**Motivation**

A graph search system that finds relevant entities for given query entities in a big graph efficiently. The returned entities are scored by a similarity metric function.

**System Design Requirements**
- High query performance;
- No knowledge of graph structure and content when issuing queries;
- Visualization panel exhibits results and allows exploration.

**Contribution**

GQFast discovers relevant entities efficiently and uses small space. It also provides the following features:
- Context-aware query completion feature instantly gives a list of suggested queries based on the current context.
- Type-ahead-search feature instantly visualizes search results during the query generation period to allow users interaction.

**System Architecture**

- **Query Generator**: Communicate with the Autocompletion Manager to help users create meaningful queries.
- **Visual Space Explorer**: Visualize results and allow interactions with users.
- **Autocompletion Manager**: Return suggested queries based on the current context and the existing partial query.
- **GQFast Query Processor**: Look up corresponding indices to retrieve relevant entities for queries.
- **Index Builder**: Read graph data (stored in relation tables) and create in-memory index.

**Data Model & Index Structure**

- GQFast deals with typed graphs where each node has a type. In addition,
  - Each node refers to an entity with several attributes.
  - Each edge refers to a relationship with several attributes.

**Experiment Results**

**Experimental setup**

Dataset: Biomedical literature graph (PubMed), which contains about 30 million nodes and 1 billion edges.

Machine: 4th generation Intel i7-4770 processor (8M Cache, 8 cores, 3.6 GHz) running Ubuntu 14.04.1 with 16GB RAM.

**(a) Running time**

<table>
<thead>
<tr>
<th></th>
<th>Neo4j</th>
<th>Postgres</th>
<th>MonetDB</th>
<th>GQ-Fast</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running</td>
<td>5546.5</td>
<td>29590.5</td>
<td>4474.8</td>
<td>3.69</td>
</tr>
</tbody>
</table>

**(b) Space cost**

<table>
<thead>
<tr>
<th></th>
<th>Neo4j</th>
<th>Postgres</th>
<th>MonetDB</th>
<th>GQ-Fast</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space</td>
<td>34.36</td>
<td>20.92</td>
<td>3.69</td>
<td>1.47</td>
</tr>
</tbody>
</table>

Chunbin Lin, Jianguo Wang, Yannis Papakonstantinou: Fast In-Memory SQL Analytics on Typed Graphs. PVLDB 10(3): 265-276 (2016)