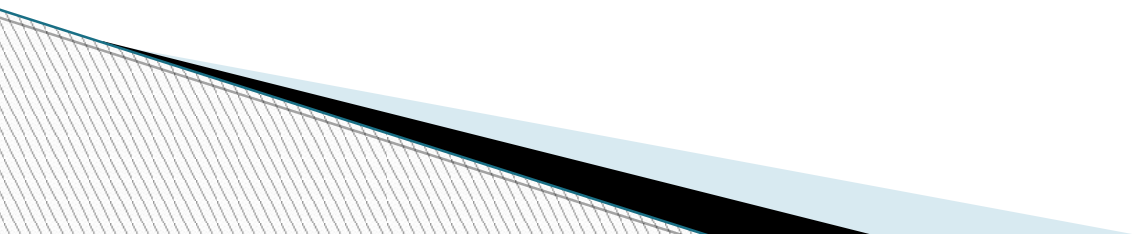
Neo4j

Sarvesh Nagarajan

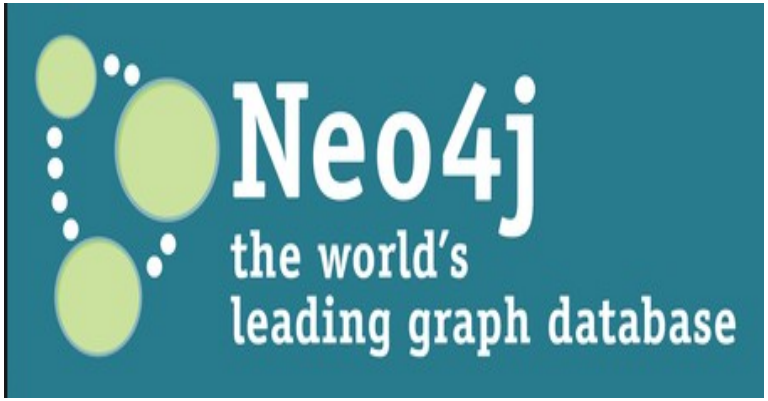
Outline

- What is Neo4j?
- Graph Databases
- Cypher
- Application Domains



What is Neo4j

- Developed by Neo Technologies
- Most Popular Graph Database
- Implemented in Java
- Open Source

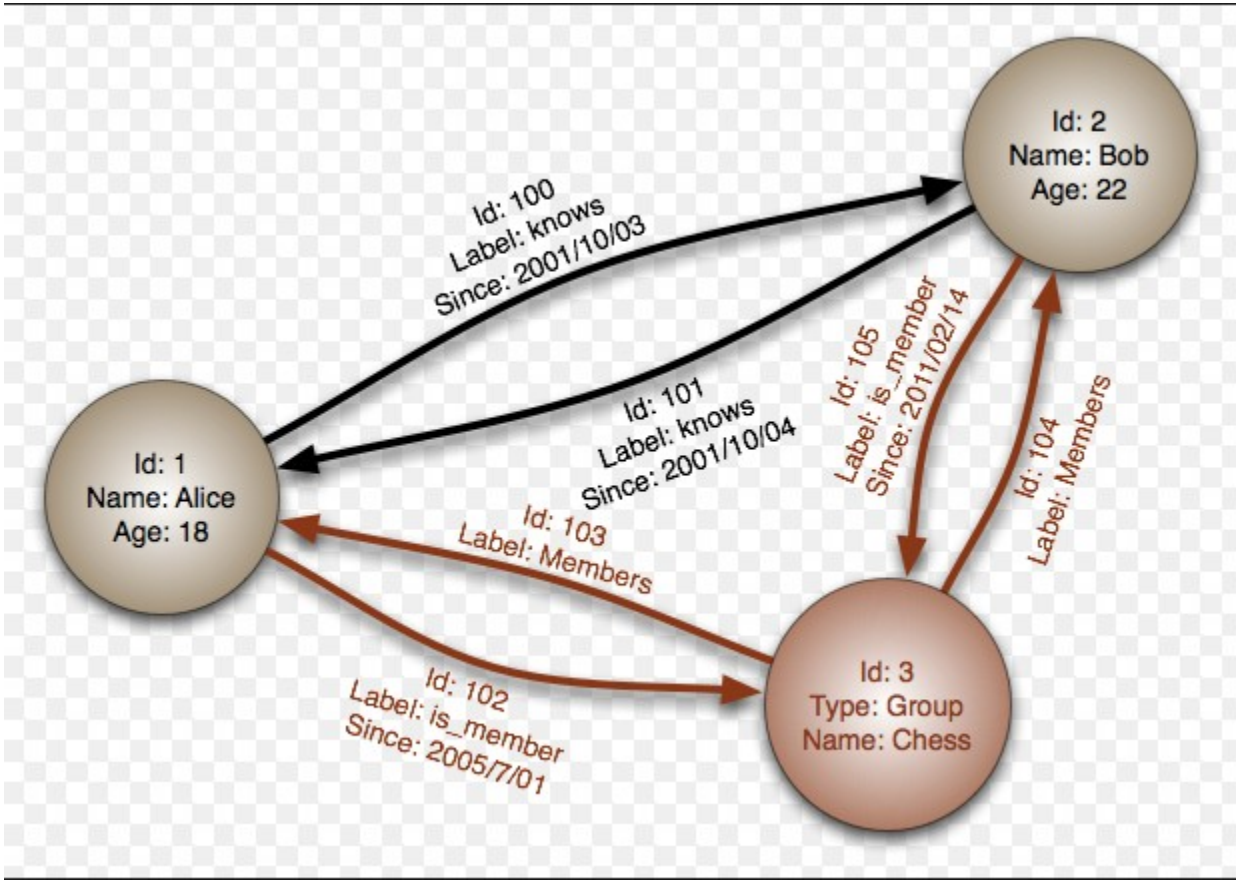


(www.neo4j.org)

Graph Databases

- Database that uses graph structures with nodes, edges and properties to store data
- Provides index-free adjacency
  - Every node is a pointer to its adjacent element
- Edges hold most of the important information and connect
  - nodes to other nodes
  - nodes to properties

Graph Databases



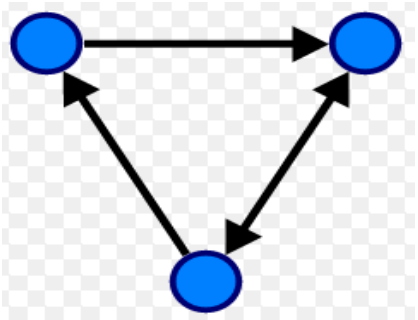
(Wikipedia)

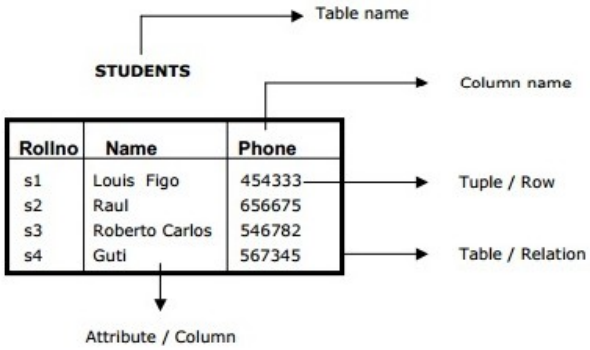
Advantage of Graph Databases

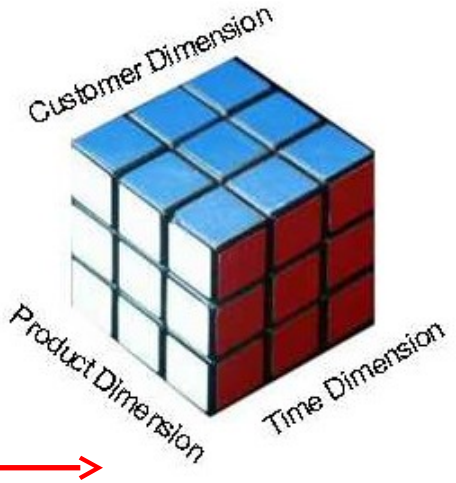
- When there are relationships that you want to analyze Graph databases become a very nice fit because of the data structure
- Graph databases are very fast for associative data sets
  - Like social networks
- Map more directly to object oriented applications
  - Object classification and Parent->Child relationships

Disadvantages

- If data is just tabular with not much relationship between the data, graph databases do not fare well
- OLAP support for graph databases is not well developed
  - Lots of research happening in this area







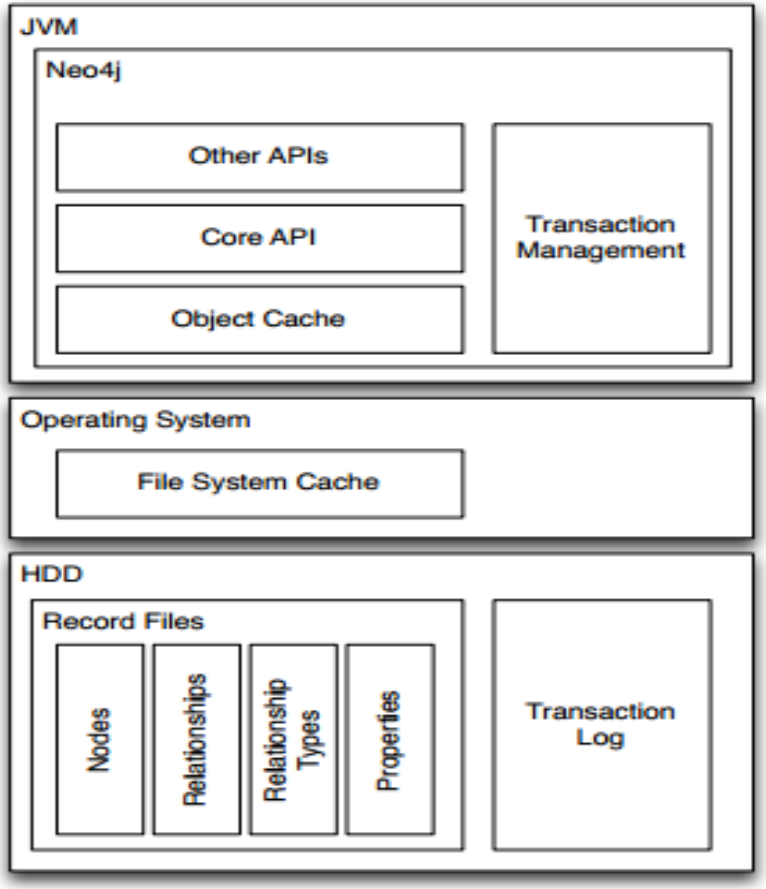
Ease of aggregation

Salient features of Neo4j

- Neo4j is schema free – Data does not have to adhere to any convention
- ACID – atomic, consistent, isolated and durable for logical units of work
- Easy to get started and use
- Well documented and large developer community
- Support for wide variety of languages
  - Java, Python, Perl, Scala, Cypher, etc

(Mistry, Deep, 2013)

Neo4j Software Architecture



(Bachman, 2013, p.11)

Cypher

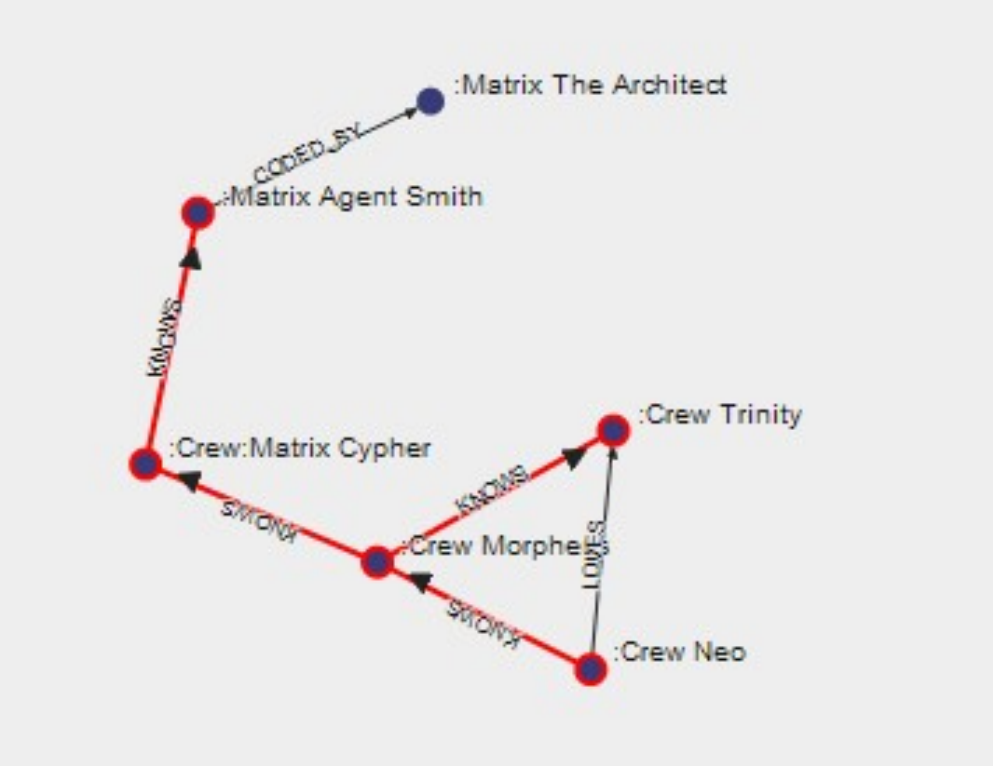
- Query Language for Neo4j
- Easy to formulate queries based on relationships
- Many features stem from improving on pain points with SQL such as join tables

(Hunger, Michael 2013)

Cypher

```
CREATE (Neo:Crew { name:'Neo' })
```

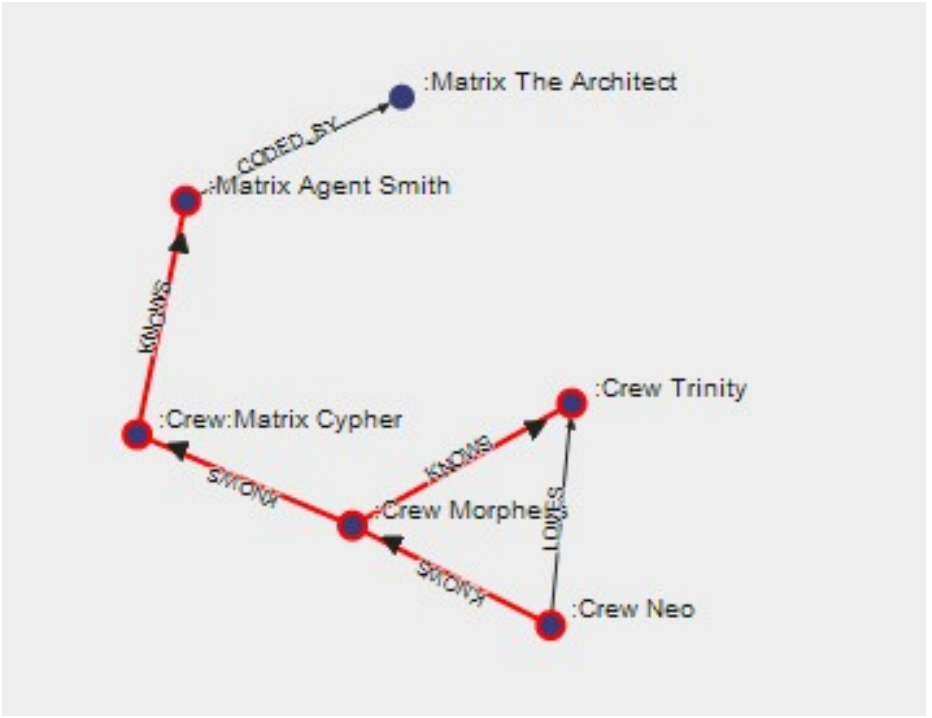
```
(Neo)-[:KNOWS]->(Morpheus)
```



(www.neo4j.org/learn/cypher)

Cypher

```
Query:
MATCH (n:Crew)-[r:KNOWS*]-m
WHERE n.name='Neo'
RETURN n AS Neo,r,m
```



Neo	r	m
{name:"Neo"}	[(0)-[0:KNOWS]->(1)]	(1:Crew {name:"Morpheus"})
{name:"Neo"}	[(0)-[0:KNOWS]->(1), (1)-[2:KNOWS]->(2)]	(2:Crew {name:"Trinity"})
{name:"Neo"}	[(0)-[0:KNOWS]->(1), (1)-[3:KNOWS]->(3)]	(3:Crew:Matrix {name:"Cypher"})
{name:"Neo"}	[(0)-[0:KNOWS]->(1), (1)-[3:KNOWS]->(3), (3)-[4:KNOWS]->(4)]	(4:Matrix {name:"Agent Smith"})

(www.neo4j.org/learn/cypher)

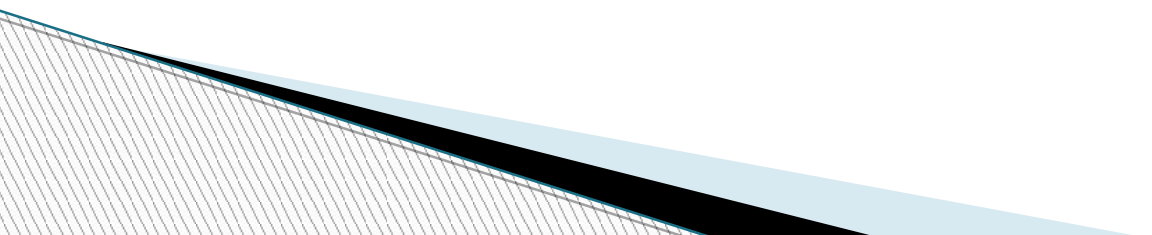
Application Domains

(www.neo4j.org)

Conclusion

- Key questions to ask yourself
  - Is my data going to have a lot of relationships?
  - What sort of questions would I like to ask my database?
- Neo4j is a fantastic Graph Database

Questions?



References

- http://www.neo4j.org
- http://www.neo4j.org/learn/cypher
- Bachman, Michal (2013). GraphAware: Towards Online Analytical Processing in Graph Databases
  - http://graphaware.com/assets/bachman-msc-thesis.pdf
- Hunger, Michael (2012). Cypher and Neo4j
  - http://vimeo.com/83797381
- Mistry, Deep (2013). Neo4j: A Developer's Perspective
  - http://osintegrators.com/opensoftwareintegrators%7Cneo4jadevelopersperspective
- Wikipedia (Neo4j, Graph Database)