SHORT SURVEY ON GRAPHICAL DATABASE

Harsha R Vyavahare[1]

Sipna COET, Computer Department, SGBAU University ,Amravati,
Amravati, Maharastra 444607, India
hvyawahare01@gmail.com

Dr.P.P.Karde[2]

HOD, GOVT polytechnic
Yavatmal, Maharastra,India
p_karde@Rediffmail.com

Abstract

This paper explores the features of graph databases and data models. The popularity towards work with graph models and datasets has been increased in the recent decades .Graph database has a number of advantage over the relational database. This paper take a short review on the graph and hyper graph concepts from mathematics so that graph so that we can understand the existing difficulties in the implantation of graph model. From the Past few decades saw hundreds of research contributions their vast research in the DBS field with graph database. However, the research on the existence of general purpose DBS managements and mining that suits for variety of applications is still very much active. The review is done based on the Application of graph model techniques in the database within the framework of graph based approaches with the aim of implementation of different graphical database and tabular database

Keywords: Graph Dataset ,Relational dataset, Graph Model, Tabular datasets.

1. Introduction

Only a database that embraces relationships as a core aspect of its data model is able to store, process, and query connections efficiently. A graph database stores connections and allows you to quickly traverse millions of connections and relations within a fraction of time. It is easy to solve the complex queries among the total size of your dataset, graph databases. The property graph contains connected entities (the nodes) which can hold any number of attributes (key-value-pairs). Nodes can be tagged with labels representing their different roles in your domain.In a graph database, each record has to be examined individually during a query in order to determine the structure of the data. For many reasons researchers are moving towers graph database, Some of them are-Graph databases are much faster than relational databases for connected .Graph databases make modeling and querying much more pleasant meaning faster development and fewer WTF moments.

2. Related work

[Roberto De Virgilio2013] Decades of engineering have made relational databases fast, reliable, and flexible. There are number of research has been done research over a conversion of relational data into graph modeled data . Plenty of those proposals focus on mapping relational databases to Semantic Web stores, a problem that is more focused than converting relational to general, graph databases is implementation of graph data model , which is our concern. On the other hand, some approaches have been proposed to the general problem of database translation between different data models. By this observation Roberto De Virgilio tried to implement a algorithm for creation of graph database. Also developed a system that implements the translation technique to show the feasibility of his approach and the efficiency of query answering. He had designed and developed a tool for migrating data from a relational to a graph database management system . He consider a different scenario where the database needs to be built from scratch. [Virgilio 2013]

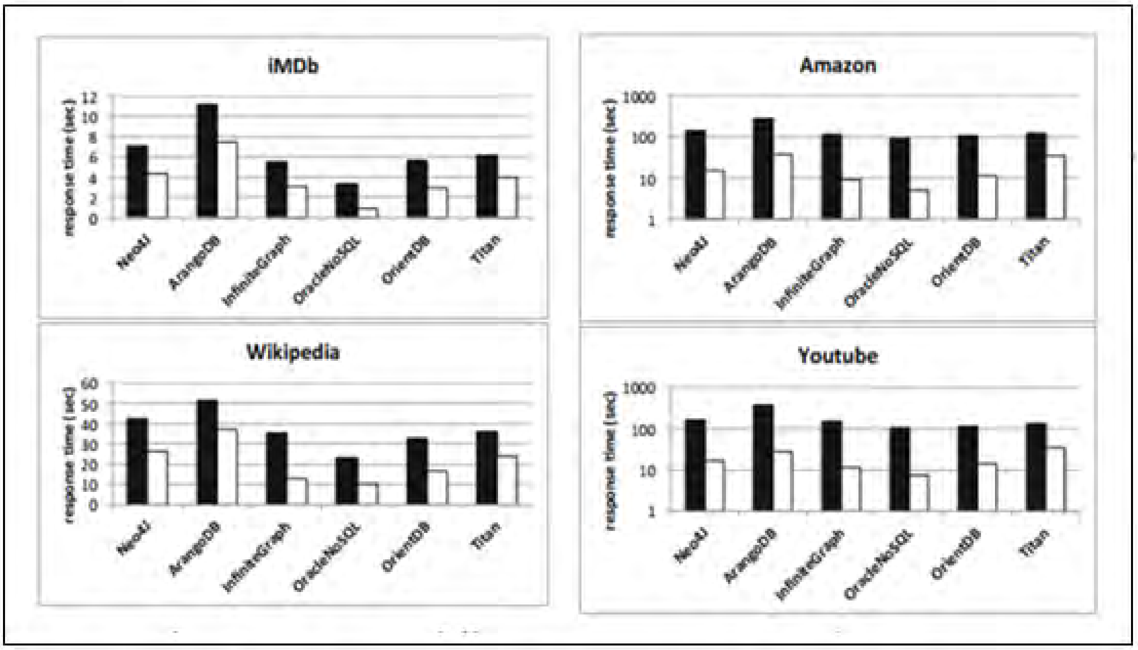
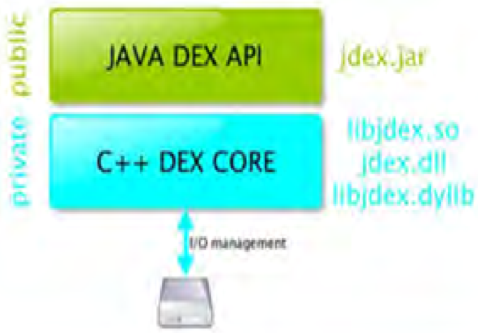


Figure 1: (Results from R. De Virgilio et;el Research) . Performance on Verious data sources

[Jasper 2011] done a survey on graph database and gives some important points in the form of comparison . graph databases are one of four major categories of NoSQL databases. Also, seven products are listed in the category of graph store: Neo4J, Infinite Graph, DEX, Info Grid, Hyper GraphDB, Trinity and AllegroGraph. hypergraph is the most generic form of graphs, a graph database supporting hypergraph should also support property graphs theoretically. Some of them as shown in figure 1, figure 2 and figure 3



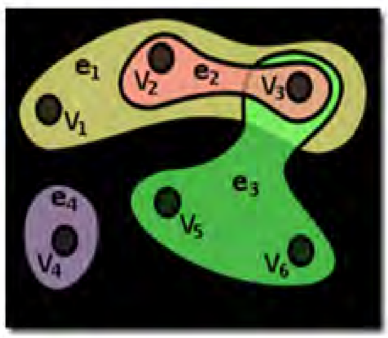


Figure 2(1): Architecture of DEX Figure 2(2): Hypergraph

[P. Atzeni 2008] ,The MIDST project has the goal of developing tools for the translation of database schemas and instances from a model to another. The approach is based on a "metamodel" notion: data models are described with reference to a small set of metaconstructs, and translations are specified on metaconstructs as well, so that they are reusable. The project started in its present form in 2003, as a follow-up of a previous project carried out in 1991-1997. In MIDST, new techniques have been proposed for database translations from a model to another, for example from object oriented to SQL or from SQL to XML schema descriptions

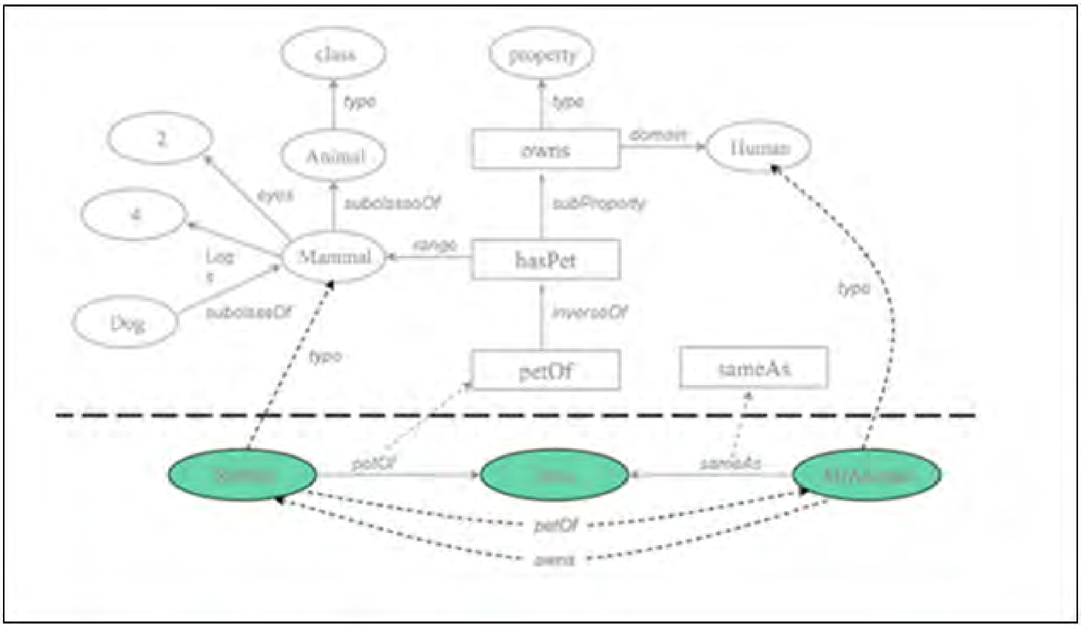


Figure 3: Example of RDF graph

[M. Hert, and etal;12] His research had shown that Semantic Web technologies are useful beyond the Web, especially if data from different sources has to be exchanged or integrated. Many mapping languages and approaches were explored leading to the ongoing standardization effort of the World Wide Web Consortium (W3C) carried out in the RDB2RDF Working Group (WG). Also he had classification proposes four categories of mapping languages: direct mapping, read-only general-purpose mapping, read write general-purpose mapping, and special-purpose mapping.

[Robinson 2013] Current approaches mainly rely on best practices and guidelines based on typical design patterns, published by practitioners in blogs [5] He Said that Unlike hypergraphs, the concept of a single directed edge still exists. It’s just that the relationships, as represented by a directed edge, can be associated with other incoming/outgoing relationships.

[Batini et al. 1992] introduce a logical design for the Network model [Radhakrishnan Delhibabu and et al 2015] that follows a sparse-like strategy for mapping an ER schema into a Network schema In [S. Chaudhuri and et al 1995]], the author gathers different design patterns for various NoSQL data stores, including one for graph databases called application side joins. This design pattern is based on the join operations that need to be performed over the database. Conversely, we do not make any assumption on the way in which the database under development is accessed and our approach relies only on the knowledge of conceptual constraints that can be defined with the ER model. Moreover, it provides a system-independent intermediate representation that makes it suitable for any GDBMS.

Juan Reutter[7] contribution is to show elementary tight bounds for the containment problem for regular queries. Specifically, we show that this problem is 2Expspace –complete. His results show that regular queries achieve a good balance between expressiveness and complexity, and constitute a well-behaved class . He also note that Checking query containment is crucial in several contexts, such as query optimization, view-based query answering, querying incomplete databases, and implications of dependencies [S. Chaudhuri 1995 ,R. Fagin,, Katsov 2012, M. Friedman 1999 , T. Imielinski 1984, Marcus Paradies 2015].

[Yufan Liu 2014] had shown in his survey in 2014 that Graph data can be stored in a relational table with two columns. Labels and attributes of nodes and edges can be managed separately in other tables and referred by foreign keys. As the dominated database management system form the 60s, RDBMS has its advantages to store graph data: well developed indexing system, sophisticated transaction support, the query language: SQL is a long-established standard and has fast learning cycle. Also he shows traversal performance comparison between relational database and graph database as shown in Table 1 and Table 2 and observe that Another big issue is more and more graph databases now start supporting distribute storage as the increasing size of graph.

Table1: Traversal performance comparison between relational database and graph database

Depth	RDBMS Execution time (seconds)	Neo4j Execution time (seconds)	Records returned
2	0.016	0.01	2500
3	30.267	0.168	125000
4	1543.505	1.359	600000
5	Not finished	2.132	800000

Table 2: Comparison of supported features.

Graph database	Costumed Index	Key index	Transaction
Neo4j	True	True	Fully
Orient DB	True	True	Fully
DEX	False	True	Partial
Titan	True	True	Fully
Infinite Graph	True	True	Fully
Allegro Graph	False	False	Fully

[N.R. Prasanth and K. Arul 2014] He found that relational database is not suitable for web applications, computer networks, geographical structure. Also there is a problem of complex join operations. To overcome some such problem He had proposed Graph Database Management Systems which provide an excellent method to store the data. In His proposed system converted into the graph database to provide efficiency of query answering. He shows that This approach supports the translation of conjunctive SQL queries over the source into graph traversal operations over the target. At the last from the result he conclude that The graph database is more effective form of database system. The time taken to add manage and update a query in database gets very simple in graph database. It also requires only less coding to perform such operation when comparing to relational database.

3 Motivation

Motivations for this approach include simplicity of design, performance for huge graphs when we enjoy the flexibility. Relational set of elements and data create the topological structure of the graph. And also found that it provides higher data complexity. There are general query languages for different graph databases, while there are also some graph database specified query languages. Here we will try to implement graph database for reducing the complexity in tabular database with existing query language.

3.1. Short term and Long term Goals:

(1) Makes it easier to express many kinds of data that require significant kludging to fit in a relational database.

(2) Implementation of Graphical database suited to the irregular, complex data involved in mapping the "real world."

(3) comparing the performance of graph algorithms between distribute graph processing platforms on large-scale datasets.

References

[1] Alexander Wendt, Benjamin Dönz, and et el;Evaluation of Database Technologies for Usage in Dynamic Data Models - A Comparison of Relational, Document Oriented and Graph Oriented Data Models. ICAART (1) 2012: 219-224
[2] AllegroGraph, http://www.franz.com/agraph/allegrograph
[3] Batini, C., Ceri, S., Navathe, S.B.: Conceptual Database Design: An EntityRelationship Approach. Benjamin/Cummings (1992)
[4] Billy Pik Lik Lau,and et el A review on dependence graph in social reasoning mechanism. Artif. Intell. Rev. 43(2): 229-242 (2015)
[5] De Virgilio, R., Maccioni, A., Torlone, R.: Converting relational to graph databases. In: SIGMOD Workshops - GRADES (2013)
[6] De Virgilio, R., Maccioni, A., Torlone, R.: R2G: A tool for migrating relations to graphs. In: EDBT (2014)
[7] Jasper 'A Survey on Graph Databases 'Jasper Tech Blogs 2011-11-25 jasperpeilee.wordpress.com.
[8] Jennifer Sleeman, Tim Finin, Anupam Joshi:Entity Type Recognition for Heterogeneous Semantic Graphs. AI Magazine 36(1): 75-86 (2015)
[9] Katsov, I.: NoSQL data modeling techniques (2012), http://highlyscalable. wordpress.com/2012/03/01/nosql-data-modeling-techniques
[10] Luis Enrique Sucar: Probabilistic Graphical Models - Principles and Applications. Advances in Computer Vision and Pattern Recognition, Springer 2015, ISBN 978-1-4471-6698-6, pp. 3-246
[11] M. Friedman, A. Y. Levy, T. D. Millstein. Navigational plans For data integration. In AAAI/IAAI 1999, pages 67–73.
[12] M. Hert, G. Reif, and H. C. Gall. A comparison of rdb-to-rdf mapping languages. In I-SEMANTICS , pages 25 {32, 201}

[13] Marcus Paradies, Wolfgang Lehner, Christof Bornhövd:GRAPHITE: an extensible graph traversal framework for relational database management systems. SSDBM 2015: 29
[14] N.R. Prasanth and K. Arul "Converting Employee Relational Database into Graph Database" , ISSN 1990-9233,DOI: 10.5829/idosi.mejsr.2014.22.11.21524,Middle-East Journal of Scientific Research 22 (11): 1618-1621, 2014
[15] P. Atzeni, P. Cappellari, R. Torlone, P. A. Bernstein, and G. Gianforme. Model-independent schema translation. VLDB J., 17(6):1347{1370, 2008
[16] P. Barceló, L. Libkin, A. Lin, P. Wood. Expressive languages for path queries over graph-structured data. In ACM TODS 38(4), 2012
[17] P. Barceló. Querying graph databases. In PODS 2013, pages 175–188
[18] R. Fagin, M. Y. Vardi. The theory of data dependencies - An overview. In
[19] Radhakrishnan Delhibabu, Andreas Behrend:A new rational algorithm for view updating in relational databases. Appl. Intell. 42(3): 466-480 (2015)
[20] Robin Hewitt, Alberto Gobbi, Man-Ling Lee:A Searching and Reporting System for Relational Databases Using a Graph-Based Metadata Representation. Journal of Chemical Information and Modeling 45(4): 863-869 (2005)
[21] Robinson, I.: Designing and building a graph database application with neo4j. In: Graph Connect (2013)
[22] S. Chaudhuri, R. Krishnamurthy, S. Potamianos, K. Shim. Optimizing queries with mate-rialized views. In ICDE 1995, pages 190-200
[23] T. Imielinski, W. Lipski Jr. Incomplete information in relational databases. J. of the ACM 31(4), pages 761–791, 1984
[24] Taylor, R.W., Frank, R.L.: Codasyl data-base management systems. ACM Comput. Surv. 8(1), 67–103 (1976)
[25] Yufan Liu 'A survey of persistant graph database' M.S., Kent State University, 2014