Middle-East Journal of Scientific Research 22 (11): 1618-1621, 2014
ISSN 1990-9233
© IDOSI Publications, 2014
DOI: 10.5829/idosi.mejsr.2014.22.11.21524

Converting Employee Relational Database into Graph Database

N.R. Prasanth and K. Arul

Saveetha University Saveetha Nagar, Thandalam, Chennai-602105, India

Abstract: Graph Database Management Systems provide an effective and efficient solution to data storage in current scenarios where data are more and more connected, graph models are widely used and systems need to scale to large data sets. In this framework, the conversion of the persistent layer of an application from a relational to a graph data store can be convenient but it is usually an hard task for database administrators. In this proposal a methodology to convert a relational to a graph database by exploiting the schema and the constraints of the source. The approach supports the translation of conjunctive SQL queries over the source into graph traversal operations over the target. The experimental results are provided to show the feasibility of the solution and the efficiency of query answering over the target database.

Key words: RDBMS · GDBMS · Cipher

INTRODUCTION

There are various application that requires large data storage for storing their appropriate data. To do so the database management system (DBMS) is used. We use a software called SQL to create a database. Sometimes it is necessary to relate the two or more table in a database. To do so we use relational database system (RDBMS). However this relational database is not suitable for web applications, computer networks, geographical structure etc.., moreover in these highly connected data applications requires complex join operation which can make typical operation on this kind of data inefficient application hard to scale. To overcome this problem we use graph database management system (GDBMS). In GDMS data are natively stored as graph and queries are expressed as graph traversal operation. This allows application to scale very large graph based data sets. In addition GDMS do not rely on any schema they provide more flexible solution in scenarios where the organization of data evolves rapidly. By using graph database rather than using the relational database is more beneficial. In graph database it follows a naive approach where tuples are mapped to nodes and foreign key is mapped into edges. In this paper the employee relational database is converted into the graph database for high performance. Specifically the relational database query is converted into the graph database query. The general graph model and generic query language for graph structures.

System Analysis

Existing System: Arelational databaseis adatabase that has a collection oftablesof data items, all of which is formally described and organized according to therelational model. Column to column relationship requires primary key. Row to row relationship requires foreign key

Proposed System: Graph Database Management Systems provide an excellent method to store the data.The graph models are widely used where the systems need to scale to large data sets.Here the relational database is converted into the graph database to provide efficiency of query answering. The approach supports the translation of conjunctive SQL queries over the source into graph traversal operations over the target.

Requirement Specification

Introduction: The requirements specification is a technical specification of requirements for the software products. It is the first step in the requirements analysis process it lists the requirements of a particular software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative perspective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target audience and its user

Corresponding Author: N.R. Prasanth, Saveetha University Saveetha Nagar, Thandalam, Chennai-602105, India.

interface, hardware and software requirements. It defines how the client, team and audience see the project and its functionality.

Hardware and Software Specification
Hardware Requirements

- Hard Disk: 80 GB and above.
- RAM: 2 GB DDR3
- Processor: i3 and above.
- Software Requirements
- Mysql(Rdbms)
- Neo4j
- JVM
- 64 bit operating system

Technology Used

Cypher: Cypher is a declarative graph query language that allows for expressive querying and updating of graph store. Cypher is a relatively simple and powerful language for expressing complex queries. This allows to focus on domain instead of getting lost in database. This approach makes query optimization and implementation detail instead of burdening the user with it and requiring her to update all traversals just because the physical database structure has changed. Its queries are built using various clauses. The clauses are chained together and they feed intermediate result sets between each other.

System Design

Architecture Diagram: It gives the basic architecture of the developing project

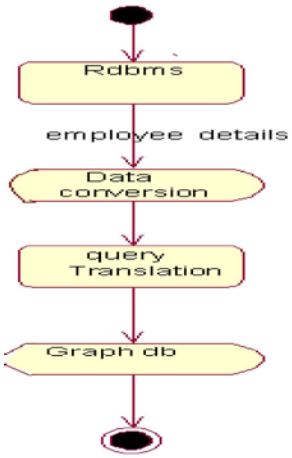

Fig. 4.1: Architecture Diagram

Syetem Implementation
Module Explanation

Rdbms: A type ofdatabase management system (DBMS)thatstoresdatain the form of relatedtables. Relational databases are powerful because they require few assumptions about how data is related or how it will be extracted from the database. As a result, the same database can be viewed in many different ways. An important feature of relationalsystemsis that a single database can be spread across several tables. This differs from flat-file databases, in which each database is self-contained in a single table. Almost all full-scale database systems are RDBMS's. Small database systems, however, use other designs that provide less flexibility in posingqueries.

Data Conversion: In this module the conversion of relational database to graph database is performed. The data in relational database in converted into graph database conversely, in our approach we try to aggregate values of different tuples in the same node to speed-up traversal operations. The basic idea is to try to store in the same node of g data values that are likely to be retrieved together in the evaluation of queries.

Query Translation: Our mechanism for translating conjunctive (that is, select-join-projection) queries, expressed in SQL, into path traversal operations over the graph database exploits the schema of the source relational. For the sake of simplicity, we consider an intermediate step in which we map the SQL query in a graph-based internal structure, that we call query template (QT for short). Basically, a QT denotes all the sub-graphs of the target graph database that include the result of the query. A QT is then translated into a path traversal query.

Graph DB: A graph database, also called a graph-oriented database, is a type of NoSQLdatabase that usesgraph theoryto store, map and query relationships. A graph database is essentially a collection of nodes and edges. Each node represents an entity (such as a person or business) and each edge represents a connection or relationship between two nodes. Every node in a graph database is defined by aunique identifier, a set of outgoing edges and/or incoming edges and a set of properties expressed askey/value pairs. Each edge is defined by a unique identifier, a starting-place and/or ending-place node and a set of properties. Graph databases are well-suited for analyzing interconnections, which is why there has been a lot of interest in using

graph databases tomine datafromsocial media. Graph databases are also useful for working with data in business disciplines that involve complex relationships anddynamicschema.

Coding and Testing

Coding: Once the design aspect of the system is finalizes the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system.

Coding Standards: Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blue print developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. *6.3 Test Procedure*

System Testing: Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walk through. Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

Vii Related Works: The need to convert relational data into graph modeled data [1-7] emerged particularly with the advent of Linked Open Data (LOD) [8-9] since many organizations needed to make available their information, usually stored in relational databases, on the Web using RDF. For this reason, several solutions have been proposed to support the translation of relational data into RDF. Some of them focus on mapping the source schema into an ontology [5-10-13] and rely on a naive transformation technique in which every relational attribute becomes an RDF predicate and every relational values becomes an RDF literal. Other approaches, such as R2O [11] and D2RQ [3], are based on a declarative language that allows the specification of the map between relational data and RDF. As shown in [8], they all provide rather specific solutions and do not fulfill all the requirements identified by the RDB2RDF (http://www.w3.org/TR/2012/CR-rdb-direct-mapping-20120223/) Working Group of the W3C. Inspired by draft methods defined by the W3C, the authors in [13] provide a formal solution where relational databases are directly mapped to RDF and OWL trying to preserve the semantics of information in the transformation. All of those proposals focus on mapping relational databases to Semantic Web stores, a problem that is more specific than converting relational to general, graph databases, which is our concern. On the other hand, some approaches have been proposed to the general problem of database translation between different data models (e.g., [2]) but, to the best of our knowledge, there is no work that tackles specifically the problem of migrating data and queries from a relational to a graph database management system. Actually, existing GDBMSs are usually equipped with facilities for importing data from a relational database, but they all rely on naive techniques in which, basically, each tuple is mapped to a node and foreign keys are mapped to edges. This approach however does not fully exploit the capabilities of GDBMSs to represent graph-shaped the information. Moreover, there is no support to query translation in these systems. Finally, it should be mentioned that some works have done on the problem of translating SPARQL queries to SQL to support a relational implementation of RDF databases [13-14]. But, this is different from the problem addressed in this paper.

CONCLUSION

In this proposed system the employee relational database is converted into the graph database system.The graph database is considered to be the one of the emerging technology. The graph database is more

effective form of database system. The time taken to add, manage and update a query in database gets very simplein graph database.It also requires only less coding to perform such operation when comparing to relational database. The comparison of relational database and graph database is shown in chart form and the above mentioned things are verified. In future works we intend to refine the technique proposed in this paper to obtain a more compact target database.

REFERENCE

1. Angles, R. and C. Gutierrez, 2008. Survey of graph database models. ACM Comput. Surv., 40: 1.
2. Atzeni, P., P. Cappellari, R. Torlone, P.A. Bernstein and G. Gianforme, 2008. Model-independent schema translation. VLDB J., 17(6): 1347-1370.
3. Bizer, C., 2003. D2r map - a database to rdf mapping language. In WWW (Posters).
4. Bizer, C. and A. Schultz, 2009. The berlin sparql benchmark. Int. J. Semantic Web Inf. Syst., 5(2): 1-24.
5. Cerbah, F., 2008. Learning highly structured semantic repositories from relational databases. In ESWC, pp: 777-781.
6. Coman, J. and A.C. Weaver, 2010. A framework for evaluating database keyword search strategies. In CIKM, pp: 729-738.
7. *et al*, S. B., 2011. Keyword search over relational databases, a metadata approach. In SIGMOD, pp: 565-576.
8. Hert, M., G. Reif and H.C. Gall, 2011. A comparison of rdb-to-rdf mapping languages. In I-SEMANTICS, pp: 25-32.
9. Holzschuher, F. and R. Peinl, 2013. Performance of graph query languages - comparison of cypher, gremlin and native access in neo4j. In EDBT/ICDT Workshops, pp: 195-204.
10. Hu, W. and Y. Qu, 2007. Discovering simple mappings between relational database schemas and ontologies. In ISWC/ASWC, pp: 225-238.
11. Rodrguez, J.B. and A. Gomez-Perez, 2006. Upgrading relational legacy data to the semantic web. In WWW, pp: 1069-1070.
12. Rodriguez, M.A. and P. Neubauer, 2010. Constructions from dots and lines. CoRR, abs/1006., 2361.
13. Sequeda, J., M. Arenas and D.P. Miranker, 2012. On directly mapping relational databases to rdf and owl. In WWW, pp: 649-658.
14. Wood, P.T., 2012. Query languages for graph databases. SIGMOD Record, 41(1): 50-60.