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]:

(1) Data structure types
(2) Query operators
(3) Integrity rules
Types and characteristics of the Most Influential Database models /comparison/

<table>
<thead>
<tr>
<th>Database Model</th>
<th>Abstraction level</th>
<th>Base data structure</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>physical</td>
<td>Pointers + records</td>
<td>records</td>
</tr>
<tr>
<td>Relational</td>
<td>Logical</td>
<td>Relations</td>
<td>Data + attributes</td>
</tr>
<tr>
<td>Semantic</td>
<td>User</td>
<td>Graph</td>
<td>schema</td>
</tr>
<tr>
<td>Object-Oriented</td>
<td>Physical/logical</td>
<td>Objects</td>
<td>Objects + methods</td>
</tr>
<tr>
<td>Semi structured</td>
<td>Logical</td>
<td>Tree</td>
<td>Data + components</td>
</tr>
<tr>
<td>Graph</td>
<td>Logical/user</td>
<td>Graph</td>
<td>Data + relations</td>
</tr>
</tbody>
</table>

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
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

<table>
<thead>
<tr>
<th>Year</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1975</td>
<td>R&amp;M</td>
</tr>
<tr>
<td>1984</td>
<td>LDM</td>
</tr>
<tr>
<td>1987</td>
<td>G-Base</td>
</tr>
<tr>
<td>1988</td>
<td>O2</td>
</tr>
<tr>
<td>1989</td>
<td>Tompa</td>
</tr>
<tr>
<td>1990</td>
<td>GOOD</td>
</tr>
<tr>
<td>1991</td>
<td>GROOV</td>
</tr>
<tr>
<td>1992</td>
<td>GMOD</td>
</tr>
<tr>
<td>1993</td>
<td>PaMal</td>
</tr>
<tr>
<td>1994</td>
<td>GraphDB</td>
</tr>
<tr>
<td>1995</td>
<td>G-Log</td>
</tr>
<tr>
<td>1996</td>
<td>GRAS</td>
</tr>
<tr>
<td>1999</td>
<td>GOQL</td>
</tr>
</tbody>
</table>

...  

R. Angles and C. Gutierrez.  

“Survey of Graph Database Models”.  

... contd
Top Reasons People Use Graph Databases


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

Neo4j
the world’s leading graph database
What is Neo4j

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

\[ \text{Query Response Time}^* = f(\text{graph density, graph size, query degree}) \]

- **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

**Connectedness of Data Set**

- **RDBMS**
  - Degree: < 3
  - Size: Thousands
  - # Hops: < 3

- **Neo4j**
  - Degree: Thousands+
  - Size: Billions+
  - # Hops: Tens to Hundreds
The Zone of SQL Adequacy

Graph Database
Optimal Comfort Zone

- Social
- Network / Data Center Management
- Master Data Management
- Geo
- Social
- Salary List
- ERP
- CRM

Connectedness of Data Set vs. Performance

SQL database
Requirement of application
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)
MATCH (person)-[:KNOWS]-(friend),
(friend)-[:KNOWS]-(foaf)
WHERE person.name = "Joe"
AND NOT (person-[:KNOWS]-foaf)
RETURN foaf

<table>
<thead>
<tr>
<th>foaf</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ name: &quot;Anna&quot; }</td>
</tr>
</tbody>
</table>
MATCH (person1)-[[:KNOWS]]-(friend),
    (person2)-[[:KNOWS]]-(friend)
WHERE person1.name = "Joe"
    AND person2.name = "Sally"
RETURN friend

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

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

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

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)

<table>
<thead>
<tr>
<th>Accenture</th>
<th>Adobe</th>
<th>Global 500 Telcommunication</th>
<th>Global 500 Manufacturing</th>
<th>Career Arc Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Career Builder.com</td>
<td>Chip Online.de</td>
<td>Cisco</td>
<td>clasmates.com</td>
<td>Compete</td>
</tr>
<tr>
<td>Curaspan Health Group</td>
<td>Global 500 Logistics</td>
<td>decibel</td>
<td>die Bayerische</td>
<td>Dingli.com</td>
</tr>
<tr>
<td>Splink</td>
<td>DOWN</td>
<td>DRAKER</td>
<td>Drw Trading Group</td>
<td>Dsini</td>
</tr>
<tr>
<td>Electra</td>
<td>ePals</td>
<td>Equilar</td>
<td>Era7</td>
<td>First Data</td>
</tr>
<tr>
<td>Fovea</td>
<td>fuseworks/&gt;</td>
<td>gamesys</td>
<td>gen</td>
<td>Glassdoor</td>
</tr>
<tr>
<td>GoodStart Genetics</td>
<td>HealthUnlocked</td>
<td>H3g</td>
<td>HP</td>
<td>Huawei</td>
</tr>
<tr>
<td>ice</td>
<td>entropy</td>
<td>id mission</td>
<td>Impact Technologies</td>
<td>Indiatimes</td>
</tr>
<tr>
<td>isar software</td>
<td>janssen</td>
<td>juice Plus</td>
<td>Justdial</td>
<td>Infojobs</td>
</tr>
<tr>
<td>kitedesk</td>
<td>Kiwirail</td>
<td>Laureate Education, Inc</td>
<td>Lifecurch.tv</td>
<td>LifeWay</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**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 Employees worldwide in 2011

> 50 Countries

> 58 bn. € 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
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 provides 18 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
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
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
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
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
- 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)