

A STUDY ON GRAPH STORAGE DATABASE OF NoSQL

Smita Agrawal¹ and Atul Patel²

¹CSE Department, Institute of Technology, Nirma University, Ahmedabad

²Dean of CMPICA, CHARUSAT University, Changa

ABSTRACT

Big Data is used to store huge volume of both structured and unstructured data which is so large and is hard to process using current / traditional database tools and software technologies. The goal of Big Data Storage Management is to ensure a high level of data quality and availability for business intellect and big data analytics applications. Graph database which is not most popular NoSQL database compare to relational database yet but it is a most powerful NoSQL database which can handle large volume of data in very efficient way. It is very difficult to manage large volume of data using traditional technology. Data retrieval time may be more as per database size gets increase. As solution of that NoSQL databases are available. This paper describe what is big data storage management, dimensions of big data, types of data, what is structured and unstructured data, what is NoSQL database, types of NoSQL database, basic structure of graph database, advantages, disadvantages and application area and comparison of various graph database.

KEYWORDS

Big Data, Graph Database, NoSQL, Neo4j, graph

1. INTRODUCTION

Big Data is used to store huge volume of both structured and unstructured data which is so large and is hard to process using current / traditional database tools and software technologies. Data may be falls into textual and non – textual. Non – textual may include images, video, audio, signals, emails, social media posts, large binary files etc.[1] The goal of Big Data Storage Management is to guarantee a great level of data quality and availability for business intellect and big data analytics applications.[2,11]

2. DIMENSION OF BIG DATA

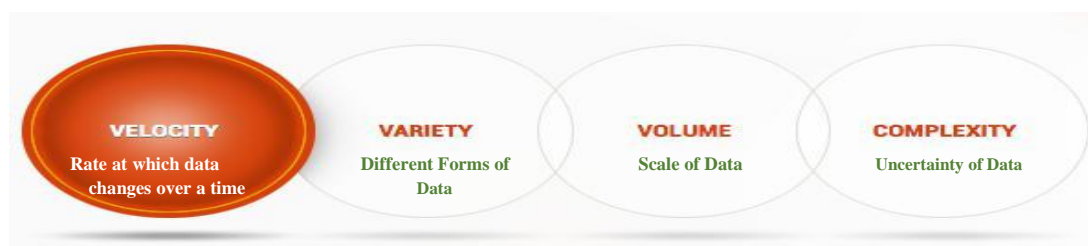


Figure 1. Dimension of Big Data [10]

Figure-1 describes the dimension of Big data Velocity, Variety, Volume and Complexity which are describe in details as follow:

2.1. High Data Velocity

Velocity means rate at which data changes over time. Velocity is the amount at which structure of database gets change. In other word, if value of any property of given structure gets change overall structure which is hosting those data and property will also get change. This thing gets happen due to two reasons. First is, now days business are changing very quickly, and as per business requirement changes, its data need will also be change. Second is, that data gaining is often an untried affair, some property are brought as per need. The most valuable to business will stay around and other will moves out.

2.2. Data Variety

Now a day data is exist in all type of data format. Data may be varies from structured, unstructured and semi – structured.

2.3. Data Volume

Due to large data comes from different location, datasets will become awkward when warehoused into relational database. Query Execution time will surge as database size grows hence more and more database joins has to be performed which is really time consuming which is called join pain. 12 TB of data transferred over internet every day.

2.4. Data Complexity

Data which is stored at different places and managed by multiple systems, it is difficult to manage those database hence complexity will surge.

3. TYPES OF DATA

Big Data may be varies from structured to unstructured database. In this section describe types of data exist now a days.

3.1. About Structured Data

Structured data is statistics, which is usually comes into text files, displayed in columns and rows format (Tabular format) which can effortlessly be tidy and administered by data withdrawal tools. **Examples:** - Database, XML Data, Data warehouses, Enterprise system (CRM and ERP)

3.2. About Unstructured Data

Unstructured data, which is usually comes in binary data format that is branded, is that which has no recognizable interior structure. Unstructured data is raw material, chaotic and companies store it all. However all this unstructured info will be converted into structured info first, which is very costly and time consuming process. Also not all types of unstructured data will be converted into structured data model. For example, an email contains info such as the when it is sanded, subject of email, and sender info but the gratified of the message is not so easily wrecked down and categorized. **Example:** - images, video, audio, signals, emails, social media posts, large binary files, RSS feed.

4. ABOUT NOSQL (NOT ONLY SQL) DATABASE

A Not only SQL (NoSQL) [9] database environment, makes non – relational and widely distributed database system which allows quick, ad – hoc query association and investigation of tremendously high – amount of data and dissimilar data types. Not only SQL databases are also sometimes referred as a Cloud databases, non – relational databases, Big Data databases and an innumerable of some other definition are also prepared in response to the pure volume of data which is generated, warehoused and examined by users (data generated by user) and their applications (data generated by machines). Not only SQL databases had become the alternative to the relational database management system which provides scalability, obtainability and fault detections and lenience which is key deciding factors. NoSQL is very flexible and schema less database model, which is horizontal scalability, supports distributed architecture, and language and interface usage that are “not only” SQL.

4.1. Document Store

At its most fundamental level the model is simply that we pile and fetch documents, just the same as an electronic filing cabinet. Documents incline to comprise the usual property which is having key-value pairs, but where values itself can be tilts, charts, or alike allowing for usual orders in the document just as we are used to with arrangements like in JSON and XML. Documents can be saved and fetched by their unique identification. Document store provides and ID's to be remember by application to which it is interested in. But in general document store relies on indexes to access of document based on their property and attributed. For example: - In e-commerce it is very useful to distinct different products types of that it can be easily map to their respective seller. **Example:** - MongoDB [1] and CouchDB.

4.2. Key-Value Store

Key-value store database is friends of the document store family. Key – value store database is intended for storage of data in a schema less way. In this database, all of its data consists on an indexed key – value pair. **Example:** - Azure Table Storage, Riak, BerkeleyDB, Cassandra, DyanmoDB.

4.3. Column Store / Big Table

The main four basic building blocks on the Big Table or Column Store are.

- o Column

- o Super Column
- o Column Family
- o Super Column Family

Column consists of name and value pair which is similar to the columns of relational databases. Number of columns are grouped into super column. Columns are warehoused into rows and when rows contains column only it is known as column family. When rows contains super columns it is known as super column family.

4.4. Graph Database

Graph database is a collection of vertexes and edges. Graph database is a set of nodes and relationship between them.[4]

5. BASIC STRUCTURE OF GRAPH DATABASE

Here in this section figure – 2 describe the basic structure of Graph Database. In structure of Graph database a graph is having records of nodes and its relationships. For each property graph nodes and relationships having properties of it[7,12]. AllegroGraph[14], Neo4j[5],G-store[15],DEX[16] ,VertexDB[17] and et al[6,8] are example of NoSQL Graph database.

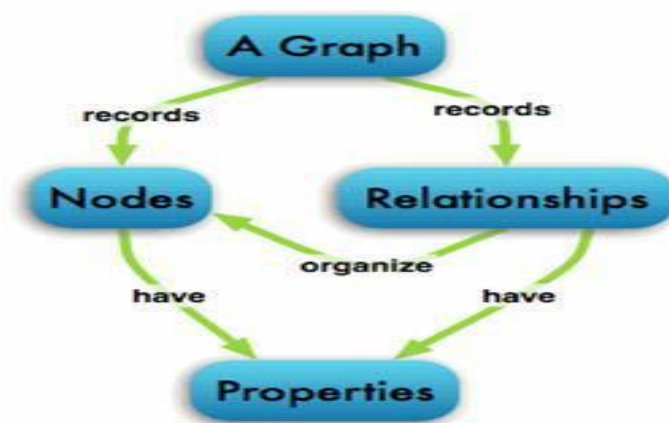


Figure 2. Basic Structure of Graph Database [5]

5.1. Nodes

The most important part of a graph database is nodes and their relationships. Nodes in graph database is used to represent entity, sometimes it is also used for represent purpose as well. Nodes, relationship and properties can be label with more than zero or more label.

5.2. Relationships

Relationships connect nodes. Relationships are always connected and directed (Always have a starting node and an ending node). In addition, every relationship has a label. A relationship's label and its direction enhance semantic simplicity to the connections between nodes. We can add properties to relationship which provides additional metadata for the graph algorithm, and for constraining queries at run time.

5.3. Properties

Nodes and relationships can have properties which is key – value pair. Properties are key-value pairs where the key and value both is a string, sometimes value are also numeric and other.[3] Property values can be either a primeval or an array of one primitive type.

5.4. Path

The path is same as road through which we can Travers. In graph database path may contains more than one nodes.

5.5. Traversal

Traversal means visiting the nodes of the graph database based on their relationships.

6. MORE ABOUT GRAPH DATABASE

This section describe advantages, disadvantages and application area for graph database.

6.1. Advantages of Graph Database

“Minutes to Milliseconds” performance.
Drastically-accelerated development cycles.

Extreme business responsiveness. Successful applications rarely stay still; changes in business conditions, user behaviors and technical and operational infrastructures drive new requirements.

Enterprise ready. Data is important.

6.2. Disadvantages of Graph Database

Very easy to describe data inconsistency.

Not widely used in business environment yet compare to relational database management system.

Can be conceptually difficult to understand at very first look.

6.3. Application/ Research area of Graph Database

Route finding (going from point A to point B)
Logistics
Authorization and Access Control
Network Impact Analysis
Network and IT Operations Management
Social Networking

7. COMPARISON OF VARIOUS GRAPH DATABASE

A comparison of databases is typically done by either using a set of common features or by defining a general model. In this section is describe the comparison of data model provided by each graph database based on it data structures (See Table-1). Here comparison of different Graph Databases based on in Data Storing Features, structure of Graph Data and support of ACID Properties described in Table 1. We consider some general features for data storing (see Table 1) in which identifies the support for main memory, external memory and backend storage as well as the implementation of indexed. Hence the support for external memory storage is a main requirement. Also, indexes are needful to improve data retrieval operations [13]. From the survey identified that Nested graphs is not supported by most of the graph database and if graph database support Nested graph in that case it support hyper graph also.

Graph data structures strongly recommended that graph nodes should be labelled and edges of graphs are labelled as well as directed. DEX[16] graph database partially support ACID property.

Graph Databases	Data Storing Features				Graph Data Structures									ACID Property			
	MainMemory	ExternalMemory	RelationalStorage	Indexes	Graphs				Nodes			Edges		Atomicity	Consistency	Isolation	Durability
					Simple	Hyper	Nested	Attributed	Labelled	Attribution	Directed	Labelled	Attribution				
AllegroGraph	√	√		√	√				√		√	√		√	√	√	√
Neo4j	√	√		√			√	√	√	√	√	√	√	√	√	√	√
G-Store		√			√			√			√	√		√	√	√	√
DEX	√	√		√	√		√	√	√	√	√	√	√	○	○	√	√
VertexDB		√	√		√			√		√	√		√	√	√	√	√

Table 1 Comparison of existing graph database by its storing features, data structure and ACID Property (In Table 1 √ indicate supports and ○ indicate partial support)

8. CONCLUSIONS

Here at the end we conclude that we have studied what is Big Database Storage Management, Dimension of Big Data Storage Model, Types of Data, What is NoSQL database, Types of NoSQL Databases, Basic structure of graph database, Advantages and Disadvantages of graph database and application/research area and comparison of various graph database. Comparison shows that Nested graphs is not supported by most of the graph database and if graph database supports Nested graph in that case it support hyper graph also.

REFERENCES

- [1] Smita Agrawal, Jai Prakash Verma, Brijesh Mahidhariya, Nimesh Patel and Atul Patel , 2015. "Survey On MongoDB: An Open-Source Document Database.International" Journal of Advanced Research in Engineering and Technology (IJARET).Volume:6,Issue:12,Pages:1-11.
- [2] Raj, Pethuru, et al. "High-Performance Big-Data Analytics."
- [3] Castellort, Arnaud, and Anne Laurent. "Representing history in graph-oriented nosql databases: A versioning system." Digital Information Management (ICDIM), 2013 Eighth International Conference on. IEEE, 2013.
- [4] Bajpayee, Roshni, Sonali Priya Sinha, and Vinod Kumar. "Big Data: A Brief investigation on NoSQL Databases." (2015)
- [5] <http://neo4j.com/>
- [6] http://en.wikipedia.org/wiki/Graph_database
- [7] E-Book of Graph Database by Ian Robinson, Jim Webber & Emil Eifrem
- [8] <http://en.wikipedia.org/wiki/NoSQL>
- [9] <http://www.3pillarglobal.com/insights/exploring-the-different-types-of-nosql-databases>
- [10] <http://www.looiconsulting.com/home/enterprise-big-data/>
- [11] Kanchi, Sravanthi, et al. "Challenges and Solutions in Big Data Management--An Overview." Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on. IEEE, 2015.

- [12] Buerli, Mike, and C. P. S. L. Obispo. "The current state of graph databases." Department of Computer Science, Cal Poly San Luis Obispo, mbuerli@calpoly.edu (2012): 1-7.
- [13] Angles, Renzo. "A comparison of current graph database models." Data Engineering Workshops (ICDEW), 2012 IEEE 28th International Conference on. IEEE, 2012.
- [14] "AllegroGraph," <http://www.franz.com/agraph/allegrograph/>.
- [15] "G-Store," <http://g-store.sourceforge.net/>.
- [16] N. Mart´inez-Bazan, V. Munt´es-Mulero, S. G´omez-Villamor, J. Nin, M.A. S´anchez-Mart´inez, and J.-L. Larriba-Pey, "DEX: High-Performance Exploration on Large Graphs for Information Retrieval," in Proceedings of the 16th Conference on Information and Knowledge Management (CIKM). ACM, 2007, pp. 573–582.
- [17] "vertexdb," <http://www.dekorte.com/projects/opensource/vertexdb/>.

AUTHORS

Smita Agrawal received Bachelor in Science (B.Sc. in Chemistry) degree from Gujarat University, Gujarat, India in 2001 and Master’s Degree in Computer Applications (M.C.A) from Gujarat Vidhyapith, Gujarat, India in 2004. She is pursuing PhD in Computer Science and Applications from Charotar University of Science and Technology (CHARUSAT) in the area of distributed processing. She is associated with Computer Science and Engineering Department of Institute of Technology - Nirma University since 2009. Her research interests include Parallel Processing, Object Oriented Analysis & Design and Programming Language(s).



Atul Patel received Bachelor in Science B.Sc (Electronics), M.C.A. Degree from Gujarat University, India. M.Phil. (Computer Science) Degree from Madurai Kamraj University, India. He has received his Ph.D degree from S. P. University. Now he is Professor and Dean, Smt Chandaben Mohanbhai Patel Institute of Computer Applications – Charotar University of Science and Technology (CHARUSAT) Changa, India. His main research areas are wireless communication and Network Security.

