GRAPH DATABASE

NEO4J

Yunghans Irawan (yirawan@nus.edu.sg)

Agenda

- Graph Database
- Neo4J
- Query with Cypher
- Visualization
- Case Study

Graph Database

- Graph is made of nodes and edges
- Nodes represent entities and edges represent relationship between entities
- One type of databases that put a lot of focus in the relationship between entity data
  - Suitable for very dense network of the entities

Example of graph related problems

- Social studies
  - Relationship between people
  - Social network
- Biological studies
- Flow problems
  - Optimal path
  - Bottleneck
- Routing problem
  - Shortest distance
- Web Search
  - PageRank is a graph algorithm

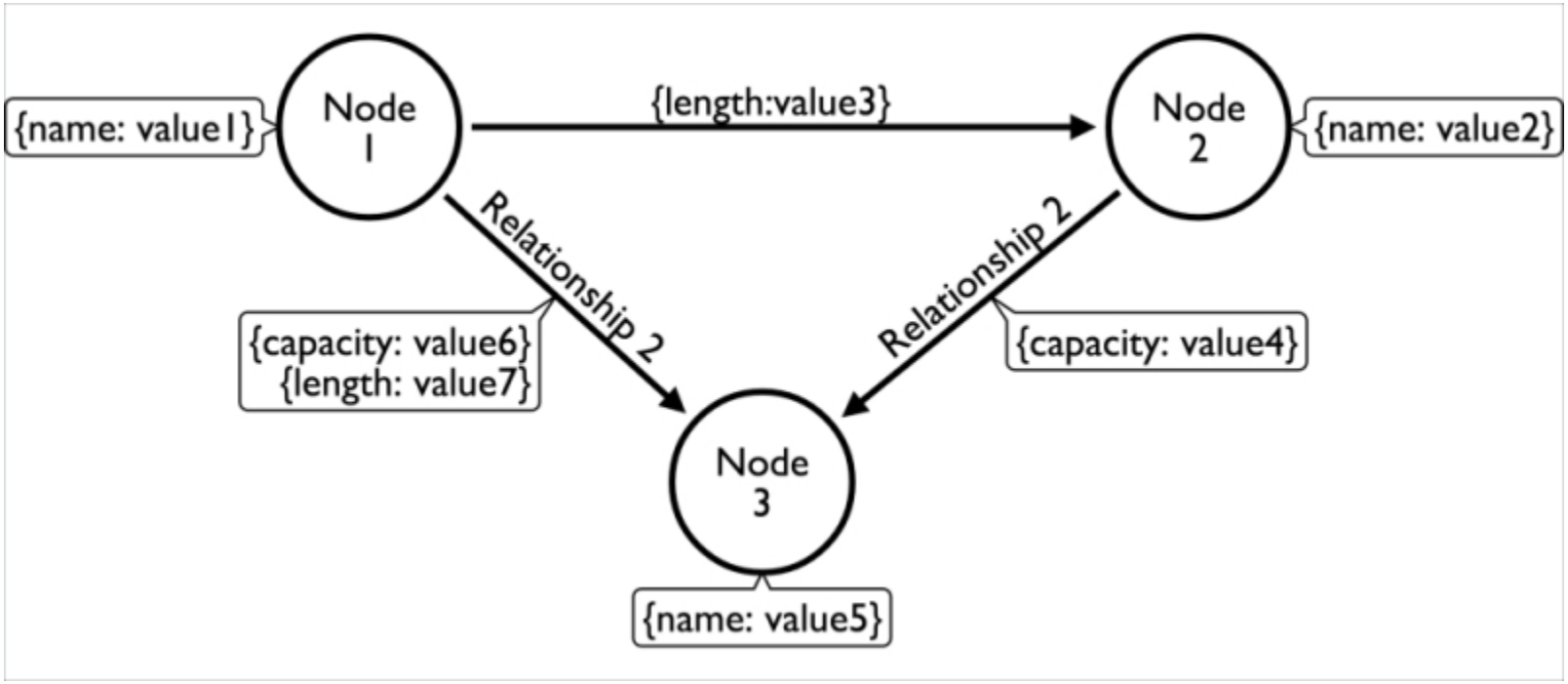
Graph Databases

Name	Initial Release	Latest Version	License
Neo4J	2007	3.3.5 April 2018	Open Source
Datastax Enterprise	2011	6.0	Commercial – creator of MongoDB, bought Titan GraphDB
Giraph	2012	1.2.0 Mar 2016	Open Source

Graph Model

- Graph is made of nodes (vertices) and relationship (edges)
- Directed Graph
  - Edges have a direction
- Multirelational graph
  - There can be multiple relationships between two nodes
- Properties of nodes and relationships is stored as key-value pairs

Graph Model



Graph Model

- Node labels
  - A node can have zero to many labels assigned to it
  - Similar to: hashtags, Gmail labels, tag
  - Allow us to quickly create a subgraph in our database
- Relationship types
  - Mandatory property for relationship

When to use

- Pattern matching complex queries
- Path finding queries
- Complex query on live data

When not to use

- Large set-oriented queries
  - RDBMS may be more suitable
- Aggregate oriented queries
- Graph global operations
  - More optimized for local operations

Neo4J

- Build from ground up to deal with graph data
  - Initially the engine was build to run on RDBMS, but not good enough!
- Transactional, ACID compliant database
- Built for OLTP (online transaction processing)
  - Short and fast insert and updates
  - Simple queries that needs to be very fast
- Support clustering of database instances for high availability and fault tolerance
  - Master-Slave architecture
  - Available in Enterprise Edition

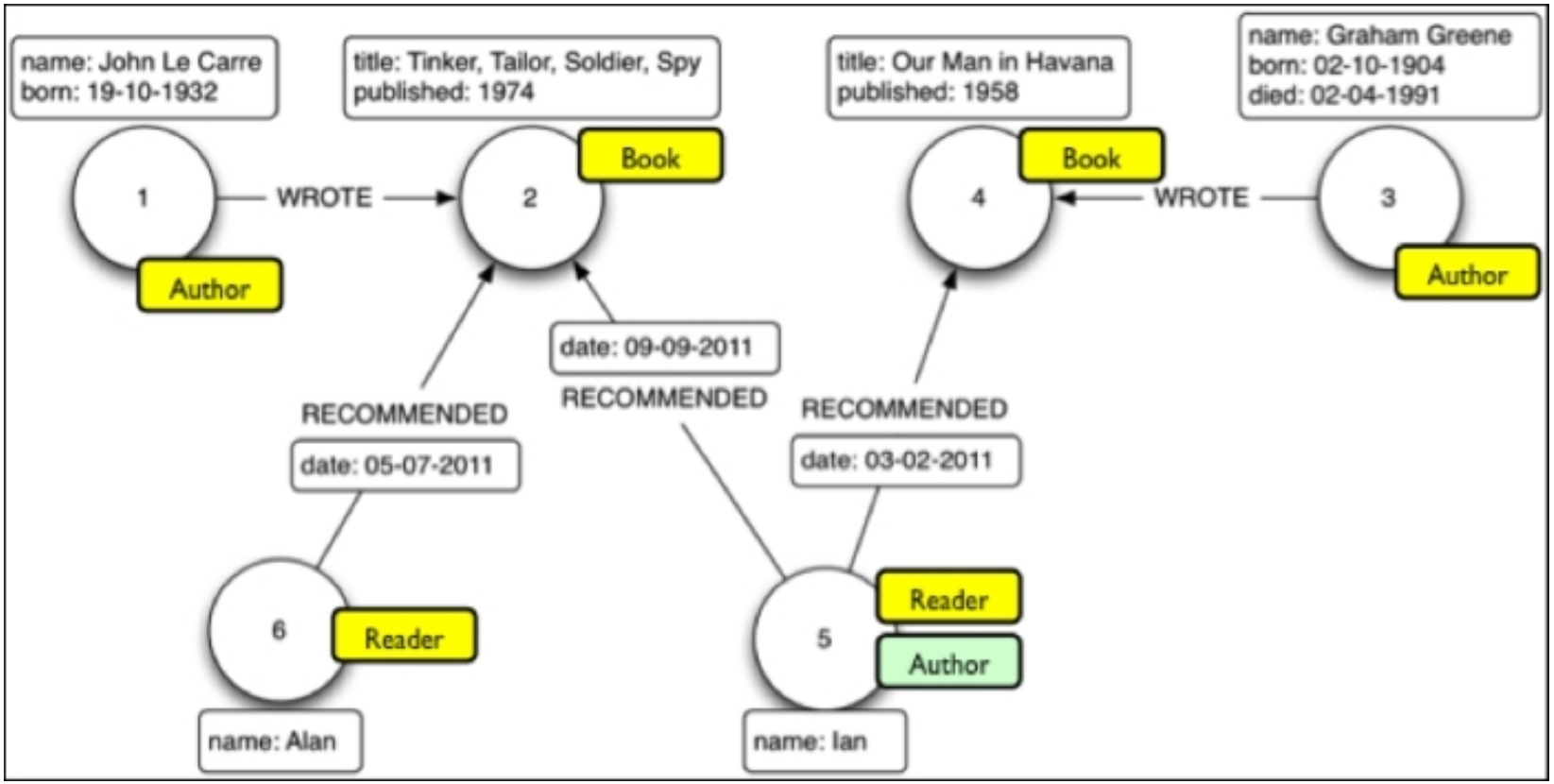
Example of Neo4J sweet spots

- Check for connection between two data elements
- Look for optimal path (lowest cost) between two things
- Look for variability of the path if a certain component of the path changes

Data Constructs

- Nodes
  - Used to store entity information
- Relationships
  - Connect nodes to one another explicitly
  - Type + start node + end node + direction
- Properties
  - Name/value pairs attached to nodes and relationships
- Labels
  - Can be assigned to nodes to quickly create subgraphs

Example



Cypher

- Cypher is a declarative, pattern-matching query language to work on graph data
- Live demo
  - http://console.neo4j.org/
- Reference Card
  - http://neo4j.com/docs/pdf/neo4j-cypher-refcard-stable.pdf

Cypher Examples

- Create a new person

```
CREATE (you:Person {name:"You"})
RETURN you
```

- Create a new node Neo4J and create a like relationship

```
MATCH  (you:Person {name:"You"})
CREATE (you)-[like:LIKE]->(neo:Database {name:"Neo4j" })
RETURN you,like,neo
```

Cypher Examples

- Create friends

```
MATCH (you:Person {name:"You"})
FOREACH (name in ["Johan","Rajesh","Anna","Julia","Andrew"] |
  CREATE (you)-[:FRIEND]->(:Person {name:name}))
```

- Find your friends

```
MATCH (you {name:"You"})-[:FRIEND]->(yourFriends)
RETURN you, yourFriends
```

- Create second degree friends

```
MATCH (neo:Database {name:"Neo4j"})
MATCH (anna:Person {name:"Anna"})
CREATE (anna)-[:FRIEND]->(:Person:Expert {name:"Amanda"})-
[:WORKED_WITH]->(neo)
```

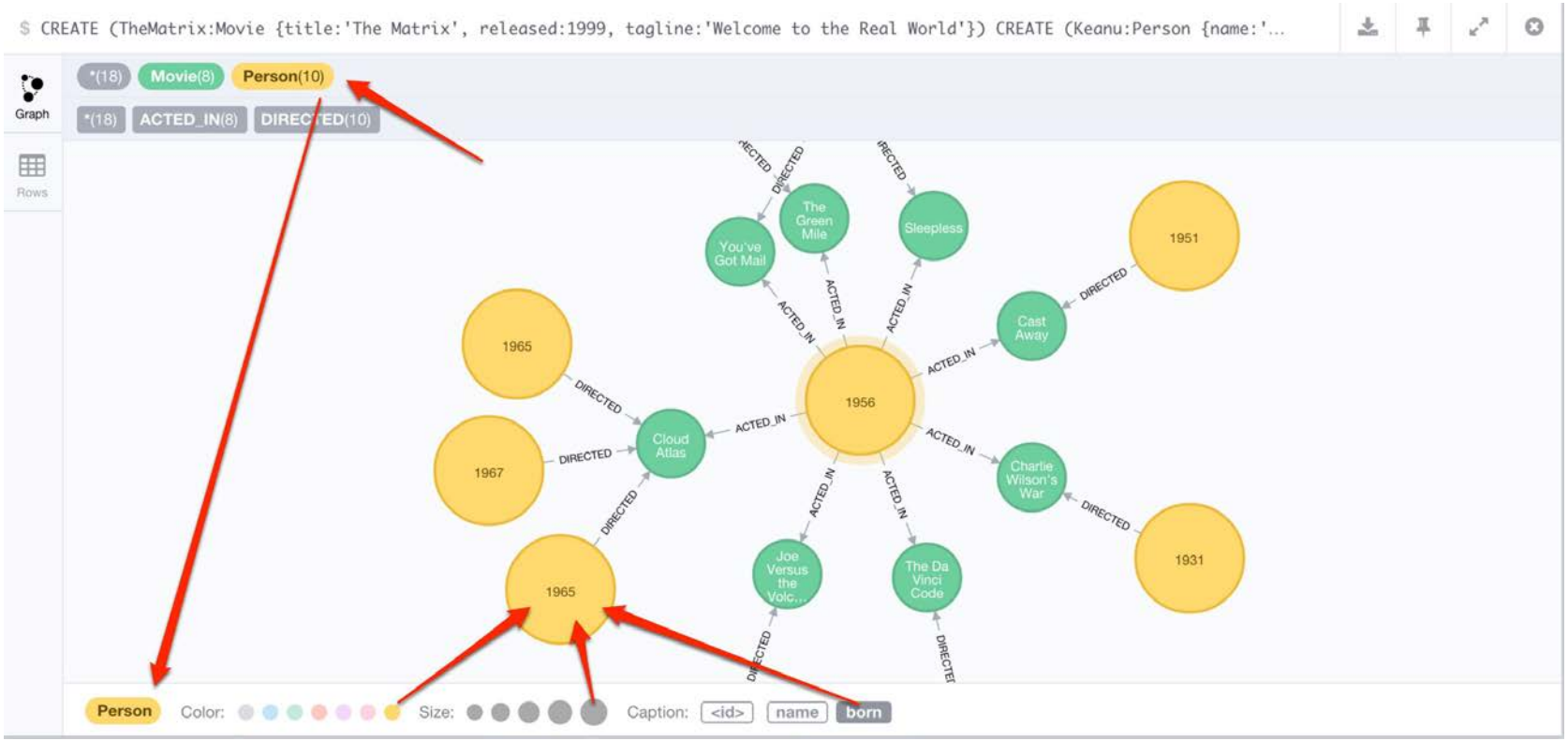
Cypher Examples

- Find someone in your network who can help you learn Neo4J

```
MATCH (you {name:"You"})
MATCH (expert)-[:WORKED_WITH]->(db:Database {name:"Neo4j"})
MATCH path = shortestPath( (you)-[:FRIEND*..5]-(expert) )
RETURN db,expert,path
```

Neo4J Browser

- Allow you to query and visualize the graph

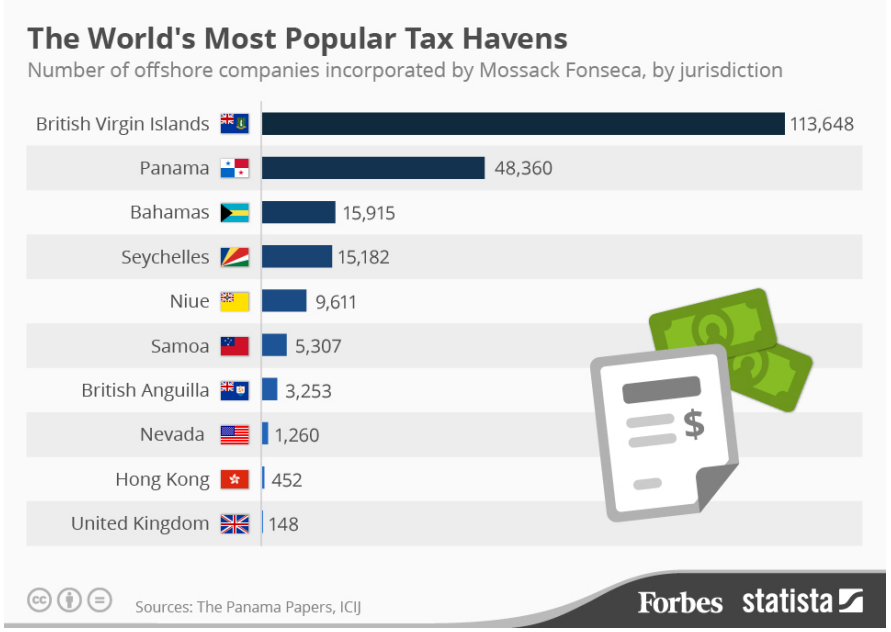


Other tools

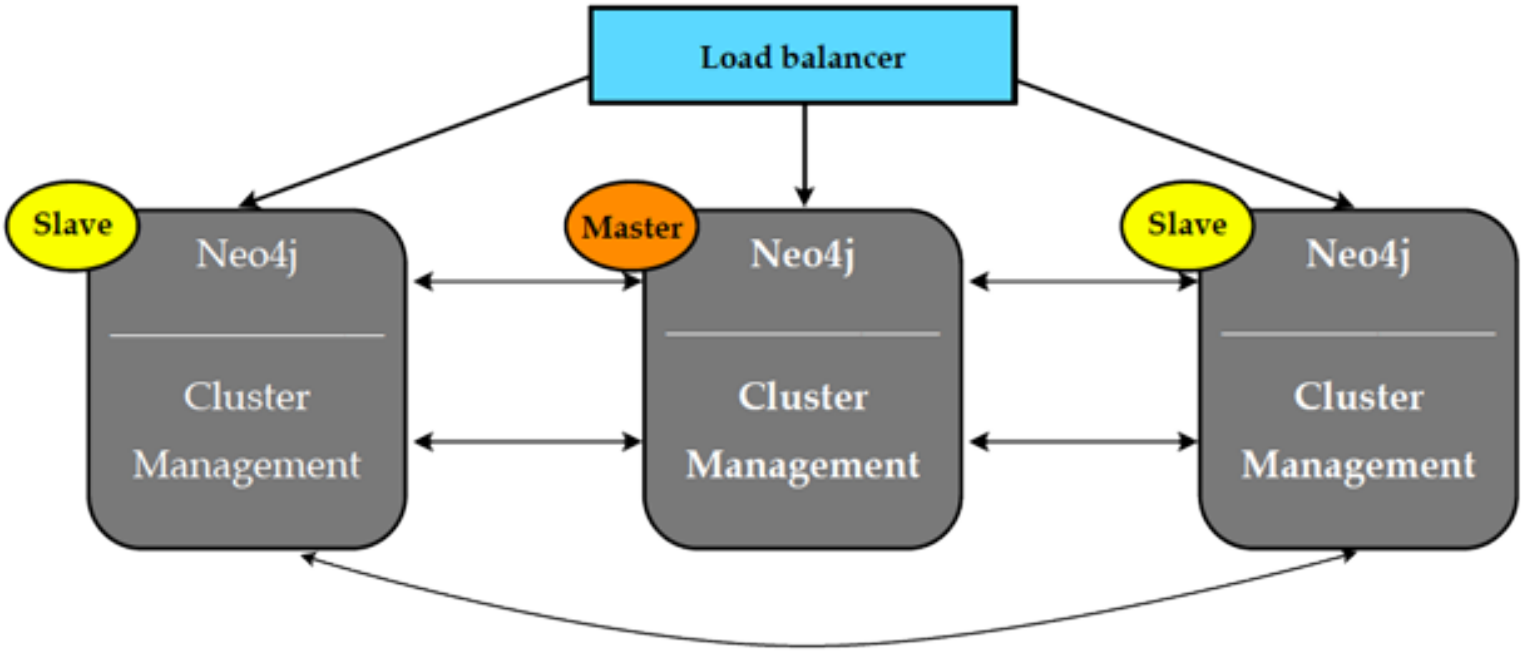
- Tom Sawyer Perspective
- Gephi
- Linkurio.us
- Keylines

Case Study Panama Papers

- The Panama Papers are 11.5 million leaked documents that detail financial and attorney–client information for more than 214,488 offshore entities.
  - The leaked documents were created by Panamanian law firm and corporate service provider Mossack Fonseca
- The data is analyzed using Neo4J and Linkurious for visualization



High Availability



Summary

- Graph database offers a unique tool and features to handle graph data
- Very useful tool to solve problems that may be very hard to solve and analyze using traditional RDBMS

References

- Neo4J Documentation
  - https://neo4j.com/docs/
- Cypher RefCard
  - http://neo4j.com/docs/pdf/neo4j-cypher-refcard-stable.pdf
- Learning Neo4J
  - Rik Van Bruggen, Packt Publishing, 2014
- Neo4J and Panama Paper article in DZone
  - https://dzone.com/articles/analyzing-the-panama-papers-with-neo4j-data-models