MAX CLIQUE PROBLEM

PARAROARS

Yogesh Jagadeesan
Dler Ahmad
MAXIMUM CLIQUE

- Largest sub-graph in the graph that is complete. i.e. all its vertices are connected to each other.
Given an undirected graph $G=(V,E)$

- $V=\{1,2,...,n\}$ is the set of vertices in $G$
- $n=|V|$ is the size of the set $V$
- $E \subseteq V \times V$ is the set of edges in $G$.

A clique $C$ is a subset of $V$, such that graph $G(C)$ is complete, i.e. every $i,j \in V$ such that $i \neq j$, $(i,j) \in E$.

The cardinality of $C$ is the number of vertices in the $C$. The maximum clique problem is the problem of finding clique $C$ with maximum cardinality in graph $G$. 
**SEQUENTIAL PROGRAM**

- Runs on a single core of a single node
- N number of attempts are performed sequentially to find a clique in the graph
- In each attempt:
  - A random initial vertex is picked as CC (current clique)
  - Any neighbor vertex that is connected to all vertices in CC is candidate
  - A candidate with highest degree will be added to CC
  - Procedure is repeated for newly added vertex.
  - CC is returned when no more vertex can be added
- Final result is the largest clique found by all attempts
MULTICORE PARALLEL PROGRAM

- Runs on all cores of a single node
- Attempts are distributed among the cores using `parallelFor` loop
- Each attempt is performed on a single core
- Final result is largest clique found at all cores
CLUSTER PARALLEL PROGRAM

- Runs on all cores of multi nodes in a cluster
- Attempts are distributed among cores using Master/Worker paradigm
- Each attempt is performed in a single core of a single node in the cluster
- Final result is largest clique found in all cores of all nodes in the cluster
TEST GRAPHS

Sequential

For multicore and cluster version random graph of size 100 is used.
STRONG SCALING

<table>
<thead>
<tr>
<th>N</th>
<th>K</th>
<th>T</th>
<th>Speedup</th>
<th>Eff</th>
</tr>
</thead>
<tbody>
<tr>
<td>1500000</td>
<td>Seq</td>
<td>60861</td>
<td>1.054</td>
<td>1.054</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>57734</td>
<td>1.054</td>
<td>1.054</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>30325</td>
<td>2.001</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>20155</td>
<td>3.02</td>
<td>1.006</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>16128</td>
<td>3.774</td>
<td>0.943</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>13075</td>
<td>4.655</td>
<td>0.931</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>11214</td>
<td>5.427</td>
<td>0.904</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>9615</td>
<td>6.33</td>
<td>0.904</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>8415</td>
<td>7.232</td>
<td>0.904</td>
</tr>
<tr>
<td>4000000</td>
<td>Seq</td>
<td>59097</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>50805</td>
<td>1.163</td>
<td>1.163</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>26669</td>
<td>2.216</td>
<td>1.108</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>18405</td>
<td>3.211</td>
<td>1.07</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>14239</td>
<td>4.15</td>
<td>1.04</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>11294</td>
<td>5.23</td>
<td>1.05</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>9816</td>
<td>6.02</td>
<td>1.003</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>8476</td>
<td>6.972</td>
<td>0.996</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>7724</td>
<td>7.65</td>
<td>0.956</td>
</tr>
</tbody>
</table>
### STRONG SCALING

<table>
<thead>
<tr>
<th>N</th>
<th>K</th>
<th>T</th>
<th>Speedup</th>
<th>Eff</th>
</tr>
</thead>
<tbody>
<tr>
<td>5000000</td>
<td>Seq</td>
<td>65569</td>
<td>1.001</td>
<td>1.001</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>65502</td>
<td>1.976</td>
<td>0.988</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>33185</td>
<td>2.937</td>
<td>0.979</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>22326</td>
<td>3.646</td>
<td>0.912</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>17981</td>
<td>4.599</td>
<td>0.92</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>14256</td>
<td>5.403</td>
<td>0.9</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>12135</td>
<td>6.328</td>
<td>0.904</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>10361</td>
<td>7.216</td>
<td>0.902</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>9086</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6000000</td>
<td>Seq</td>
<td>62745</td>
<td>1.057</td>
<td>1.057</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>59329</td>
<td>1.057</td>
<td>1.019</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>30789</td>
<td>2.038</td>
<td>1.019</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>20911</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>16118</td>
<td>3.893</td>
<td>0.973</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>16328</td>
<td>4.743</td>
<td>0.949</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>11300</td>
<td>5.553</td>
<td>0.925</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>9549</td>
<td>6.571</td>
<td>0.939</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>8583</td>
<td>7.31</td>
<td>0.914</td>
</tr>
</tbody>
</table>
## Strong Scaling

<table>
<thead>
<tr>
<th>N</th>
<th>K</th>
<th>T</th>
<th>Speedup</th>
<th>Eff</th>
</tr>
</thead>
<tbody>
<tr>
<td>40000000</td>
<td>Seq</td>
<td>62601</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>56670</td>
<td>1.104</td>
<td>1.104</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>27256</td>
<td>2.296</td>
<td>1.148</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>19341</td>
<td>3.237</td>
<td>1.079</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>14812</td>
<td>4.226</td>
<td>1.056</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>11958</td>
<td>5.235</td>
<td>1.047</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>9320</td>
<td>6.717</td>
<td>1.195</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>9019</td>
<td>6.941</td>
<td>0.991</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>7882</td>
<td>7.942</td>
<td>0.993</td>
</tr>
</tbody>
</table>
STRONG SCALING

Running Time vs. Cores

Efficiency vs. Cores
## Weak Scaling

<table>
<thead>
<tr>
<th>N</th>
<th>K</th>
<th>T</th>
<th>Speedup</th>
<th>Eff</th>
</tr>
</thead>
<tbody>
<tr>
<td>1500000</td>
<td>Seq</td>
<td>60100</td>
<td>1.041</td>
<td>1.041</td>
</tr>
<tr>
<td>1500000</td>
<td>1</td>
<td>57734</td>
<td>1.041</td>
<td></td>
</tr>
<tr>
<td>3000000</td>
<td>2</td>
<td>60640</td>
<td>1.982</td>
<td>0.991</td>
</tr>
<tr>
<td>4500000</td>
<td>3</td>
<td>59119</td>
<td>3.05</td>
<td>1.016</td>
</tr>
<tr>
<td>6000000</td>
<td>4</td>
<td>62173</td>
<td>3.867</td>
<td>0.967</td>
</tr>
<tr>
<td>7500000</td>
<td>5</td>
<td>62617</td>
<td>4.8</td>
<td>0.96</td>
</tr>
<tr>
<td>9000000</td>
<td>6</td>
<td>66122</td>
<td>5.453</td>
<td>0.909</td>
</tr>
<tr>
<td>10500000</td>
<td>6</td>
<td>66404</td>
<td>6.335</td>
<td>0.905</td>
</tr>
<tr>
<td>12000000</td>
<td>7</td>
<td>6695</td>
<td>7.209</td>
<td>0.901</td>
</tr>
<tr>
<td>4000000</td>
<td>Seq</td>
<td>59097</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4000000</td>
<td>1</td>
<td>50805</td>
<td>1.163</td>
<td>1.163</td>
</tr>
<tr>
<td>8000000</td>
<td>2</td>
<td>52710</td>
<td>2.242</td>
<td>1.121</td>
</tr>
<tr>
<td>12000000</td>
<td>3</td>
<td>53236</td>
<td>3.33</td>
<td>1.11</td>
</tr>
<tr>
<td>16000000</td>
<td>4</td>
<td>56972</td>
<td>4.149</td>
<td>1.037</td>
</tr>
<tr>
<td>20000000</td>
<td>5</td>
<td>56225</td>
<td>5.255</td>
<td>1.051</td>
</tr>
<tr>
<td>24000000</td>
<td>6</td>
<td>59270</td>
<td>5.982</td>
<td>0.997</td>
</tr>
<tr>
<td>28000000</td>
<td>7</td>
<td>59172</td>
<td>6.99</td>
<td>0.998</td>
</tr>
<tr>
<td>32000000</td>
<td>8</td>
<td>63339</td>
<td>7.464</td>
<td>0.933</td>
</tr>
</tbody>
</table>
### WEAK SCALING

<table>
<thead>
<tr>
<th>N</th>
<th>K</th>
<th>T</th>
<th>Speedup</th>
<th>Eff</th>
</tr>
</thead>
<tbody>
<tr>
<td>5000000</td>
<td>Seq</td>
<td>65569</td>
<td>1.001</td>
<td>1.001</td>
</tr>
<tr>
<td>5000000</td>
<td>1</td>
<td>65502</td>
<td>1.978</td>
<td>0.989</td>
</tr>
<tr>
<td>10000000</td>
<td>2</td>
<td>66280</td>
<td>2.946</td>
<td>0.982</td>
</tr>
<tr>
<td>15000000</td>
<td>3</td>
<td>66780</td>
<td>3.796</td>
<td>0.95</td>
</tr>
<tr>
<td>20000000</td>
<td>4</td>
<td>69091</td>
<td>4.637</td>
<td>0.927</td>
</tr>
<tr>
<td>25000000</td>
<td>5</td>
<td>70706</td>
<td>5.427</td>
<td>0.904</td>
</tr>
<tr>
<td>30000000</td>
<td>6</td>
<td>72523</td>
<td>6.392</td>
<td>0.913</td>
</tr>
<tr>
<td>35000000</td>
<td>7</td>
<td>71807</td>
<td>7.327</td>
<td>0.916</td>
</tr>
<tr>
<td>40000000</td>
<td>8</td>
<td>71592</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60000000</td>
<td>Seq</td>
<td>62745</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6000000</td>
<td>1</td>
<td>59329</td>
<td>1.057</td>
<td>1.057</td>
</tr>
<tr>
<td>12000000</td>
<td>2</td>
<td>61242</td>
<td>2.05</td>
<td>1.024</td>
</tr>
<tr>
<td>18000000</td>
<td>3</td>
<td>61412</td>
<td>3.065</td>
<td>1.022</td>
</tr>
<tr>
<td>24000000</td>
<td>4</td>
<td>66892</td>
<td>3.752</td>
<td>0.94</td>
</tr>
<tr>
<td>30000000</td>
<td>5</td>
<td>66694</td>
<td>4.704</td>
<td>0.941</td>
</tr>
<tr>
<td>36000000</td>
<td>6</td>
<td>68362</td>
<td>5.507</td>
<td>0.918</td>
</tr>
<tr>
<td>42000000</td>
<td>7</td>
<td>68799</td>
<td>6.384</td>
<td>0.912</td>
</tr>
<tr>
<td>48000000</td>
<td>8</td>
<td>69224</td>
<td>7.251</td>
<td>0.906</td>
</tr>
</tbody>
</table>
## Weak Scaling

<table>
<thead>
<tr>
<th>N</th>
<th>K</th>
<th>T</th>
<th>Speedup</th>
<th>Eff</th>
</tr>
</thead>
<tbody>
<tr>
<td>400000000</td>
<td>Seq</td>
<td>62601</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40000000</td>
<td>1</td>
<td>56670</td>
<td>1.104</td>
<td>1.104</td>
</tr>
<tr>
<td>80000000</td>
<td>2</td>
<td>54125</td>
<td>2.312</td>
<td>1.157</td>
</tr>
<tr>
<td>120000000</td>
<td>3</td>
<td>54551</td>
<td>3.443</td>
<td>1.147</td>
</tr>
<tr>
<td>160000000</td>
<td>4</td>
<td>57653</td>
<td>4.343</td>
<td>1.086</td>
</tr>
<tr>
<td>200000000</td>
<td>5</td>
<td>54011</td>
<td>5.795</td>
<td>1.159</td>
</tr>
<tr>
<td>240000000</td>
<td>6</td>
<td>55377</td>
<td>6.783</td>
<td>1.13</td>
</tr>
<tr>
<td>280000000</td>
<td>7</td>
<td>62446</td>
<td>7.012</td>
<td>1.002</td>
</tr>
<tr>
<td>320000000</td>
<td>8</td>
<td>62501</td>
<td>8.013</td>
<td>1.001</td>
</tr>
</tbody>
</table>
WEAK SCALING

Running Time vs. Cores

Efficiency vs. Cores
OUTCOMES

- Less than optimal solution possible for maximum clique problem with heuristic search
- Large number of well-diversified attempts increases the chance of finding a solution close to optimal
- Not all solutions can be easily parallelized
- As long as there are no sequential dependencies in the solution, good speedups with parallelization is possible.
FUTURE WORK

- Applying realistic graph problems
- Experiment GPU accelerated version
- Consider more constrains in selecting candidate vertices
THANK YOU