Graph storage: How good is CSR really?

Mahammad Valiyev
Technische Universität München
Informatics
12. December 2017
Agenda

- Introduction
- Implementation
- Evaluation
Graph

- Vertices
- Edges:
  - Directed/Undirected
  - Weighted/Unweighted
Representations of graph

- Adjacency Matrix
- Adjacency List
- Compressed Sparse Row
Adjacency Matrix

\[
\begin{pmatrix}
0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 \\
0 & 1 & 1 & 0
\end{pmatrix}
\]
Adjacency List using std::list

- head of 1
- head of 2
- head of 3
- head of 4

Diagram:

- 1
- 2
- 3
- 4
Adjacency List using \texttt{std::vector}

\begin{align*}
\text{edges[1]} & : & 2 & 3 & 4 \\
\text{edges[2]} & : & 4 \\
\text{edges[3]} & : & 1 & 4 \\
\text{edges[4]} & : & 2 & 3
\end{align*}

\[1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 1\]
Compressed Sparse Row

edges

offsets

1 2 3 4 5 6 7
2 3 4 4 1 4 2 3
0 3 4 6 8

1 2 3 4

2 2 2 2
Implementation

- **Graph Containers:**
  - **Compressed Sparse Row:**
    - Simple Update
    - Light Update
  - **Adjacency List:**
    - Implemented with `std::list`
    - Implemented with `std::vector`

- **Algorithms:**
  - Depth-First Search
  - Breadth-First Search
  - Dijkstra Algorithm
Evaluation: Platform

- OS: Ubuntu
- Processor: Intel(R) Core(TM) i7-3930K
- Frequency: 3.20GHz
- Memory: 64 Gb

- Dataset:
  - # vertices: 1,000,000
  - # neighbors per node differs from 50 to 200
  - Neighbors are selected randomly
  - Directed and weighted
Evaluation

DFS

BFS
Evaluation

Dijkstra

Memory Consumption ratio

<table>
<thead>
<tr>
<th># neighbors per node</th>
<th>mem(std::list) ÷ mem(CSR)</th>
<th>mem(std::vector) ÷ mem(CSR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>7.33</td>
<td>1.41</td>
</tr>
<tr>
<td>65</td>
<td>7.46</td>
<td>2.01</td>
</tr>
<tr>
<td>80</td>
<td>7.55</td>
<td>1.66</td>
</tr>
<tr>
<td>95</td>
<td>7.62</td>
<td>1.42</td>
</tr>
<tr>
<td>110</td>
<td>7.67</td>
<td>1.23</td>
</tr>
<tr>
<td>125</td>
<td>7.7</td>
<td>1.15</td>
</tr>
<tr>
<td>140</td>
<td>7.73</td>
<td>1.15</td>
</tr>
<tr>
<td>155</td>
<td>7.76</td>
<td>1.15</td>
</tr>
<tr>
<td>170</td>
<td>7.78</td>
<td>1.54</td>
</tr>
<tr>
<td>185</td>
<td>7.79</td>
<td>1.42</td>
</tr>
<tr>
<td>200</td>
<td>7.81</td>
<td>1.31</td>
</tr>
</tbody>
</table>
Evaluation

Update time

Real World Example
Evaluation

- Dataset:
  - # nodes: 1,000,000
  - # neighbors per node differs between 50 and 200
  - Neighbors have closer ids

![Graphs showing performance comparison between DFS and BFS]

- DFS
- BFS
Thanks for your attention