Neo4j – what is it?

- Java based, embeddable, data-local
- GPL/AGPL
- ACID, JTA compliant
- Indexing framework
- 24/7 since 2003
- High Availability clustering support
- Great community
- Tinkerpop pipes processing stack
The Neo4j model: Property Graph

Core abstractions:

- Nodes
- Relationships between nodes
- Properties on both

```
name = “Emil”
age = 29
sex = “yes”
```

```
type = KNOWS
time = 4 years
```

```
type = car
vendor = “SAAB”
model = “95 Aero”
```
Building a node space (core API)

GraphDatabaseService graphDb = ... // Get factory

// Create Thomas 'Neo' Anderson
Node mrAnderson = graphDb.createNode();
mrAnderson.setProperty( "name", "Thomas Anderson" );
mrAnderson.setProperty( "age", 29 );

// Create Morpheus
Node morpheus = graphDb.createNode();
morpheus.setProperty( "name", "Morpheus" );
morpheus.setProperty( "rank", "Captain" );
morpheus.setProperty( "occupation", "Total bad ass" );

// Create a relationship representing that they know each other
mrAnderson.createRelationshipTo( morpheus, RelTypes.KNOWS );
// ...create Trinity, Cypher, Agent Smith, Architect similarly
Building a node space

GraphDatabaseService graphDb = ... // Get factory
Transaction tx = graphDb.beginTx();

// Create Thomas 'Neo' Anderson
Node mrAnderson = graphDb.createNode();
mrAnderson.setProperty( "name", "Thomas Anderson" );
mrAnderson.setProperty( "age", 29 );

// Create Morpheus
Node morpheus = graphDb.createNode();
morpheus.setProperty( "name", "Morpheus" );
morpheus.setProperty( "rank", "Captain" );
morpheus.setProperty( "occupation", "Total bad ass" );

// Create a relationship representing that they know each other
mrAnderson.createRelationshipTo( morpheus, RelTypes.KNOWS );
// ...create Trinity, Cypher, Agent Smith, Architect similarly
tx.commit();
Code (2): Traversing a node space

// Instantiate a traverser that returns Mr Anderson's friends
Traverser friendsTraverser = mrAnderson.traverse(
    Traverser.Order.BREADTH_FIRST,
    StopEvaluator.END_OF_GRAPH,
    ReturnableEvaluator.ALL_BUT_START_NODE,
    RelTypes.KNOWS,
    Direction.OUTGOING );

// Traverse the node space and print out the result
System.out.println( "Mr Anderson's friends:" );
for ( Node friend : friendsTraverser )
{
    System.out.printf( "At depth %d => %s%n",
        friendsTraverser.currentPosition().getDepth(),
        friend.getProperty( "name" ) );
}
Ruby

gem install neo4j

require "rubygems"
require 'neo4j'

class Person
  include Neo4j::NodeMixin
  property :name, :age, :occupation
  index :name
  has_n :friends
end

Neo4j::Transaction.run do
  neo = Person.new :name=>'Neo', :age=>29
  morpheus = Person.new :name=>'Morpheus', :occupation=>'badass'
  neo.friends << morpheus
end

neo.friends.each { |p| ...}
Neo4j 1.3 news

- GPL Community Edition
- 128 Billion primitives address space
- More graph algos
- Short string → long
- Web visualization
- Gremlin 0.9 built-in
- HA improvements
Web admin
Neo4j High Availability

Cluster Manager

Load Balancer/REST API

App Server
HA GraphDB
Neo4jSlave 1
Machine 1

App Server
HA GraphDB
Neo4jSlave 2
Machine 3

App Server
HA GraphDB
Neo4jSlave 2
Machine 4

App Server
HA GraphDB
Neo4jSlave X
Machine X

App Server
HA GraphDB
Neo4jMaster
Machine 2

ZooKeeper Service

ZooKeeper Server 1

ZooKeeper Server 2

ZooKeeper Server 3

Neo4j Highly Available Cluster
Neo4j – what do do people with it?

- Graphs over 1M nodes
- Network Management
- Master Data Management
- Social
- Finance
- Spatial
- Other
  - Bioinformatics
  - RDF
  - Routing, Logistics
  - Product config etc
Call Data Records (CDR)

- Forming a (social) graph
- Location based
- Possible uses:
  - Find clusters (better plans)
  - Build social connections
  - Find influencers
Financial data – fraud detection

1. name = “Mr Godfather”
   karma = veeeery-low
   cash = more-than-you

2. amount = $1000

3. amount = $1000

4. name = “Emil”
   cash = always-too-li’l

5. name = “The Tavern”
   lat = 1295238237
   long = 234823492

6. title = “ATM @ Wall St”
   id = 230918484233
   cash_left = 384204
Routing
Social graphs

- Recommendations
- Location based services
- Influencers
- Shortest path
Recommendations and big graphs

- Global heuristics
  - Page rank
- Local recommendations
  - Shortest paths
  - Hammock functions
  - Random walks
  - Dijkstra, A*, Shooting star etc
Impact Analytics, CMDB, Network Management, Provisioning

[Diagram showing relationships between power generators, servers, instances, and applications]
Impact Analytics, CMDB, Network Management, Provisioning
Master Data Management
Multiple indexes - GIS
Neo4j dynamic layers

Connected domain data  Neo4j Spatial  GIS and Spatial stacks

- Geometry Encoder
- Dynamic Query
- Dynamic Styles
- Dynamic Meta-Inf

Layer1  Layer2  Layer3
Network Topology analysis

- Analytics of network coverage and frequencies
  - Cell towers
  - Drive data
  - Infrastructure
- Analytics
  - Spatial signal strength
  - Antenna placement and azimuth
  - Frequency planning
  - Network differences over time
  - Reporting and charting
Cell network analysis
Cell network analysis
Simulations (Energy market)

- agents, markets, power plants, bids, substances, technologies

nodes: 171'130
edges: 650'691
ticks: 10
Relevant research areas

- Traversals/Querying
  - Query optimization (Sameh)
  - Parallel (BSP) and Local (Neo4j)
  - Crossing shards during traversal (context etc)
  - Multithreaded
  - Subgraph matching (Mattias)

- Graph sharding
  - Upfront sharding utils
  - Runtime sharding algos
  - Partial sharding/replication

- Parallel garbage collection under high load
- Big graph algos and heuristics
Questions?

Image credit: lost again! Sorry :(  