Graph Storage and Query: An Industrial Perspective

Dr. Jim Webber
Chief Scientist, Neo Technology
@jimwebber
Common Perception: Data storage/query is a done deal
- Commercial developers already have the golden hammer of RDBMS
- Lots of academic research still in RDBMS systems
  - Query optimistion, “semantic” stores, etc
- Graph is just a subset of relational, so...
My RDBMS/triple store/etc already does this

There’s even a book about it (sort of)
The diagram illustrates the relationship between Customers, Customers_Accounts, and Accounts. Each layer represents a customer account with associated values. The Customers table contains a row for Alice with an ID of 143. The Customers_Accounts table has records linked to this customer, with values for 143, 326, and 725. The Accounts table further details these values with IDs 326 and 981, and associated amounts of $100 and $212.
Customers  Customers_Accounts  Accounts

Alice

143

326

725

981

143

143

143

326

981

981

$100

$632

$212
Customer

name: Alice

Relationships

Account
balance: $100

Account
balance: $632

Account
balance: $212
What’s Neo4j?

• It’s a graph database
• OLTP-friendly (low latency)
• ACID transactional
• Clusterable
• Friendly query language - Cypher
• Programmatic APIs
  — For arbitrary graph algorithms
A Good Man Goes to War

- appeared in episode
- loves character

- A Good Man Goes to War
- appeared in episode
- species

- species
- appeared in episode
- enemy

- enemy
- stole from planet
- prop

- prop
- character

- character
- appeared in episode

- episode
- appeared in episode
- species
Neo4j models the real world

Semi-structured, messy, inherently connected

http://www.bbc.co.uk/london/travel/downloads/tube_map.html
Neo4j supports humane queries with Cypher.

(Andres Taylor not always humane)
Mechanical sympathy makes Neo4j fast for graph traversals

Millions rels/sec/sec/core on commodity hardware

Neo4j is operationally boring
Which is a good thing!
Neo4j is an ACID transactional database. It values and protects your data.
Neo4j is optimised for graph structures

- Good graph databases exhibit *Index-free Adjacency*
  - Neubauer and Rodriguez, 2013

- We explicitly store:
  - Nodes
  - Relationships
  - Labels
  - Properties

- **Not** tables, **not** keys and values, **not** triples
  - Triples suck.
  - RDF is for the Web, not the database
But Research Challenges Remain

• Query planning and optimisation for Cypher
  – Cypher Algebra exists, pattern matcher constantly improving
• Very large scale mutable graphs
  – Transaction processing (OLTP)
  – Partitioned storage (domain-specific)
  – Performant distributed querying (caches! c.f. Beehive)
• I/O
  – Low-latency, safe storage to durable media
  – Low-contention, low resource use access to graph data
• Runtime Adaptation
  – Ensure fast traversals (gather heuristics from cache misses)
  – Graph-affined caching (shape, volatility, connectivity)
What do we need?

• We’re (mostly) consumers of research right now
• We need research on:
  – Query language design/optimisation
  – Efficient data structures for graph storage/query/localised indexing
  – Highly available (ACID) transaction/coordination protocols
  – Very large scale graph storage/query
  – Data structures
  – Efficient I/O
  – Etc.
Better tools/perspectives

• Relational algebra not very useful
  – We treat graph storage/query more like data structure/algorithm analysis
  – Except in our query language team
• RDF is not gaining much traction commercially
  – Triple model is far less expressive than labelled property graph model
DE MUNDI CREATIONE, RERUM & ornatis: deque hominis formatione, cui...
Neo4j Tutorial
Wednesday @ 15:45

• Neo4j is the leading commercial graph database
  – It’s open source, so you can see how it works
• Hands-on, practical tutorial
  – Bring your laptop
  – Focus on Cypher query language
• Free Teaching materials and free text book you can use
Thanks for listening
@jimwebber

Free Full O’Reilly eBook!
http://graphdatabases.com