Graph Data Management

A survey on addressing The BigData Challenge With A Graph

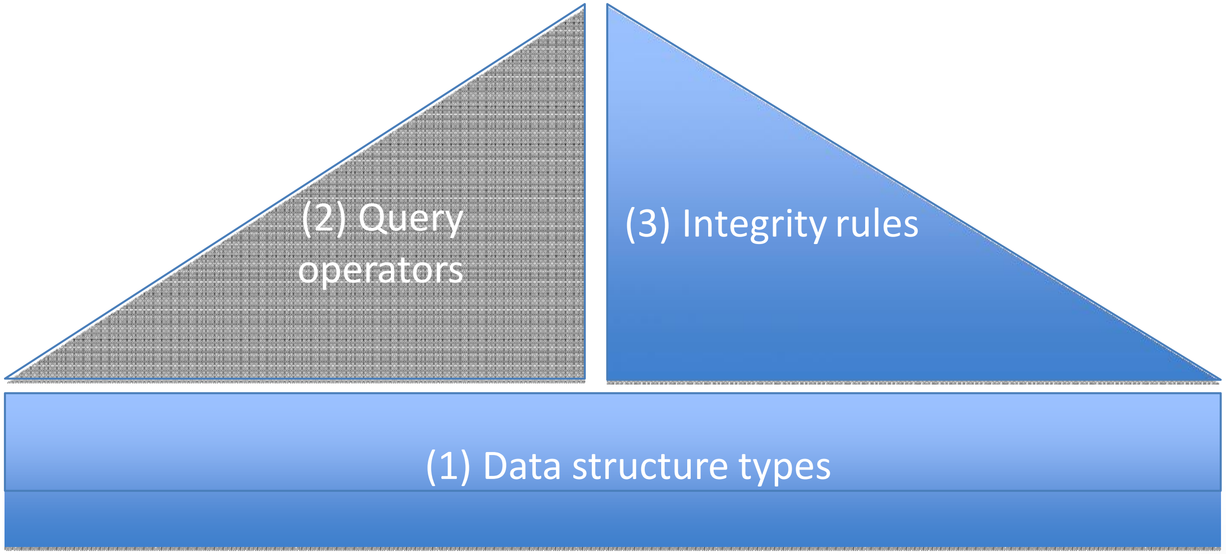
Sore Shewangizaw Dogda

Outline

- Introduction
- Graph data modeling
- Graph database models Historical view
- The Neo4j
  - What is Neo4j?
  - Graph Databases
  - Cypher
  - Application Domains

Introduction to Database models

- A data model is a collection of conceptual tools used to model real-world entities and the relationships among them [Silberschatz et al. 1996].
- A DB model consists of 3 components [Codd 1980]:



Types and characterstics of the Most Influential Database models /comparison/

DataBase Model	Abstraction level	Base data structure	Information
Network	physical	Pointers + records	records
Relational	Logical	Relations	Data + attributes
Semantic	User	Graph	schema
Object –Oriented	Physical /logical	Objects	Objects + methods
Semi structured	Logical	Tree	Data + components
Graph	Logical/user	Graph	Data + relations

Motivation for the Graph Datamodel:

- Attempt to overcome limitations imposed by traditional DB models with respect to capturing the inherent graph structure of data appearing in applications such as hypertext or GIS..

Graph data modeling:

1) Data and/or schema:
   -Represented by graph. e.g. simple graphs (nodes + edges + labels + direction)
   -Or by data structure generalizing the notion of graph.
2) Data manipulation:
   - Expressed by graph transformation/ operations. (main primitives are on graph features)
   -Operations like - Paths, neighborhood, sub graphs, patterns, connectivity, statistics.
3) Integrity constraints: Constraints grouped into
   - schema-instance consistency,
   - Identity and reference integrity,
   - Function and inclusion dependencies
   - ..

Why a graph data model?

- For applications where 'Interconnectivity and topology' comes.
- Allows more natural modeling visible to user.
  E.g. GIS – represents information as nodes, relations as arcs.
- Queries can refer directly to graph structure. So, we can do specific graph operations like – shortest path, sub graph determining etc.

- Implementation – to GraphDB- may provide special graph storage structure and efficient graph algorithm for realizing specific operations.

Graph Databases

- Databases that use 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 are Designed to:

1. Store inter-connected data
2. Make it easy to make sense of that data
3. Enable extreme-performance operations for:
   - Discovery of connected data patterns
   - Relatedness queries > depth 1
   - Relatedness queries of arbitrary length
4. Make it easy to evolve the database

…more on Graph Databases

- **There are two important properties of graph database technologies:**
- **Graph Storage**
  - Some graph databases use native graph storage that is specifically designed to store and manage graphs, while others use relational or object-oriented databases instead. Non-native storage is often much more latent.
- **Graph Processing Engine**
  - Native graph processing (a.k.a. “index-free adjacency”) is the most efficient means of processing graph data since connected nodes physically “point” to each other in the database. Non-native graph processing uses other means to process CRUD operations.

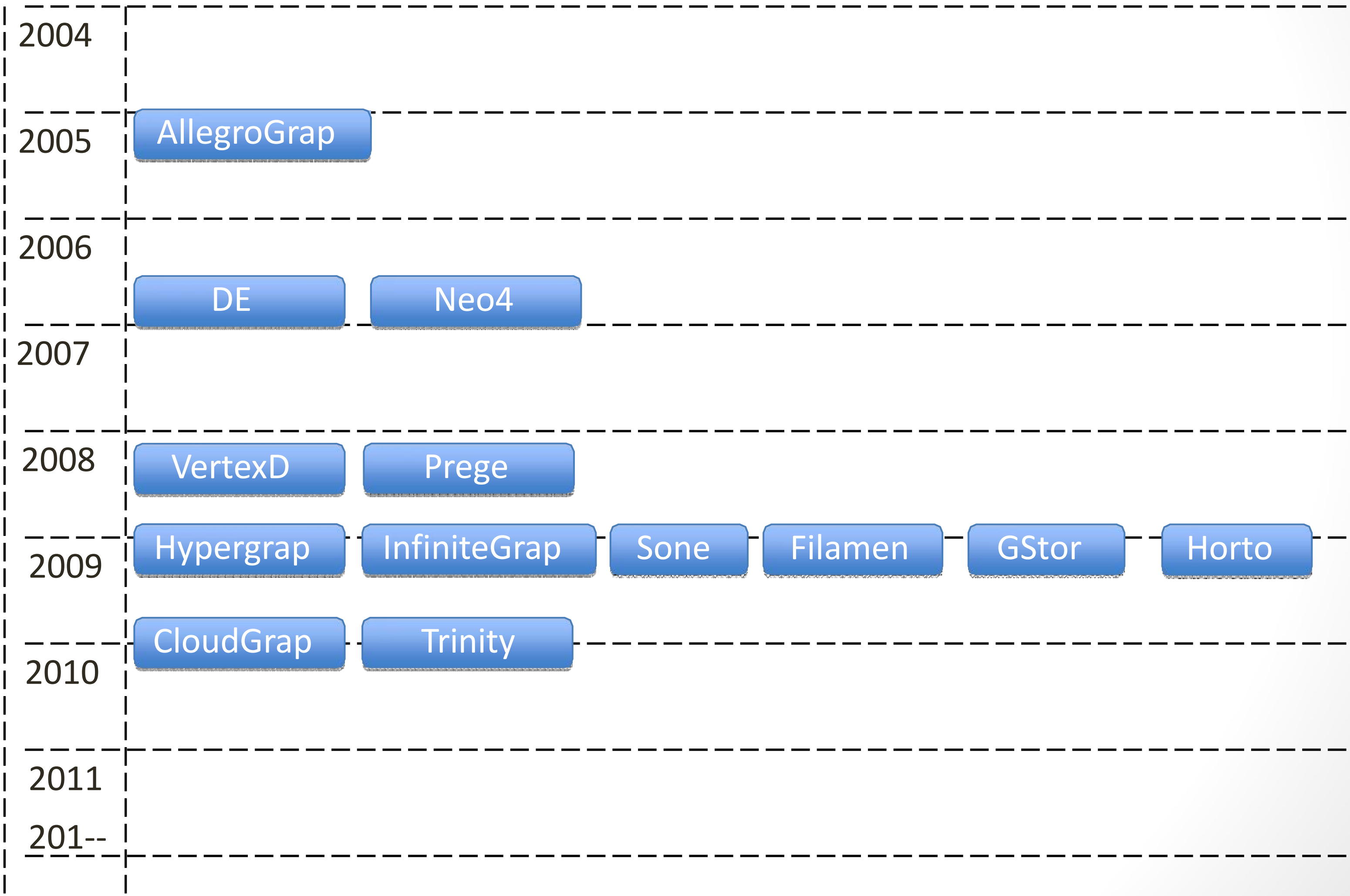
https://neo4j.com/why-graph-databases/

Graph databases

1975	R&M		
…			
1984	LDM		
. . .			
1987	G-Base		
1988	O2		
1989	Tompa		
1990	GOOD	W&X	Hypernode
1991	GROOV		
1992	GMOD	Gram	Simatic-XT
1993	PaMal	GOAL	Hy+
1994	GraphDB	DGV	Hypernode2
1995	G-Log	GGL	Hypernode3
1996	GRAS		
…			
1999	GOQL		
…			
2002			

R. Angles and C. Gutierrez. “Survey of Graph Database Models”. ACM Comp. Surveys, 2008.

… contd



2004
2005
AllegroGrap
2006
DE
Neo4
2007
2008
VertexD
Prege
2009
Hypergrap
InfiniteGrap
Sone
Filamen
GStor
Horto
CloudGrap
Trinity
2010
2011
201--

Top Reasons People Use Graph Databases

1. **Problems with Join performance.**
2. **Continuously evolving data set** (often involves wide and sparse tables)
3. The **Shape of the Domain** is naturally a graph
4. **Open-ended business requirements** necessitating fast, iterative development.

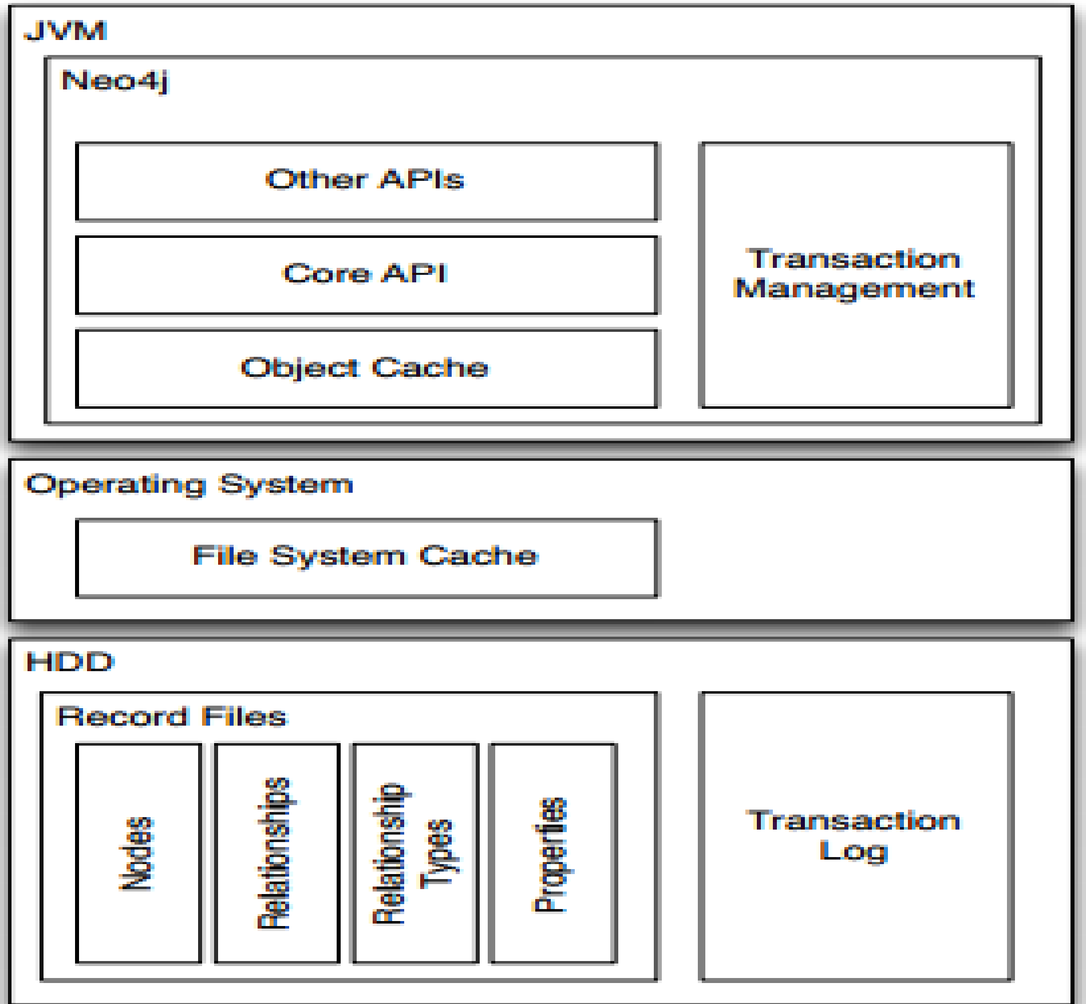
Neo4j

What is Neo4j

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

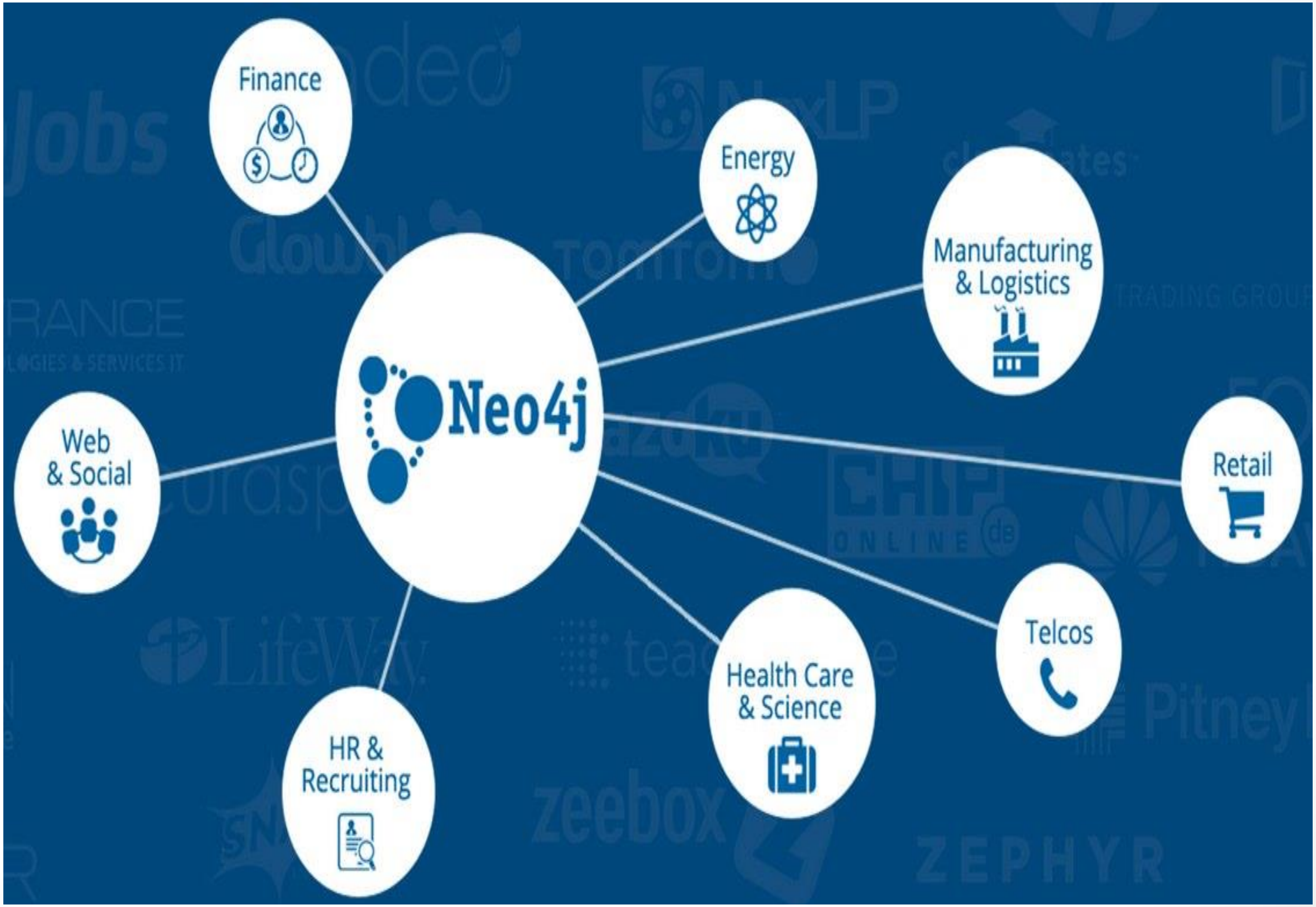
(www.neo4j.org)

Neo4j Software Architecture



(Bachman, 2013, p.11)

Use cases of neo4j



Cypher

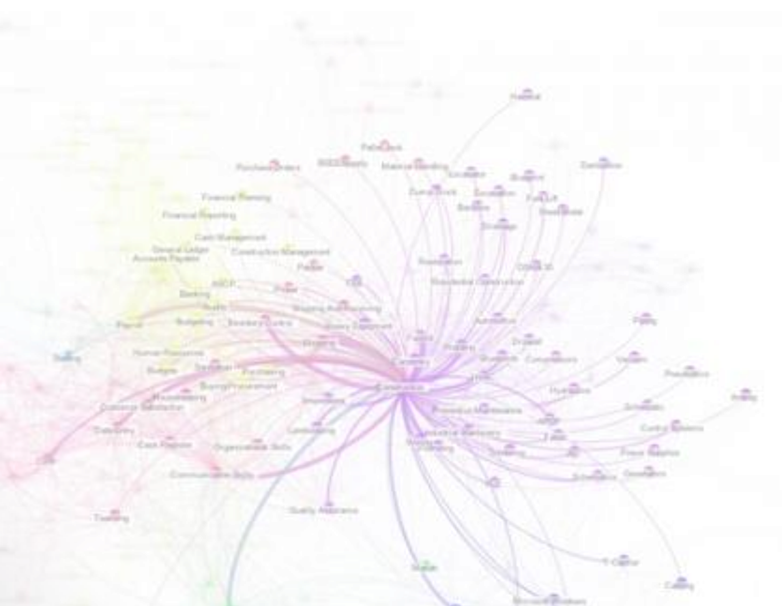
- Query Language for Neo4j
- Easy to formulate queries based on relationships
- Many features stem from improving on pain points with SQL such as join tables

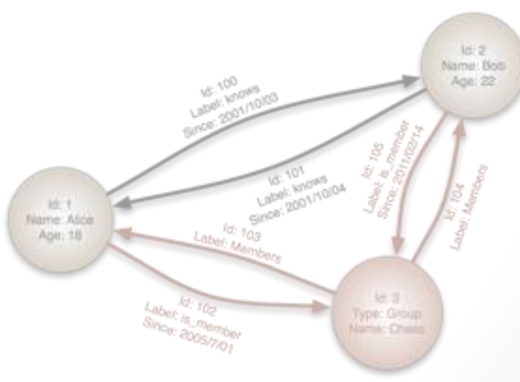
(Hunger, Michael 2013)

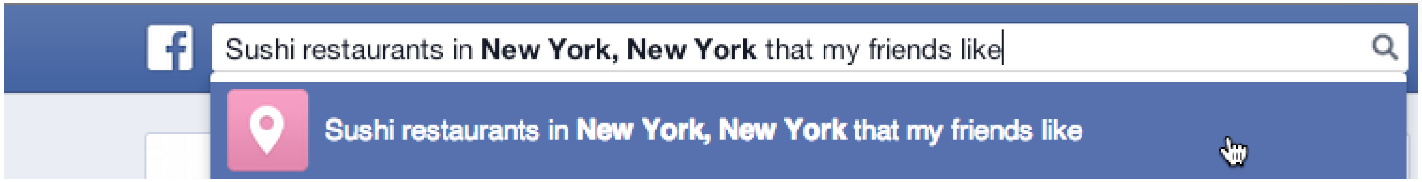
Working with Graphs

Use Cases & Working Examples

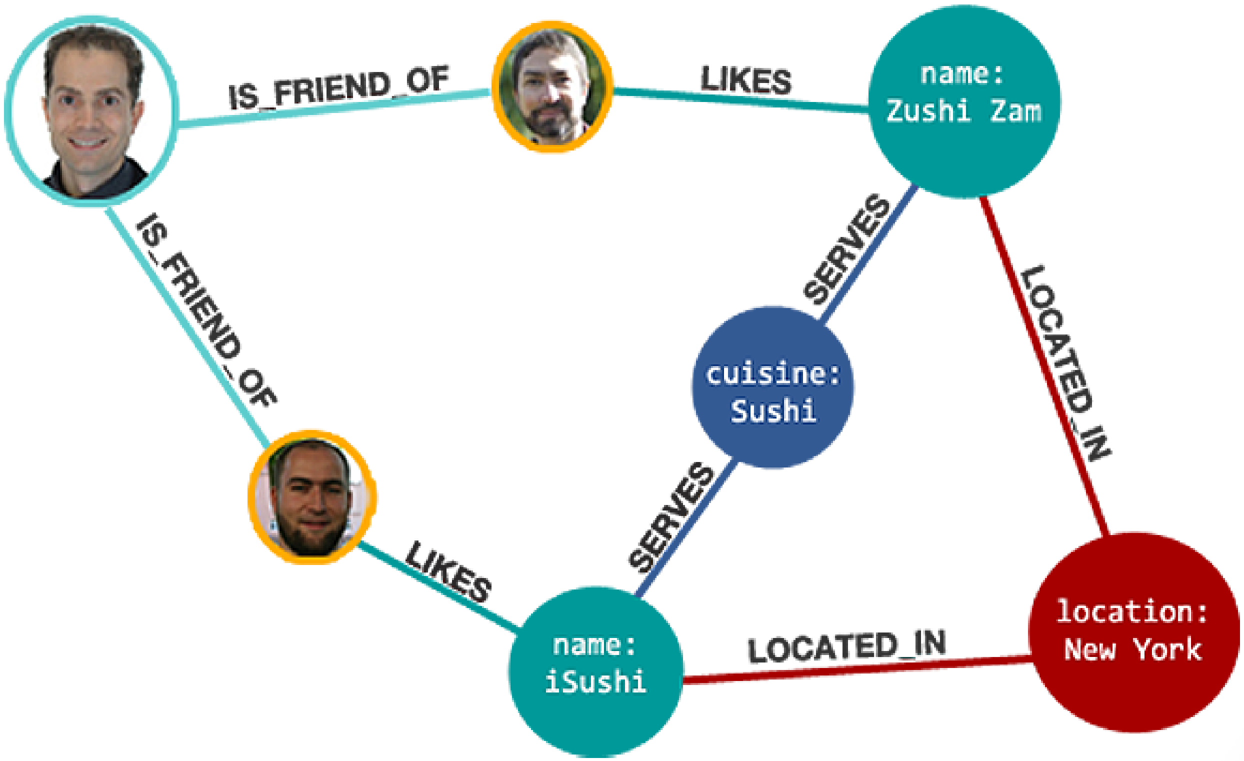
Social Example



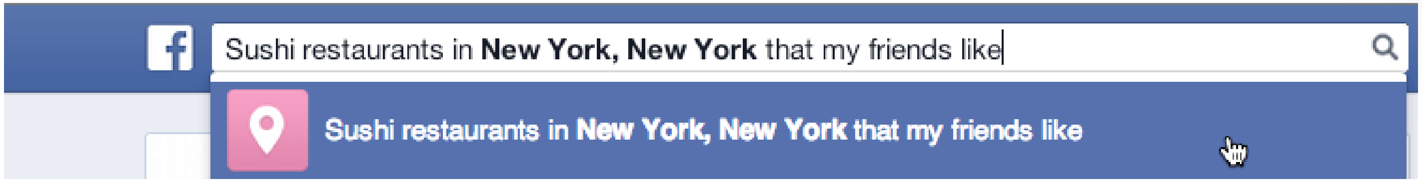




Sushi restaurants in New York, New York that my friends like
Sushi restaurants in New York, New York that my friends like



IS_FRIEND_OF
LIKES
name: Zushi Zam
IS_FRIEND_OF
SERVES
LOCATED_IN
cuisine: Sushi
SERVES
LIKES
name: iSushi
LOCATED_IN
location: New York




```
MATCH (me:Person)-[:IS_FRIEND_OF]->(friend),
    (friend)-[:LIKES]->(restaurant),
      restaurant)-[:LOCATED_IN]->(city:Location),
      (restaurant)-[:SERVES]->(cuisine:Cuisine)

WHERE me.name = 'Philip' AND city.location='New York' AND
      cuisine.cuisine='Sushi'

RETURN restaurant.name
```

http://maxdemarzi.com/?s=facebook

** Cypher query language example*

Connected Query Performance

Query Response Time = f(graph density, graph size, query degree)*

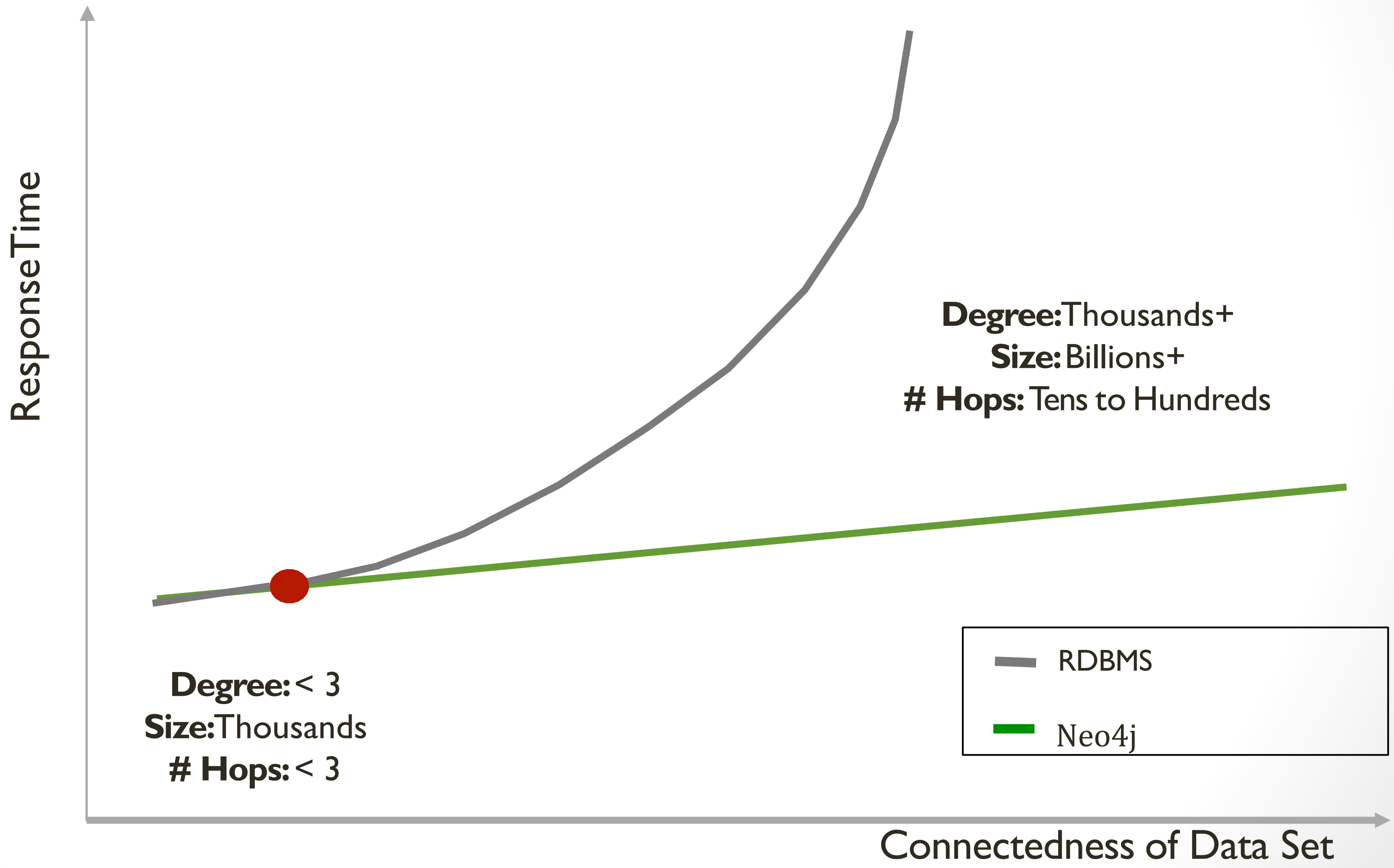
- **Graph density** (avg # rel's / node)
- **Graph size** (total # of nodes in the graph)
- **Query degree** (# of hops in one's query)

RDBMS:
>> exponential slowdown as each factor increases

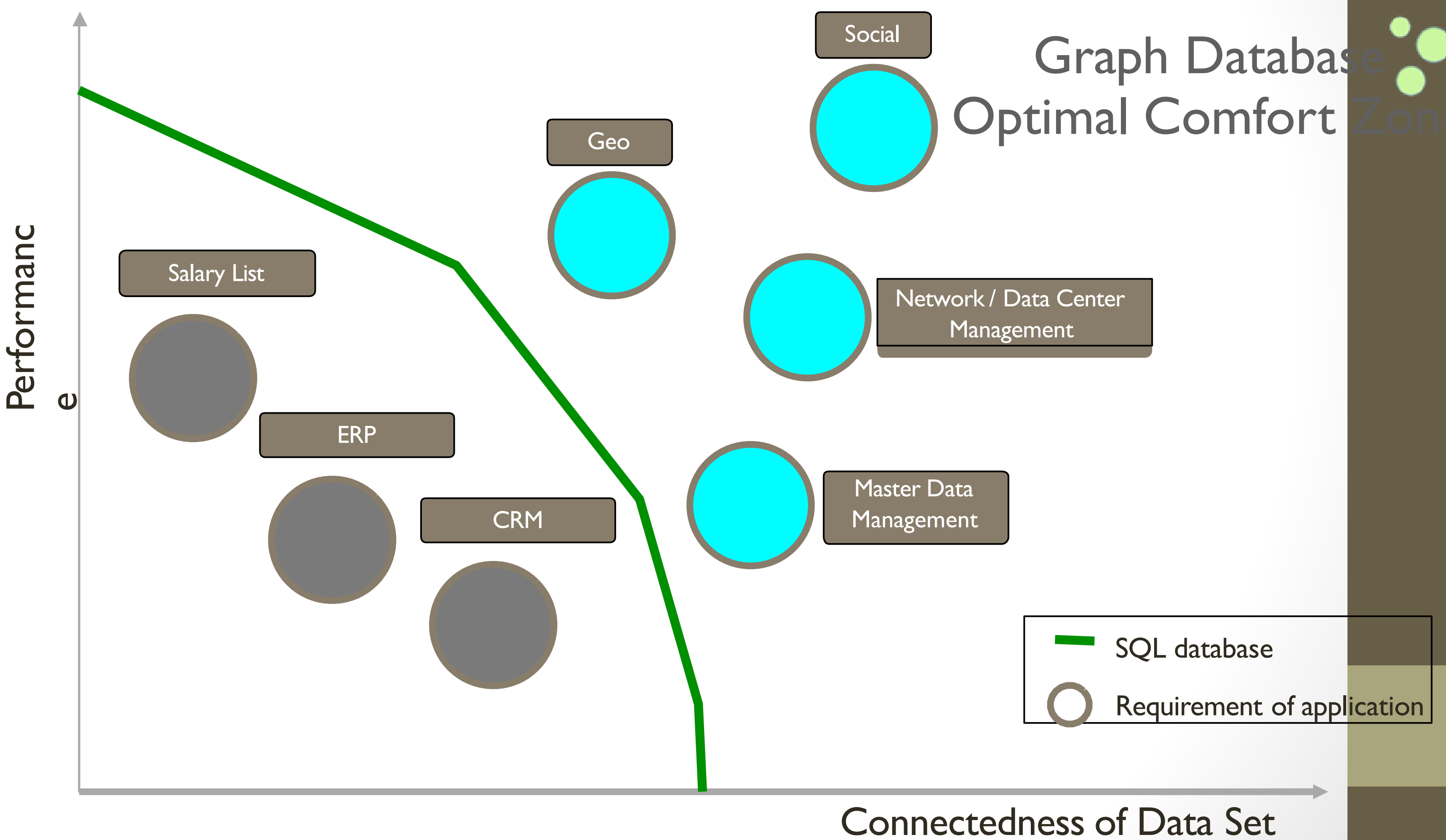
Neo4j:
>> Performance remains constant as graph size increases
>> Performance slowdown is linear or better as density & degree increase

Connected Query Performance

RDBMS vs. Native Graph Database



The Zone of SQL Adequacy

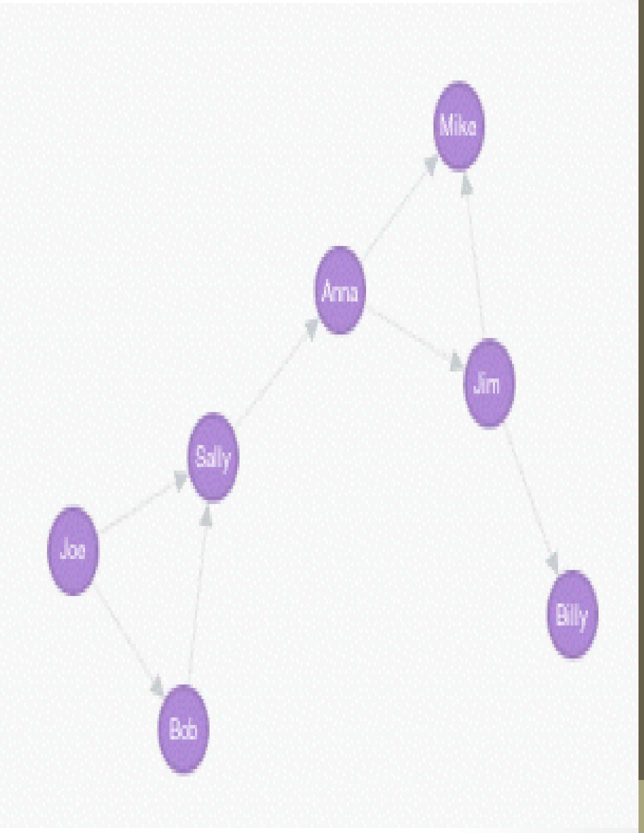


Practical Cypher

Social Graph - Create

```
CREATE
    (joe:Person  {name:"Joe"}),
    (bob:Person  {name:"Bob"}),
    (sally:Person {name:"Sally"}),
    (anna:Person  {name:"Anna"}),
    (jim:Person   {name:"Jim"}),
    (mike:Person  {name:"Mike"}),
    (billy:Person {name:"Billy"}),

    (joe)-[:KNOWS]->(bob),
    (joe)-[:KNOWS]->(sally),
    (bob)-[:KNOWS]->(sally),
    (sally)-[:KNOWS]->(anna),
    (anna)-[:KNOWS]->(jim),
    (anna)-[:KNOWS]->(mike),
    (jim)-[:KNOWS]->(mike),
    (jim)-[:KNOWS]->(billy)
```

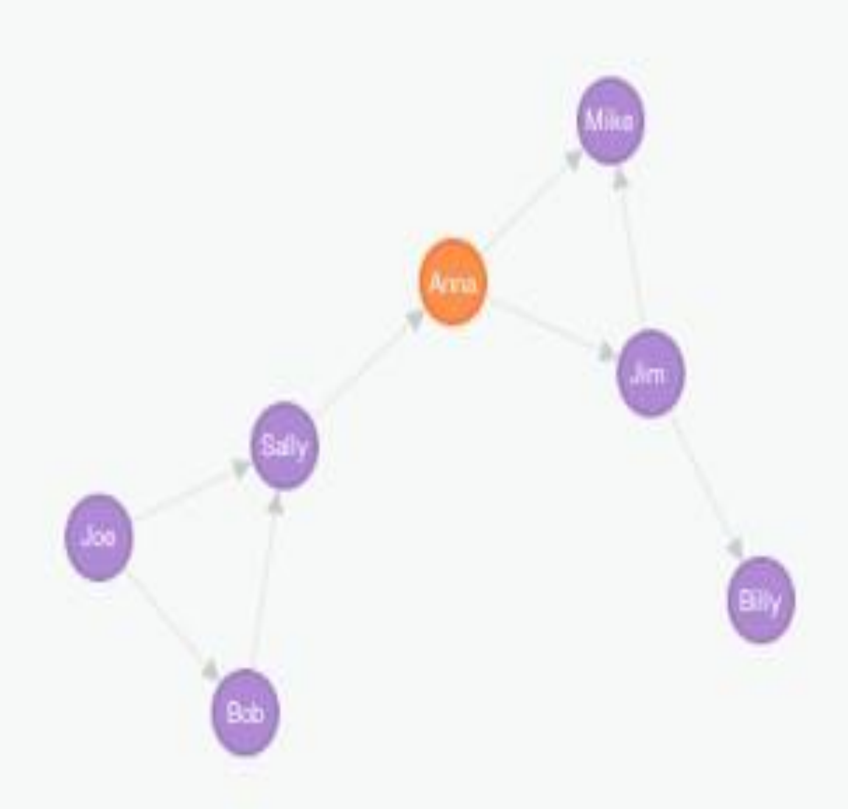


Practical Cypher

Social Graph - Friends of Joe's Friends

```
 MATCH (person)-[:KNOWS]-(friend),
       (friend)-[:KNOWS]-(foaf)
 WHERE person.name = "Joe"
   AND NOT(person-[:KNOWS]-foaf)
RETURN  foaf
```

foaf
{name:"Anna"}



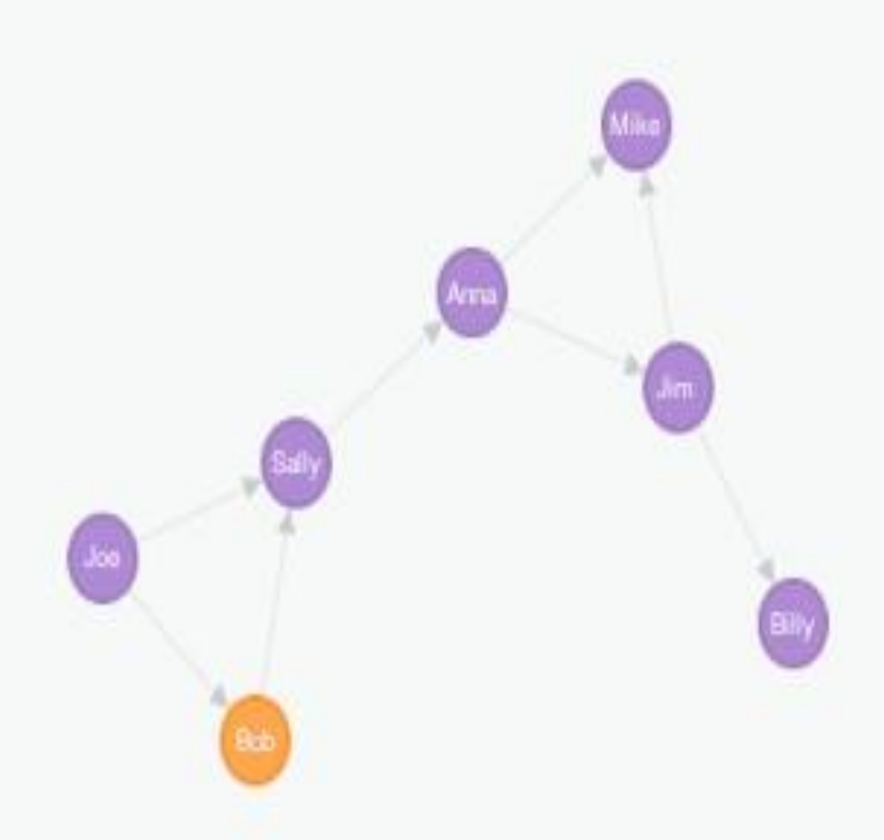
Practical Cypher

Social Graph - Common Friends

```
MATCH (person1)-[:KNOWS]-(friend),
      (person2)-[:KNOWS]-(friend)

WHERE person1.name = "Joe"
  AND person2.name ="Sally"
RETURN friend
```

friend
{name:"Bob"}



Practical Cypher

Social Graph - Shortest Path

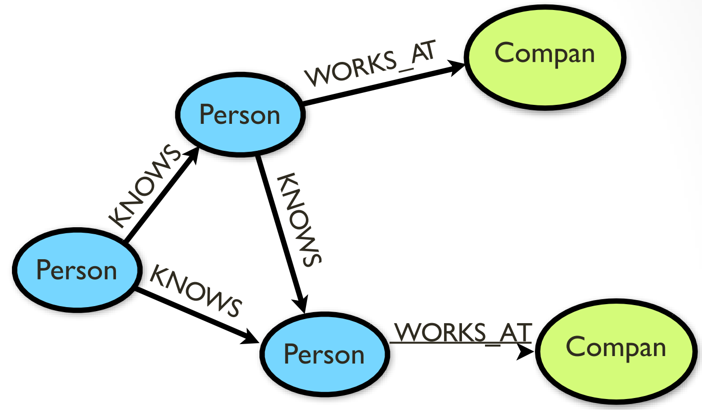
```
MATCH path = shortestPath (
  (person1)-[:KNOWS*..6]-(person2))
WHERE person1.name = "Joe"
   AND person2.name = "Billy"
RETURN path
```

path
{start:"13759", nodes:["13759","13757","13756","13755","13753"], length:4, relationships:["101407","101409","101410","101413"], end:"13753"}

Industry: Online Job Search
Use case: Social / Recommendations
Sausalito, CA

Background

- Online jobs and career community, providing anonymized inside information to job s



Business problem

- Wanted to leverage known fact that most jobs are found through personal & professional connections
- Needed to rely on an existing source of social network data. Facebook was the ideal choice.
- End users needed to get instant gratification
- Aiming to have the best job search service, in a very competitive market

Solution & Benefits

- First-to-market with a product that let users find jobs through their network of Facebook friends
- Job recommendations served real-time from Neo4j
- Individual Facebook graphs imported real-time into Neo4j
- Glassdoor now stores > 50% of the entire Facebook social graph
- Neo4j cluster has grown seamlessly, with new instances being brought online as graph size and load have increased

Application Domains (more graphs on the real world)

(www.neo4j.org)

Deutsche Telekom

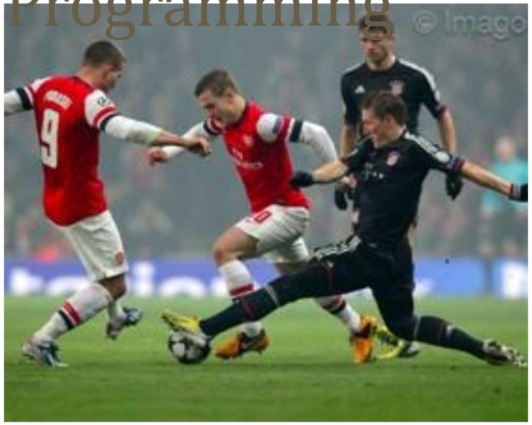
Industry: Communications
Use case: Social gaming
Frankfurt, Germany

Background

- Europe's largest communications company
- Provider of mobile & land telephone lines to consumers and businesses, as well as internet services, television, and other services

> 236,000	50	> 58 bn. €
Employees worldwide in 2011	Countries	Revenue in 2011

Interactive Television Programming



Business problem

- The Fanorakel application allows fans to have an interactive experience while watching sports
- Fans can vote for referee decisions and interact with other fans watching the game
- Highly connected dataset with real-time updates
- Queries need to be served real-time on rapidly changing data
- One technical challenge is to handle the very high spikes of activity during popular games

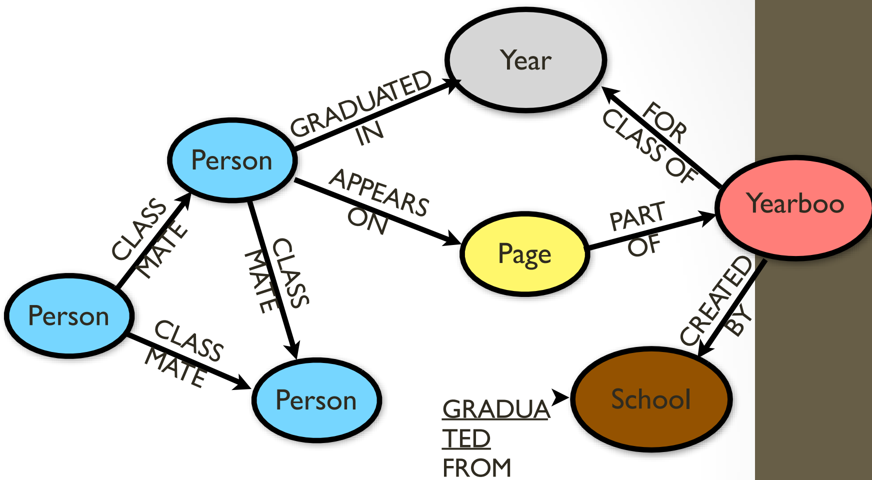
Solution & Benefits

- Interactive, social offering gives fans a way to experience the game more closely
- Increased customer stickiness for Deutsche Telekom
- A completely new channel for reaching customers with information, promotions, and ads
- Clear competitive advantage

Industry: Social Network
Use case: Social / Recommendations
Seattle,WA

Background

- Memory Lane, Inc. was founded in 1995 and based in Seattle, Washington. Subsidiary of United Online, Inc.
- Classmates.com, operates an online yearbook that connects members in the United States and Canada with friends and acquaintances from school, work, and the military.
- Evolving toward more sophisticated social networking capability



Business problem

- Develop new Social capabilities to help monetize Yearbook-related offerings
  - Show me all the people I know in a yearbook
  - Show me yearbooks my friends appear in most often (i.e. "Top yearbooks to look at")
  - Show me sections of a yearbook that your friends appear most in (i.e. "8 of your friends are on page 12 with the football team)
  - Show me other high schools that my friends went to (i.e. friends you made in other schools)

Solution & Benefits

- 3-Instance Neo4j Cluster with Cache Sharding + Disaster-Recovery Cluster
- Neo4j provides18 ms response time for the top 4 queries
- Initial graph size: 100M nodes and 600M relationships
  - People, Images, Schools, Yearbooks, Yearbook Pages
- Projected to grow to 1B nodes & 6B relationships

maaii
Let's connect

Industry: Communications
Use case: Social, Mobile
Hong Kong

Background

- Hong Kong based telephony infrastructure provider (aka M800 aka Pop Media)
- Exclusive China Mobile partner for international toll-free services. SMS Hub & other offerings
- 2012 Red Herring Top 100 Global Winner

Business problem

- Launched a new mobile communication app "Maaii" allowing consumers to communicate by voice & text (Similar to Line, Viber, Rebtel, VoxOx...)
- Needed to store & relate devices, users, and contacts
- Import phone numbers from users' address books. Rapidly serve up contacts from central database to the mobile app
- Currently around 3M users w/200M nodes in the graph

Solution & Benefits

- Quick transactional performance for key operations:
  - friend suggestions ("friend of friend")
  - updating contacts, blocking calls, etc.
  - etc.
- High availability telephony app uses Neo4j clustering
- Strong architecture fit: Scala w/Neo4j embedded

Industry: Logistics
Use case: Parcel Routing

Background

- One of the world's largest logistics carriers
- Projected to outgrow capacity of old system
- New parcel routing system
    - *Single source of truth for entire network*
    - *B2C & B2B parcel tracking*
    - *Real-time routing: up to 5M parcels per day*

Business problem

- 24x7 availability, year round
- Peak loads of 2500+ parcels per second
- Complex and diverse software stack
- Need predictable performance & linear scalability
- Daily changes to logistics network: route from any point, to any point

Solution & Benefits

- Neo4j provides the ideal domain fit:
    - a logistics network is a graph
- Extreme availability & performance with Neo4j clustering
- Hugely simplified queries, vs. relational for complex routing
- Flexible data model can reflect real-world data variance much better than relational
- "Whiteboard friendly" model easy to understand

Industry: Health Care
Use case: Recommedatations
Newton, Massachusetts

Background

- Founded in 1999. Widely considered the industry leader in patient management for discharges & referrals
- Manage patient referrals for more than 4600 health care facilities
- Connects providers, payers and suppliers via secure electronic patient-transition networks, and web-based patient management platform

Business problem

- Satisfy complex "Graph Search" queries by discharge nurses and intake coordinators, e.g.: *"Find a skilled nursing facility within n miles of a given location, belonging to health care group XYZ, offering speech therapy and cardiac care, and optionally Italian language services"*
- Real-time Oracle performance not satisfactory
- New functionality called for more complexity, including granular role-based access control

Solution & Benefits

- Fast real-time performance needs now satisfied
- Queries span multiple hierarchies, including provider graph & employee permissions graph
- Graph data model provided a strong basis for adding more dimensions to the data, such as insurance networks, service areas, and ACOs (Accountable Care Organizations)
- Some multi-page SQL statements have been turned into one simple function with Neo4j

Industry: Health Care
Use case: ioinformatics
Cambridge, Massachusetts

Background

- Clinical diagnostics company specializing in genetic carrier screening for inherited diseases
- Founded in 2008 by Harvard Business School & Harvard Medical School graduates
- Two sides of the business: Clinical and R&D
- Particularly strong in the detection of rare alleles and measuring frequency in the population

Business problem

- Clinical data split across several operational databases that are not structured for discovery
- Needed an easy query mechanism for scientists who are not data scientists."Graph search" for bioinformatics.
- Much in Bioinformatics remains unknown: having to specifying a schema ahead of time can range from difficult to impossible.

Solution & Benefits

- New R&D database build atop Neo4j to support information discovery by scientists
- Lightweight web front end allows simple Cypher queries to be constructed ad hoc
- RawVCF sequence data imported into Neo4j, along with clinical data from Oracle database
- Time to answer new questions went from days of ad-hoc information gathering to hours or minutes

Conclusion

- Key questions to ask yourself to use GraphDB
  - Is my data going to have a lot of relationships?
  - What sort of questions would I like to ask my database?

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)

Thank You !!