Cypher for Apache Spark

Graph processing workloads on OLAP and OLTP
Mats Rydberg
mats@neotechnology.com
Cypher for Apache Spark

- Apache Spark: computational platform (OLAP)
- Neo4j: transactional graph database (OLTP)
  - Query language: Cypher

Wouldn't it be lovely to be able to execute a Spark job on a Neo4j graph? How do we integrate? What is a graph when it isn't in Neo4j anymore?

==> *Cypher is the bridge!*
Schematic dataflow
Example use case

- Graph of financial transactions
- Snapshot subgraph of transactions made during last month
- Do computationally heavy graph analytics on transaction patterns
  - Consume results as report (for humans)
  - Feed back results as new data to original graph
  - Deploy results as new graph
- Neo4j still operational for incoming transactions due to analytics off-loaded to Spark
- Fully integrated OLTP + OLAP
Apache Spark -- overview / characteristics

- DataFrames are abstractions of tables
  - Based of RDD (Resilient Distributed Dataset)
  - SQL type system deployed in a non-type safe way (Scala code)
- SQL and API that compiles to lazily executed plans
  - Catalyst plan optimiser
- Distributed architecture for scalability
Key developments

- Extend Cypher with the ability to *return graphs*
  - Cypher becomes *closed over graphs*
  - True compositionality of queries
- Modelling dynamic Cypher type system on strict table-based, SQL-aligned Spark DataFrames
  - Using DataFrames to make use of Catalyst optimiser
  - No support for type inheritance (compare Cypher's ANY type)
Key developments -- type system

- Represent entities as flat maps
  - One column per property and label / rel type
  - Requires exact type information of all properties
    - Acquired during import of graph
    - Read-only setting allows immutable schema
Key developments -- return graphs

- Interpret query results as a *graph* rather than table
  - Round-trip: graph to graph; can execute another query
  - No focus on syntax
- Pipeline of queries lazily evaluated on top of one another
  - Maximum utilisation of Catalyst to reorder operations
- Complementary API for injecting other operations in-between queries
  - Based on Spark DataFrame API
Demo of prototype