Computers and Electrical Engineering 000 (2018) 1–15

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Data allocation optimization for query processing in graph databases using Lucene☆

Anita Brigit Mathew

Department of Computer Science and Engineering, NIT, Calicut, India

ARTICLE INFO

Article history:
Received 25 August 2017
Revised 16 January 2018
Accepted 17 January 2018
Available online xxx

Keywords:
Big Data
Query retrieval
Graph NoSQL databases
Data allocation
Best Fit Decreasing
Ant Colony Optimization

ABSTRACT

Methodological handling of queries is a crucial requirement in social networks connected to a graph NoSQL database that incorporates massive amounts of data. The massive data need to be partitioned across numerous nodes so that the queries when executed can be retrieved from a parallel structure. A novel storage mechanism for effective query processing must to be established in graph databases for minimizing time overhead. This paper proposes a metaheuristic algorithm for partitioning of graph database across nodes by placement of all related information on same or adjacent nodes. The graph database allocation problem is proved to be NP-Hard. A metaheuristic algorithm comprising of Best Fit Decreasing with Ant Colony Optimization is proposed for data allocation in a distributed architecture of graph NoSQL databases. Lucene index is applied on proposed allocation for faster query processing. The proposed algorithm with Lucene is evaluated based on simulation results obtained from different heuristics available in literature.

1. Introduction

In today's big data era, storage of massive data sets is a serious issue to be considered [1]. NoSQL graph databases, provide a pathway for unstructured data to be stored in structured form offering significant advantage in processing. It meets the needs of data holders based on the growing demand of storage. This data is stored in a distributed setup in graph NoSQL database. The distributed data in graph NoSQL databases allows load balancing based on their capacities. The data distributed across nodes is accessed by the controller at the time of query processing. Query process invokes nodes based on the query request made by the user. Hence there is a requirement that the data locality of the nodes to be geographically near so that query retrieval can be made faster. The data stored in each of these nodes represent relationships because the data dealt here is from social sites. Hence there is a need for efficient storage of related and replicated data for effective query processing.

In relational data model, the theory of sharing has a long way to go, and different procedures have been examined for partitioning tabular data into shards and assign these shards to other nodes [2]. Many other NoSQL databases like key-value store, column family, and document databases use range-based fragmentation for distribution of data [3]. Apart from conventional methodologies of query answering, a flexible query processing provides mechanisms for the intellectual answering of user queries. This provides user to retrieve any type of queries whether it is range or related. Sometimes the data to be searched may be present in database but due to lack of efficient related storage, the query retrieval process fails to retrieve

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. J. D. Peter.

E-mail address: anita_p120024cs@nitc.ac.in

https://doi.org/10.1016/j.compeleceng.2018.01.022
0045-7906/

the data. Internally a flexible query processing system revise query process at time of failure and return results in an informative manner compared to empty results. Efficient placement of all related terms with an index structure leads to flexible query processing without any failure in retrieval.

The rest of the paper is structured as follows, Section 2 Background, gives a review on related works of existing data storage techniques available in the literature. Section 3 gives a sketch of graph database architecture. Section 4 illustrates the graph database allocation problem with replication is a variation to Bin Packing Problem (BPP) which is a NP-Hard problem. Graph database allocation problem is solved using 0 1 Integer Linear Programming(ILP). To solve the hard allocation strategy by taking replication and relation factors into consideration a metaheuristic based Ant Colony Optimization BFD integration is modeled, further an index named Lucene is also implemented and computational results are analyzed in Section 5. Section 6 presents the experimental setup and implementation details. Section 7 concludes the work with the conclusion and future scope.

2. Background

Many researchers have studied the problems of query retrieval over relational databases. Ning et al. [1] develop data allocation scheme based on the prior knowledge of type of related data and grouping this data by encapsulation schemes available. This form of data allocation try to store all related or shared data to a single node. If a raw data could not be inserted into a single node, data is compressed and stored on the same node. This strategy used can cause loss of data during data retrieval process. Sun et al. [4] discuss a data allocation strategy by balancing the data across various nodes. They use a next fit technique of data distribution. But this technique failed to allocate duplicate data in a distributed environment and relations were also not considered. How to query for keywords in a distributed network which is connected to data sources are covered by Kavitha et al. [5] The data allocation scheme pre-owned by them is based on load balancing technique across all available nodes. Mathew et al. [6] suggest a new allocation scheme for social data in NoSQL database. This technique reflects on the depreciation of search procedure thereby increasing query performance. This procedure deeply constrained to keywords associated to user file data like names of files, other data including record and field information are unable to search. Here each search dimension file keyword moves across nodes in a distributed peer to peer model. Li et al. [7] proves in their paper distributed NoSQL systems, object, and graph NoSQL databases are lightweight and they support both low latency and high throughput for fast data allocation. Distributed computing in cloud environment provides platform services to users to allocate the Big Data using their own algorithms in a pay-as-you-go structure [8]. The adaptation from relational to NoSQL databases in cloud environment resulted in significant profits in Big Data allocation and cost savings with respect to node usage but lack of data retrieval techniques forced for a research in query processing techniques [9]. Maheswari et al. [2] discuss about the resource allocation in mobile cloud environment is a tedious task hence various techniques to resolve these issues are studied by them. Among the different techniques, FFD strategy was used in order by them to resolve the issue of allocation in a distributed environment of cloud. Zhang et al. [10] survey how the recent advancements in NoSQL databases support for Big Data. They also focus on the challenges neglected and results obtained for the development of data computing applications across homogenously distributed nodes. David Gil et al. [11] is a framework for data load optimization of Big Data storage and computation in nodes. In this approach amount of data transferred between the nodes, structure of communication, memory requirements, I/O activity, hard disk, affinity of relationships, etc. are all considered. These characteristics are analyzed and efficient techniques are implemented. The main drawback seen here is replication and relationships are not considered. Gondhi et al. [12] discuss how feature selection is done in text categorization. To upgrade the performance of text classification, they present a new algorithm depended on Ant Colony Optimization. This algorithm is based on the real ants movement in search for shortest path to reach the source of food. Here the movement of ants in search of food through shortest path indicate the relation between blocks stored in different nodes but replication of blocks is not considered.

Social big data allocation analytics consists of terabytes or petabytes of data. To store these data, many nodes are required. So for efficiently storing this input data based on relations between them and with replication, social data input needs more than one node which means it needs a distributed environment to store this big data. Execution time of query processing depends on the optimal store of this social data in the distributed environment. Since here we are using graph NoSQL databases integrated with social network sites like Twitter and Facebook for data allocation, the motivation is that the data allocated should be optimal in distributed architecture by minimizing the query processing overhead based on replications and relations.

3. Architecture of graph NoSQL databases

Graph NoSQL databases architecture consists of numerous nodes in a distributed environment. Each node constitutes a different configuration. These nodes are set with distributed configuration setup to support graph social data allocation. The replicated data are placed on different nodes and data related to each other are placed on the same node based on the proposed method to achieve efficient query processing. Social data storage in the era of big data is a high requirement for effective query processing from NoSQL graph databases. Cypher Query Language (CQL) is used to fetch queries from graph NoSQL databases architecture after storing across numerous adjacent node clients.

Please cite this article as: A.B. Mathew, Data allocation optimization for query processing in graph databases using Lucene, Computers and Electrical Engineering (2018), https://doi.org/10.1016/j.compeleceng.2018.01.022