An Efficient Distributed Approach for Querying And Retrieving in NOSQL Graph Database

Subin Babu
Dept of Computer Science and Engineering, Mount Zion College of Engineering, Kadammanitta, Kerala, India

Dr. Smita C Thomas
Dept of Computer Science and Engineering, Mount Zion College of Engineering, Kadammanitta, Kerala, India

Reshma Suku
Dept of Computer Science and Engineering, Mount Zion College of Engineering, Kadammanitta, Kerala, India

***Abstract*— With the development of NoSQL stores, large amount of data can be stored in a highly-available manner through CRUD operations, i.e., Create, Read, Update, and Delete. These operations in NoSQL storage systems provide high throughput and low latency. NoSQL systems are generally more efficient than relational databases where a system administrator has to grapple with system logs by parsing flat files that store these operations. The main aim of our project is to develop a scalable and efficient storage and querying of large collections of graphs serialized as database. In this project we proposed a distributed indices that enable extremely fast querying of provenance graphs along causality lines. We have also designed an algorithm for building these indices, as well as updating them when adding new triples to an already indexed graph. Finally, we have implemented our approach and measured the query performance of our indices using a cluster in the Distributed cloud. This work focuses on two things, one is about the scalability of NoSQL graph management and the other is about minimizing causality lines in latency of graph traversals. In our project, metadata system intended for NoSQL data stores functions as a component of the data store and leverages the underlying NoSQL functionalities to deliver its services. Distributed Querying approaches have to address three major challenges. First, it must collect distributed dataset without imposing overhead on the CRUD operations arriving from clients side. Second, it allows a user to specify which metadata is collected and how to use it. Finally, Distributed query approaches must scale with: i) the size of the cluster, ii) the size of the data being stored, iii) the rate of incoming CRUD operations, and iv) the size of the metadata itself.**

Keywords—Data Mining, NoSQL, Query, Database

I. INTRODUCTION

Graph databases are part of the so-called NoSQL DBMS ecosystem, in which the information is not organized by strictly following the relational model. The structure of graph databases is well-suited for representing some types of relationships within the data, and their potential for distribution makes them appealing for applications requiring large-scale data storage and massively parallel data processing. Natural example applications of such database systems are social network analysis or the storage and querying of the Semantic Web. The main focus of our research in this work is on the efficient and scalable storage and querying of large collections of provenance graphs serialized as RDF graphs in an database. With the development of user-friendly and powerful tools, such as scientific workflow management systems we are able to design and repeatedly execute workflows with different input datasets and varying input parameters with just a few mouse clicks. Each workflow execution generates a provenance graph that will be stored and queried on different occasions. A single provenance graph is readily manageable as its size is correlated with the workflow size and even workflows with many hundreds of processes produce a relatively small metadata footprint that fits into main memory of a single machine. The challenge arises when large number of provenance graphs constitute a provenance dataset. Managing large and constantly growing provenance datasets on a single machine eventually fails and we turn to distributed data management solutions. We design such a solution for large provenance datasets, while we deploy and evaluate our solution on a small cluster of commodity machines, Dataset is readily available in cloud environments suggesting virtually unlimited elasticity.

The main contributions of this work are: (i) novel storage and indexing schemes for RDF data in distributed column-oriented database that are suitable for provenance datasets, and (ii) novel and efficient querying algorithms to evaluate SPARQL queries in distributed column-oriented database that are optimized to make use of bitmap indices and numeric values instead of triples.

II. DEFINITIONS

A. Big Data

It is huge, large or voluminous data, information or the relevant statistics acquired by the large organizations and ventures. Many software and data storage created and prepared as it is difficult to compute the big data manually. It is used to discover patterns and trends and make decisions related to human behaviour and interaction technology.

B. Data Mining

Data Mining is a technique to extract important and vital information and knowledge from a huge set/libraries of data. It derives insight by carefully extracting, reviewing, and processing the huge data to find out pattern and co-relations which can be important for the business.

C. Knowledge Discovery in Databases

The term Knowledge Discovery in Databases, or KDD for short, refers to the broad process of finding knowledge in data, and emphasizes the "high-level" application of particular data mining methods. It is of interest to researchers in machine learning, pattern recognition, databases, statistics, artificial

intelligence, knowledge acquisition for expert systems, and data visualization. The unifying goal of the KDD process is to extract knowledge from data in the context of large databases.

It does this by using data mining methods (algorithms) to extract (identify) what is deemed knowledge, according to the specifications of measures and thresholds, using a database along with any required pre-processing, sub-sampling, and transformations of that database.

D. Abbreviations and Acronyms

SWfMS Scientific Workflows Management Systems
RDBMS Relational Database Management System
SaaS Software as a Service
PaaS Platform as a Service
IaaS Infrastructure as a Service

III CONVOLUTIONAL NEURAL NETWORK ALGORITHM AND ITS IMPROVEMENTS

A. Related Works

According to Daren Fadolalkarim and Elisa Bertino [1],they proposed A-PANDDE which is a system for detecting insiders attempts to exfiltrate data from the database. The system mainly tracks such attempts at the file system level when the data is piped from the database to files. It works in two phases: training phase in which the system collects provence records related to files containing data from the database and builds profiles based on these records, and detection phase in which profiles are used to detect suspicious actions. Our system detects advanced attacks and collects more information about the data being exfiltrated; e.g., the syntactic features of the data, and the amount of data. Also, it tracks the frequencies of users' actions and warns security admins when a user exceeds the time threshold. In our system, we consider multiple administrators with separation of duty policy in order to protect against abusing the profiles by a malicious security administrator. An outsider or insider can break the security properties to gain access to user profiles. Nevertheless, profiles can be protected by tracking accesses to the files that store the user profiles and isolating them on secure storage.

According to Sunitha Ramanujam, Anubha Gupta and Latifur Khan [2], This paper extends the R2D framework by including the ability to represent complex provenance information available in RDF stores, through the process of reification, in a relational format accessible through traditional relational tools. Specifically, this paper presents a methodology for relationalizing provenance information stored in complex RDF structures called blank nodes. A JDBC interface aimed at accomplishing this goal through a mapping between RDF reification constructs and their equivalent relational counterparts was presented. Graphs highlighting response times for map file generation and query processing obtained during performance evaluation of the framework using databases of various sizes, both with and without reification data, were also included. Future directions for R2D include improving the normalization process for complex reification blank node.

According to Paolo Missier, Simon Woodman, Hugo Hiden and Paul Watson [3], they have focused specifically on workflow-based e-science, and on a scenario where attempts to reproduce earlier results translates into new runs of the same workflow at a later time. We assume that the workflow is still executable, but it may produce unexpected results or may have been made dysfunctional by changes in its system environment, or by uncontrolled evolution of the workflow itself or of its input data.

According to Szabolcs Rozsnyai, Aleksander Slominski, Yurdaer Doganata [4], introduced a cloud-based business provenance solution as a technology to track a consistent and accurate history, the relationships and derivations between artifacts in order to be able to monitoring and analyze business processes. This opens wide range of opportunities such as process discovery, verification and process improvement.

However, one of the major issues with tracking arbitrary artifacts for provenance purposes is that the data can grow to tremendous amounts, which makes it a very resource consuming effort to process, store, organize, retrieve and analyze the data. Therefore the goal of this paper was to introduce and discuss the components of a large-scale distributed provenance solution built around HBase/Hadoop that overcomes these problems. It demonstrates how cloud storage specific advantages can be utilized and demonstrated how certain related problems, such as the representation of relationships by graphs, affecting the capabilities of a business provenance can be solved. The work in this paper is part of a larger research effort aiming at developing and evaluating a comprehensive set of technologies relating to business provenances and its applications covering the full life-cycle starting from data integration, to data management and analytics.

According to Philippe Cudr´e-Mauroux and Paul Groth [5], they presented the first attempt to translate theoretical insight from the database provenance literature into a high-performance triple store. Our techniques enable not only simple tracing of lineage for query results, but also considers fine-grained multilevel provenance and allow us to tailor the query execution with provenance information. We introduced two storage models and five query execution strategies for supporting provenance in Linked Data management systems. From their experiment, it says that the more data we have to process the slower queries are executed and the more selective is the provenance query the more we gain in performance. Less selective workload queries are more sensitive to those aspects than queries that allow to early prune intermediate results. The more advanced query execution strategies that take advantage of the selectivity of the provenance information are more sensitive to the number elements returned by the provenance query. A user of a system like TripleProv can easily restrict what data to be used in a query by providing a SPARQL query scoping the data provenance. Additionally, the user will receive detailed information about exact pieces of data were used to produce the results. Such a provenance trace can be used to compute the quality of the results or to (partially) invalidate the results in case some parts of the data collection turn out to be

incorrect. Moreover, a provenance trace can be leveraged to partially reevaluate the query in case new data arrives, i.e., we can reevaluate only the part that is influenced by the new data since we know what data exactly has been used in the query.

B. Existing System

Science aims to produce knowledge from reproducible experiments. Even for in silico experiments, it is not always a trivial task due to the lack of activities management. This is the reason why reproducibility and provenance are closely related. When dealing specifically with provenance in Scientific Workflows Management Systems (SWfMS), studies have collaborated on suitable solutions such as a taxonomy definition to classify those systems and the understanding of the workflow life cycle. Not only SQL systems have emerged as an alternative to traditional and well established Relational Database Management Systems (RDBMS). NoSQL database systems are scalable solutions that enable distributed processing and high availability. In addition, these systems consider a flexible schema that may deal with both structured and unstructured data. The provisioning infrastructure is hosted in geographically distributed data-centers that are accessible via the Internet. This model represents a convergence of several paradigms of computing, such as virtualization, distributed systems and service orientation. It allows the improvement of fault tolerance methods, high availability for data and programs and, especially, elasticity. Elasticity is the ability to provision resources quickly and sometimes automatically, either to increase or to decrease the number of computational resources. Cloud computing users utilize the resources on demand, in accordance with their service provider. There are two models that describe the cloud environment under different aspects: deployment models and service models. The association between these models is given by the access level to the underlying infrastructure, which is presented to the customer. Service models define an architectural standard for cloud-based solutions. These models are defined as Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS).

In existing system, we cluster, group, and analyze data without SQL queries. It takes time to process the data at the beginning, though the processing speed gradually increases. The existing system becomes disadvantageous when there are more number of clusters. The more the number of clusters, the more it takes time to process them. Our focus is on bringing a change here. Another limitation of the existing system is, when there are more number of data rows we cannot make use of the system directly. If a machine learning algorithm is developed data processing becomes more accurate.

C. Proposed System

The main aim of enhancing this project is developing an improved version of a machine learning algorithm like CNN (Convolutional Neural Network). Currently CNN cannot be applied directly to data processing and analysis, but it should be modified to optimal CNN algorithm, which we are supposed to develop. The basic intention is analyzing data using machine learning algorithm. At present, the data is divided into different clusters and analyzing them. Data analysis should be made stronger using the optimal algorithm CNN. This can replace the existing SVM (Support Vector Machine) thereby creating more data clusters thus making the analysis of data a speedy process. Developing optimal CNN algorithm can graphically display data without any database. The main aim is to increase the speed of processing and analyzing data.

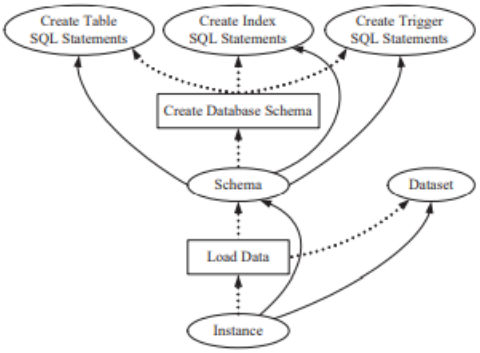

Fig 1: Proposed System Architecture

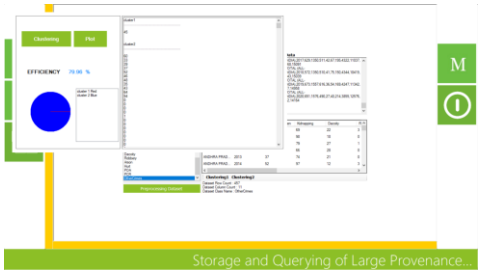

Fig 2: Clustering data based on column and graph generation

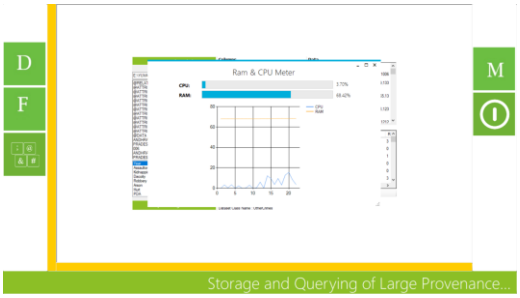

Fig 3: Graph based memory analysis

D.Optimal Convolutional Neural Network Algorithm

The steps involved in the application of the optimal CNN algorithm are, we make a large number of clusters, and regroup them according to the relationship between them. These are called convolution layers. These convolution layers are again re clustered into dense layers. In this manner we create optimal number of clusters and process and analyze them. Usually we access data from database using SQL. SQL is a common structure which helps access data stored in varied manners. Optimal CNN is a non-SQL algorithm. It has

a mechanism which can cluster, group, process, and analyze data without a common structure.

III. RESULT ANALYSIS

In data aggregation module, the aggregation is done by using the aggregation query functions such as sum, count, difference. The execution of these queries can be done faster than the existing Pull based mechanism. The aggregate value of the data item is obtained from the data base using the aggregate queries. In the proposed model, the Server can view the active and inactive systems and can also find the participating systems/clients in the network. This module also includes another functionality called system monitoring to find out the CPU status, Memory status and Hard disk status of the system. In Pull based mechanism, the server communicates with each client, get the data and perform data aggregation. Then the server contacts the next client and performs data aggregation with the previous aggregated value and so on. This process works in a circular manner. But this process takes much time and lot of data aggregations. In pull based mechanism data sources send messages to the client only when the client makes a request. In Push based mechanism, the server send update messages to clients on their own. Clients send data directly to the server when there is an updation. We assume push based mechanism for data transfer between data sources and clients. For scalable handling of push based data dissemination, network of data aggregators are used. In such network of data aggregators, data refreshes occur from data sources to the clients through one or more data aggregators. Server needs just to add this value to the server's aggregated value. This mechanism is suitable for real time calculation. Here refresh messages is send from the client to the server. In case of frequent updation of data there will be lot of messages of different clients.

IV. CONCLUSION

In this work we proposed an improved version of a machine learning algorithm like CNN (Convolutional Neural Network). The basic intention is analyzing data using machine learning algorithm. The main aim is to increase the speed of processing and analyzing data. Usually we access data from database using SQL. SQL is a common structure which helps access data stored in varied manners. Optimal CNN is a non-SQL algorithm. It has a mechanism which can cluster, group, process, and analyze data without a common structure.

REFERENCES

[1] D. Fadolalkarim, E. Bertino, "A-PANDDE: Advanced Provenance- based ANomaly Detection of Data Exfiltration," Comput. Security, vol. 84, pp. 276-287, 2019.
[2] S. Ramanujam, A. Gupta, L. Khan, S. Seida, B. M. Thuraisingham, "Relationalization of provenance data in complex RDF reification nodes," Electronic Commerce Research, vol. 10, no. 3-4, pp. 389- 421, 2010.
[3] P. Missier, S. Woodman, H. Hiden, and P. Watson, "Provenance and data differencing for workflow reproducibility analysis," Concurrency and Computation: Practice and Experience, vol. 28, no. 4, pp. 995- 1015, 2016.
[4] S. Rozsnyai, A. Slominski and Y. Doganata, "Large-Scale Distributed Storage System for Business Provenance," IEEE International Con- ference on Cloud Computing, 2011, pp. 516-524.
[5] Morshedzadeh, J. Oscarsson, A. Ng, M. Jeusfeld, and J. Sillanpaa, "Product lifecycle management with provenance management and virtual models: an industrial use-case study," Procedia CIRP, vol. 72, 2018, pp. 1190-1195.
[6] "A new approach to harnessing manufacturing data," report, Cisco, 2018.