now maintained on GitHub

## Integrated Cloud

Applications & Platform Services

## Appendix

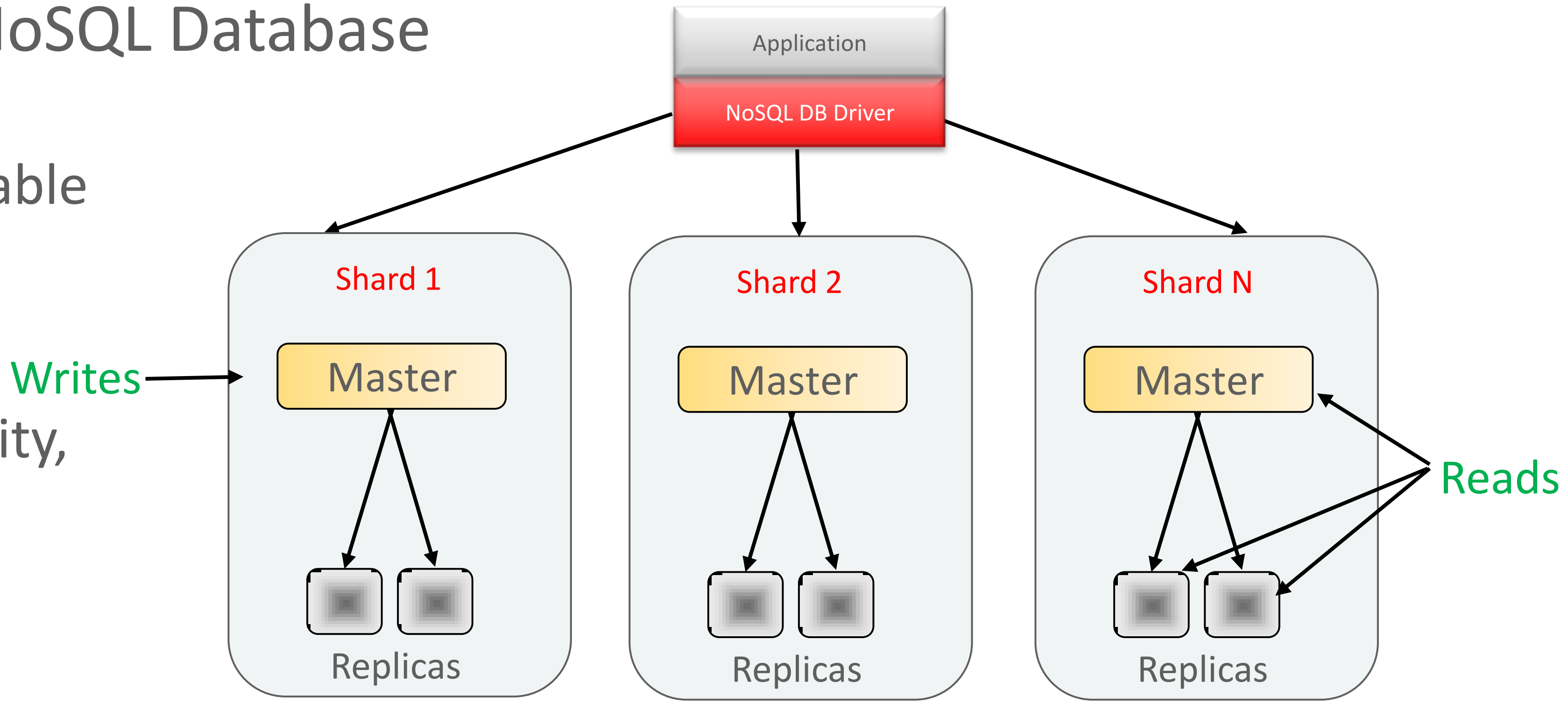
# Key features of Oracle Big Data Spatial and Graph

## Graph capabilities

- Complete platform:
  - Secure databases + Text indexing/search + Built-in analytics + Open source Java APIs & scripting languages for developers + Groovy console + integration w/ relational & SQL-based analytics
- Scalable:
  - Distributed database and text indexing/search; parallel in-memory analytics are concurrent & multiuser; filter queries refine graph data read into memory for analysis
- Fast: Parallel everywhere - load, query and in-memory analytics
- Flexible:
  - Deploy on-premise or in the Cloud, store in Hbase & NoSQL, text search w/ Lucene & SolrCloud, 3 ways to deploy in-memory analytics, extensible analytics architecture
- Open Source-based: Apache, Java, Tinkerpop APIs; Groovy, Python… scripting languages

# Key Features: Optimized Schema for Oracle NoSQL Database

- Overview of Oracle NoSQL Database
  - Distributed, highly scalable key-value store
  - Based on Berkley DB
  - KVStore, SN, RN, Capacity, Shard, RF, Partition
  - Secondary indices
  - Parallel scanner



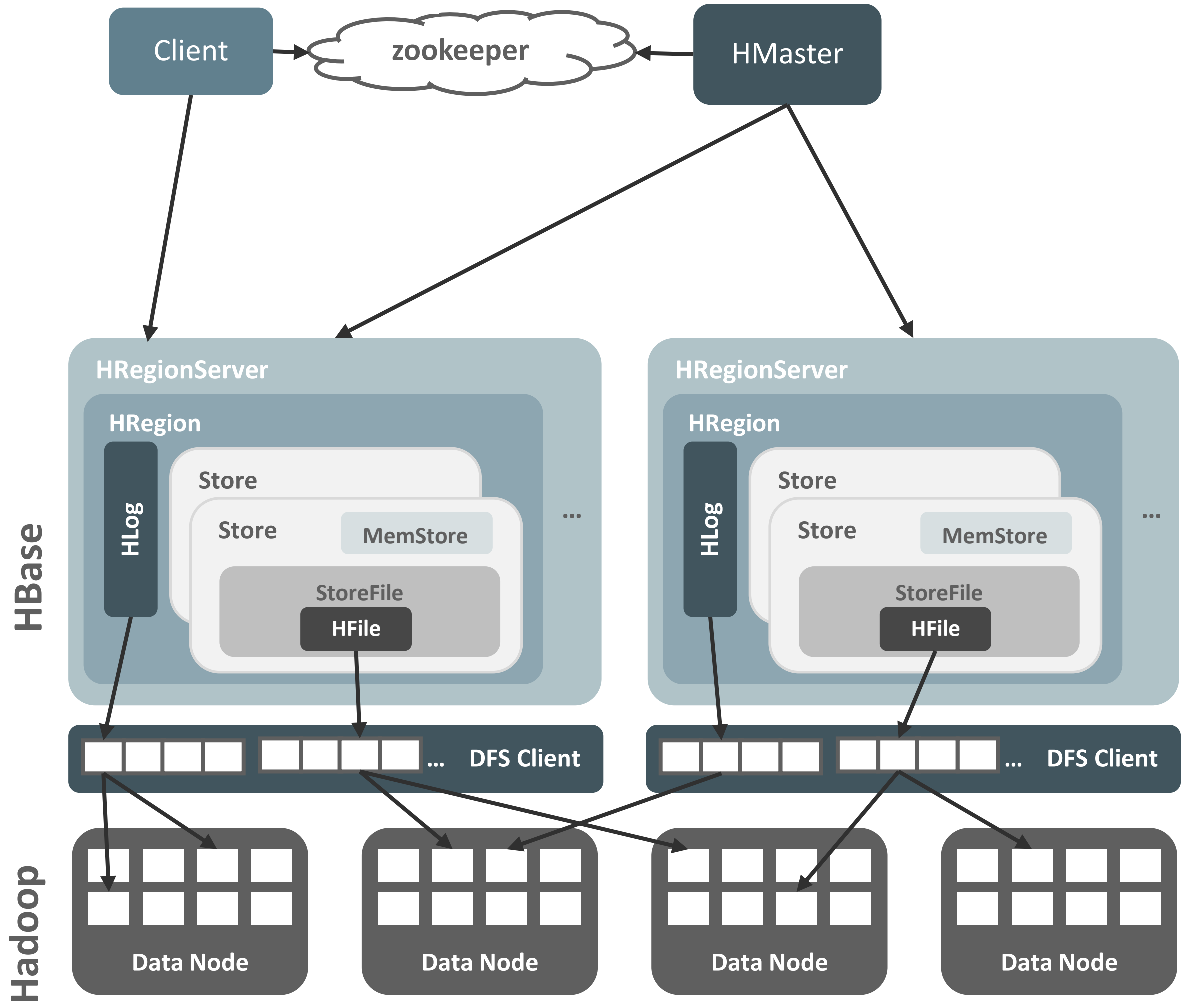
- Elastic shards (split, add, contract)
- Write to elected leader (Master)
- Read from any node (Replica)

# Key Features: Optimized Schema for Oracle NoSQL Database

- Tables used for managing property graph data
  - Vertex, Edge and Index tables
    - <graph>VT_
      - With secondary index on K/V
    - <graph>GE_
      - With secondary indices on (svid, dvid, el), (dvid, svid, el) and K/V
    - <graph>IT_
      - Child table and secondary index used
- Major features used
  - Secondary Indexing
  - Parallel scan
  - Customized scan
  - Parent/Child Table

# Key Features: Optimized Schema for Apache HBase

- Distributed column-oriented data store built on top of HDFS
- Based on Google's Bigtable model
- HBase files are internally stored in HDFS
- Automatically organize data in **Regions**, a subset of a table's rows, like horizontal range partitioning
- Data regions handled by **Region Servers** that serve data for reads and writes (using a log)
- A **Master** is responsible for coordinating the Region Servers, assigns regions, detects failures, etc.



# Key Features: Optimized Schema for Apache HBase

- Vertex Table (<graph>VT.)
  - Salting, K/V pairs, Adj. edges
- Edge Table (ET.)
  - SaltEID + EID acts as Row key
    - Salting for better data distribution
  - Edge label
    - Qualifier: l / Value: Edge label
  - In/out vertices
    - Qualifier: i/o   Value: SVID/DVID
  - Key/Value pairs
    - Qualifier: "k" + key
    - Value: Value + Datatype_ID
- Text Index Metadata (IT.)
- Secondary Indices (EI. And VI.)

| EID | |
|---|---|
| 100 | |

| Column Family | Qualifier | Value |
|---|---|---|
| e | ***l*** | collaborates |
| e | ***i*** | 1 |
| e | ***o*** | 2 |

| Column Family | Qualifier | Value |
|---|---|---|
| e | ***k***since | 2009-01-20***5*** |
| e | ***k***relation | political***1*** |
| e | ***k***weight | 1***2*** |

## Key Features: Lucene-based Text Search

- Integrated with Apache Lucene
  - open source text search engine library
  - inverted index, ranked searching, fuzzy matching …
- Supports manual and auto indexing of Graph elements
  - Auto Index

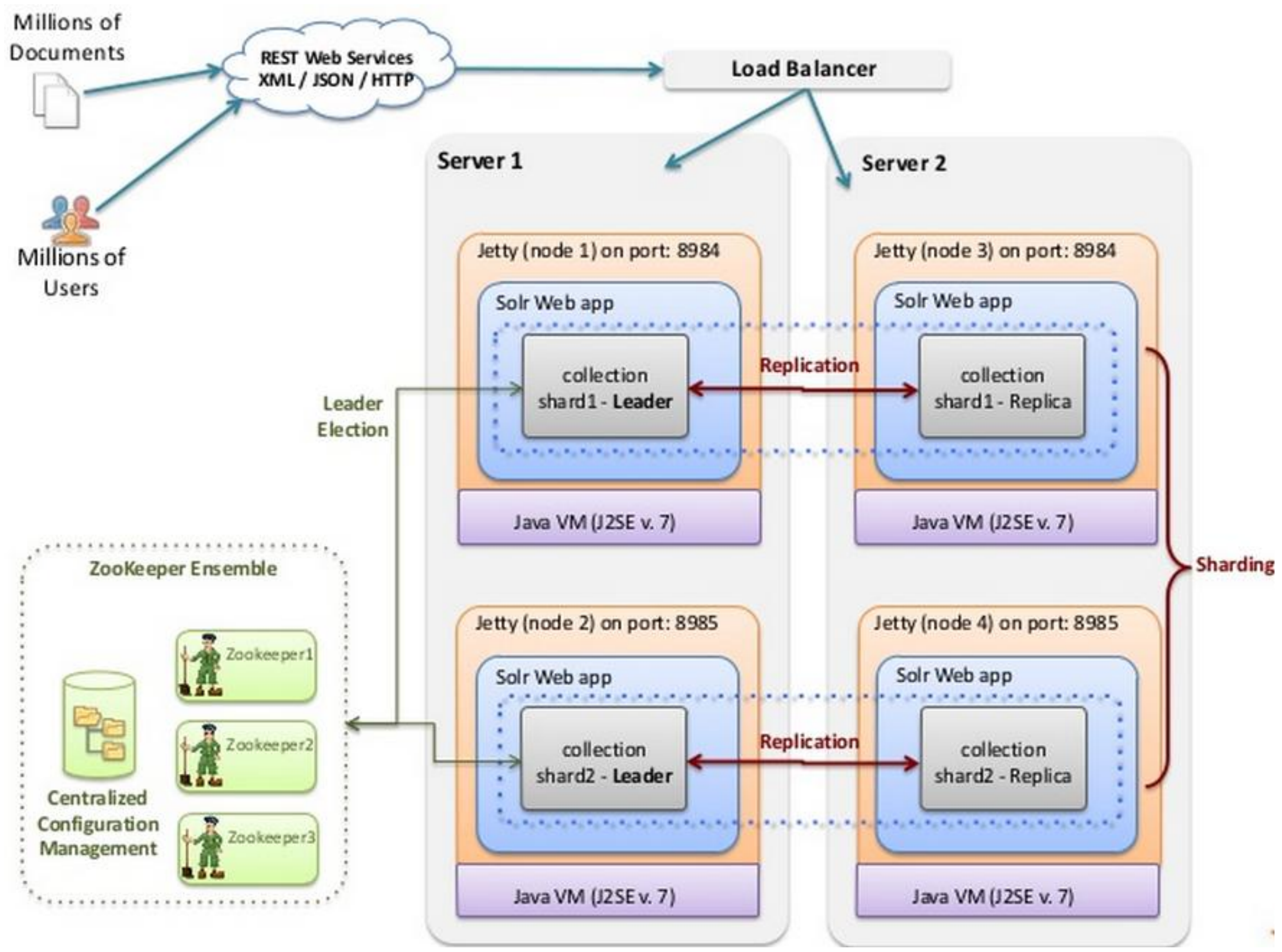
```
oraclePropertyGraph.createKeyIndex(“name”, Vertex.class);
oraclePropertyGraph.getVertices(“name”, “*Obama*”, true);
```

  - Manual index

```
oraclePropertyGraph.createIndex(“my_index", Vertex.class);
indexVertices = oraclePropertyGraph.getIndex(“my_index” ,  Vertex.class);
indexVertices.put(“name”, “Barack Obama”,  v);
```

# Key Features: SolrCloud Based Text Search

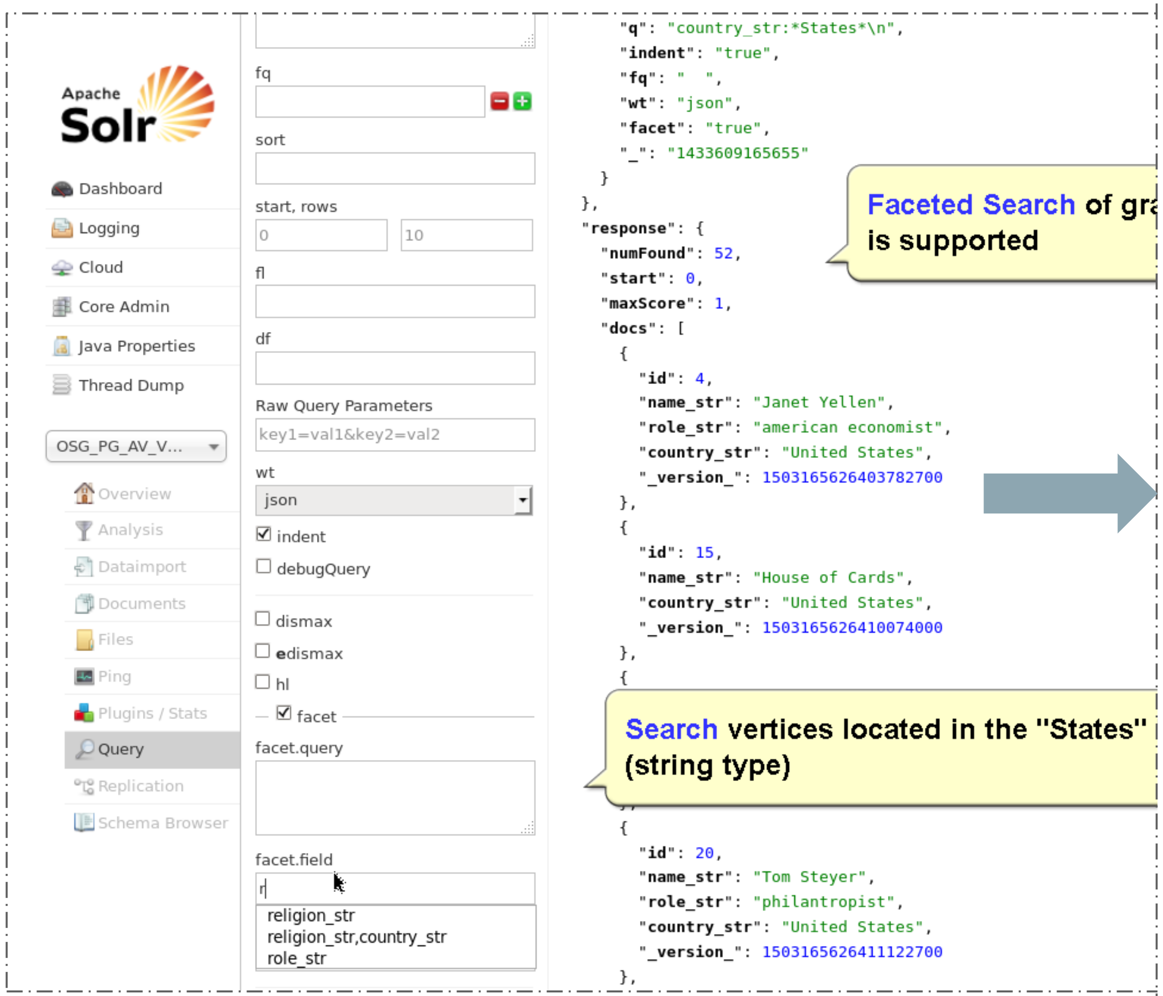
- Integrate with Apache Solr/SolrCloud
  - SolrCloud enables horizontal scaling of text index with
    - Sharding
    - Replication



Millions of Documents
Millions of Users
REST Web Services XML / JSON / HTTP
Load Balancer
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
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
ZooKeeper Ensemble
Zookeeper1
Zookeeper2
Zookeeper3
Centralized Configuration Management

http://www.slideshare.net/thelabdude/solr-exchange-introtosolrcloud

# SolrCloud: Fuzzy and Faceted Search of Graph Elements



fq

sort

start, rows
0 10

fl

df

Raw Query Parameters
key1=val1&key2=val2

wt
json

☑ indent
☐ debugQuery

☐ dismax
☐ edismax
☐ hl
☑ facet

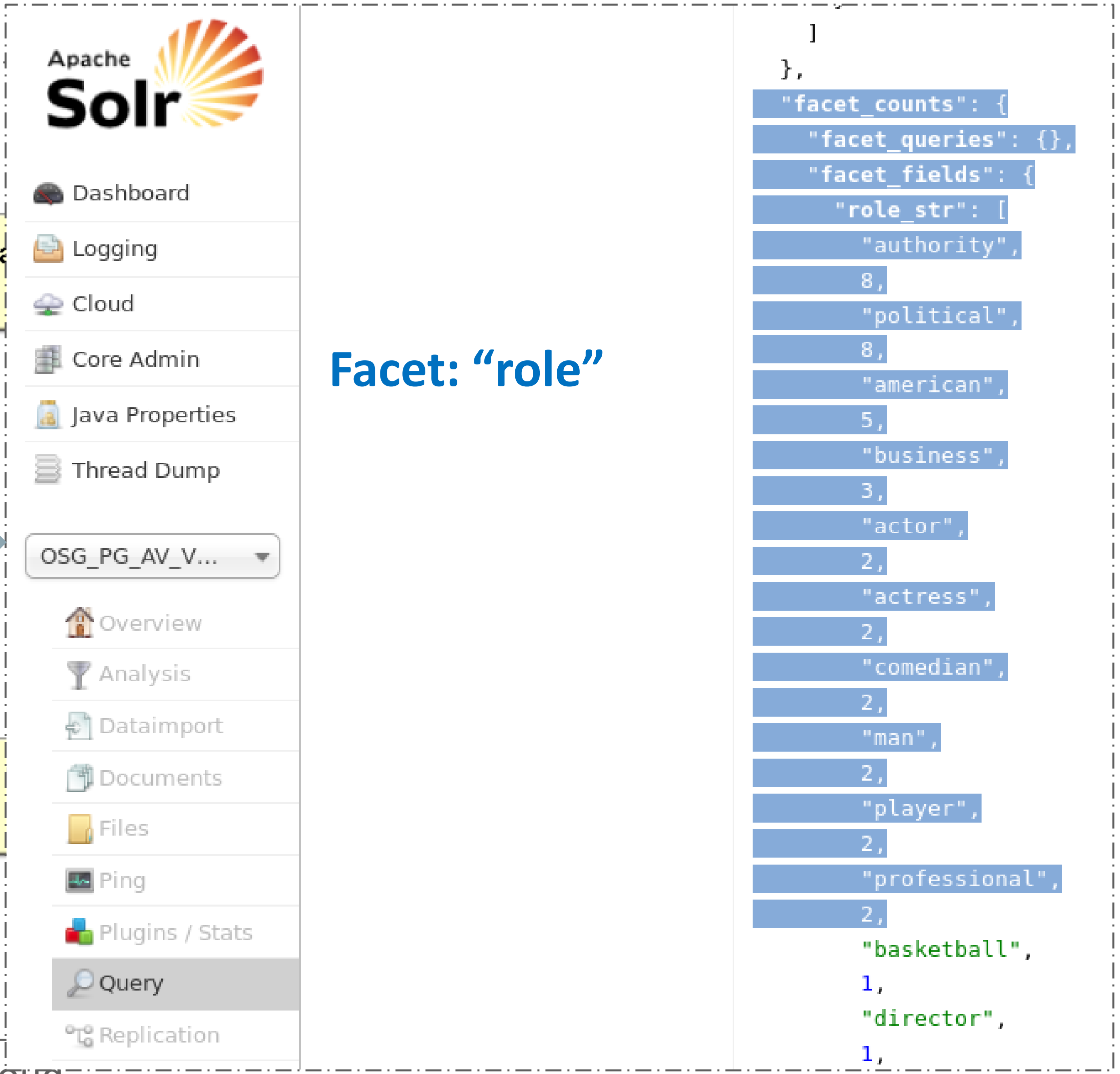
facet.query

facet.field
r
religion_str
religion_str,country_str
role_str

```
      "q": "country_str:*States*\n",
      "indent": "true",
      "fq": "  ",
      "wt": "json",
      "facet": "true",
      "_": "1433609165655"
    }
  },
  "response": {
    "numFound": 52,
    "start": 0,
    "maxScore": 1,
    "docs": [
      {
        "id": 4,
        "name_str": "Janet Yellen",
        "role_str": "american economist",
        "country_str": "United States",
        "_version_": 1503165626403782700
      },
      {
        "id": 15,
        "name_str": "House of Cards",
        "country_str": "United States",
        "_version_": 1503165626410074000
      },
      {
      ...
      {
        "id": 20,
        "name_str": "Tom Steyer",
        "role_str": "philantropist",
        "country_str": "United States",
        "_version_": 1503165626411122700
      },
```

**Faceted Search** of gra
is supported

**Search** vertices located in the "States"
(string type)



**Facet: "role"**

```
      ]
    },
    "facet_counts": {
      "facet_queries": {},
      "facet_fields": {
        "role_str": [
          "authority",
          8,
          "political",
          8,
          "american",
          5,
          "business",
          3,
          "actor",
          2,
          "actress",
          2,
          "comedian",
          2,
          "man",
          2,
          "player",
          2,
          "professional",
          2,
          "basketball",
          1,
          "director",
          1,
```

http://www.slideshare.net/thelabdude/solr-exchange-introtosolrcloud

# Key Features: Parallel Property Graph Data Loader

- 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

```
OraclePropertyGraphDataLoader opgdl = OraclePropertyGraphDataLoader.getInstance();
vfile = "/home/alwu/pg-bda-nosql/demo/connections.opv";
efile = "/home/alwu/pg-bda-nosql/demo/connections.ope";
opgdl.loadData(opg, vfile, efile, 48);
```

ORACLE

# Key Features: Multiple Property Graph Data Format Support

- GML, GraphML, GraphSON
- Oracle-defined Property Graph flat files
  - Vertex file, Edge file
  - Supports basic data types + Date with Timezone + Serializable objects
  - Allows multiple data types to be associated with one key
  - UTF8 based
  - Vertex Example

```
1,name,1,Barack%20Obama,,
1,age,2,,53,
1,likes,5,,,2009-01-20T00:00:00.000-05:00
1,occupation,1,44th%20president%20of%20United%20States%20of%20America,,
2,likes,1,soccer,,
```

**Same Key, different value types allowed!**

## Key Features: Groovy Scripts

- Supports groovy scripts to allow easy property graph data exploration

```
$ ./gremlin-opg-nosql.sh
opg-nosql> hhosts = new String[1];
opg-nosql> hhosts[0] = "bigdatalite:5000";
opg-nosql> cfg =
  GraphConfigBuilder.forNosql().setName("myGraph").setHosts(Arrays.asList(hhosts)).setStoreName("mystore"
  ).addEdgeProperty("lbl", PropertyType.STRING, "lbl").addEdgeProperty("weight", PropertyType.DOUBLE,
  "1000000").build();

opg-nosql> opg = OraclePropertyGraph.getInstance(cfg);
opg-nosql> opgdl = OraclePropertyGraphDataLoader.getInstance();
opg-nosql> opgdl.loadData(opg, new FileInputStream("../../data/connections.opv"),
                                    new FileInputStream("../../data/connections.ope"), 1, 1, 0, null);
opg-nosql> opg.getVertices();
opg-nosql> opg.getEdges();
```

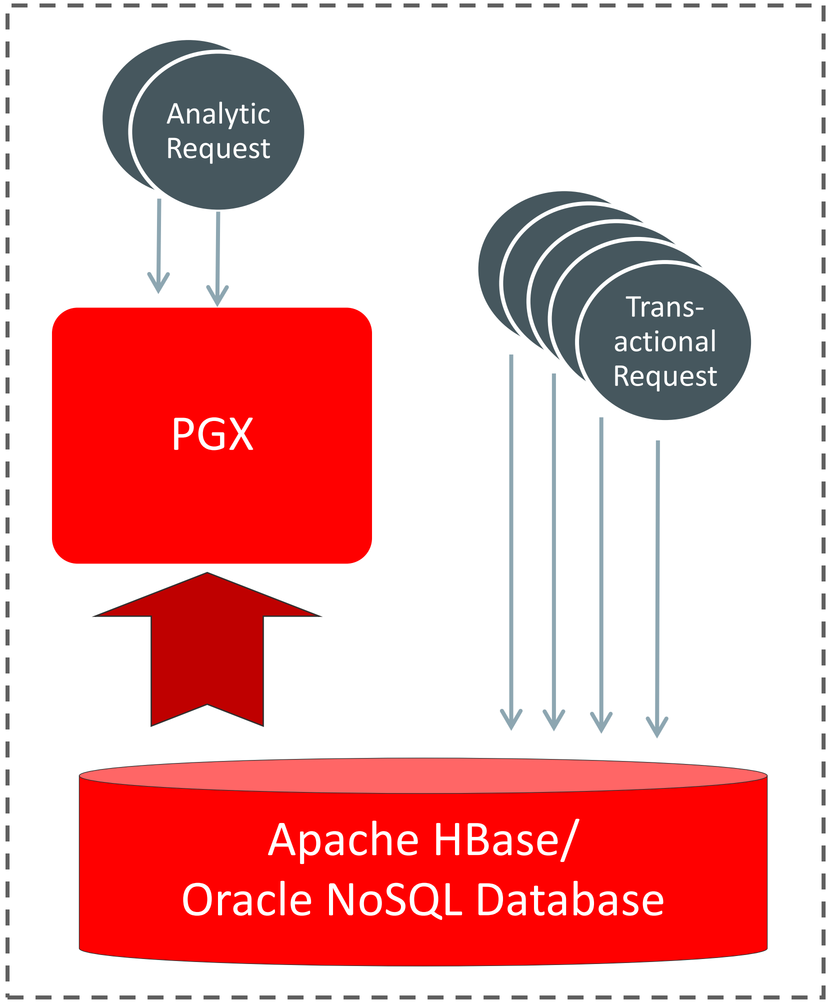
**No Compilation Required!**

ORACLE

## Key Features: In-Memory Parallel Graph Analytics

### Parallel Graph AnalytiX (**PGX**)

- An in-memory, parallel framework for fast graph analytics
  - Exploits the architecture of modern servers
    - The computation is parallelized using multiple CPU cores
    - The non-sequential data-access is mitigated with large DRAMs
  - Can read a graph (through data access layer) from Oracle NoSQL Database or Apache HBase
    - Parallel reading supported
  - Handles analytic workloads while the data access layer handles transactional workloads
  - Supports concurrent sessions and multiple users/graphs

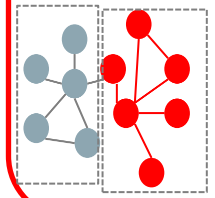


# Key Features: In-Memory Parallel Graph Analytics

## Built-in Algorithms and Graph Mutation

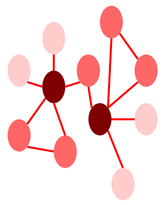
### A rich set of built-in, parallel algorithms

**Detecting Components and Communities**



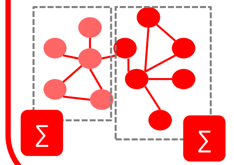
Tarjan's, Kosaraju's, Weakly Connected Components, Label Propagation (w/ variants), Spasification

**Ranking and Walking**



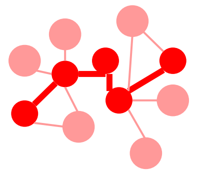
Pagerank, Personalized Pagerank, Betwenness Centrality (w/ variants), Closeness Centrality, Degree Centrality, Eigenvector Centrality, HITS, Random walking and sampling (w/ variants)

**Evaluating Community Structures**



Conductance, Modularity, Clustering Coefficient (Triangle Counting)

**Path-Finding**

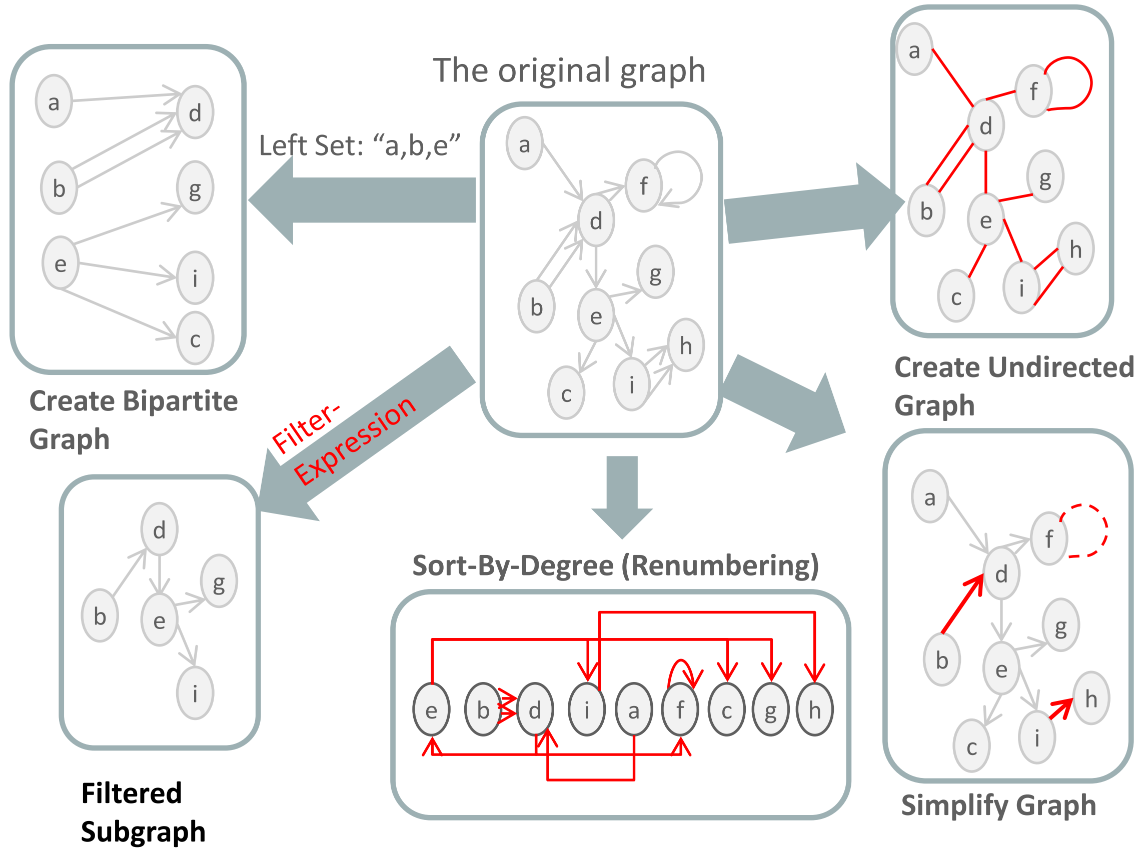


Hop-Distance (BFS)
Dijkstra's,
Bi-directional Dijkstra's
Bellman-Ford's

**Link Prediction** SALSA (Twitter's Who-to-follow)

**Other Classics** Vertex Cover

### Parallel graph mutation operations



## Key Features: Flexible Graph Analytics Deployment

- 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
  - Yarn Integration
    - Let Yarn find the necessary CPU/RAM to deploy the in memory graph analytics

ORACLE

## Key Dependencies

- Hadoop Distribution
  - CDH 5.2.x, 5.3.x, 5.4.0
- Database Versions
  - Oracle NoSQL Database (3.2.5)
  - Apache HBase (0.98.6, 1.0)
- Big Data Appliance (4.2, 4.3)
- J2EE Web Container
  - WebLogic Server (11g and 12c), Apache Tomcat (7.x, 8.x), Eclipse Jetty (9.2.x)
- Groovy
  - v1.8.9 for data access layer
  - v2.5 (installation required) for in-memory analytics shell

ORACLE

## Key Dependencies

- TinkerPop Blueprints 2.3.0
  - Blueprints Java APIs 2.3.0
  - Gremlin 2.3.0
  - Pipes 2.3.0 (a dataflow framework)
- Apache Lucene 4.5.0
- Apache Solr (SolrCloud)  4.4.0 and 4.10.3
- OracleJDK 8u45
- Release notes – My Oracle Support  Doc ID 2010123.1

ORACLE

## Hardware and Software
## Engineered to Work Together