GRAPH DATABASE SYSTEMS
Use Case: Logistics

Source: Neo Technology, Inc.
Use Case: Social Network

Source: Neo Technology, Inc.
Use Case: Recommendations

Source: Neo Technology, Inc.
Graph Database: Example

Source: Neo Technology, Inc.
Graph Database Systems: Definitions

• “A graph database is a database whose data model conforms to some form of graph (...) structure. The graph data model usually consists of nodes (...) and (...) edges (...), where the nodes represent concepts (...) and the edges represent relationships (...) between these concepts (...). ..” Encyclopedia of Database Systems (Springer)

• “A graph database management system (henceforth, a graph database) is an online database management system with Create, Read, Update, and delete (CRUD) methods that expose a graph data model ...” Robinson/Webber/Eifrem

• “A graph database is any storage system that provides index-free adjacency.” Rodriguez
Relational vs. Graph (Cont.)

Source: Neo Technology, Inc.
Example: Social Network “path exists” Performance
- a sample social graph with ~1,000 persons
- average 50 friends per person
- pathExists(a,b) limited to depth 4
- caches warmed up to eliminate disk I/O

<table>
<thead>
<tr>
<th># persons</th>
<th>query time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relational DBMS</td>
<td>1,000</td>
</tr>
<tr>
<td>Neo4j</td>
<td>1,000</td>
</tr>
<tr>
<td>Neo4j</td>
<td>1,000,000</td>
</tr>
</tbody>
</table>

Source: Neo Technology, Inc.
Graph Databases ...

Are good for ...
• Highly connected data (social networks)
• Recommendations (e-commerce)
• Path Finding (how do I know you?)
• A* (Least Cost path)
• ...

If you’ve ever ...
• Joined more than 7 tables together
• Modeled a graph in a table
• Tried to write some crazy stored procedure with multiple recursive self and inner joins

... you should use a graph database
Graph Database Management Systems

- First graph database: R&M (1975)
- Until 2005: development of different graph models and (research) graph libraries and graph database
- Since 2005: development of new graph databases (commercial and open source)
  - AllegroGraph (2005)
  - DEX ($\Rightarrow$ Sparksee), Neo4j (2006)
  - Tinkerpop Graph Processing Stack (2007)
  - VertexDB, Pregel (Google)(2008)
  - HyperGraphDB, InfiniteGraph, (sones Graph DB), Filament, Horton (Microsoft), ... (2009)
  - CloudGraph, Trinity, FlockDB (Twitter), OrientDB, ... (2010)
  - Titan, ArangoDB, Sqrrl, Giraph (Apache) ... (2012)
  - ...
• **Graph Database Management Systems differ strongly** in
  – Supported graph data structures
  – Data storing features
  – Operation and manipulation features
  – Query Features
  – Schema and integrity constraints
  – Support for essential graph queries

• **Further reading:**
  – Renzo Angles: *A Comparison of Current Graph Database Models*, 3rd Int. Workshop on Graph Data Management: Techniques and applications (GDM 2012), 5 April, Washington DC, USA
Graph Database Management Systems (Cont.)

- „Popularity“
Graph Database Systems – Topics of Interest

• Graph Data Models

• Graph Query Languages (not discussed here)

• Graph Storage and Indexing

• Scalability, Availability and Consistency
Literature

• Edlich, S., Friedland, A., Hampe, J., Brauer, B., Brückner, M. NoSQL Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken, Carl Hanser Verlag, 2011 (2nd ed.)

• Robinson, I., Webber, J., Eifrem, E. Graph Databases, O’Reilly, 2015 (2nd ed.)
free e-Book on http://graphdatabases.com/
Introduction

- Graph Data Models
- Graph Storage and Indexing
- Scalability, Availability and Consistency
Orthogonal graph characteristics
• Directed vs. undirected
• Simple vs. multi
• Weighted vs. unweighted
• Unlabeled vs. edge-labeled vs. vertex-labeled
• …

Property Graph Model
• Most popular graph data model in graph databases today (Tinkerpop, InfiniteGraph, InfoGrid, Neo4j etc.)
• Directed labeled multigraph
• Edge properties: key/value pairs
Property Graph Data Model

Source: Neo Technology, Inc.
Property Graph Data Model

• **Nodes**
  - Entities

• **Relationships**
  - Connect entities and structure domain

• **Properties**
  - Attributes and metadata

• **(Labels)**
  - Group nodes by role
Property Graph Data Model: Nodes

Source: Neo Technology, Inc.
Property Graph Data Model: Relationships

Source: Neo Technology, Inc.
Property Graph Data Model: Relationships

Nodes can have more than one relationship

Self relationships are allowed

Source: Neo Technology, Inc.
Property Graph Data Model: Labels

Source: Neo Technology, Inc.
Property Graph Data Model

Source: Neo Technology, Inc.
• Easy to design and model direct representation of the model – “Whiteboard Friendliness”
Graph Data Modeling (Cont.)

Source: Neo Technology, Inc.
Graph Data Modeling: Best Practice

- Nodes for Things
- Relationships for Structure
- Represent Complex value types as nodes

- Iterative and incremental development
- Test-driven data model development
Graph Data Modelling: Cross-Domain Models

Source: Robinson/Webber/Eifrem: 2013
Graph Data Modelling: Cross-Domain Models

Diagram showing models involving theatrical, literary, and geospatial domains.

Source: Robinson/Webber/Eifrem: 2013
Graph Databases and Schema

• Most graph databases are schema-less
• However, first graph databases started to introduce schema constructs ...

• Example: Neo4j 2.0
  – Necessary requirement: Labels
  – Unique Constraints:
    • Unique constraints do not mean that all nodes have to have a unique value for the properties — nodes without the property are not subject to this rule.
    • CREATE CONSTRAINT / DROP CONSTRAINT

```
CREATE CONSTRAINT ON (book:Book) ASSERT book.isbn IS UNIQUE
```
Graph Data Models

• Further graph data model approaches
  – Hypergraphs with hyperedges (e.g. HyperGraphDB, sonesGraphDB)
  – …

• Related approaches
  – RDF
    • Triples (subject, predicate, object)
    • Standardized (W3C)
    • Optimized for reasoning
    • However, e.g. AllegroGraph: RDF store and graph database
  – …

• Further reading for graph data models:
Graph Databases

✓ Introduction

✓ Graph Data Models

• Graph Storage and Indexing

• Scalability, Availability and Consistency
Graph Storage

Adjacency Matrix vs. Adjacency List

Advantages? Disadvantages?

Source: Edlich et al.:2011
Graph Storage: Example

• Concrete Example: Neo4j – Architecture

Source: Robinson/Webber/Eifrem: 2015
Graph Storage: Example

- Concrete Example: Neo4j – Physical Storage of a Graph

Source: Robinson/Webber/Eifrem: 2015
Graph Storage: Example (Cont.)

- Concrete Example: Neo4j – File Record Structure

<table>
<thead>
<tr>
<th>Node (15 bytes)</th>
<th>Relationship (34 bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>inUse</td>
<td>inUse</td>
</tr>
<tr>
<td>nextRelId</td>
<td>firstPrevRelld</td>
</tr>
<tr>
<td>nextPropId</td>
<td>secondPrevRelId</td>
</tr>
<tr>
<td>labels</td>
<td>nextPropId</td>
</tr>
<tr>
<td>extra</td>
<td>firstNode</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>secondNode</th>
</tr>
</thead>
<tbody>
<tr>
<td>relationshipType</td>
<td>firstNextRelId</td>
</tr>
<tr>
<td></td>
<td>secondNextRelId</td>
</tr>
<tr>
<td></td>
<td>firstInChainMarker</td>
</tr>
</tbody>
</table>

Source: Robinson/Webber/Eifrem: 2015
Graph Indexing

• Do we really need indexes in graph databases?
  – Graphs are their own indexes!

  – But sometimes we want short-cuts to well-known nodes

  ⇒ Indexes for efficient lookup of specific properties of nodes or relationships

  – AND: Indexes for uniqueness constraints!
    (properties of nodes or relationships)
Graph Indexing (Cont.)

- Index data structures
  - B-Trees
  - Hash Indexes
  - Quadtrees
  - R-Trees
  - ...

- Some graph databases use search engines like Apache Lucene as index backend
  - Supports exact and regex-based matching
  - Supports scoring
    - Number of hits in the index for a given item
    - Great for recommendations!
  - ...

h_da Prof. Dr. Uta Störl
Big Data Technologies: Graph Database Systems - SoSe 2016
Graph Databases

✓ Introduction

✓ Graph Data Models

✓ Graph Storage and Indexing

• Scalability, Availability and Consistency
Example: Neo4J

- Transactions
  - ACID
  - Write-Ahead logging

- Replication
  - Master-Slave Replication (asynchronous!)
  - Neo4j also supports writing through slaves
    - the slave first ensures that it is consistent with the master
    - thereafter, the write is synchronously transacted across both instances.
Graph Database Systems: Pros & Cons

- **Strengths**
  - Fast, for connected data
  - Whiteboard friendly, agile development

- **Weaknesses**
  - Global Queries / Number Crunching
  - Binary Data / Blobs
  - Requires conceptual shift
  - No standardization yet
Graph Database Systems: Outlook

• Standardization?!

• Integration of graph database features and native graph storage in RDBMS?
  – RDF already supported in IBM DB2 and Oracle ...
  – What will be next?

• Polyglot Persistence?!
Big Data Technologies

- Introduction
- NoSQL Database Systems
- Column Store Database Systems
- In-Memory Database Systems
- Graph Database Systems

• Conclusion & Outlook