Implementing And Analysing An Approximate Data Extraction Algorithm From Graph Databases

Vakkalagadda Gautame | gv5412@g.rit.edu
Advisor: Prof. Carlos Rivero

INTRODUCTION

Graph databases are one of the leading drivers in these currently emerging heterogeneous database management systems. It gives lot of flexibility and can serve real time scenarios efficiently.

ALGORITHM

Steps:

- Evaluate important Nodes of the query graph.
- Probe the index structure to find all the matches.
- Grow the Matches by examining the neighbour’s of the graph in the Database.
- Output all the resulted approximate matches in the graph database.

INDEX STRUCTURE

DESIGN

- Main
- Data Parser
- Graph Generator
- Index
- Probe Algorithm
- Grow Algorithm

Input

Output

Map Dataset:

Yeast Dataset:

Total Size of the data Graph: 23429
Time Taken for building B+ Tree index: 55004ms
Final number of node matches : 255

 Yeast Data

<table>
<thead>
<tr>
<th>Data Nodes</th>
<th>Data Edges</th>
<th>Query Size</th>
<th>Probing Time(ms)</th>
<th>Growth Time(ms)</th>
<th>Final Node matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1076</td>
<td>10</td>
<td>11001980</td>
<td>568</td>
<td></td>
</tr>
</tbody>
</table>

| Map Data
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1</td>
<td>43</td>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PERFORMANCE MEASURE

CONCLUSION

- The algorithm works well with sparse graphs compare to dense graphs.
- The algorithm has good pruning power because of the index structure and the probe algorithm.
- It does not guarantee that the final resultant graph is the best match to the query graph.