Multiple Graph Querying

Alastair Green
alastair.green@neotechnology.com
Multiple graph processing

Some important use-cases for multiple-graph processing

- Neo4j-Spark-HDFS **Snapshots** for analytics
- **Views** (graph projections) over Neo4j for access control, aggregation
- **Logical Databases** for owner or geographical segregation

All imply that a graph has an **Identity**, and a **Location** or Context
T → T ... G → T ... G → G’

The world of data management is often tabular in mindset or appearance. Stores, visualization, reports, spreadsheets, **Cypher result sets** ...
Queries as functions

A query viewed as a function over a graph (tuples of graphs)

\[ G \to G' \]

A graph query language that supports that perspective is composable

```java
Graph outputGraph = lastQuery
    (graphReturningAlgorithm
    (firstQuery
    (new Graph(inputGraphIdentity))));
```

An example is a **user** query \( u \) operating over a **view** query \( v \)

\[ G' = (u \circ v) (G) \]
Cypher query operating as a graph function
Number and names

Let a database contain multiple graphs
- Snapshot a graph, process it and store the results
- Merge a delta graph into a history graph
- Name a view in a database or a session
- Name a temporary graph in a query

Multiplicity raises the question of identity, within multiple contexts
Identity within queries, within stores, across engines

Graph references

*In space*
- Containers on the network
- Containers within containers
- **Graphs** in spaces or containers

*In time*
- **Graphs** in queries, sessions, in persistence stores
## A graph:// URI scheme

<table>
<thead>
<tr>
<th>Type</th>
<th>URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opaque</td>
<td>graph:name:ec654ec1-7f8f-11e3-ae96-b385f4bc450c</td>
</tr>
<tr>
<td>RDF</td>
<td>graph:rdf+http:'http://chucknorris.com-data_-chuck~foaf_based_near'</td>
</tr>
<tr>
<td>Bolt anonymous graph</td>
<td>graph:bolt:'bolt+routing://production'</td>
</tr>
<tr>
<td>Bolt named graph</td>
<td>graph:bolt:(bolt+routing://production/west-coast/orders/snap/2016-04-12)</td>
</tr>
<tr>
<td>Graph stored in HDFS using Parquet storage with conventionally named files for edges and vertices</td>
<td>graph:gve+hdfs+parquet:(hdfs://production/west-coast/orders/snap/2016-04-12)</td>
</tr>
</tbody>
</table>

**Draft graph URI Scheme**

Copyright © 2016 Neo Technology Inc.

- [Identifying and addressing graphs](#)
- [Use of ABNF to define the syntax of a graph URI](#)
- [URI](#)
- [URI Scheme](#)
- [Path Variants](#)
Composition and Identity

Two important aspects that sketch an introduction to what follows

More detailed examinations of multiple graph data models, processing and views in the remainder of this session

In the meantime …
Multiple Graph Querying

Questions ?