Thinking in Graphs

@iansrobinson
ian.robinson@neotechnology.com
Data Complexity

\[ \text{complexity} = f(\text{size}, \text{connectedness}, \text{semi-structure}) \]
Property Graph Data Model

1. John Le Carre
   - name: John Le Carre
   - born: 19-10-1932
   - wrote: 2

2. Tinker, Tailor, Soldier, Spy
   - title: Tinker, Tailor, Soldier, Spy
   - published: 1974
   - recommended
     - date: 09-09-2011
     - recommended
       - date: 05-07-2011
       - recommended
         - name: Alan

3. Graham Greene
   - name: Graham Greene
   - born: 02-10-1904
   - died: 02-04-1991
   - wrote: 3

4. Our Man in Havana
   - title: Our Man in Havana
   - published: 1958
   - recommended
     - date: 03-02-2011
     - name: Ian
Expressive Graph Data Model

- Complex, densely-connected domains
  - Lots of join tables? Relationships
  - Lots of sparse tables? Semi-structure
- Messy data
  - Ad hoc exceptions
- Relationships as first-class elements
  - Semantic clarity: named, directed
  - Not simply constraints
Schema-Free...
Neo4j Architecture

- REST API
- Traversal API
- Cypher
- Core API
- Object Cache
- File Buffer Cache
- Filesystem
Filesystem

Neo4j uses several stores:

- Node
- Relationship
- Property
  - including short strings and small arrays
- String
  - for long strings
- Array
  - for large arrays
Caches

- File Buffer Cache:
  - Memory-mapped NIO
- Object Cache:
  - Nodes
  - Relationships
APIs

• Core API:
  – Nodes
  – Relationships

• Traversal API:
  – Lazily spider the graph

• Cypher
  – Graph pattern matching

• REST
  – JSON over HTTP
Embedded

Application
Server

Application

REST client

Neo4j Server
High Availability
Cypher – A Graph Language
Anonymous Nodes & Rels

( ) -- > ( )
ASCII Art Patterns

A - [:CONNECTED_TO] -> B
ASCII Art Patterns

A --> B <-- C
ASCII Art Patterns

A --> B --> C, A --> C
A --> B --> C <-- A
Variable Length Paths

$A \rightarrow \ast \rightarrow B$

$A \rightarrow \rightarrow \rightarrow B$

$A \rightarrow \rightarrow \rightarrow \rightarrow B$
Create Some Data

CREATE

(cc {name:'Cost Centre 1'}),
(lb {name:'Ledger Book 1'}),
cc-[:HAS_LEDGER_BOOK]->lb
Add Data

START party=node:party(name='Party 1'),
    sourcebook=node:source_book(name='Source Book 1')
CREATE UNIQUE sourcebook-[:HAS_TRADE]->(trade {id:'0001'}),
    trade-[:PARTY_ALIAS]->
        (partyalias {name:'Party Alias 1'}),
    trade-[:CURRENCY]->
        (currency {name:'GBP', _label:'currency'}),
    party-[:ALIAS]->partyalias
RETURN trade
Find Start Nodes in Indexes

```
START
  party=node:party(name='Party 1'),
  sourcebook=node:source_book(name='Source Book 1')
CREATE UNIQUE
  sourcebook-[HAS_TRADE]->(trade {id:'0001'}),
  trade-[PARTY_ALIAS]->
    (partyalias {name:'Party Alias 1'}),
  trade-[CURRENCY]->
    (currency {name:'GBP', _label:'currency'}),
  party-[ALIAS]->partyalias
RETURN trade
```
Describe New Data

START
  party=node:party(name='Party 1'),
  sourcebook=node:source_book(name='Source Book 1')
CREATE UNIQUE
  sourcebook-[:HAS_TRADE]->(trade {id:'0001'}),
  trade-[:PARTY_ALIAS]->
    (partyalias {name:'Party Alias 1'}),
  trade-[:CURRENCY]->
    (currency {name:'GBP', _label:'currency'}),
party-[:ALIAS]->partyalias
RETURN
  trade
START party=node:party(name='Party 1'),
sourcebook=node:source_book(name='Source Book 1')

CREATE UNIQUE sourcebook-[:HAS_TRADE]->(trade {id:'0001'}),
trade-[:PARTY_ALIAS]->
    (partyalias {name:'Party Alias 1'}),
trade-[:CURRENCY]->
    (currency {name:'GBP', _label:'currency'}),
party-[:ALIAS]->partyalias

RETURN trade
Querying the Graph

Graph local:
• One or more start nodes
• Explore surrounding graph
• Millions of joins per second
Pattern Matching

Pattern
Match

Pattern
Match
Non-Match
Non-Match

Pattern
Traversal
Follow Relationships
Evaluate Node
Continue Traversing
Continue Traversing
Continue Traversing

#neo4j
Continue Traversing
All Trades for Cost Centre

START cc=node:cost_centre(name='Cost Centre 1')
MATCH cc-[*3]->trade
RETURN trade;
Is Party Connected?

START cc=node:cost_centre(name='Cost Centre 1'),
    party=node:party(name='Party 1')
MATCH p=shortestPath(cc-[*..5]-party)
RETURN length(p);
Network Impact Analysis
Asset Management & Access Control
Logistics
Social Network & Recommendations
Questions

@iansrobinson
ian.robinson@neotechnology.com
32 billion nodes
32 billion relationships
64 billion properties