

ORACLE®

Analyzing Stack Exchange data using Property Graph in Oracle

Oskar van Rest (Oracle)

Eleventh LDBC TUC Meeting, University of Texas at Austin

June 8, 2018

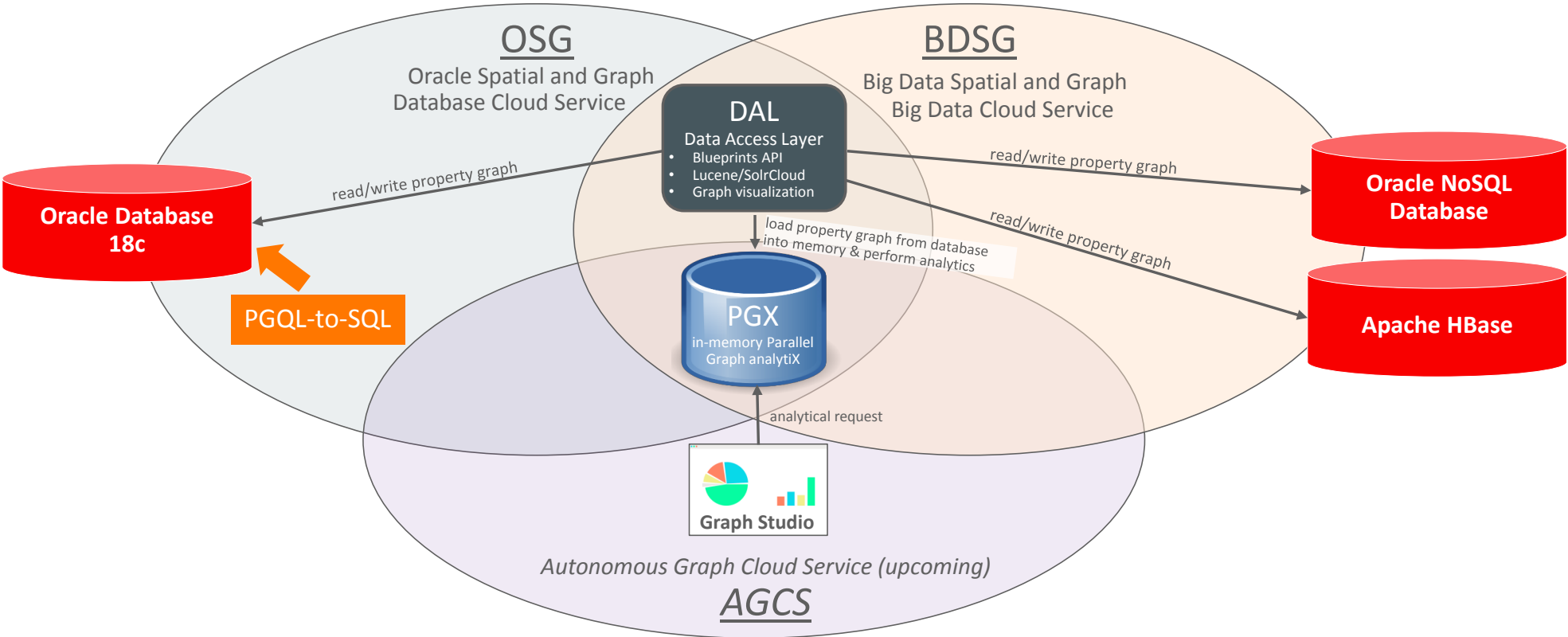
ORACLE

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. | Confidential – Oracle Internal

Safe Harbor Statement

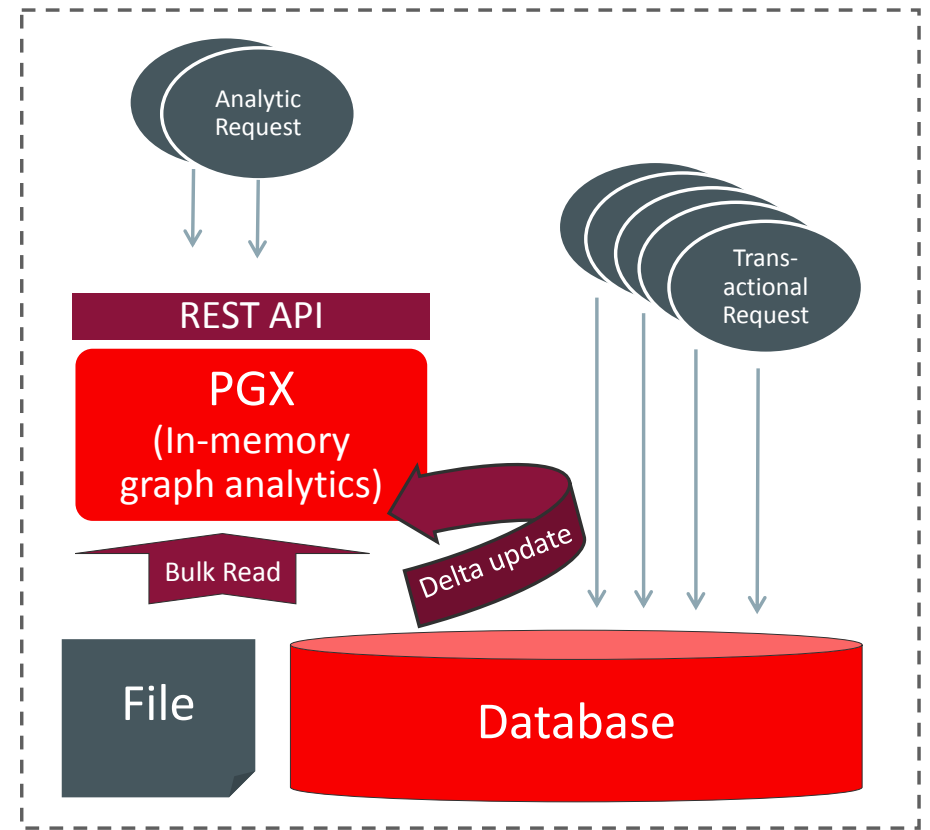
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Property Graph products at Oracle



Analytical and Transactional processing

- **Transactional processing** through **database**
- **Analytical processing through PGX** (an in-memory Parallel Graph analytiX engine)
- Approach
 - Read snapshot of graph data from database (or file)
 - Support delta-update from transactional changes in database
 - Process analytic requests efficiently in-memory
 - E.g. graph query (PGQL) or graph algorithm



Graph queries in Oracle Spatial and Graph (OSG)

Graph Query (PGQL)

```
/* find friends of friends of Clara */  
SELECT fof.name  
FROM myGraph  
MATCH (p:Person) -/:knows{2}/-> (fof:Person)  
WHERE p.name = 'Clara'
```

In-memory Analyst (PGX)

- Excels in computationally intense workloads and recursive queries
- Can combine graph algorithms with graph queries

Analytical graph query

PGQL-to-SQL

- Excels in workloads with mixtures of read and write queries
- Can query data sets that don't fit into the memory of a single machine

Transactional graph query

In-memory Analyst (PGX)



Oracle RDBMS



- ### Bulk Update
- Synchronizes an in-memory graph snapshot with graph changes from RDBMS
 - Every x seconds/minutes/hours or upon request

PGQL – Property Graph Query Language

- Core Features

- **SQL** alignment

- **SELECT .. FROM .. WHERE ..**
- Grouping and aggregation: **GROUP BY, COUNT, AVG, MIN, MAX, SUM**
- Sorting: **ORDER BY, ASC, DESC**

- Graph **pattern matching**

- Define a high-level pattern and match all the instances in the data graph

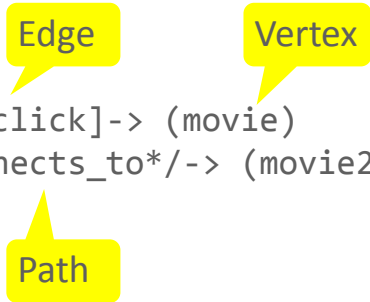
- **Regular path expressions**

- Typically recursive in nature
 - E.g. can I reach from vertex A to vertex B via any number of edges?

Example query:

```
PATH connects_to AS (m1) -> () <- (m2)

SELECT customer.first_name, movie2.title
FROM myMovieGraph
MATCH (customer) -[:click]-> (movie)
      , (movie) -/:connects_to*/-> (movie2)
GROUP BY ..
ORDER BY ..
LIMIT ..
OFFSET ..
```



Example: Network Impact Analysis

- How does **network disruption** impacts reachability between electric devices?

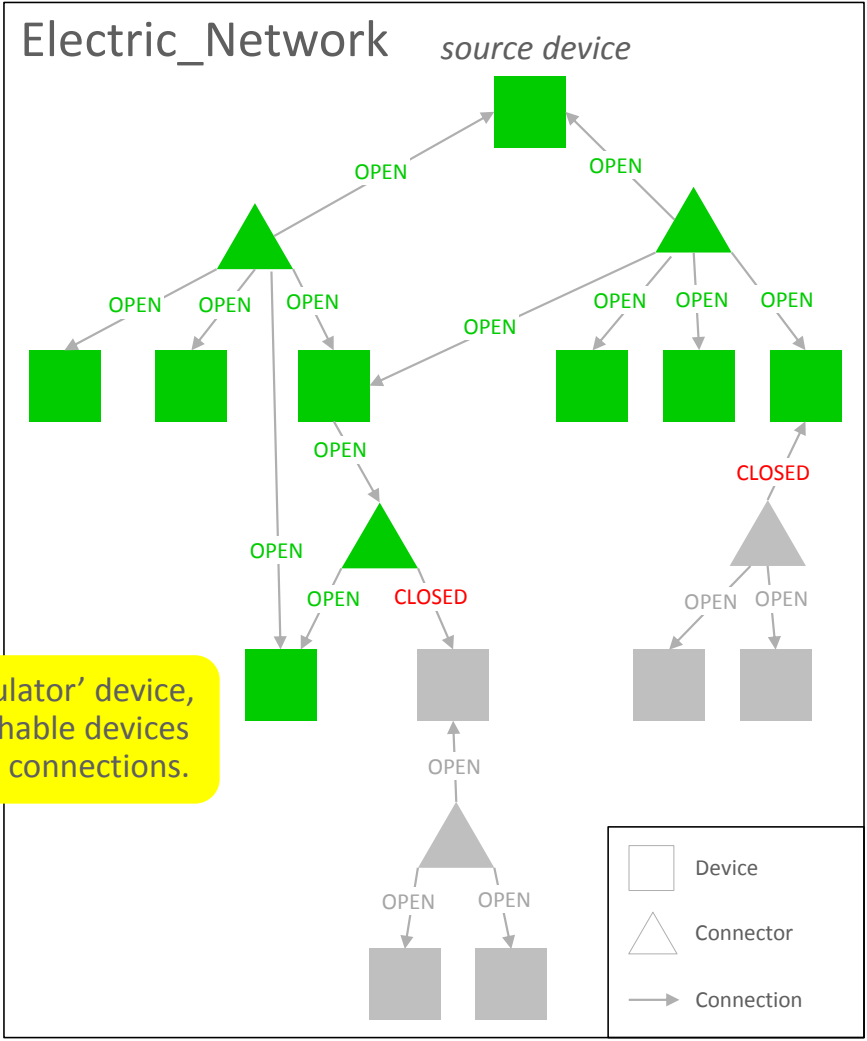
```

PATH connects_to
  AS (from) <-[c1]- (connector) -[c2]-> (to)
  WHERE c1.status = 'OPEN'
        AND c2.status = 'OPEN'
SELECT n.nickname, COUNT(m)
FROM Electric_Network
MATCH (n:Device) -/:connects_to*/-> (m:Device)
WHERE java_regexp_like(n.nickname, 'Regulator')
      AND n <> m
GROUP BY n
ORDER BY COUNT(m) DESC, n.nickname
    
```

Query: For each 'Regulator' device, show number of reachable devices following only 'OPEN' connections.

Example result

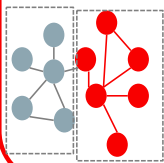
n.nickname	COUNT(m)
Regulator, VREG2_A	1596
Regulator, VREG4_B	1537
Regulator, VREG4_C	1537
Regulator, HVMV_Sub_RegA	3
Regulator, HVMV_Sub_RegB	3



Built-in Analytics and Graph Mutations

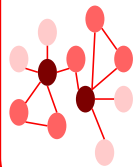
- Rich set of built-in (parallel) graph algorithms

Detecting Components and Communities



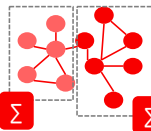
Tarjan's, Kosaraju's, Weakly Connected Components, Label Propagation (w/ variants), Soman and Narang's Sparcification

Ranking and Walking



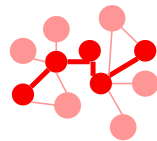
Pagerank, Personalized Pagerank, Betweenness Centrality (w/ variants), Closeness Centrality, Degree Centrality, Eigenvector Centrality, HITS, Random walking and sampling (w/ variants)

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

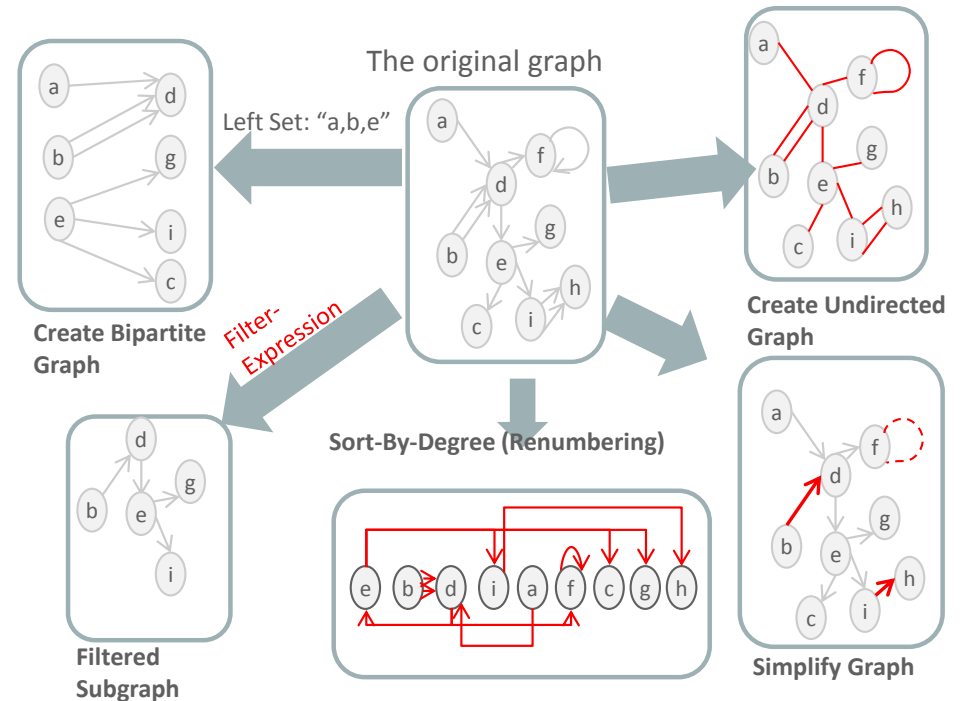
Link Prediction

SALSA (Twitter's Who-to-follow)

Other Classics

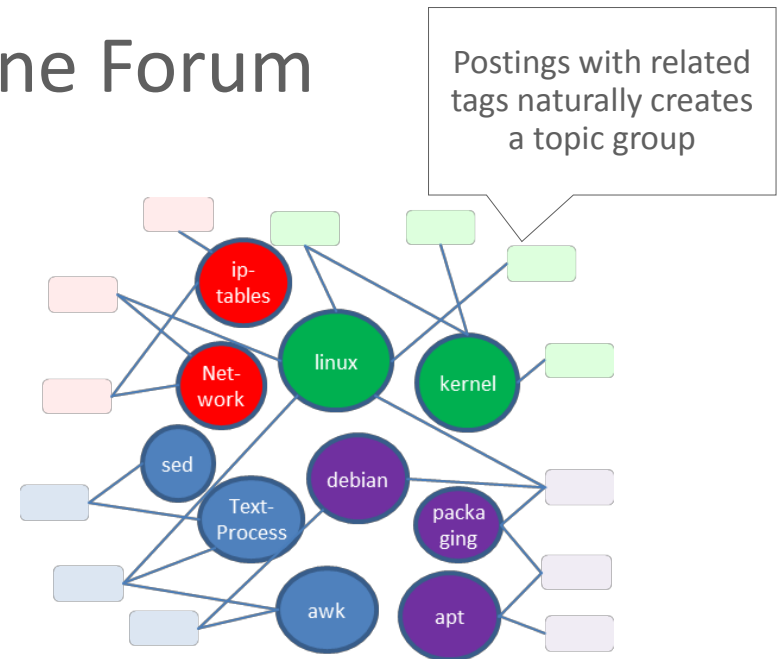
Vertex Cover, Minimum Spanning-Tree (Prim's)

- as well as parallel graph mutation operations



Example: Topic analysis in an Online Forum

- Analysis Goals:
 - Identify popular *topics* in on-line forum
 - Understand how these topics evolve
 - Detect expert users in certain topics
- Graph Approach
 - Create graph from postings and tags
 - Apply graph partitioning (**community detection**) algorithms



Comparing to traditional ML approach (e.g. LDA), this approach often results better quality of answer, with less susceptibility to hyper-parameters

Topic (tags)
Bash, shell-script, shell, scripting
Linux, ssh, grep, linux-kernel, files, kernel, regular-expression
Networking, network-interface, dns, ip, raspberry-pi, raspbian, routing
Centos, python, yum, rpm, mysql, php, postgresql, software-installation, repository
Permissions, sudo, users, root, sort, aix, chmod, group, executable, acl

Topic (tags)
Bash, shell-script, scripting, mmv
Text-processing, awk, sed, grep, perl
Centos, rhel, yum, rpm, repository, rpmbuild, redhat-satellite, drupal
Networking, ip, routing, dhcp, tcp, router, iproute, isc-dhcp, pcap
Ssh, openssh, sshd, ssh-tunneling, key-authentication, ssh-config

Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

ORACLE®