Logo

Description automatically generated

# ANALYSIS OF COMPUTER

**ALGORITHMS**

**PROJECT PROGRESS REPORT**

**Nth level Search of connections for a User in a Social Networking**

**Using Breadth First Search and Neo4j**

**BY:**

AKSHAY RAJ SINGH ANGUWALA.

SRI VAESHNAVI NAGULA.

Project Summary:

To find the nth level connections for a user in social network by using the Breadth first Search and the neo4j Graph Database.

Here we are using the Graph Database to implement the large amount of Data.

Understanding the Nth level Finding :

Let here be a graph with edges and vertices connected to one another.

Here we could see that the graph has connections to each other at different levels hence here we would want to know the 2nd level friends for the Node 1.

Chart

Description automatically generated

Then the second level friends for the node 1 could be the 1st level friends of the direct connected friends of Node 1 .

A picture containing chart

Description automatically generated

Here as we could observe the second level friends would be 4,6 for node 1.

Accomplishments and Milestones Achieved:

We were able to work on the concept on searching nth level friends using a Breadth first search.

Able to onboard the Neo4j a Graph Database into the Research to handle the large data sets.

Were able to know the working of the nodes and edges in a network by implementing the BFS algorithm.

We were able to create a network of nodes and edges in the neo4j using the Cypher query language also adding to it we were able to map the corresponding relation between the two nodes as well such that we could further work on searching the nth level connections for a User.

Here we are using the neo4j because it is one of the best Graph Database and it could also be used for making a connection and its Cypher query language is simple as My Sql which inherits the queries given and responds with the output.

Here Neo4j is also compatible with the modern programming languages as well, hence we have opted to make our hands dirty by using the Neo4j Application which is also used by many product-based organizations for their projects.

Challenges:

To get the basic picture on how the project should work and what resources are to be used to achieve the required Solution.

How and what type of graph database to be used to deal with large amount of data sets as social networking could have a lot of data being used.

Understanding the Neo4j and trying to implement the network of nodes and edges into the Database.

Building a relationship between the nodes and to see their visualized representation and get the output in an expected format.

Understanding and implementing the Cypher query language to:

Create the data.

Make the relationships to that Data (relation between nodes)

To get the finalized output for the created dataset of Social Network.

Solutions for the Challenges:

Worked on knowing different approaches to be used to implement the problem statement for the project.

Reading different Blogs and relevant articles and visiting the websites with similar content on BFS and Graph Databases to get an idea on how the problems can be addressed.

Visited the Neo4j official website to learn the CQL queries and was able to implement them.

Progress on work plan:

The Progress is:

* We created a Nodes, Able to find a node.
* Able to make the relationship between the Nodes.
* Able to Match the Patterns.

We are also trying to implement the BFS Algorithm for the Project in python and c++.

Working on the Connectivity between the Neo4j and Python.

Please find the below snips and CQL queries and the Algorithm we are working on:

Graphical user interface, text, application

Description automatically generated

//Create the user in the Database with the keys user id and name the examples are mentioned for the same.

create (UserName1:User {id : 1,name:'UserName1'})

create (UserName2:User {id : 2,name:' UserName2'})

//Make the Entitled relationship between the users to make a =n edge between the node the examples are mentioned for the same.

create (UserName1)-[:knows]->( UserName2),( UserName1)-[:knows]->( UserName3),

(UserName1)-[:knows]->( UserName2),

(UserName3)-[:knows]->( UserName4),

Some CQL queries used in Graph Database Neo4j:

CREATE CONSTRAINT ON (n:User) ASSERT (n.id) IS UNIQUE

CREATE INDEX FOR (m:User) ON (m.name)

 MATCH (UserName1:User {id: 1}) RETURN UserName1

 MATCH (n:User) RETURN n LIMIT 25

This is how the network looks when we try to retrieve the created Graph:

A picture containing necklet, screenshot

Description automatically generatedChart

Description automatically generated

When we need some kth level value of friends for any user, we use match query as follows:

Match (p: User)- [\*3]->(n: User) where p.name = 'UserName'  return distinct n

Resulted output would be the nth level friends of a user:

Graphical user interface, text, application

Description automatically generated

Future Milestones To be Achieved:

* To inter-link the Python/C++ programming language interface with Graph DB.
* To complete and implement the Algorithm of BFS to achieve the feat of searching nth level users in the network.
* To Get the Dataset Having the username and id with their specific links.
* To use the google cloud into the Problem statement.
* To implement the code in the Linux machine, commit it.
* Document the source code and README file .
* Prepare a Project Documentation.

Still must make 100% progress for the Implementation and Completion quarter to get the project to the conclusion.

References and Citations:

[1]Yuede Ji .,and H. Howie Huang.2020.AQUILA: Adaptive Parallel Computation of Graph Connectivity Queries. HPDC ’20, June 23–26, 2020, Stockholm, Sweden.149-160

[2] Bharati, A. (2019, October 3). *Social Networking with BFS and Neo4j*. Nuclei. https://gonuclei.com/resources/social-networking-with-bfs-and-neo4j/

[3] *Understanding Neo4j’s data on disk - Knowledge Base*. (n.d.). Neo4j Graph Data Platform. https://neo4j.com/developer/kb/understanding-data-on-disk/