ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

An Overview on Graph Database Model

Harsha R.Vyawahare, Dr P.P.Karde

Assistant Professor, Dept. of CSE, Sipna College of Engineering and Technology, Amravati, Amravati University, India

Head, Dept. of CSE, Govt. Polytechnic, Yavatmal, Maharashtra, India

ABSTRACT: The relational model has been a dominant database model in the computer industry since the 1980s mainly for storing and retrieving data. However, due to exponential growth of data it is becoming complicated to work with relational model as joining a large number of tables is not working efficiently , also relational model has dependence on rigid schema. One of the proposed solutions is to shift to the Graph databases as they aspire to overcome such type of problems. Graph Database Models is increasingly a topic of interest in the database industry. The representation of data in the form of a graph lends itself well to structured data with a dynamic schema. This paper goes over survey, use cases and popular implementations of graph databases.

KEYWORDS: Graph Database, Rigid Schema, dynamic schema.

I. INTRODUCTION

Until recently, back-end hardware and software treated all data identically. Customer demographics, press releases, product photos, and web data were all considered slight variations of the same kind of information. And regardless of the form of that information, it was all shredded, pummeled, and re-boxed into something resembling a collection of spreadsheets. This old way is known as the relational database.

The relational database is robust, effective, and proven. It is logically consistent and easy to understand i.e think lots of tables with rows and columns & is just like a spreadsheet, where a cell can point to data in another table. Decades of engineering have made relational databases fast, reliable, and flexible. Almost all the popular business database technologies — most Oracle, Microsoft, and SAP products, and anything containing the letters “SQL” — are relational databases. For many kinds of data, especially data that might easily fit in a spreadsheet (demographics, inventory, sales leads), nothing beats a relational database. Thus relational data model has dominated database management systems for more than thirty years. It has proved to be a powerful platform for business applications and has spawned, in the form of SQL, a standard language for querying databases.

But as digitization has progressed, data has grown in both size and scope. There are several questions like, what if your data isn’t so neat and clean? What if you have millions of documents you need to search at a moment’s notice? What if your data is better represented as a sort of social network, where the relationships between data are just as important as the data itself? What if your data does not fit clearly into any sort of logical structure? Also companies like Google and Amazon have long been generating massive amounts of data using countless numbers of servers. With the resulting data spread across multiple machines, traditional relational **SQL JOIN** (used to combine rows from multiple tables) operations are just not possible. Thus demand for alternatives to the relational model has grown.

Graph databases manage maps of interconnections between objects. With the rise of Social Media, Graph databases are increasingly being used because they are a natural fit for modelling things like interconnected web links, recommendations, tags, and friend and contact relationships. Traditional databases just aren’t able to store this kind of information easily in tables and rows of data. Core standard relational databases are the concepts of entities and relationships. In graph databases, the corresponding concepts are nodes and edges. A node holds the property and data for an object. Diagrams which show the relationships maintained in graph databases resemble the diagrams used in object-oriented programming, and because of the similarity, graph databases easily can model the relationships between the objects used when programming an application. Speed is another big reason graph databases are gaining traction.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

II. GRAPH BASED DATABASES

Graph databases address one of the great macroscopic business trends of today, leveraging complex and dynamic relationships in highly connected data to generate insight and competitive advantage. Whether we want to understand relationships between customers, elements in a telephone or data center network, entertainment producers and consumers, or genes and proteins, the ability to understand and analyse vast graphs of highly connected data will be key in determining which companies outperform their competitors over the coming decade. For data of any significant size or value, graph databases are the best way to represent and query connected data.

A graph is just a collection of vertices and edges i.e a set of nodes and the relationships that connect them. Graphs represent entities as nodes and the ways in which those entities relate to the world as relationships. This general-purpose, expressive structure allows us to model all kinds of scenarios, from the construction of a space rocket, to a system of roads, and from the supply-chain or provenance of foodstuff, to medical history for populations. A graph database is one that stores data in terms of entities and the relationships between entities. Thus graph database management system (a graph database) is an online database management system with Create, Read, Update, and Delete (CRUD) methods that expose a graph data model. Graph databases are generally built for use with transactional (OLTP) systems. Search or query with Graph DBs is called "traversal". These queries are designed to start at a specific node and explore its relationship with other nodes based on the relationships requested.

There are three types of graph database: true graph databases, triple stores and conventional databases that provide some graphical capabilities. Triple stores are often referred to as RDF databases. The difference between a true graph product and a triple store is that the former supports index free adjacency (which means you can traverse a graph without needing an index) and the latter doesn't. The former are designed to support property graphs (graphs where properties may be assigned to either entities or their relationships, or both) but recently some triple stores have added this capability. Both graph and RDF databases may be native products or they may be built on top of other database types. It was previously (more or less) the case that graph databases needed to be distinguished from RDF (resource description framework) and triple stores, where the latter were primarily designed to support the semantic web and the former for more general-purpose use. From a technical point of view the latter do not have the inferencing capabilities that are associated with the former.

There are three generic use cases for graphs CRUD (create, read, update, delete) applications that are focused on transaction processing; query processing—reporting, business intelligence and real-time analytics; and what we might call deep analytics (typically in batch mode) or data discovery. Different vendors in the graph market focus on one or more of these.

III. LITRETURE REVIEW

Emil Eifrem, CEO of Neo Technology has ran some tests in which he compared the speed of relational to graph databases. He created a "friends of friends" query and found that when the query of relationships went three levels deep that the graph database beat the relational one by a factor of 150, and when the query depth was increased to four the graph database bested the relational one by a factor of 1000.

Philip Howard, analyst at Bloor Research reported that Graph databases are critical when the degree of separation [ie, I know x who knows y who is related to z who used to live in the same house as w etc.]between entities becomes too great to handle using conventional technology. Oracle or DB2, for example, can reasonably handle up to three degrees of separation but not the six or seven

Howard commented on the limitations of graph databases, saying that "The major limitation is that while these are technically NoSQL databases, in practice they cannot be implemented across a low-cost cluster (at least not a present) but have to run on a single machine, the reason being that performance degrades rapidly across a network. Another potential drawback is that either you have to write your own queries using Java or whatever — which means employing expensive programmers — or you use SparcQL or one of the other query languages that have been developed to support graph databases, but this means learning a new skill."

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

Angles, R.; et al presents a survey of earlier work (pre-NOSQL) in graph databases. i.e prior to 2002,particularly geographical, spatial and semistructured database models. Older data models focused heavily on semistructured and XML data in a traditional database. The authors synthesized the notion of a "graph database model" and compare proposals available at the moment.

Angles, R.; performs comparison & performance analysis of different graph database models ,compares current graph databases concentrating on their data model features, that is data structures, query facilities, and integrity constraints. Author shows that most graph database models provide an innate support for different graph structures, query facilities in the form of APIs (most of the models) and query languages (a few of them), and basic notions of integrity constraints.

Jouili, S.;et al empirically compares graph databases ie presents Graph Database Benchmark, to compare four graph databases: Neo4j, DEX, Titan (BerkeleyDB and Cassandra) and OrientDB (local) on different types of workloads, each time identifying which database was the best and the less adapted. Based on measure, the database that obtained the best results with traversal workloads is definitely Neo4j: it outperforms all the other candidates, regardless the workload or the parameters used.

IV. THE POWER OF GRAPH DATABASES

[1] Performance
Graph database performance increases when dealing with connected data versus relational databases and NOSQL stores. In contrast to relational databases, where join-intensive query performance deteriorates as the dataset gets bigger, with a graph database performance tends to remain relatively constant, even as the dataset grows.

[2] Flexibility
Graph database allows structure and schema to emerge in tandem with our growing understanding of the problem space .Graphs are naturally additive, meaning we can add new kinds of relationships, new nodes, and new subgraphs to an existing structure without disturbing existing queries and application functionality. Because of the graph model's flexibility, we don't have to model our domain in exhaustive detail ahead of time.

[3] Agility
Modern graph databases equip us to perform frictionless development and graceful systems maintenance. In particular, the schema-free nature of the graph data model, coupled with the testable nature of a graph database's application programming interface (API) and query language, empower us to evolve an application in a controlled manner. At the same time, precisely because they are schema free, graph databases lack the kind of schema-oriented data governance mechanisms we're familiar with in the relational world.

V. USE CASES FOR GRAPH DATABASES

Graph databases lend themselves very well to a range of different uses. The most obvious one is social networks; Twitter, Facebook and LinkedIn all use a graph approach; but really they can apply to any kind of connected data:
Website link structure: Using URLs as nodes and hyperlinks as links, it's possible to find authoritative and popular websites.
Path finding: Their traversal efficiency make graph databases an effective path-finding mechanism. Links can be weighted, or assigned relative distances or times, to ascertain the shortest and most efficient routes between two nodes in a network.
Mapping dependencies: networks of computers and hardware can be modelled as graphs to find components with many dependents that may be potential weak points or vulnerabilities. Other dependency networks, for example corporate or investment structures can be mapped in a similar manner.
Communications: Communications between people can be stored as graphs. Applying network analysis measures can help find influential individuals.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2015

VI. POPULAR GRAPH DATABASES

I. InfiniteGraph by Objectivity is a high performance, scalable graph database that allows the storage of information in an entity-link-property model. It can interact with KeyLines through a Restlet server.
II. Neo4J by Neo Technologies is a popular open source graph database technology that is also available under commercial terms. Using Neo's own Cypher query language, users can visually analyse Neo4J graphs in KeyLines.
III. Titan by Aurelius is an open source database technology, capable of integrating seamlessly with KeyLines using Rexter.
IV. Google Cayley is a relative new-comer, an open source database developed by the search giant.
V. FlockDB Twitter, has developed its own graph database software to represent the links between its members.

VI. CONCLUSION

Many new database technologies have come onto the market in the last few years. While these alternative databases are not as time-tested as many relational database products, they offer distinct advantages over relational databases in some common business situations. It is worth understanding that in the era of "big data" not everything is the nail it appears to be when you're wielding only a relational database hammer. Many organizations are switching over old relational databases to these new technologies and enjoying the advantages over their competition. Among them, the Graph Databases [3] are calling the attention of the database community. This paper gave an overall survey of the current state of graph databases.

REFERENCES

[1] Database Trends and Applications. Available http://www.dbta.com/Articles/Columns/Notes-on-NoSQL/Graph-Databases-and-the-Value-They-Provide-74544.aspx,2012.
[2] M. Kleppmann, Should you go beyond Relational databases ? Available http://carsonified.com/blog/dev/should-you-go-beyond-relational-databadat.
[3]http://architects.dzone.com/articles/performance-graph-vs
[4] http://blog.octo.com/en/graph-databases-an-overview/
[5] http://keylines.com/graph-databases-data-visualization.
[6] Jadhav, P.; Oberoi ,R. ,"Comparative Analysis of Different Graph Databases", International Journal of Engineering Research & Technology e-ISSN: 2278-0181 ,vol. 3 ,no 9,pp.820-824 September 2014.
[7]Angles, R., "A comparison of current graph database models", IEEE 28th International Conference on Data Engineering Workshops
[8]Angles,R.;Gutierrez, C., "Survey of graph database models" , ACM Computing Surveys (CSUR) , vol. 40, no. 1, pp. 1–39, Feb 2008.

BIOGRAPHY

Harsha R.Vyawahare, working as Astt.Prof in the department of Computer Science & Engineering , Sipna CET, Amravati, Amravati University. Done M.E(Computer Science & Engineering) in 2012.Member of ISTE,IETE,IE,CSI.

Dr P.P.Karde, working as Head, Computer Department, Govt. Polytecnique, Yavatmal. He is Phd in Computer Sceince & Engineering from S.G.B. Amravati University, Amravati. He is a research guide for Ph.D. at S.G.B. Amravati University, Amravati.