Domain Suitable Graph Database Selection: A Preliminary Report

Vijay Kumar A, and Anjan Babu G

Abstract- This work is proposed based on the saying "as the time passing priority will be given to the new one than old". The concept on which this work focuses is the percentage of unstructured data getting introduced by various sources when compare to structured data was increasing exponentially day by day. As we know the fact that relational databases are the better room which satisfies structured data needs, but they must accept their inability in dealing with unstructured data due to some limitations which turned as advantage to NoSQL family, the new trend. According to marketplace, the NoSQL demand is anticipated to produce $5.6 billion in returns by 2015. Based on the current market needs we picked up the NoSQL component graph databases which are very popular in current epoch to investigate on various parameters in comparison with each other are presented both in tabular and graphical representations which will be added advantage to the database designers to opt suitable graph database based on their application needs.

Keywords--Exponentially, Investigate, Limitation, unstructured, marketplace.

I. INTRODUCTION

THE limitations of relational databases made a path to the database designers for inventing an alternative one for managing the unstructured data. One such alternative is graph database, which is from **No**SQL category. The major focus of **No**SQL databases is on unstructured data like images, videos etc. graph databases follows graph theory techniques for managing the data which uses nodes, properties and edges and provides index-free adjacency. Every node contains a direct link to its adjacent node and there is no need of index lookups in graph databases. The organization of paper is done as follows; Section 2 lists all the 16 graph databases which are presented in this article, Section 3 explains some specific query languages supported by graph databases with some sample queries, Section 4 presents and explains comparison matrix, finally concluded and suggested for future work.

II. GRAPH DATABASES

Due to the advancements in data storage technology, the need for accommodating new changes dynamically is playing a crucial role in the database realm which is the major drawback of relational databases. The scalability, natural representation and white board friendly nature of graphs aids to achieve that property very easily.

Vijay Kumar A, and Anjan Babu G, are with Sri Venkateshwara University, Tirupati,India.

Neo4j, AllegroGraph, Titan, BigData, Sones, Dex, InfiniteGraph, HyperGraphDB, Trinity, InfoGrid, G-Store, OrientDB, CloudGraph, VertexDB, ArangoDB, FlockDB are some of the graph databases which are compared here in this article with respect to their general features, ACID support, graph types and usage.

III. QUERY LANGUAGES

A. CYPHER

It is declarative query language specifically designed for managing the data which was stored in Neo4j graph database. Like all declarative query languages, Cypher too focuses on the output which it wants to retrieve rather than the technique used to retrieve it. The motivations of some of the query languages like SQL act as a base to develop the Cypher query language [1].

The structure of the Cypher query language is identical to SQL. In SQL first we select the area, in which our desired data stored, then we plan how to retrieve it, in the same way in Cypher first we select the starting node through START command, then we reach the destination through several techniques.

Sample Queries:

SQL: SELECT * FROM "person" WHERE Name = 'alice'.

Output:

NAME	ID	AGE	GENDER
alice	1	25	male

Cypher:

START person=node: Person (name='alice') RETURN person.

Output: Node[0]{name:"alice",id:1,age:20,gender:"male"}

B. SPARQL

SPARQL is an RDF query language which manages the data that was stored in the form of triples. World Wide Web Consortiums RDF DAWG made SPARQL as a standard for semantic web. It retrieves information from any type of source, may be structured or unstructured data, and performs intricate joins from different databases very naturally by traversing.

Major graph databases which uses SPARQL querying the data is AllegroGraph, Trinity, Bigdata. SPARQL uses "? or $" before a string which acts as a variable to store and display the data which was fetched from triplestore [2]. Suppose if we want to visit the home page of any friend of us from our facebook home page, we can do that with the following

SPARQL query.

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
PREFIX card: < https://www.facebook.com/
profile.php?id=100000685414282 >
SELECT? homepage
FROM < https://www.facebook.com/
profile.php?id=100000685414282>
WHERE {
   card: i foaf: knows ?known.
      ?known foaf: homepage? homepage
          }
```

FOAF is a standard RDF vocabulary for describing people and relationships. SPARQL uses foaf for retrieving the names and homepages.

C. GREMLIN

Gremlin is a functional language in which traversal operators are linked together for generating path-like expressions. It traverses level by level from source node to the desired node for fetching the requisite information. It can be natively used in various JVM languages. Some of the graph databases which use Gremlin for accessing the data are Titan, and InfiniteGraph [3].

The following sample query explains how the traversing procedure will be carried out in Gremlin query language.

Gremlin> g.V('name', 'Cambridge').out(' Department'). out('Computer Science').out ('HOD').name.

In the above query "g" indicates that, from the current graph, get all the vertices with name property "Cambridge", then traverse outgoing "Department" edge's from "Cambridge", then traverse outgoing "Computer Science" edge's then "HOD" edge's and finally get the name property of the HOD of Computer Science Department in Cambridge.

D. AQL

ArangoDB Query Language (AQL) is designed for handling the data that was especially stored in ArangoDB graph database. Storage and retrievals will be done using collections. It is a declarative query language, which focuses on what output to be produced not on how the output is produced. AQL supports client independency property, which tells that language and syntax are same for all the clients, whatever the programming language clients may use. Sample query was given below which displays top 3 ranked companies in the companies' collection [4].

```
FOR c IN companies   LIMIT 3
RETURN {"rank": c.id, "name": c.name}

Output:    [{"rank": 1, "name": "Neo"},
        {"rank": 2, "name": "Franz"},
        {"rank": 3, "name": "Sparsity"}]
```

In the above query 'c' treated as an object for the collection 'companies', and that object will be used for retrieving the data from the collection companies.

IV. COMPARISON MATRIX

We presented the comparison matrix as Table 1 which projected the information by comparing 16 graph databases. In this article we collected and presented various parameters which differentiate the graph databases in various aspects. First we compared general parameters, ACID support parameters, type of graphs supported by the various graph databases, the usage criteria of different graph databases.

A.General Parameters

The parameters which we presented in the comparison matrix include the language used for designing the graph database, the mode of availability whether it is open or closed source, the owner or maintainer of the graph database, the company by whom it was designed and maintained, the license of graph database, the ability of working in different environments, ability of distributing [5], query language used, API and various paths used for getting the desired information. Most of the graph databases which we presented in the comparison matrix support the distributing property, which is the major drawback of the relational databases. Based on this limitation of relational database it can be concluded that graph database is the best alternative database. The majority of the graph databases support portability, with which they can get the capability to work with different environments.

B.ACID Support

The trustworthiness of database transactions is guaranteed by Atomicity, Consistency, Isolation, Durability properties. The graph databases which support all ACID properties are indicated with full support, and those which support some of the ACID properties are indicated with partial support and the remaining which doesn't support either are [6] not preferable as the application domain requires fully transactional support.

The databases which are not to prefer are Sones, InfoGrid, G-Store, CloudGraph, and VertexDB because the transactions carried out by these databases won't guarantee the reliability and consistency. The word reliability is more suitable to Neo4j, AllegroGraph and some more which are listed under full ACID support in comparison matrix.

C. Types of Graphs

This parameter indicates various types of graphs supported by all graph databases. These include simple graphs, attributed graphs and hypergraphs as shown in Fig.1. The hypergraph concept won't put any constraints on edges to point to number of nodes.

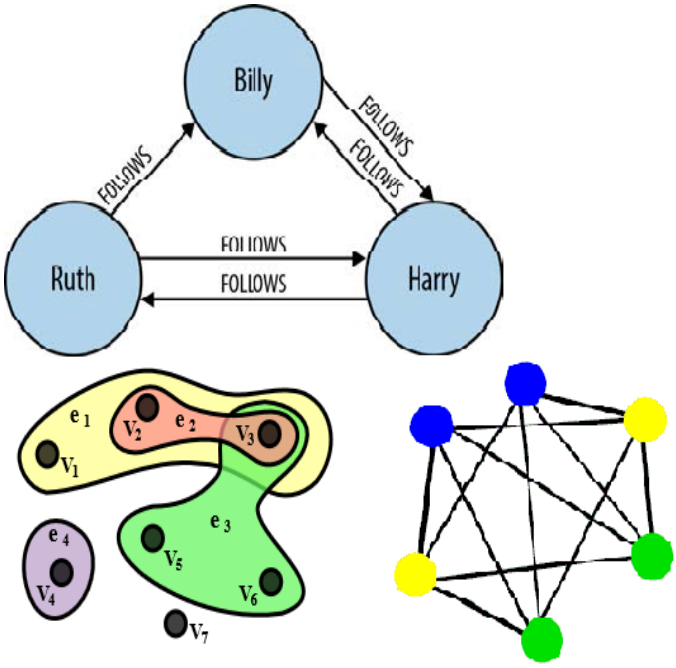

Fig 1: Simple, Attributed, HyperGraphs

In many real-world problems however the relationships between the various objects are very complex, simple graphs are not sufficient for representing such complex relationships; in such cases these hypergraphs place a vital role. One of the areas where such complex relationships exist is Artificial Intelligence. From the comparison matrix there are 3 graph databases which support these hypergraphs are HypergraphDB, Sones, Trinity. Graphs will be more expressive when the attributes are allowed. These attributed graphs are well suited for social networking sites which involve social interactions of individuals. From the comparison matrix it seems eleven graph databases support attributed graphs among Neo4j is the world's most popular graph database.

D. Usage

The usage levels of various graph databases are segregated into three; whether the graph databases are used for just retrieving the data or provide the facility to do some analysis and reasoning. Among the databases which we listed AllegroGraph supports analysis and reasoning [7, 8] from the data retrieved, there by extending its ability to support the decision making paradigm. All 16 graph databases by default supports the retrieving property out of which 4 graph databases shown their presence in analysis are Sones, Dex, InfoGrid and AllegroGraph.

V. CONCLUSIONS

Responsiveness and goodwill of a database technology are indirectly proportional. The more time it takes to give response the less chance of electing such technology as storage medium will be. By keeping this into consideration we made some suggestions which help the database designers in opting suitable technology. Since every graph database has their own advantages in various aspects like distributing capability, portability, transaction support, etc, the application area decides which graph database will fit exactly by verifying the mandatory features must be supported to reach the goals of the domain. It will act as a manual to the database designers and reduces searching time for the mandatory requirements to be needed in their application domains like concurrency control support, complex relations support, reasoning and decision making support etc. In above cases which one to opt and which not to opt were clearly explained in this document. Since the selection of suitable graph database also places an important role in achieving the performance, it is an added advantage to the designers to be worry free from the performance issues.

VI. FUTURE WORK

Due to the white-board friendly nature of the graphs, any data can be represented naturally in the form of graphs. Since geographic networks are inherently graphs, this work can extend the support to linkup with geo data models, where locations and route segments are stored as nodes and relationships consecutively. The details about each location and segment are stored as properties associated with each node and relationship. By storing one's geo network as a graph, one avoids the extra effort spent converting a graph into tables and then back again. Bioinformatics is one more area where we can deal with graph structures for storing various data sets like, transformations involved in DNA sequencing forms a network like structure, which involves nodes and edges in the form of graph.

REFERENCES

[1] http://neo4j.com/docs/pdf/neo4j-manual-2.0.4.pdf.
[2] http://www.w3.org/2009/Talks/0615-qbe/
[3] http://en.wikipedia.org/wiki/Gremlin_ (programming_language).
[4] http://docs.arangodb.org/AqlExamples/RE- ADME.html.
[5] Robert McColl, David Ediger, Jason Poovey, Dan Campbell, David A. Bader, "A Performance Evaluation of Open Source Graph Databases", Georgia Institute of Technology.
[6] RND Irena Holubova, Department of Software engineering, master Thesis on "Analysis and Experimental Comparison of Graph Databases" Charles University in Prague-2013.
[7] Darshana Shimpi, Sangita Chaudhari, "An Overview of Graph Databases" International Conference in Recent Trends in Information technology and Computer Science (ICRTITCS-2012), Proceedings published in International Journal of Computer Applications (IJCA) (0975-8887).
[8] Renzo Angles, "A Comparison of Current Graph Database Models" 3rd International workshop on Graph Data Management: techniques and applications (GDM 2012) 5th April, Washington DC, USA.

TABLE I
COMPARISON MATRIX OF 16 GRAPH DATABASES

GRAPH DATABASE	GENERAL PARAMETERS									ACID SUPPORT			GRAPH TYPES			USAGE		
	Written In	Availability Model	Owner/ Maintainer	License	Portability	API	Reach ability	Distributed	Query Language	Full	Partial	Not Preferable	Simple	Hyper	Attributed	Retrieval	Reasoning	Analysis
Neo4j	Java	Open Source	NEO TECHNOLOGIES	GPL/ PROPRIETARY	Yes	Java, JPython JRuby	Fixed Length, Regular Simple, Shortest Path	Yes	CYPHER	Yes	No	No	Yes	No	Yes	Yes	No	No
AllegroGraph	Likely Java	Commercial	FRANZ & INC.	PROPRIETARY	No	Java	Fixed Length	Yes	SPARQL	Yes	No	No	Yes	No	Yes	Yes	Yes	Yes
Titan	Java	Open Source	AURELIUS	APACHE		Java, Blueprints		Yes	GREMLIN	Yes	No	No	Yes	No	Yes	Yes	No	No
BigData	Java	Commercial	SYSTAP, LLC.	GPLV2	Yes	Java	Fixed Length	Yes	SPARQL	Yes	No	No	Yes	No	No	Yes	No	No
Sones	C#	Commercial	SONES GMBH	AGPL/PROPRIE TARY	Yes	C#		No	SQL BASED GRAPHQL	No	No	Yes	Yes	Yes	Yes	Yes	No	Yes
Dex	Java, C++	Commercial	SPARSITY;TECHN OLOGIES	PROPRIETARY	Yes	Java, C++	Fixed Length, Regular Simple, Shortest Path	Yes	SQL BASED	No	Yes	No	Yes	No	Yes	Yes	No	Yes
InfiniteGraph	Java, C++	Commercial	OBJECTIVITY	PROPRIETARY	No	Java	Fixed Length, Regular Simple, Shortest Path	Yes	GREMLIN BLUEPRINT SUPPORT	Yes	No	No	Yes	No	Yes	Yes	No	No
HyperGraphDB	Java	Open Source	KOBRIX&SOFTW ARE	LGPL	Yes	Java	-	Yes	SQL STYLE	No	Yes	No	Yes	Yes	Yes	Yes	No	No
Trinity	C,C#	Commercial		-	Yes	C#	Fixed Length, Shortest Path	Yes	SPARQL	No	Yes	No	Yes	Yes	Yes	Yes	No	No
InfoGrid	Java	Open Source	JOHANNES & ERNST	AGPL/ PROPRIETARY	Yes	Java	-	No	WEB USER INTERFACE WITH HTML	No	No	Yes	Yes	No	Yes	Yes	No	Yes
G-Store		Commercial	-	-	Yes	C/C++	Fixed Length, Regular Simple, Shortest Path	Yes	SQL BASED	No	No	Yes	Yes	No	No	Yes	No	No
OrientDB	Java	Open Source	NUVOLABASE& LTD	APACHE2	Yes	Java	Fixed Length, Regular Simple, Shortest Path	Yes	SQL	Yes	No	No	Yes	No	Yes	Yes	No	No
CloudGraph	C#	Commercial	-	-	Yes	C#	-	No	GQL	No	No	Yes	Yes	No	No	Yes	No	No
VertexDB	C	Commercial	-	-	Yes	C, C++	Fixed Length, Regular Simple Path	No	SQL THROUGH JDB	No	No	Yes	Yes	No	No	Yes	No	No
ArangoDB	C,C++	Open Source	ARANGODB	APACHE	-	JavaScript, Blueprints	-	Yes	AQL	Yes	No	No	Yes	No	Yes	Yes	No	No
FlockDB	Java, Scala, Ruby	Open Source	TWITTER	APACHE	-		-	Yes	MY SQL BASED	No	No	No	Yes	No	No	Yes	No	No