A Practical Approach to Process Streaming Data using Graph Database

Mukul Sharma
Research Scholar
Department of Computer Science & Engineering
SBCET, Jaipur, Rajasthan, India

Narendra Singh Yadav,Ph.D
Assoc. Prof.
Department of Computer Science & Engineering
SBCET, Jaipur, Rajasthan, India

ABSTRACT

In today's information scenario, processing of data for exact knowledge has become a very important but critical task for the researchers and organizations. Involvement of Big Data and real time streaming data makes the data processing more challenging in order to extract and to visualize the exact data. In the most popular social media portals that includes streaming data like Twitter, Facebook and LinkedIn the rapidly growing information is updated several thousand times within one second.

In this research, the extraction of streaming data from most popular social media portal (Facebook) is done using different SDKs, Graph API Explorer and other APIs and data processing is done using Cypher Query Language (CQL) and Neo4jCLient (C# API for Neo4j) in Microsoft C#. Then Neo4j Graph Database along with Microsoft Visual Studio 2013 is used for the Visualization and knowledge extraction. CQL facilitate the extraction of streaming data in an efficient manner as its development intended the processing of data in a linked manner. The processing and visualization of rapidly growing streaming data is done as linked data which makes the data and knowledge extraction very easy as the relationship between data is present along with the link between them.

***Keywords**:* Graph Database, Streaming Data, Neo4j, Microsoft C#, Neo4jClient, Cypher Query Language

1. INTRODUCTION

Graph databases have become the popular choice for the processing of connected or linked data. Graph databases excel at management of highly connected data along with complex queries being independent from the size of database or datasets. Graph databases starts visualization with one node and explore the data with relationship of that node. Neo4j has the ability to explore billions of nodes at a single time in a connected manner. We choose Neo4j for this research along with Microsoft C# and Cypher Querying to process streaming Data.

1.1 Streaming Data:

Concept of Data streaming can be understand as the transfer of data with a constant but at extremely fast speed rate capable to support various applications as HD televisions or the systems that requires continuous backup facility to a storage medium of the flow of data within a single system [2].

Data streaming is an act in which transfer of high speed data is involved between a device and memory, it can be on a host system or can be on a hard drive drive. National Instruments provides the streaming devices that can perform data transfer at the rate of 750 mbps and up to 12 TBs of data. One can use several devices with data storage capabilities to support systems having up to 2.8 Giga Bytes of linked data to and from the disk with data rates being more than 48 TBs of area for storage.

1.2 Linked Data:

In big data processing linked data can be define as a method that can describe or publish structured data in order to make it to be interrelated and interlinked while being more useful with the capabilities of semantic queries. Linked Data is basically a concept in which a URI can uniquely define a data set that can describe the resources [5].

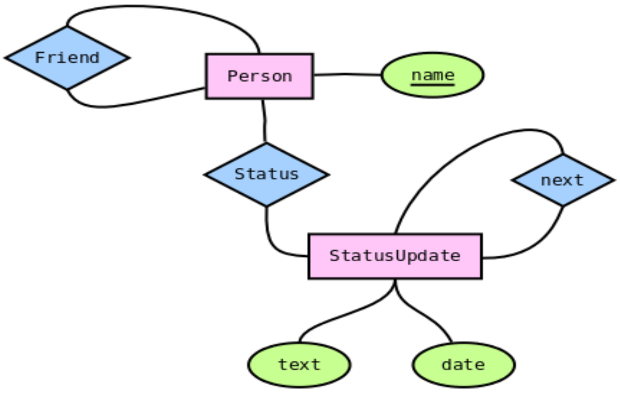


Figure 1: Graph Representation of Social network

1.3 Neo4j Graph Database:

Neo4j is most popular, open source graph database, basically developed and implemented using Java. Neo4j is an embedded and disk-based Java persistence engine with fully transactional capabilities, and is capable to store linked data as graph instead of relational tables" [6].

Neo4j version 1.0 was released in month of February, 2010. The community edition of the Neo4j is licensed under the free GNU, GPL (General Public License) and available at github. Few additional modules for the purpose of commercial uses, such as online backup and high availability options are licensed under the free AGPL. The Neo4j database, having the additional modules, is also available with a commercial license, in a dual license model [6].

1.4 Cypher Query Language:

Cypher is the official query language for the Neo4j graph database. It is basically a declarative graph query language with the capabilities of efficient and expressive querying and performing several operations on the graph stores. Cypher is a comparatively simple but still one of the most efficient and powerful query language.

1.5 Neo4jClient:

Neo4j client is a .NET client for Neo4j an open source, transactional graph database. Neo4jclient for graph database makes it very simple to write Cypher queries with the capabilities of C# with Visual Studio IntelliSense and also provide a support for the basic CRUD operations and indexing.

1.6 Microsoft C#:
Microsoft C# is the official language of Microsoft .NET. It is a programming language that is developed for the development of different kinds of applications such as windows, app store, web etc. that execute under .NET Framework. C# is object-oriented, powerful but still simple. The several innovations in C# provides rapid application development.

1.7 Facebook Graph APIs:
The Facebook Graph API is the best way to get data of Facebook's user in and out of Facebook's social graph. It's a low-level API which is based on HTTP and can be used by developer for querying data, adding new stories, uploading photos and several different tasks that an application might need to perform.

1.8 Facebook SDKs:
Facebook offers a variety of different SDKs consists of a large set of client-side operations for adding Social Plugins within websites or applications, calling and using several API and implementation of Facebook Login.

Facebook officially provides SDKs for Android, Unity, JavaScript, iOS and PHP, but there is also a large range of unofficial or third-party SDKs are available for a range of different languages and different frameworks developed by outstanding communities of active developers.

2. LITERATURE SURVEY
Jayanta Mondal et. al. in their research titled "Stream Querying and Reasoning on Social Data" have given a broad classification of various types of stream querying and reasoning tasks along with examples and further discussed about the different challenges in order to process efficient querying on streaming data and knowledge extraction. They also classify stream reasoning and querying tasks according to their input scope, temporal scope and network traversal scope [1].

Davide F. Barbieri and Emanuele Della Valle, in their research paper with title "A Proposal for Publishing Data Streams as Linked Data" proposed an extension of C-SPARQL engine that is capable to publish streaming data as linked data. In their research they explained the principle of their approach and also explained the approach for publishing RDF streaming data generated by C-SPARQL queries and the RESTful services to control these queries [7]. But they did not give any practical approach for implementation of Streaming Linked Data Server and its evaluation.

3. EXPERIMENTAL SETUP
All the experiments performed in this research are done under following configurations:

- **Host System:** Intel Core i5 processor with 6 GB RAM and 1000 GB Hard disk.
- **Operating Environment:** Windows 8.1 64 bit
- **Neo4j Version:** Neo4j 2.1.3
- **IDE:** Microsoft Visual Studio 2013
- **Framework:** .NET Framework 4.5

Following steps has been used for experimental setup in this research

3.1 Extracting data from social media:
For the purpose of accessing data from social media (Facebook) we first created the Facebook developer application and used following SDKs and APIs to access the required information from user in our web application.

3.2 Facebook SDK:
During this research we've used Facebook SDK for .NET and Facebook JavaScript SDK.

Facebook SDK for .NET is used to retrieve data from Facebook Graph API by adding its reference in .NET application and Facebook JavaScript SDK is used to get access from user.

3.3 Facebook API:
For the purpose of security, Facebook provides Access Tokens for uniquely identifying users. The Access Token changes dynamically every hour, or so. We are going to start by obtaining an access token from Facebook's Graph API tool.

3.4 Microsoft c#
After obtaining the access token using a Session State and a Handler of C# we perform an HTTP POST with the access token and redirect our user to a new page, in order to send information to the server.

Facebook Developer page provides Graph API Explorer tool for making Get() requests. We've used two Get() requests. One, for getting the basic details of the logged-in user, and the second for retrieving the likes of the user.

3.5 Processing Data
After extracting data from social media, the big challenge is to process this huge data which is being updated thousands times in a second. To process the streaming data in order to extract the useful information we used official query language of Neo4j graph database with the capabilities of C# using Neo4jClient.

3.6 Neo4jClient:
We added following namespaces after installing Neo4jClient in our application for the purpose of enabling capabilities of Cypher Query Language with Microsoft C#.

```
Using Neo4jClient;

Using Neo4jClient.Cypher;
```

3.7 Cypher Query Language
Cypher Query Language (CQL) is used to query the streaming data effectively. CQL consists a large collection of keywords for the purpose of querying linked data to form a graphical structure. Here is a small example, for the query to display a graph in which user and category are the nodes and interest areas of user is the relationship.

Query in C# within application:

```
.Match( "(x:user),(y:category)" )

.Where( "x.Username='" + facebookUser.Username + "' and y.Category_name='"

.Create( "(x)-[r:interest_area]->(y)" )

.ExecuteWithoutResults();
```

3.8 Data Visualization

After extracting the data from social media and process it using Cypher Query Language for the purpose to visualize the results in Neo4j Graph Database we use different tools that are available within the WebAdmin panel of Neo4j Graph Database.

3.9 Neo4j

Neo4j graph database provides the facility, to store and traverse billions of nodes and links with an excellent level of efficiency and robustness.

1. *Neo4j Web Admin*

The Dashboard tab provides an overview of a running Neo4j instance. Neo4j Web Admin Consists of different tabs for the specific purposes as follows:

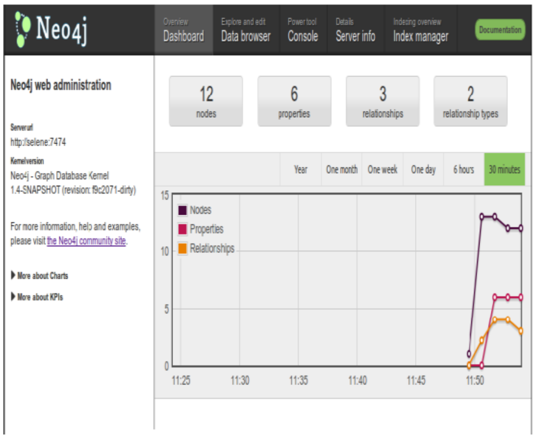
a) Status Monitoring: collection of status panels, displaying current resource usage.

b) Entity Chart: The charts show entity counts over time: node, relationship and properties.

c) Data Browser Tab: Use the Data tab to browse, add or modify nodes, relationships and their properties.

d) Console Tab: Console tab gives query access via Cypher and HTTP access via the HTTP console.

e) Server Info: Give information about active instances at server.



Neo4j web administration

Serverurl
http://selene:7474

Kernelversion
Neo4j - Graph Database Kernel 1.4-SNAPSHOT (revision: f9c2071-dirty)

For more information, help and examples, please visit the Neo4j community site.

12 nodes | 6 properties | 3 relationships | 2 relationship types

Figure 2: Neo4j WebAdmin Panel

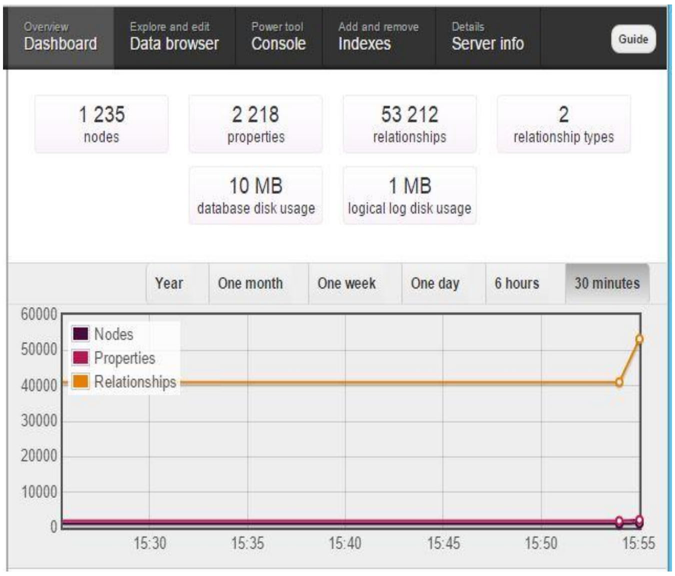
4. RESULTS AND ANALYSIS

After extracting the data from Facebook using the Access Token of the user and writing the Cypher Queries within C# for the specific relationships & nodes we start the Neo4j community server and access port 7474 at localhost and open the WebAdmin Panel.

In Dashboard of WebAdmin panel we initially get following data configuration:

Table 1: Data Configuration Details

Nodes	1,235
Properties	2,218
Relationships	53,212
Relationship Types	2
Database Disk Usage	10 MB
Logical Log Disk Usage	1 MB



1 235 nodes | 2 218 properties | 53 212 relationships | 2 relationship types | 10 MB database disk usage | 1 MB logical log disk usage

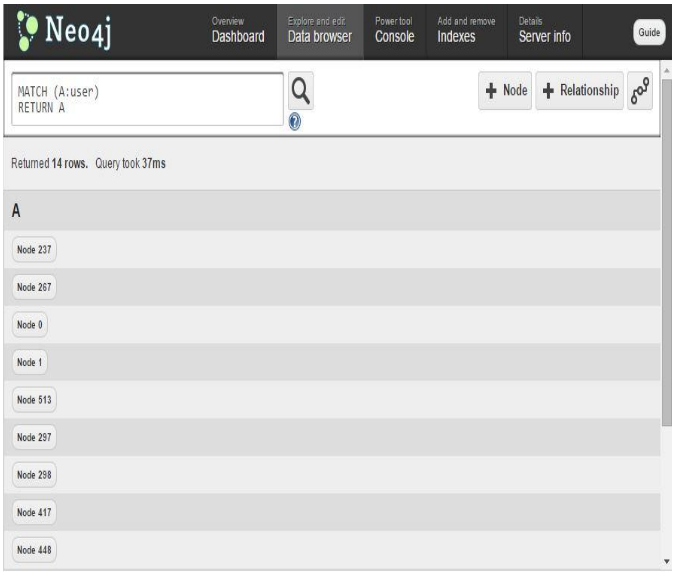
Figure 3: Data Configuration

In this WebAdmin panel we execute the following Cypher Query under Data Bowser tab

MATCH (X: user)

RETURN X

The above CQL returns a single node but as a user click on the node, user can view all its properties.



```
MATCH (A:user)
RETURN A
```

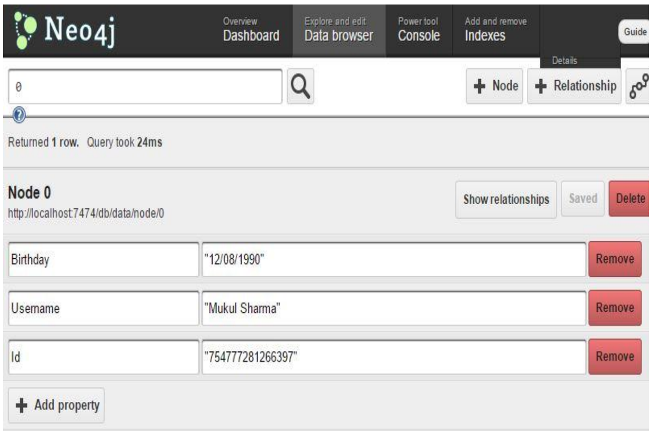
Returned 14 rows. Query took 37ms

A
Node 237
Node 267
Node 0
Node 1
Node 513
Node 297
Node 298
Node 417
Node 448

Figure 4: Cypher Query Result

In above figure, we can see that Neo4j returns the single nodes as output but as it is a graph database it consists an ability to visualize the nodes.

We further clicked on node 0 and get its details as an output as following figure.



Returned 1 row. Query took 24ms

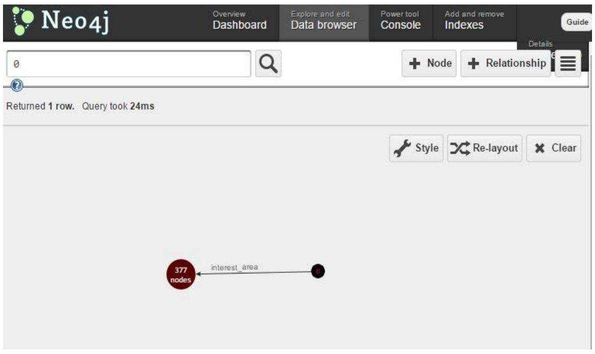
Node 0
http://localhost:7474/db/data/node/0

Birthday	"12/08/1990"
Username	"Mukul Sharma"
Id	"754777281266397"

Figure 5: Properties of Node 0

From above figure it is clear that as an output we got all the public details of user 0 such as Birthday, Username and ID.

As we clicked on the visualize button under Data Browser Tab we get the visual output for the node 0 similar to following Figure 6.

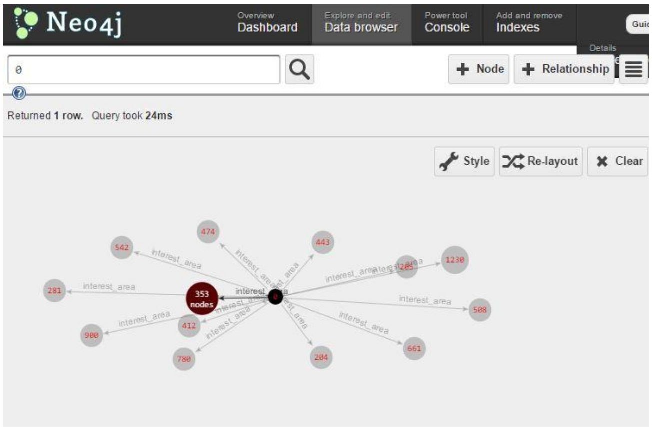


Returned 1 row. Query took 24ms

Figure 6: Visual Representation of Node 0

In the visual representation of node 0 we found 377 nodes linked with node 0 with relationship named interest_areas.

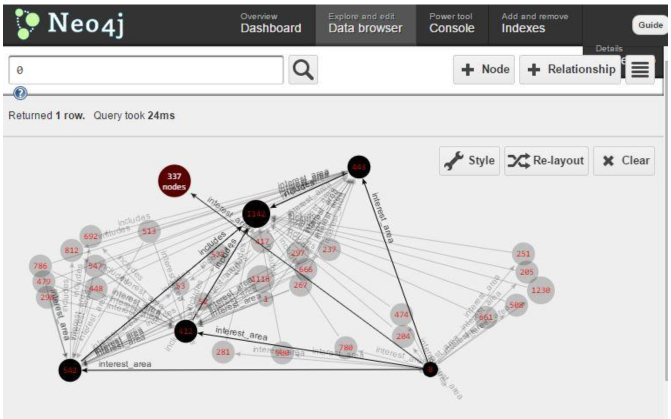
On clicking 377 nodes we get their properties and further we include 14 out of 377 nodes to be linked with the visual representation as following figure.



Returned 1 row. Query took 24ms

Figure 7: Representation of Node 0 and related nodes

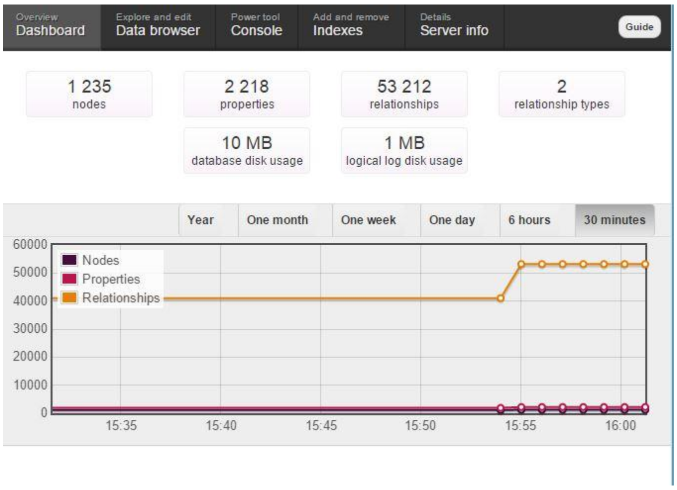
In above representation we further click on different nodes and get several links and relationships. As this data is the live streaming data from Facebook that relates with a specific user containing thousands of relationships, even then querying this data does not seem to be problem in order to access exact information.



Returned 1 row. Query took 24ms

Figure 8: Linked Streaming Data

In above figure we can see the visualization of tightly linked complex data consisting of several relationships with hundreds of nodes.



1 235 nodes
2 218 properties
53 212 relationships
2 relationship types
10 MB database disk usage
1 MB logical log disk usage

Figure 9: Data Configuration after analysis is done

Above figure shows the data configuration after the analysis of result is completed we can see as we started the analysis number of nodes, properties and relationship doesn't change.

5. CONCLUSION

This research demonstrate the power of Cypher Queries along with Neo4j Graph Database and Microsoft C#. Neo4j managed more than 53,000 relationships and respond for user's queries within few milliseconds, proved that it is possible to handle and manage the streaming data of the users of various social media portals while giving them an extreme interface and interactive way to query and respond for their queries within few milliseconds.

As in this research initially node 0 was having 377 interest areas, out of these 377 areas user selected only 14 and from these 14 user further selected one interest area which is further includes 15 nodes and 12 interest areas and so on. From this analysis we can say that Neo4j works with nodes and relationships between these nodes, this model also reduces the complexity of user information accessed from social media as a single relationship with one node can manage thousands of other nodes and allows user to further access only desired nodes and their relationships.

6. FUTURE SCOPE

In future this concept to access and visualizing streaming data from social media can be implement using different graph databases as HyperGraphDB and FlockDB to make a qualitative and quantitative analysis of these databases and for an improved user experience.

It will also be a great experience to access the useful and exact information within the most popular social portals just by clicking on the nodes. Integration of Neo4j Graph Database will be a revolutionary attempt in social media querying.

Generation of Dynamic Cypher Queries can make this concept more effective and compatible for various social media portal users in coming years.

7. REFERENCES

[1] Jayanta Mondal and Amol Deshpande, “Stream Querying and Reasoning on Social Data”, Department of Computer Science, University of Maryland, College Park MD 20742

[2] Database Trends And Applications. Available http://www.dbta.com/Articles/Columns/Notes-on-NoSQL/Graph-Dat abases-and-the-Value-They- Provide-74544.aspx,2012

[3] R. Angles and C. Gutierrez,” Survey of graph data-base models”,. ACM Comput. Surv., 40(1):1–39, 2008.

[4] Mukul Sharma and Pradeep Soni, “Quantitative Analysis and Implementation of Relational and Graph Database Technologies”, International Journal of Modern Computer Science and Applications (IJMCSA) ISSN: 2321-2632 (Online) Volume No.-2, Issue No.-5, September, 2014

[5] D. F. Barbieri, D. Braga, S. Ceri, E. Della Valle, and M. Grossniklaus. “Continuous queries and real-time analysis of social semantic data with c-sparql”, Web Workshop at the 8th International Semantic Web Conference, October 2009.

[6] The Neo4j Team , The Neo4j Manual v2.0.0-M03, Neo Technology, May 2013, Available http://www.neotechnology.com

[7] Weaver,Jesse, and Gregory Todd Williams, "Scalable RDF query processing on clusters and supercomputers." The 5th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS2009), 2009.

8. AUTHOR’S PROFILE

Mr. Mukul Sharma is a Microsoft certified Technology Specialist. He has more than two year experience in research & development with Microsoft Technologies. He is Pursuing his Master of Technology Degree in Computer Science. His area of research includes Ad Hoc Networks, Parallel Programming, Scalable Computing, Big Data and NOSQL Technologies.

Dr. Narendra Singh Yadav received M.Tech. degree in Computer Science from Birla Institute of Technology, Ranchi, India in 2002 and completed Ph.D from Malaviya National Institute of Technology, Jaipur, India in 2011. He is an Associate Professor and Head of Department of Computer Science & Engineering in SBCET, Jaipur. He is an active member of various professional bodies. His research interests include Clustering, Routing and Security in ad hoc wireless networks, wireless sensor and wireless hybrid networks.