Introduction to Graph Databases

David Montag
@dmontag
#neo4j
Agenda

- NOSQL overview
- Graph Database 101
- A look at Neo4j
- The red pill
Why you should listen

Forrester says:

“The market for graph databases will boom in 2012 as companies everywhere adopt them for social media analytics, marketing campaign optimization, and customer experience fine-tuning.”

Bloor Research says:

“Graph databases will have a bigger impact on the database landscape than Hadoop or its competitors.”
Why is this happening?
Trends in BigData & NOSQL

1. Increasing data size (big data)
   - “Every 2 days we create as much information as we did up to 2003” - Eric Schmidt

2. Increasingly connected data (graph data)
   - for example, text documents to html

3. Semi-structured data (heterogeneous)
   - individualization of data, schemaless

4. Architecture – SOA
   - from monolithic to modular, distributed applications
Four Categories of NOSQL
Key-Value Category

๏ “Dynamo: Amazon’s Highly Available Key-Value Store” (2007)

๏ Data model:
  ● Global key-value mapping
  ● Big scalable HashMap
  ● Highly fault tolerant (typically)

๏ Examples:
  ● Riak, Redis, Voldemort
Key-Value: Pros & Cons

- **Strengths**
  - Simple data model
  - Great at scaling out horizontally
  - Scalable
  - Available

- **Weaknesses:**
  - Simplistic data model
  - Poor for complex data
Column-Family Category

- Google’s “Bigtable: A Distributed Storage System for Structured Data” (2006)
  - Column-Family are essentially Big Table clones
- Data model:
  - A big table, with column families
  - Map-reduce for querying/processing
- Examples:
  - HBase, HyperTable, Cassandra
Column-Family: Pros & Cons

๏ Strengths
  • Data model supports semi-structured data
  • Naturally indexed (columns)
  • Good at scaling out horizontally

๏ Weaknesses:
  • Complex data model
  • Uns suited for interconnected data
Document Database Category

- Data model
  - Collections of documents
  - A document is a key-value collection
  - Index-centric, lots of map-reduce

- Examples
  - CouchDB, MongoDB
Document Database: Pros & Cons

◉ Strengths

● Simple, powerful data model (just like SVN!)
● Good scaling (especially if sharding supported)

◉ Weaknesses:

● Uns suited for interconnected data
● Query model limited to keys (and indexes)
● Map reduce for larger queries
Graph Database Category

- **Data model:**
  - Nodes & Relationships
  - Hypergraph, sometimes (edges with multiple endpoints)

- **Examples:**
  - Neo4j (of course), OrientDB, InfiniteGraph, AllegroGraph
Graph Database: Pros & Cons

Strengths

- Fast, for connected data
- Easy to query
- Powerful data model, as general as RDBMS

Weaknesses:

- Sharding (though they can scale reasonably well)
  - also, stay tuned for developments here
- Requires conceptual shift
  - though graph-like thinking becomes addictive
Living in a NOSQL World

Dataset complexity

90% of use cases

Dataset size

Graph Databases

Document Databases

Column Family

Key-Value Store

Thursday, May 17, 2012
Graph DB 101
A graph database?

- no: not for storing charts & graphs, or vector artwork
- yes: for storing data that is structured as a graph
  - remember linked lists, trees?
  - graphs are the general-purpose data structure

Graphs are Everywhere

- social graph, related products, the internet, your brain
- any time data is connected to other data
- hard to talk about data, without talking about connections

see http://en.wikipedia.org/wiki/Graph_(abstract_data_type)
We’re talking about a Property Graph

- Nodes
- Relationships
- Properties

+ Indexes
Some well-known named graphs

diamond
butterfly
star
bull
franklin
robertson
horton
hall-janko

see http://en.wikipedia.org/wiki/Gallery_of_named_graphs
Famous graph

The Montag graph
Compared to Key-Value

becomes

Neo4j
Compared to Column-Family

becomes

indexed
Compared to Document
Compared to RDBMS
How fast is it?

- 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></th>
<th># persons</th>
<th>query time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relational database</td>
<td>1,000</td>
<td>2000ms</td>
</tr>
<tr>
<td>Neo4j</td>
<td>1,000</td>
<td>2ms</td>
</tr>
<tr>
<td>Neo4j</td>
<td>1,000,000</td>
<td>2ms</td>
</tr>
</tbody>
</table>
Graph Queries
Cypher

- a pattern-matching query language
- declarative grammar with clauses (like SQL)
- create, update, delete, aggregation, ordering, limits
- tabular results

// get node with id 0
start a=node(0) return a

// traverse from node 1
start a=node(1) match (a)-->(b) return b

// return friends of friends
start a=node(1) match (a)--()--(c) return c
Live Cypher demo
Neo4j is a Graph Database

- A **Graph** Database:
  - a Property Graph with Nodes, Relationships and Properties on both
  - perfect for complex, highly connected data

- A Graph **Database**:
  - reliable with real ACID Transactions
  - scalable: 32 Billion Nodes, 32 Billion Relationships, 64 Billion Properties
  - Server with REST API, or Embeddable on the JVM
  - high-performance with High-Availability (read scaling)
The Red Pill
Q: What are graphs good for?
A: highly connected data

- Recommendations
- Social computing
- MDM
- Configuration management
- Network management
- Genealogy
- ....

Thursday, May 17, 2012
Use cases omitted, please contact david@neotechnology.com if you’re interested
Thanks for listening!

http://neo4j.org

http://neotechnology.com

http://console.neo4j.org