Maximum Clique: Team Presentation 3

Team Perplexingly Parallel
Team Members

Akshay Sharma  
MS Computer Science (Second year)

Srinath Obla  
MS Computer Science (Second year)

Vishal Kole  
MS Computer Science (Second year)
Problem Space: Cliqu es

Clique - A subset of vertices present in an undirected graph such that each vertex is directly connected to all the other vertices in the set, thereby making it a complete graph.
Maximal Cliques

- Maximal Clique - A clique which cannot be extended by including another adjacent vertex of the graph in the set. Some authors define cliques to be maximal by definition.

- Maximal cliques are not to be confused with maximum cliques!
The Maximum Clique problem

- A clique in a graph such that there is no clique with more vertices.
- Also called the clique number of a graph.
- A maximum clique is always maximal, the converse is not always true.
- NP-Complete Problem
Bron–Kerbosch algorithm

• David Eppstein, Maarten Loffler, and Darren Strash
  “Listing All Maximal Cliques in Sparse Graphs in Near-optimal Time”
BronKerbosch1(R, P, X):
    if P and X are both empty:
        report R as a maximal clique
    for each vertex v in P:
        BronKerbosch1(R \cup \{v\}, P \cap \text{Nbr}(v), X \cap \text{Nbr}(v))
        P := P \setminus \{v\}
        X := X \cup \{v\}

R: possibly a clique.

P: Holds vertices adjacent to every vertex currently in R, when added to R makes it maximal.

X: contains nodes already in some clique or processed (removes the duplicates)
Example

R: {}
P: \{1,2,3,4\}
X: {}
\{\},\{1,2,3,4\},\{}

\{1\},\{2,3\},\{}

\{1,2\},\{3\},\{}

\{1,2,3\},\{\},\{} - Maximal Clique - 1,2,3

\{1,3\},\{\},\{2\}

\{2\},\{3,4\},\{1\}

\{2,3\},\{\},\{1\}

\{2,4\},\{\},\{} - Maximal Clique - 2,4

\{3\},\{\},\{1,2\}

\{4\},\{\},\{2\}
Sequential Implementation

- Uses the Bron-Kerbosch algorithm to iteratively create the configurations and compute the maximal cliques.

\[
\begin{align*}
\emptyset, \{1,2,3,4\}, \emptyset \\
\{1\}, \{2,3\}, \emptyset \\
\{1,2\}, \{3\}, \emptyset \\
\{1,2,3\}, \emptyset, \emptyset &- \text{Maximal Clique - 1,2,3} \\
\{1,3\}, \emptyset, \{2\} \\
\{2\}, \{3,4\}, \{1\} \\
\{2,3\}, \emptyset, \{1\} \\
\{2,4\}, \emptyset, \emptyset &- \text{Maximal Clique - 2,4} \\
\{3\}, \emptyset, \{1,2\} \\
\{4\}, \emptyset, \{2\}
\end{align*}
\]
Parallel Implementation

- Uses the Bron-Kerbosch algorithm

- Distributing the first level of recursive calls amongst threads

- Steps include
  - Start with the root configuration and explore its children.
  - Store the children and give iterative bounds for each thread.
  - Each thread takes a child configuration and explores it completely.
  - For $m$ configurations and $n$ threads, each thread explores $m/n$ configurations.
<table>
<thead>
<tr>
<th>Thread 0</th>
<th>Thread 1</th>
<th>Thread 2</th>
<th>Thread 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1,2,3,4},{}</td>
<td>{2,3},{}</td>
<td>{1,3},{},{2}</td>
<td>{4},{},{2}</td>
</tr>
<tr>
<td>{1},{2,3},{}</td>
<td>{1,2},{3},{}</td>
<td>{1,2,3},{},{} - Maximal Clique - 1,2,3</td>
<td>{2},{3,4},{1}</td>
</tr>
<tr>
<td>{1,2},{3},{}</td>
<td>{2,3},{},{1}</td>
<td>{2,4},{},{} - Maximal Clique - 2,4</td>
<td></td>
</tr>
<tr>
<td>{3},{},{1,2}</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>{4},{},{2}</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Parallel Implementation - Multicore

- Similar configuration based approach
- Dividing recursive calls across cores
- Graph can be easily stored in shared memory for all threads
Parallel Implementation - Multicore

- Vertex index based approach
- Dividing recursive calls across cores
- Graph can be easily stored in shared memory for all threads
Parallel Implementation - Cluster

- Generates the Bron-Kerbosch configurations as tuples in a sequential manner.
- The configurations are then consumed in parallel by the workers.
Parallel Implementation - Cluster

- Generates the Bron-Kerbosch configurations as tuples in a sequential manner.
- The configurations are created by the workers during execution.
Demo Time!
Future work

● Optimizations to the Bron-Kerbosch algorithm.
  ○ Vertex ordering
  ○ Pivot based approach
  ○ $O(dn^{3d/3})$

● Handling special cases
  ○ Ensuring use of all cores; cases when there are more cores than initial configurations
Thank You!

Any questions?