Graphs in Manufacturing

mark.quinsland@neo4j.com

May 24, 2018

neo4j

Neo4j - The Graph Company

Industry's Largest Dedicated Investment in Graphs

neo4j

- Creator of the Neo4j Graph Platform
- ~200 employees
- HQ in Silicon Valley, other offices include London, Munich, Paris and Malmö (Sweden)
- $80M in funding from Fidelity, Sunstone, Conor, Creandum, and Greenbridge Capital
- Over 10M+ downloads
- 270+ enterprise subscription customers with over half with >$1B in revenue

Adoption

7/10	Top Retail Firms
12/25	Top Financial Firms
8/10	Top Software Vendors

Ecosystem

720+	Startups in program
270+	Enterprise customers
100+	Partners
53K+	Meet up members
450+	Events per year

Customers

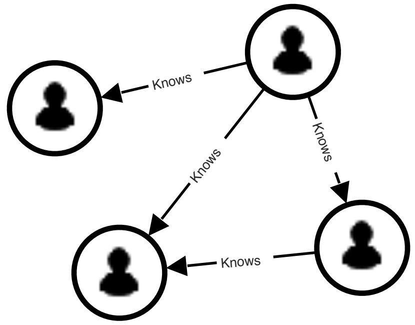
LinkedIn
UBS
COMCAST
CISCO
Lufthansa
Walmart
ebay

Partners

IBM
Microsoft Azure
LINKURIOUS
pitney bowes
GraphAware
structr

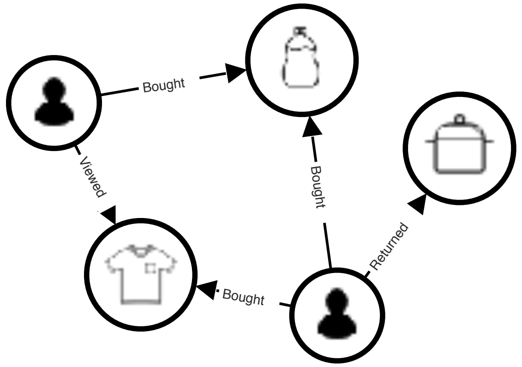
Why Graph?
The Rise of Connections in Data

Data connections are increasing as rapidly as data volumes



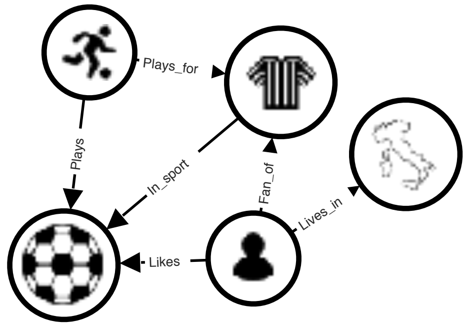
Networks of People

E.g., Employees, Customers, Suppliers, Partners, Influencers



Business Processes

E.g., Risk management, Supply chain, Payments



Knowledge Networks

E.g., Enterprise content, Domain specific content, eCommerce content

Harnessing Connections Drives Business Value

Digital Transformation Megatrends

Data Driven Discovery & Innovation

Drug Discovery
Intelligence and Crime Detection
Product & Process Innovation

Hyper Personalization

Product Recommendations
Personalized Health Care
Media and Advertising

Enhanced Decision Making

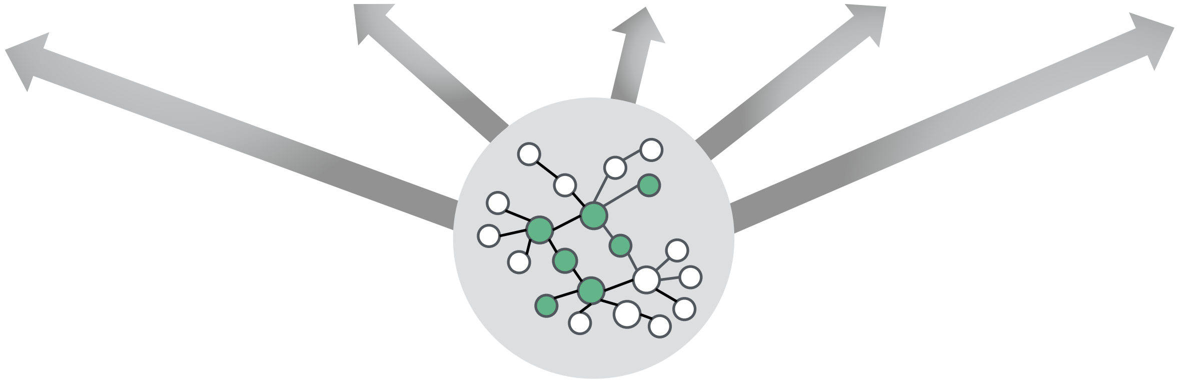
Fraud Prevention
Network Analysis
Law Enforcement

Massive Data Integration

360 view of customer
Compliance
Optimize Operations

AI & Machine Learning

Price optimization
Product Recommendations
Resource allocation



Connected Data at the Center

neo4j

Graphs Are a Logical Choice for Many Areas

Organization

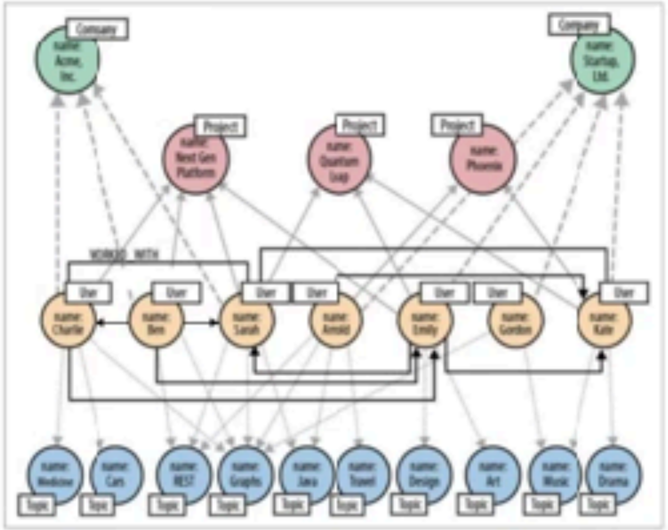


Figure 5-7. Talent.net graph enriched with WORKED_WITH relationships

Identity & Access

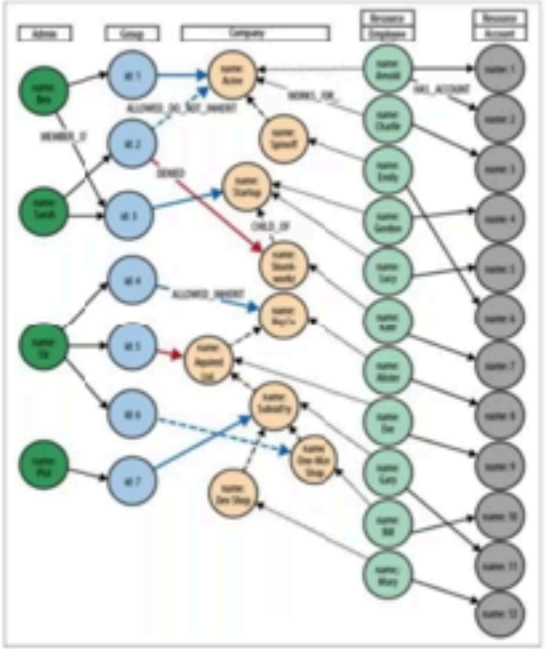


Figure 5-8. Access control graph

Network & IT Ops

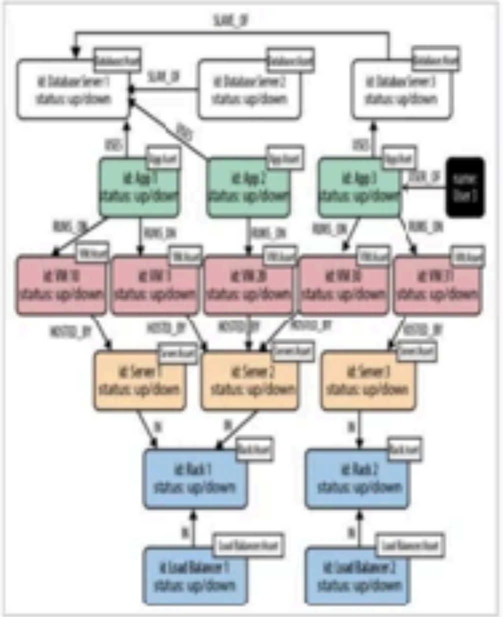
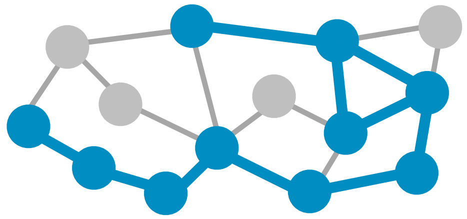


Figure 3-5. Example graph for the data center deployment scenario

Relationship Queries Strain Traditional Databases

A single query can touch a ***lot of data***



Queries can take non-sequential, ***arbitrary paths*** through data

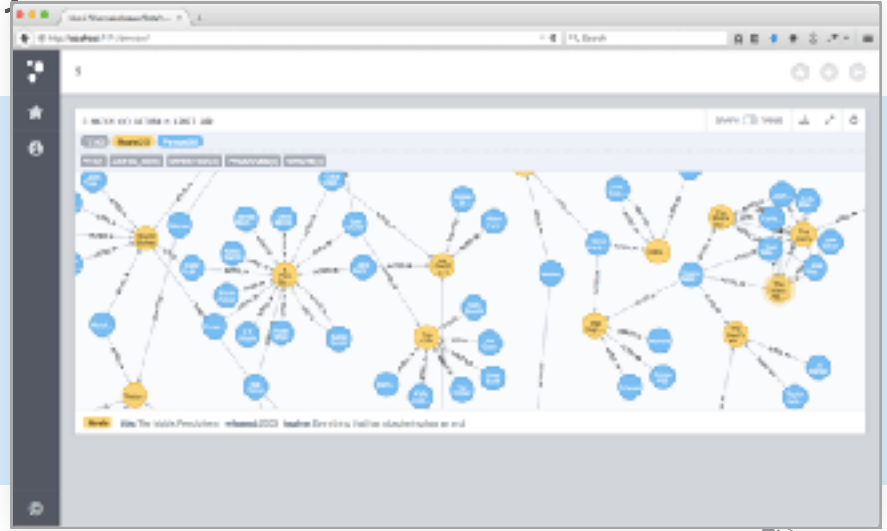
Real-time** queries need speed and **consistent response times

Queries must ***run reliably*** with ***consistent results***

Neo4j - The #1 Platform for Connected Data

Neo4j is an ***enterprise-grade <u>native</u> graph database*** that enables you to:

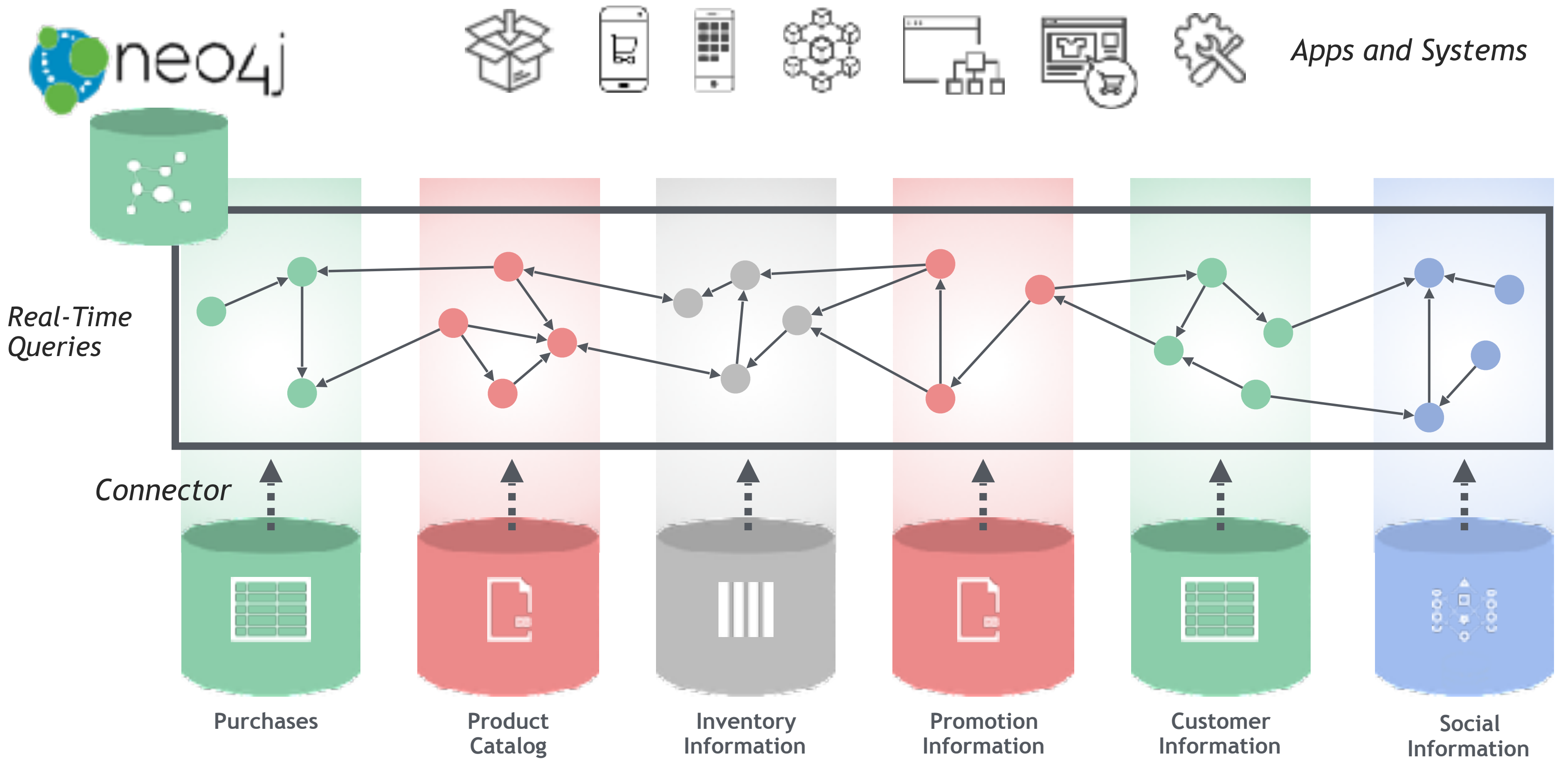
- **Store and query** data relationships
- **Traverse** any levels of depth on real-time
- **Add and connect** new data on the fly



Designed, built and tested *<u>natively</u>* for graphs from the start to ensure:

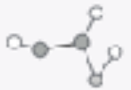
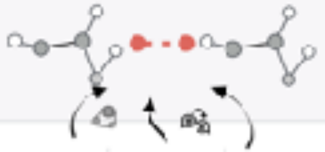
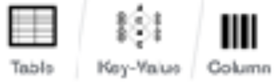
- Performance
- ACID Transactions
- Agility
- Developer Productivity
- Hardware Efficiency

Neo4j – Brings Together and Mobilizes Your Data



Lessons Learned: Ten Year Head Start

Native Connectedness Differentiates Neo4j

	neo4j Native Graph DB	Non-Native Graph DB	RDBMS
Conceive			
Code	Cypher (graphs)-[are]->(everywhere)	Cypher/Gremlin /Proprietary	SQL
Compute			
Store			

Optimized for graph workloads

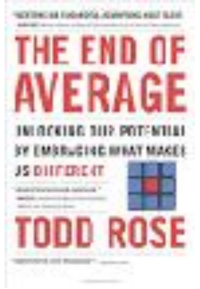
NoSQL Databases Are NOT All the Same

MongoDB was clearly the wrong NoSQL database for providing recommendations and was a significant factor in a $250 million drop in market cap.

‘…we’ve heard one former employee … describe it as a “technical s**t show.” ’

https://techcrunch.com/2017/06/09/pandora-raises-480m-from-siriusxm-sells-ticketfly-to-eventbrite-for-200m

Transformative Customer Experiences



Real-time promotion recommendations

FORTUNE 50 RETAIL

- Record “Cyber Monday” sales
- About 35M daily transactions
- Each transaction is 3-22 hops
- Queries executed in 4ms or less
- Replaced IBM Websphere commerce

Marriott’s Real-time Pricing Engine

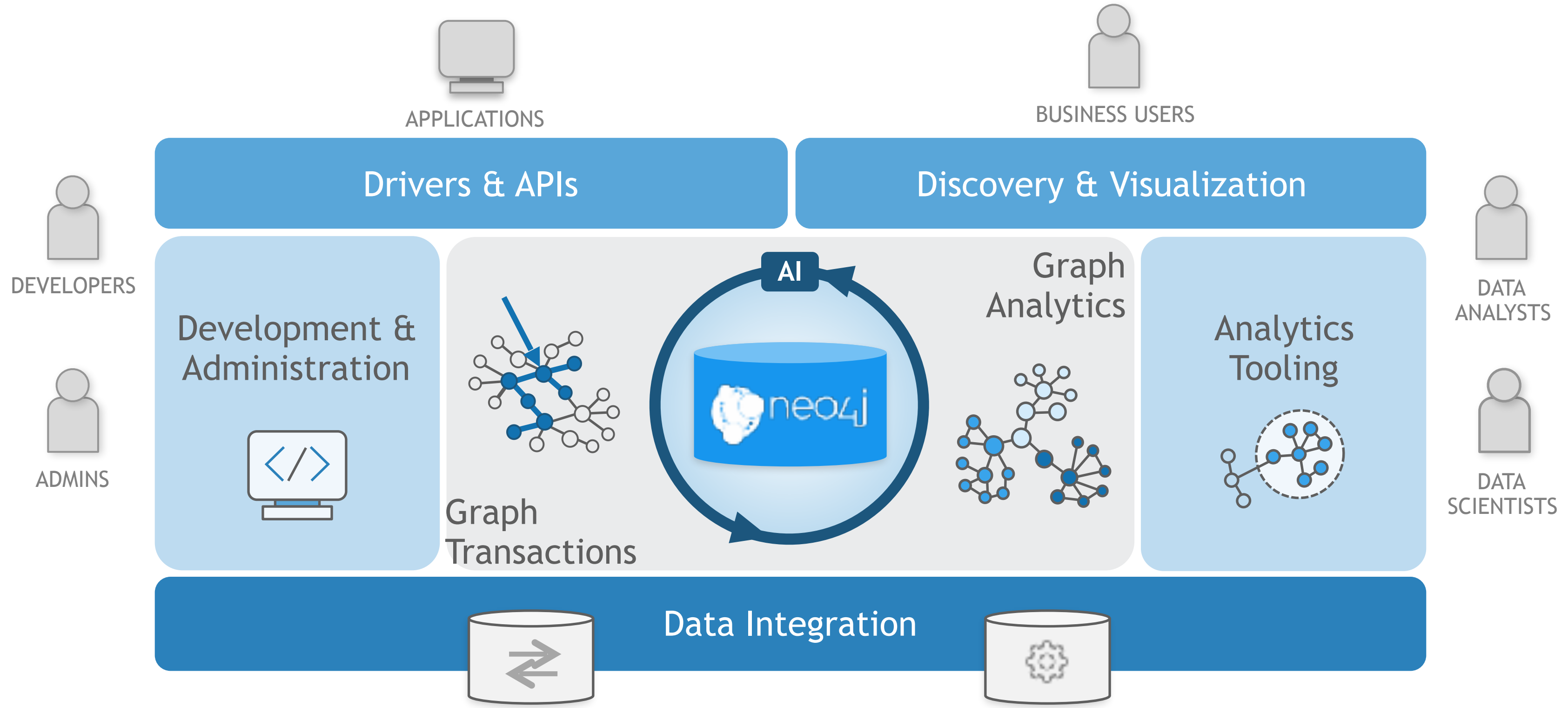
MARRIOTT

- 300M pricing operations per day
- 10x transaction throughput on half the hardware compared to Oracle
- Replaced Oracle database

Handling Package Routing in Real-Time

- Large postal service with over 500k employees
- Neo4j routes 7M+ packages daily at peak, with peaks of 5,000+ routing operations per second.

Roadmap - Neo4j Graph Platform

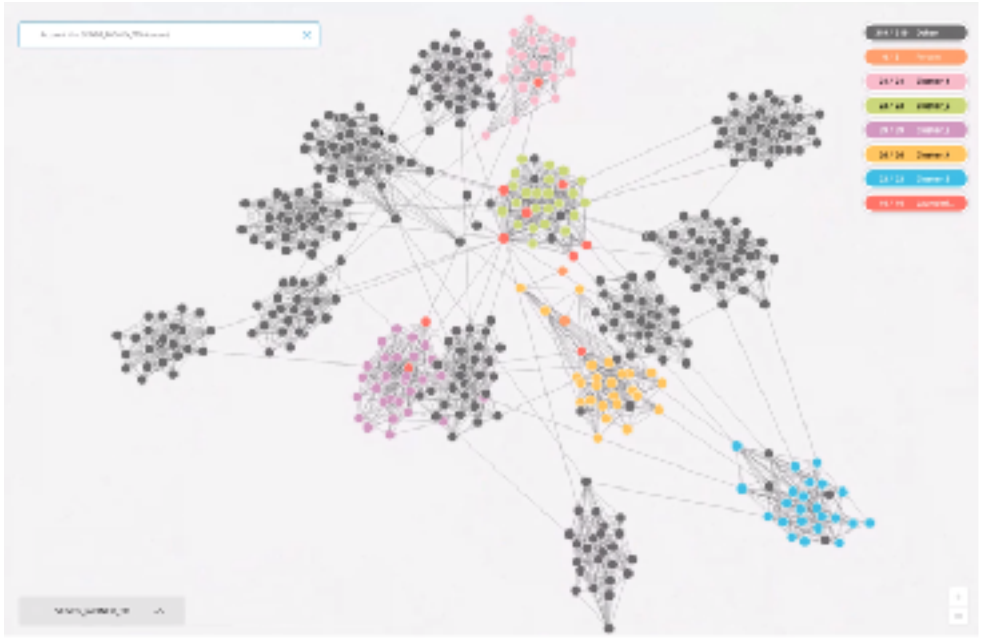


Neo4j Roadmap - Introducing Neo4j Bloom

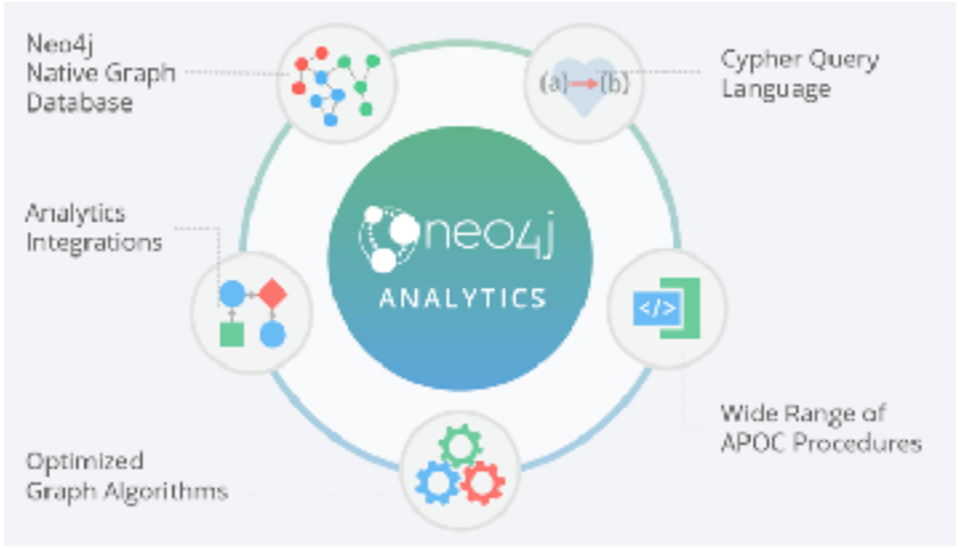
- Graph visualization accessible to the entire enterprise
- Minimal training required to derive insights from connected data using Bloom

Neo4j Bloom Planned 1.0 Features

Graph Perspectives	a business user view of the graph
Graph Visualization	high performance graph layout + rendering
Graph Exploration	navigate through direct graph interaction
Graph Inspection	browseable details of graph entities
Graph Editing	create, duplicate, edit, delete
Graph Search	extensible, idiomatic search phrases

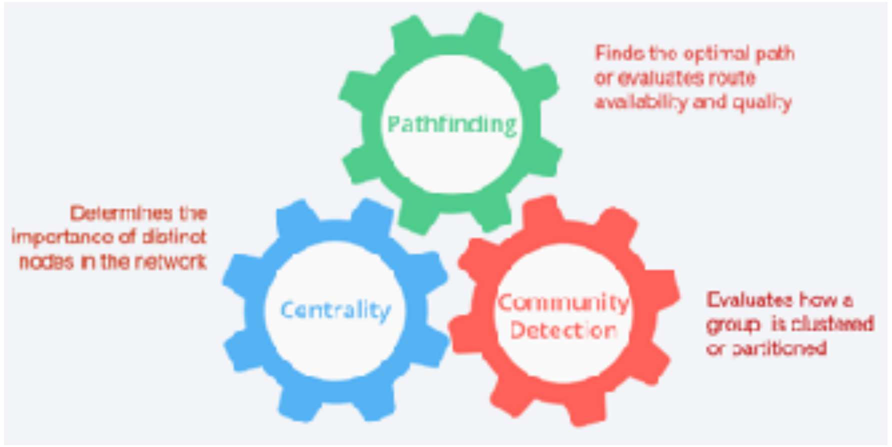


Neo4j Graph Analytics and Algorithms


Neo4j Native Graph Database
Cypher Query Language
(a)→(b)
Analytics Integrations
neo4j ANALYTICS
</>
Wide Range of APOC Procedures
Optimized Graph Algorithms

Graph Algorithms - Insights

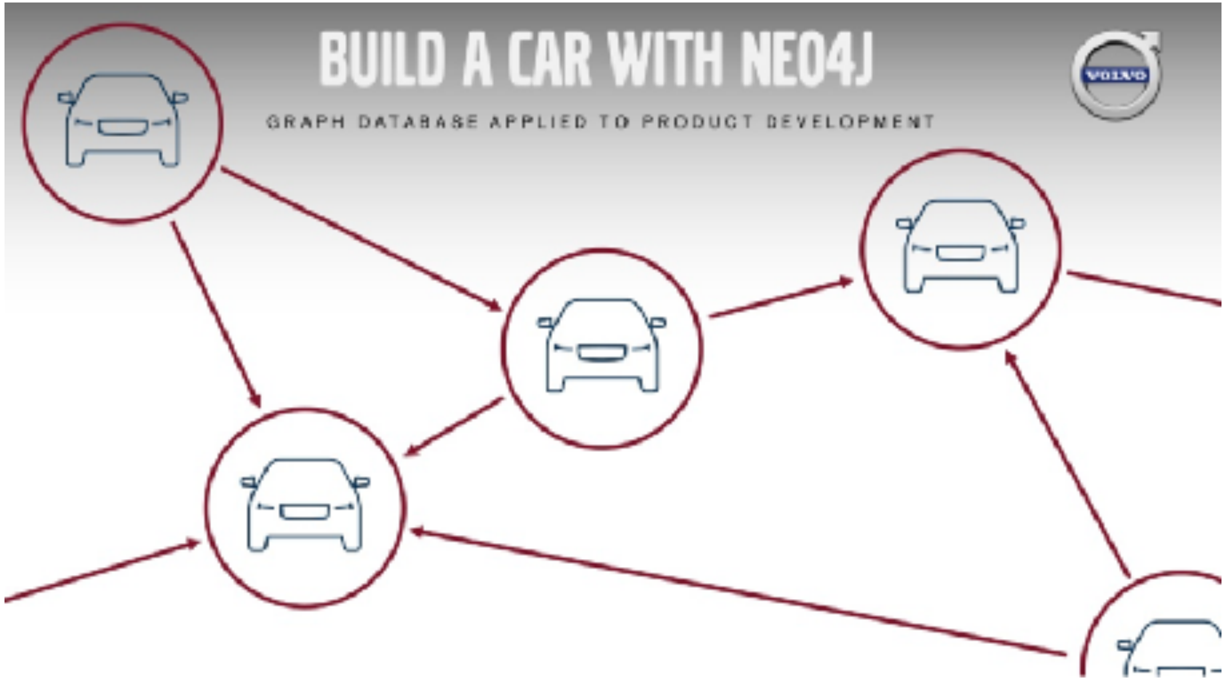
- Metrics
- Relevance
- Clustering and Classification
- Dependencies


Pathfinding
Finds the optimal path or evaluates route availability and quality
Determines the importance of distinct nodes in the network
Centrality
Community Detection
Evaluates how a group is clustered or partitioned

Graphs in Manufacturing

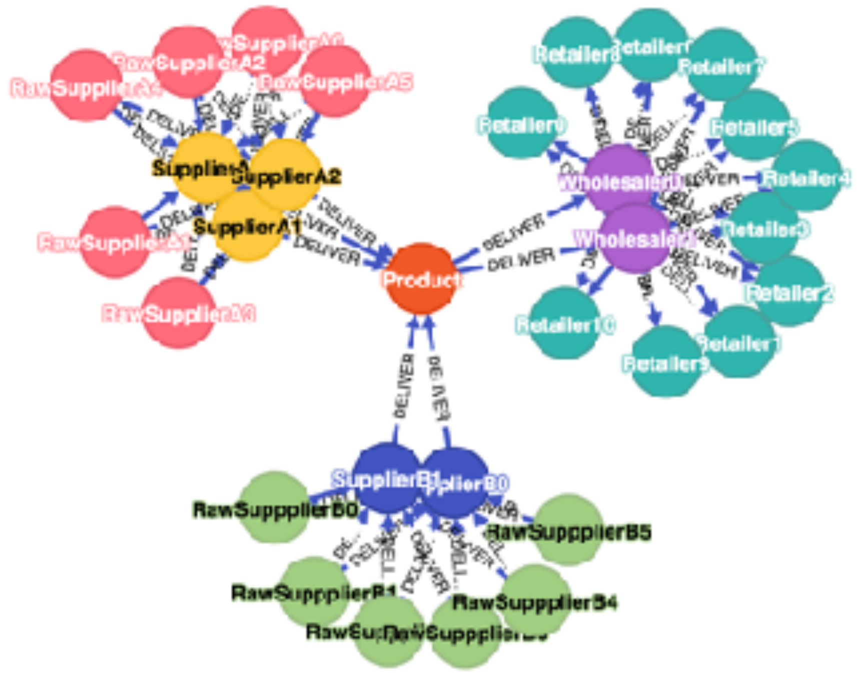
neo4j

Neo4j at Volvo



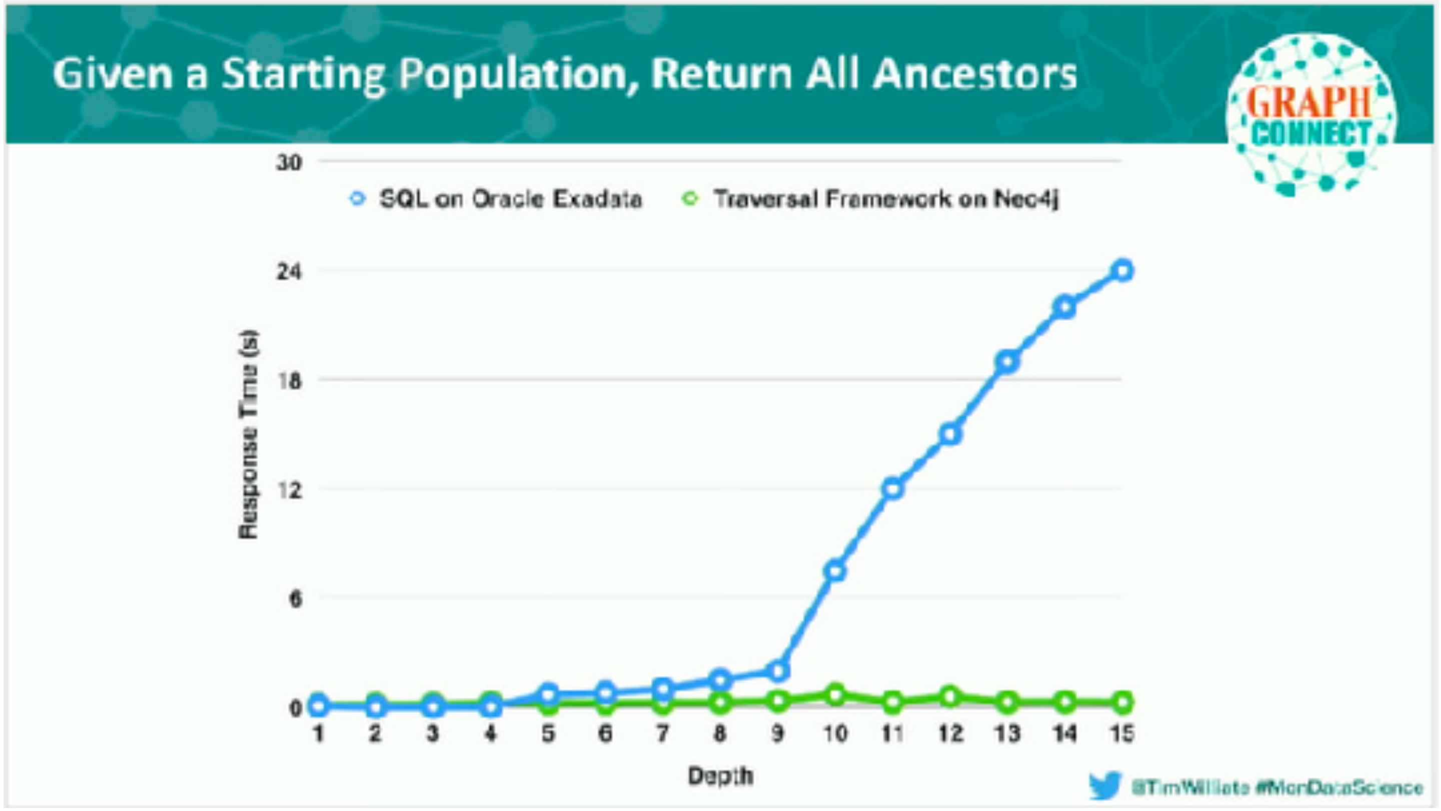
https://www.slideshare.net/neo4j/volvo-cars-build-a-car-with-neo4j

Neo4j Supply Chain Gist



https://neo4j.com/graphgist/supply-chain-management

Neo4j Replaces Oracle at Monsanto

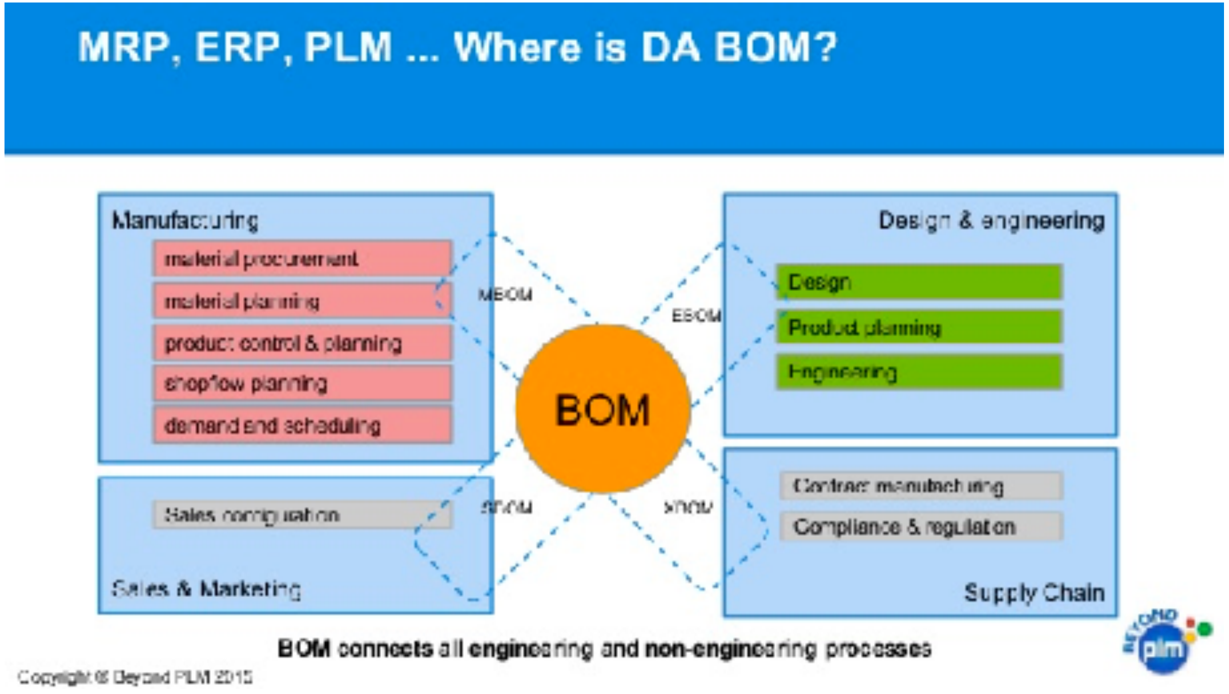


US Army Supply Logistics

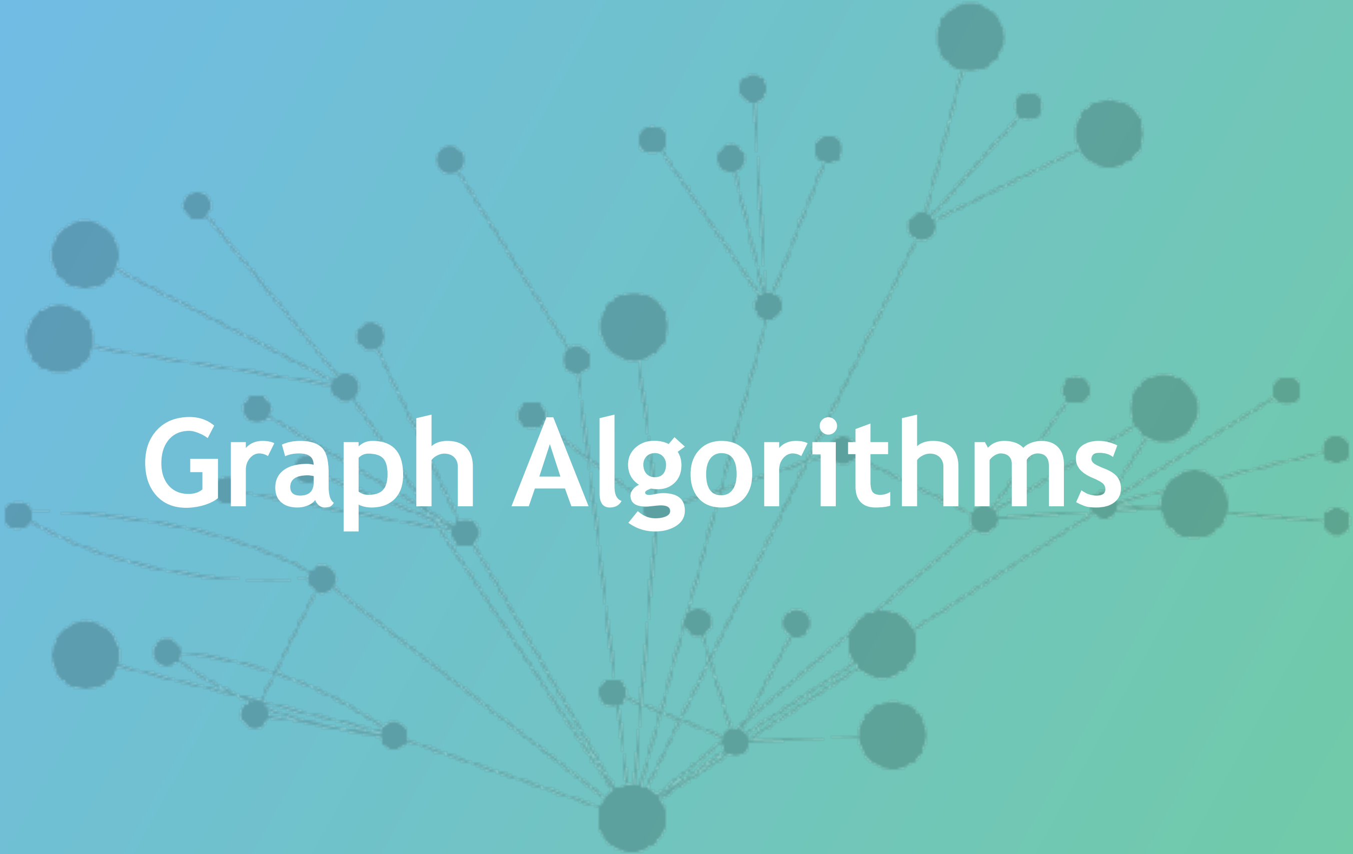
- Forecast the need for replacement parts
- Calculate mean time to failure rates
- Perform multi-dimensional cost comparison and trend analysis
- Inform the Army's budget requirements process
- Answer vital "what-if" questions such as the cost of deploying certain forces and the supporting equipment to a new war zone

https://neo4j.com/case-studies/us-army/

Bill of Materials - A Graph Specialty



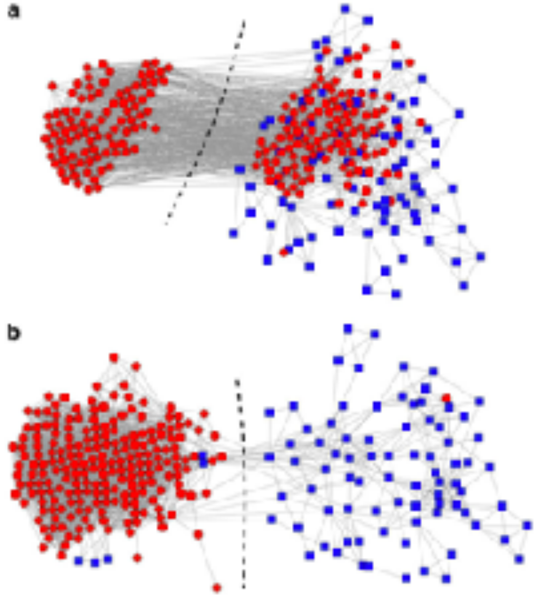
https://maxdemarzi.com/2017/11/17/bill-of-materials-in-neo4j/

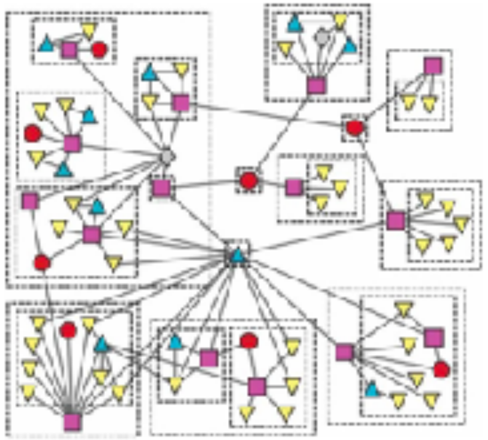


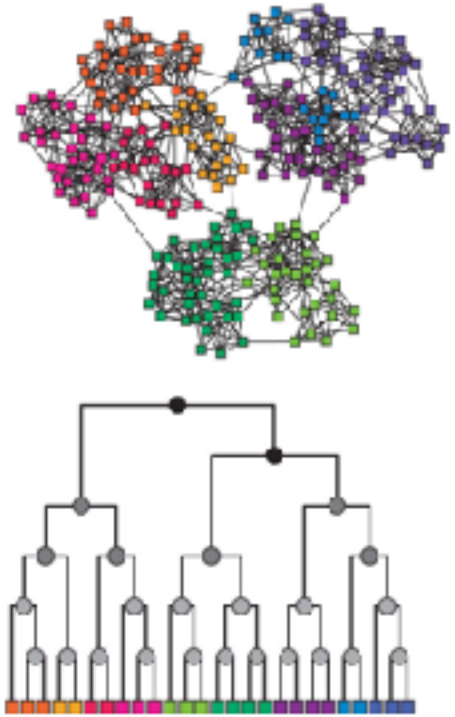
Graph Algorithms

neo4j

Use Graph Algorithms
Extract Structures and Infer Meaning







Use Graph Algorithms

Pathfinding & Search

Finds the optimal path or evaluates route availability and quality

Centrality

Determines the importance of distinct nodes in the network

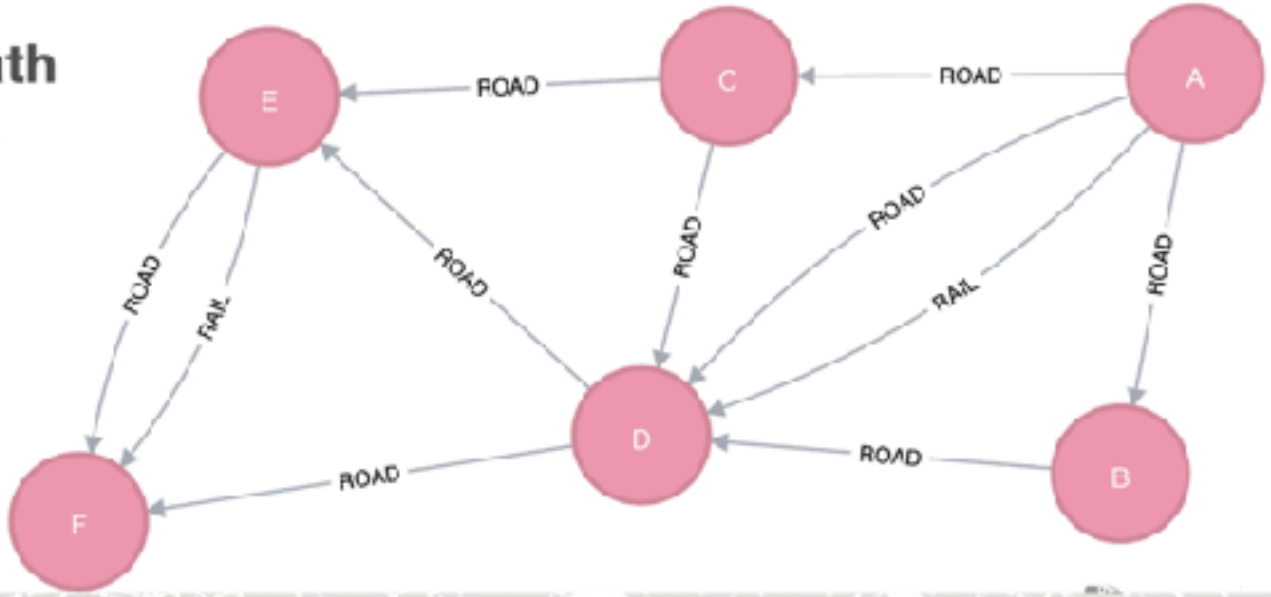
Community Detection

Evaluates how a group is clustered or partitioned

Algorithms - Pathfinding & Search

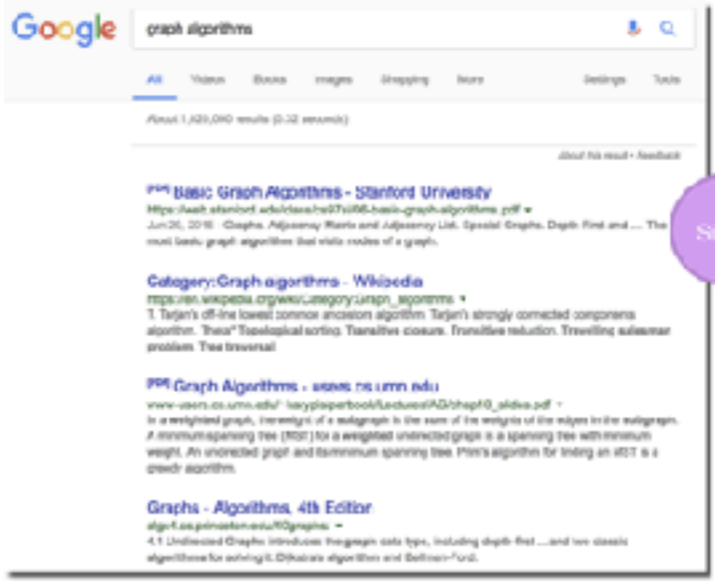
- **Single-Source Shortest Path**
  - Calculates "shortest" path between a node and all other nodes

- **All-Pairs Shortest Path**
  - Finds all shortest paths between all nodes



Algorithms - Centralities

- **PageRank**
  - Which nodes have the most overall influence



Google

graph algorithms

All Videos Books Images Shopping More Settings Tools

About 1,420,000 results (0.32 seconds)

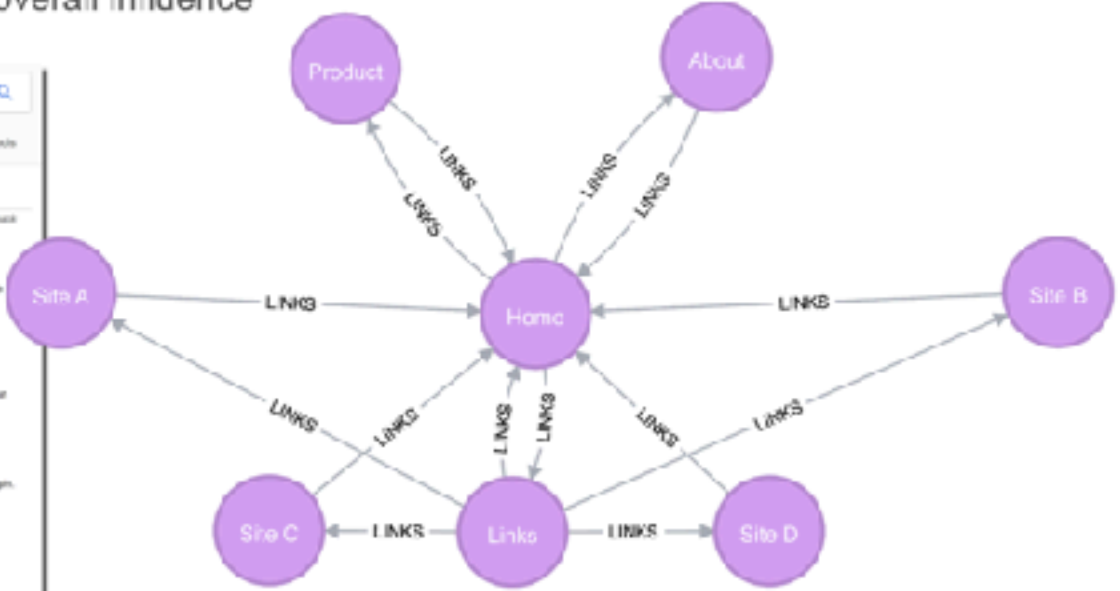
About this result • Feedback

[PDF] Basic Graph Algorithms - Stanford University
https://web.stanford.edu/class/cs97si/06-basic-graph-algorithms.pdf
Jun 26, 2016 - Graphs. Adjacency Matrix and Adjacency List. Special Graphs. Depth-First and .... The most basic graph algorithm that visits nodes of a graph.

Category:Graph algorithms - Wikipedia
https://en.wikipedia.org/wiki/Category:Graph_algorithms
1. Tarjan's off-line lowest common ancestors algorithm. Tarjan's strongly connected components algorithm. Theta* Topological sorting. Transitive closure. Transitive reduction. Travelling salesman problem. Tree traversal.

[PDF] Graph Algorithms - www.cs.umn.edu
www-users.cs.umn.edu/~karypis/parbook/Lectures/AG/chap10_slides.pdf
In a weighted graph, the weight of a subgraph is the sum of the weights of the edges in the subgraph. A minimum spanning tree (MST) for a weighted undirected graph is a spanning tree with minimum weight. An undirected graph and its minimum spanning tree. Prim's algorithm for finding an MST is a greedy algorithm.

Graphs - Algorithms, 4th Edition
algs4.cs.princeton.edu/40graphs/
4.1 Undirected Graphs introduces the graph data type, including depth-first ... and two classic algorithms for solving it. Dijkstra's algorithm and Bellman-Ford.



neo4j

Centrality

Measure of importance

PageRank

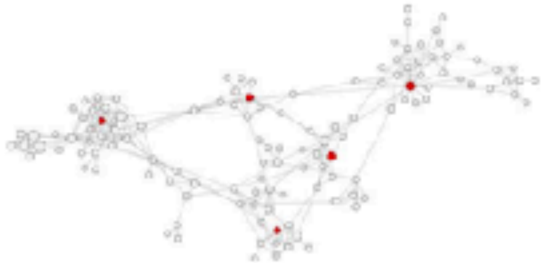
- Recursive
- Importance and number of connected nodes

Betweenness Centrality

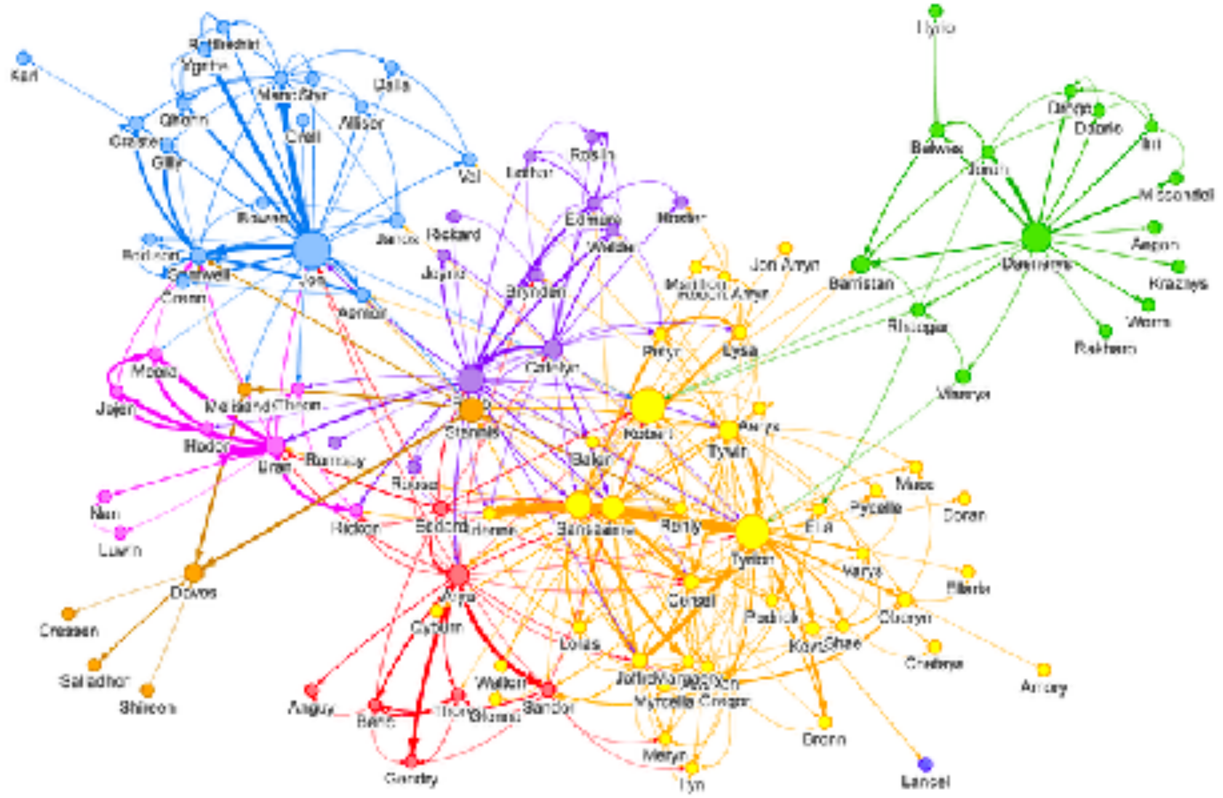
- Number of shortest paths connecting all pairs in the network

Closeness Centrality

- Inverse of distance to all other nodes in the network



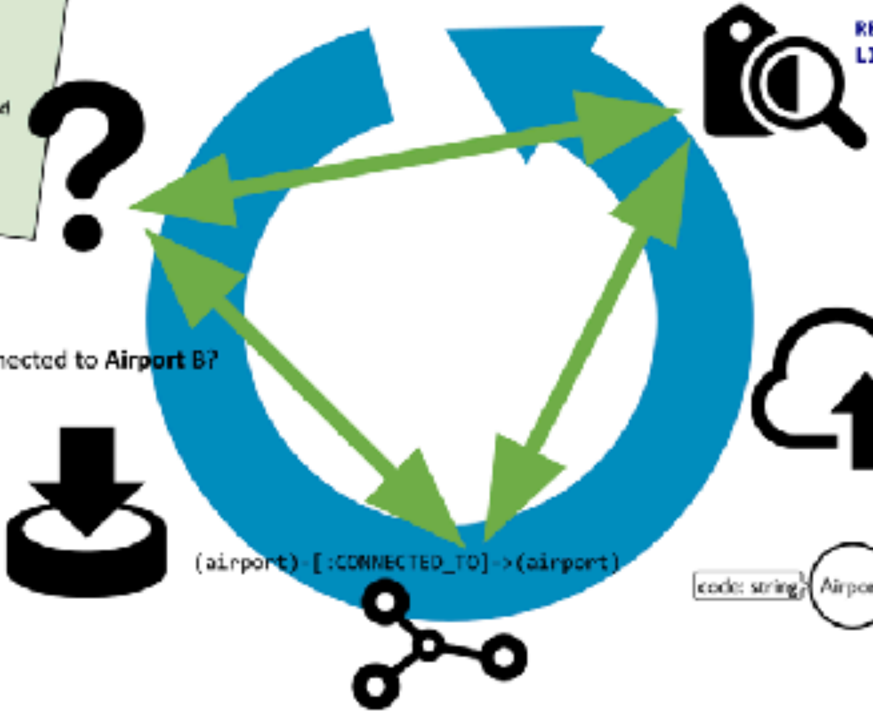
Visualization



Technical

The modeling workflow

As an air travel enthusiast
I want to know how airports are connected
So that I can find the busiest ones



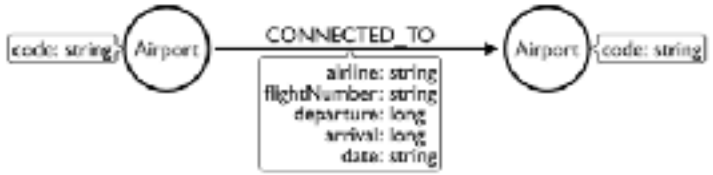
```
MATCH (origin:Airport {code: "LAX"})
      -[flight:CONNECTED_TO]->
      (destination:Airport {code: "LAS"})
RETURN origin, destination, flight
LIMIT 10
```

Is **Airport** A connected to **Airport** B?

```
CREATE (lax:Airport {code: "LAX"})
CREATE (las:Airport {code: "LAS"})
CREATE (las)-[connection:CONNECTED_TO {
    airline: "WN",
    flightNumber: "82",
    date: "2008-1-3",
    departure: 1715,
    arrival: 1820}]->(lax)
```

```
|--------+------+-----------|
| Origin | Dest | FlightNum |
|--------+------+-----------|
| IAD    | TPA  | 335       |
| IAD    | TPA  | 3231      |
| IND    | BWI  | 448       |
| IND    | BWI  | 3920      |
|--------+------+-----------|
```

(airport)-[:CONNECTED_TO]->(airport)



Cluster Architecture

Raft-based architecture

Consensus commits via "Core" servers

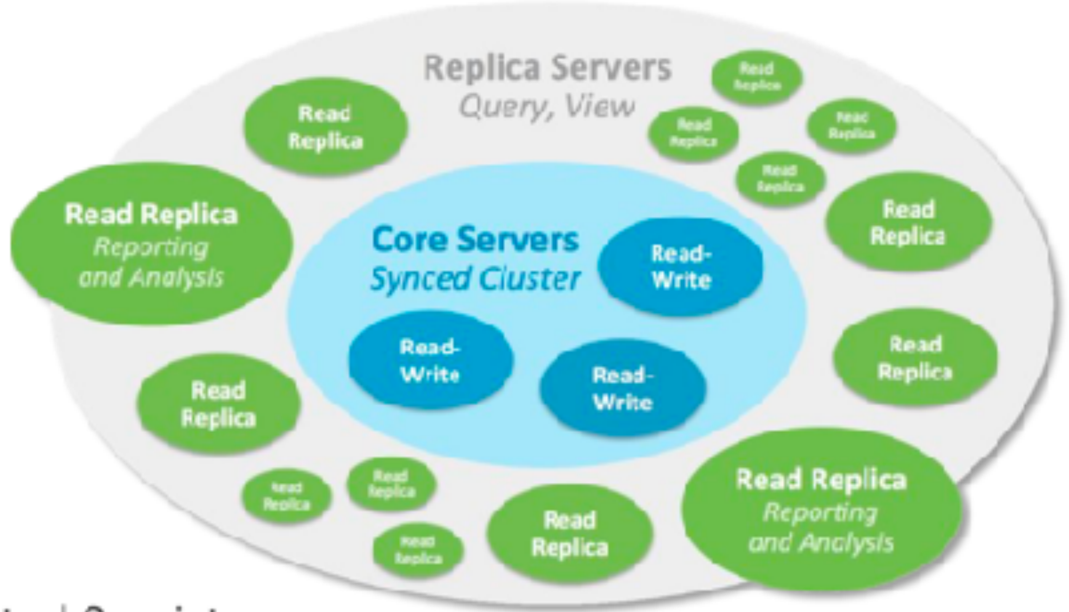
Cluster-aware drivers

No need for external load balancer

Stateful, cluster-aware sessions

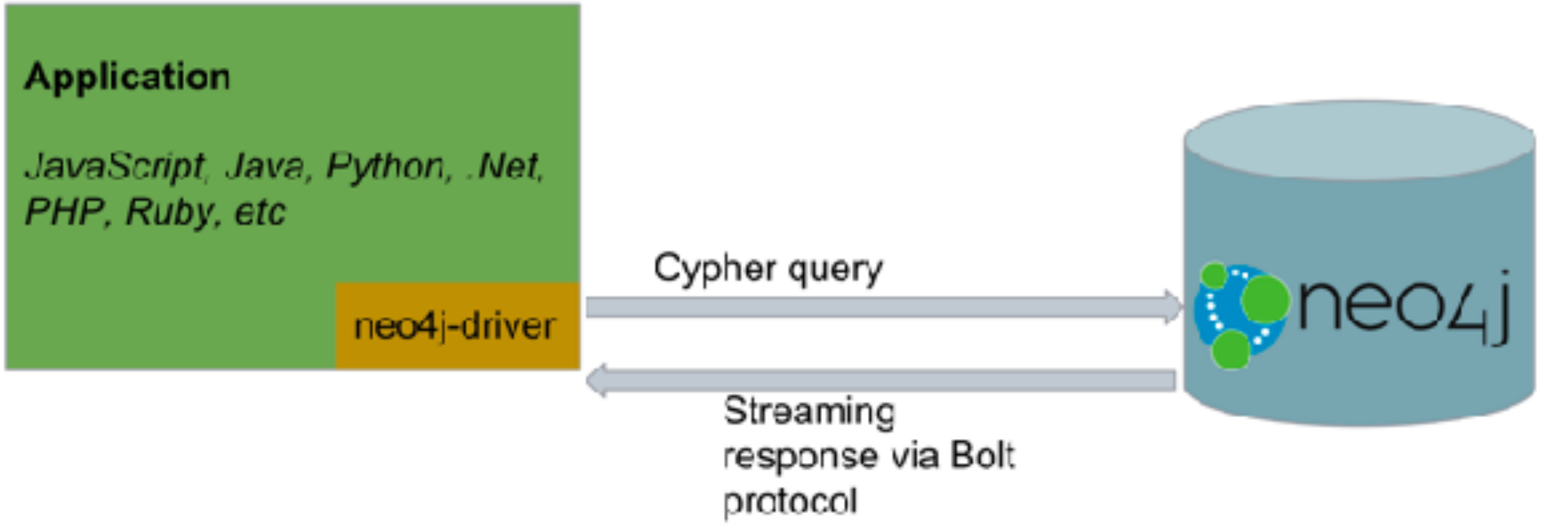
Causal Consistency

Stronger consistency model than Eventual Consistency



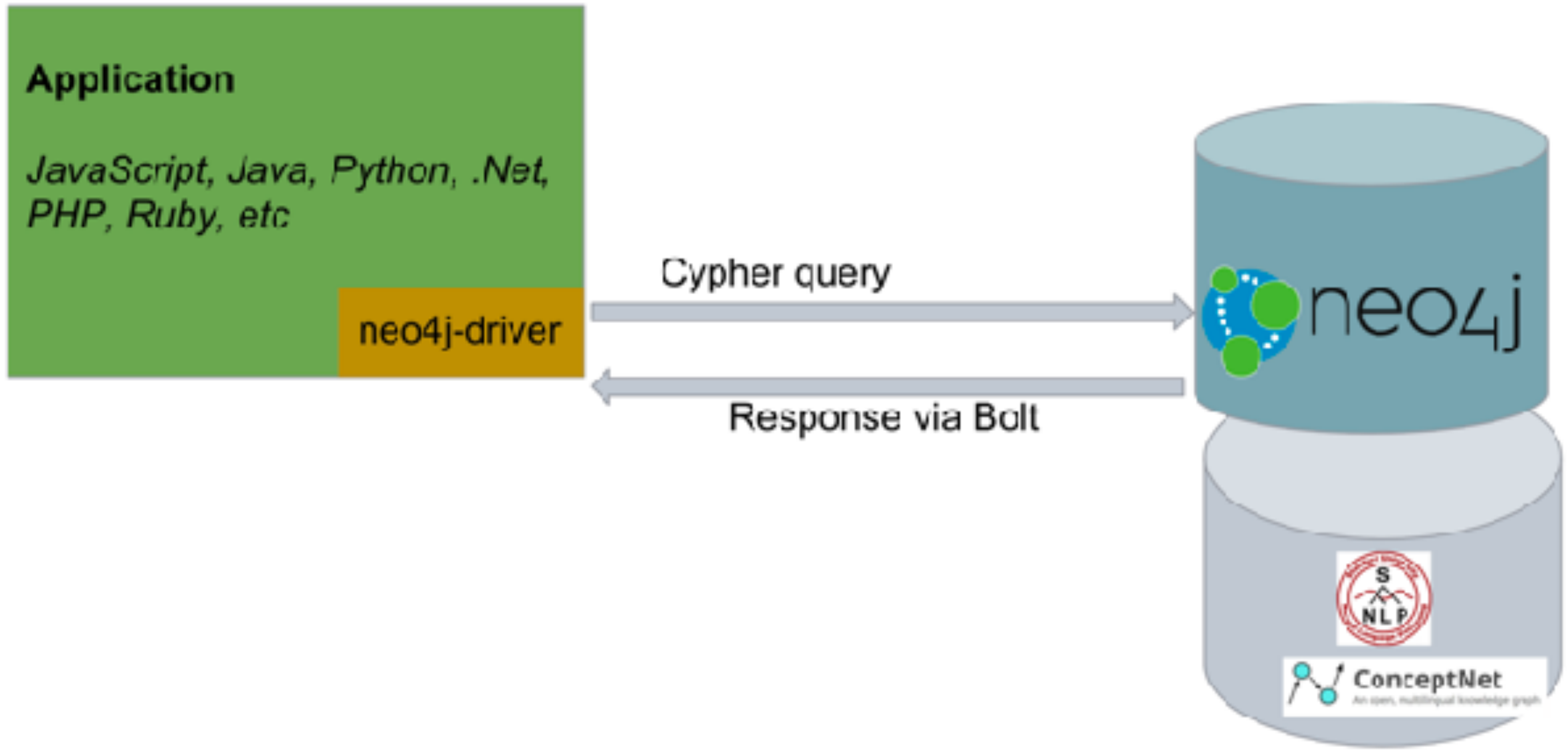
Application Architecture

Client driver libraries use Cypher over Bolt protocol

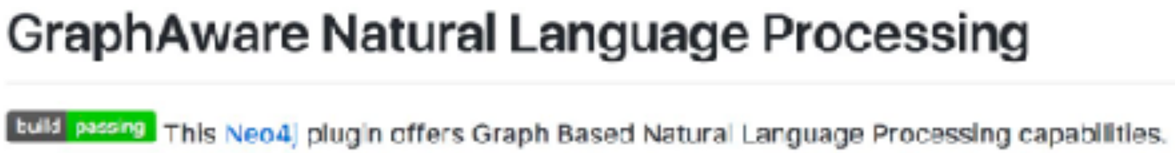

Application
JavaScript, Java, Python, .Net, PHP, Ruby, etc
neo4j-driver
Cypher query
Streaming response via Bolt protocol
neo4j

https://neo4j.com/developer/language-guides/

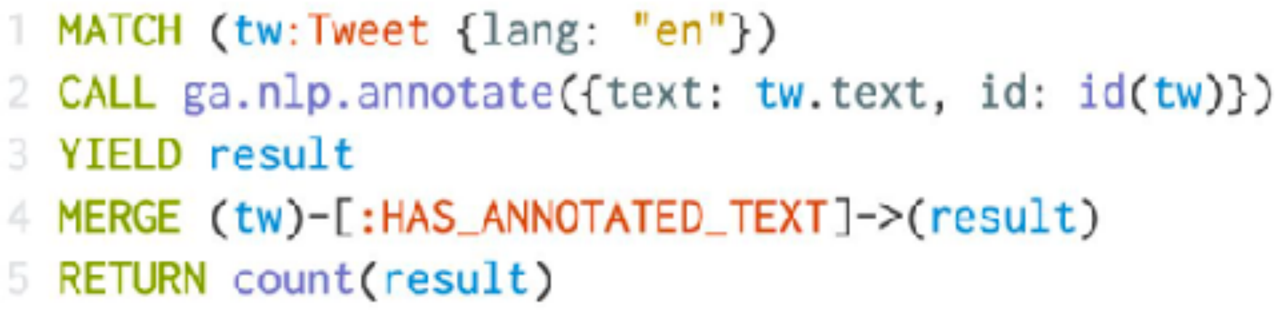
Using NLP and User-Defined Functions



Using NLP and User-Defined Functions

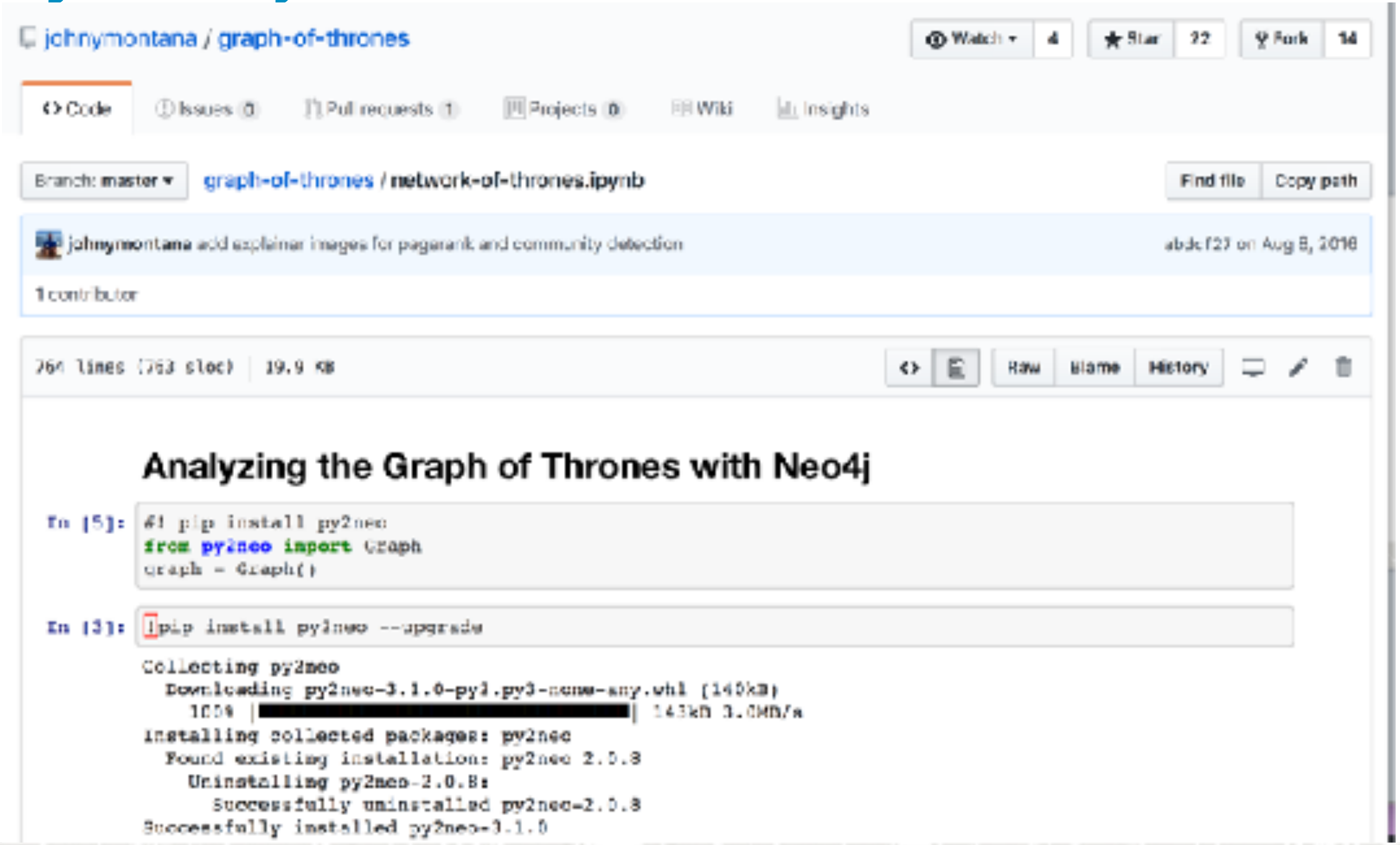

GraphAware Natural Language Processing

build passing This Neo4j plugin offers Graph Based Natural Language Processing capabilities.


```
MATCH (tw:Tweet {lang: "en"})
CALL ga.nlp.annotate({text: tw.text, id: id(tw)})
YIELD result
MERGE (tw)-[:HAS_ANNOTATED_TEXT]->(result)
RETURN count(result)
```

https://github.com/graphaware/neo4j-nlp

Neo4j With Python Notebooks



johnymontana / graph-of-thrones

Watch 4 | Star 22 | Fork 14

Code | Issues 0 | Pull requests 1 | Projects 0 | Wiki | Insights

Branch: master ▾ graph-of-thrones / network-of-thrones.ipynb

Find file | Copy path

johnymontana add explainer images for pagerank and community detection — abdcf27 on Aug 8, 2016

1 contributor

764 lines (763 sloc) | 19.9 KB

Raw | Blame | History

Analyzing the Graph of Thrones with Neo4j

```
In [5]: #! pip install py2neo
        from py2neo import Graph
        graph = Graph()
```

```
In [3]: !pip install py2neo --upgrade
```

```
Collecting py2neo
  Downloading py2neo-3.1.0-py2.py3-none-any.whl (140kB)
    100% |████████████████████████████████| 143kB 3.0MB/s
Installing collected packages: py2neo
  Found existing installation: py2neo 2.0.8
    Uninstalling py2neo-2.0.8:
      Successfully uninstalled py2neo-2.0.8
Successfully installed py2neo-3.1.0
```

Neo4j vs JanusGraph / DSE Graph / Titan

	Neo4J	Titan & Derivatives
ACID Compliance	Yes	Eventually Consistent
Graph Storage	Native Graph	Cassandra
Suitable for Multi-Hops	Yes	No
Language	Open Cypher	Gremlin

OpenCypher vs Apache Gremlin - Simplicity

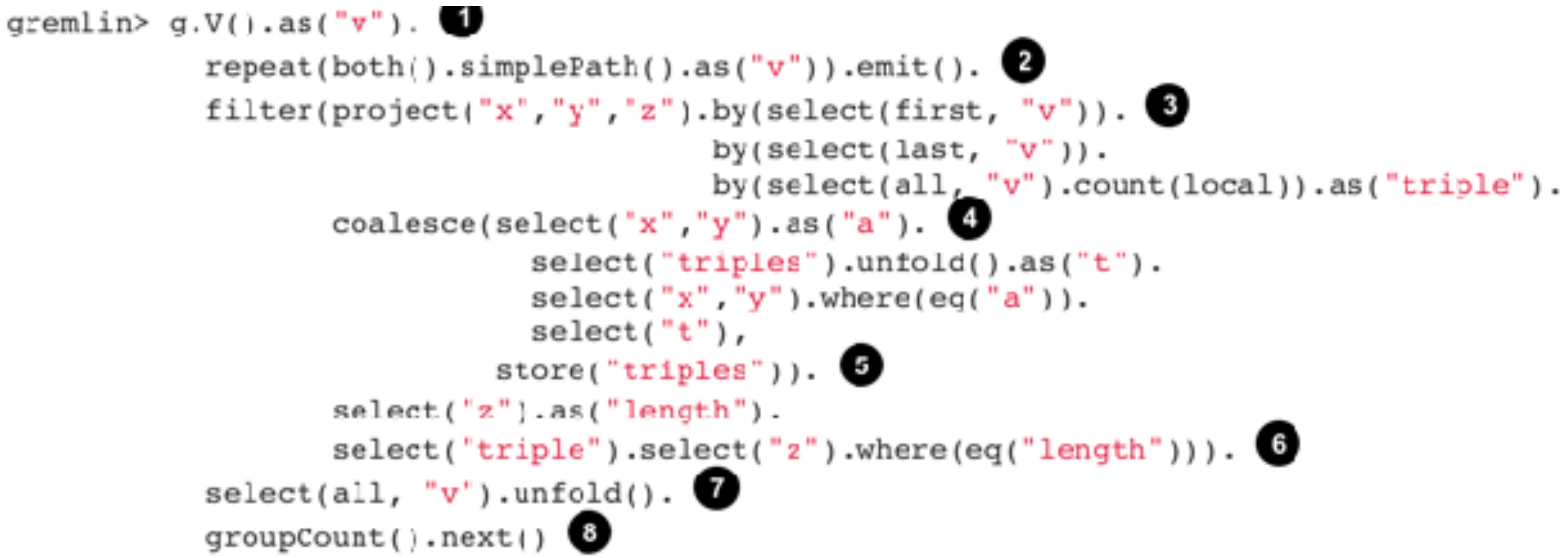
	Open Cypher	Apache Gremlin
Language Style	Declarative Pattern Matching	Imperative API Style
Ease of Use	Simple to learn	Extremely Complex
Costs to Develop and Maintain	Low	High
Suitable for Power Users	Yes	Are you kidding?

OpenCypher vs Apache Gremlin - Simplicity

CYPHER

```
CALL algo.betweenness.stream('User','MANAGE',{direction:'out'})
```

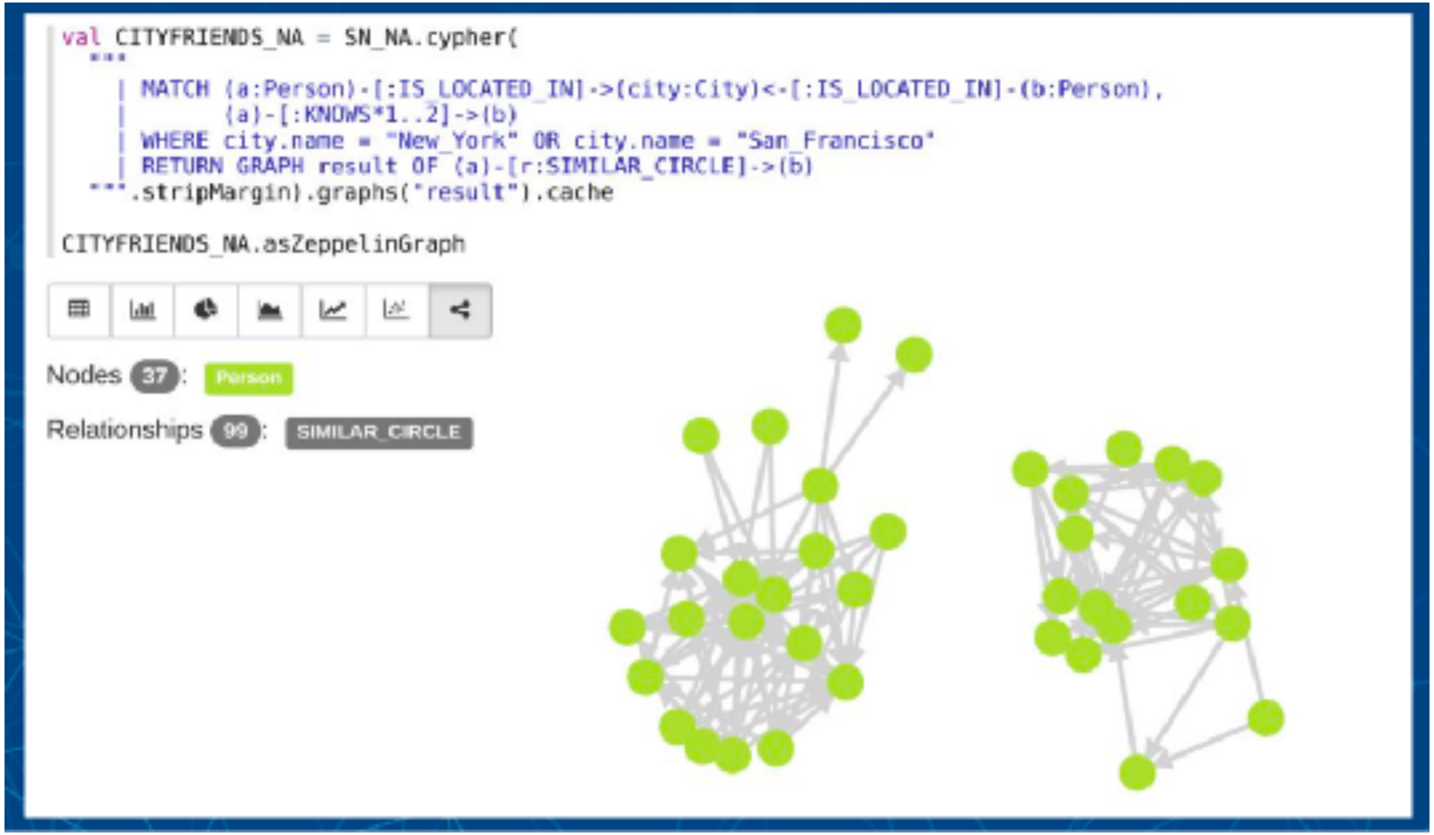
vs Gremlin



```
gremlin> g.V().as("v"). ①
           repeat(both().simplePath().as("v")).emit(). ②
           filter(project("x","y","z").by(select(first, "v")). ③
                                       by(select(last, "v")).
                                       by(select(all, "v").count(local)).as("triple").
                  coalesce(select("x","y").as("a"). ④
                               select("triples").unfold().as("t").
                               select("x","y").where(eq("a")).
                               select("t"),
                             store("triples")). ⑤
                  select("z").as("length").
                  select("triple").select("z").where(eq("length"))). ⑥
           select(all, "v").unfold(). ⑦
           groupCount().next() ⑧
```

http://tinkerpop.apache.org/docs/current/recipes/

Cypher for Apache Spark



```
val CITYFRIENDS_NA = SN_NA.cypher(
  """
    | MATCH (a:Person)-[:IS_LOCATED_IN]->(city:City)<-[:IS_LOCATED_IN]-(b:Person),
    |        (a)-[:KNOWS*1..2]->(b)
    | WHERE city.name = "New_York" OR city.name = "San_Francisco"
    | RETURN GRAPH result OF (a)-[r:SIMILAR_CIRCLE]->(b)
  """.stripMargin).graphs("result").cache

CITYFRIENDS_NA.asZeppelinGraph
```

Nodes 37 : Person

Relationships 99 : SIMILAR_CIRCLE

Thank You!

neo4j