Computers and Electrical Engineering 000 (2017) 1–16

Contents lists available at ScienceDirect

Computers and Electrical Engineering

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

Efficient query retrieval in Neo4jHA using metaheuristic social data allocation scheme☆

Anita Brigit Mathew*, S.D. Madhu Kumar, K. Murali Krishnan, Sameera M. Salam

Department of Computer Science and Engineering, National Institute of Technology Calicut, India

ARTICLE INFO

Article history:
Received 15 July 2017
Revised 29 November 2017
Accepted 2 December 2017
Available online xxx

Keywords:
Big data
Cypher query retrieval
Neo4jHA
Data allocation
Skip List
Best Fit Decreasing
Ant Colony Optimization

ABSTRACT

Large amount of data from social networks needs to be shared, distributed and indexed in a parallel structure to be able to make best use of the data. Neo4j High Availability (Neo4jHA) is a popular open-source graph database used for query handling on large social data. This paper analyses how storing and indexing of social data across machines can be carried out by placing all the related information on the same or adjacent machines, with replication. The social graph data allocation problem referred to as Neo4jHA allocation has proved to be NP-Hard in this paper. An integration of Best Fit Decreasing algorithm with Ant Colony Optimization based metaheuristics is proposed for data allocation in a distributed architecture of Neo4jHA. The evaluation of the algorithm is carried out by simulation. The query processing efficiency is compared with other heuristic algorithms like First Fit, Best Fit, First Fit Decreasing and Best Fit Decreasing found in literature. A Skip List index was constructed on Neo4jHA of every machine after the implementation of the proposed allocation strategy for enhancing the query processing efficiency. The results illustrate how the proposed algorithm outperforms other data allocation approaches in query execution with and without an index.

1. Introduction

The storage of massive data from social networks in a single machine is a tedious process [1]. NoSQL graph databases, where unstructured data is stored in structured form offers advantages in flexible query retrieval of the stored data based on the growing demand of users. Social data is stored in distributed master-slave setup in Neo4j High Availability (Neo4jHA) graph NoSQL database. A thread manager access the data distributed in Neo4j across different machines at the time of query processing in Neo4jHA. Query process invokes machines based on the query request made by the user. Accordingly, machines used for data storage should be geographically near so that the query retrieval is as fast as possible. Data stored in these machines constitutes relationships as the data dealt here is social data. Relationships of social data in Neo4j is string or integer type value, giving a semantic meaning, and *out* or *in* direction, represented as *relationout* and *relationin* in the record storage structure. The relationships in Neo4j is traversed in both directions during retrieval process [2]. There is no need to create two different relationships between the nodes as one implies the other.

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. R. C. Poonia.

* Corresponding author.
E-mail address: anita_p120024cs@nitc.ac.in (A.B. Mathew).

https://doi.org/10.1016/j.compeleceng.2017.12.005
0045-7906/

Apart from conventional methodologies of query retrieval, a flexible query processing provides mechanisms for answering user queries in an intelligent manner. This allows the user to retrieve queries whether it is range (records within an upper and lower bound) or related (relation existing between data in fragments or queries belonging to the same dataset) [3,4]. A query failure is a situation where the query retrieval fails despite the data residing in the database, often on account of inefficient storage mechanism. Traditional database systems return an empty result during query failure. Query failure makes user to check and resend the query. Efficient placement of all related data and index structure leads to flexible query processing without any failure in the retrieval process.

A review of related works of existing data storage techniques in Neo4j and various optimization techniques are presented in Section 2. Neo4jHA architecture is explained in Section 3. Section 4 discuss about the requirement of social data allocation in Neo4jHA. This problem is represented as a variation of Bin Packing Problem (BPP) and solved using 0–1 Integer Linear Programming (ILP). To solve this social data allocation problem, a metaheuristic based Best Fit Decreasing Ant Colony Optimization (BFDACO) integrated model is proposed and analyzed in Section 5. Skip List indexing technique is incorporated on Neo4jHA and description of the algorithm is presented in Section 6. The experimental setup and the result obtained are presented in Section 7. Section 8 concludes the work with scope for future expansion.

2. Literature review

Many researchers have studied the problems of query retrieval over relational databases. Xin et al. [5] proposes a single server data allocation scheme based on the prior knowledge of the type of related data and grouping by encapsulation schemes. Data compression techniques can be used for storage of data which causes loss of data during retrieval process. Poria et al. [6] discusses on next fit data allocation scheme by balancing data across various machines. The technique failed to consider relations existing between the data and its duplicate in a distributed environment. Ramakrishna et al. [7] explains how to perform query in a distributed network connected to data sources with the help of two real world instances of data allocation schemes for load balancing across all available machines. Deng et al. [8] suggest a First Fit allocation scheme for load balance on social media data in NoSQL database. First Fit allocation scheme reflects on the depreciation of search procedure thereby decreasing query performance. This procedure is deeply constrained to single keyword data like names of persons, countries or things. Here every query moves across machines in a peer to peer model. Li et al. [9] prove that distributed NoSQL systems, object and graph NoSQL databases are lightweight and support low latency and high throughput for fast data allocation. Distributed computing in cloud environment provides platform for users to allocate data using their own algorithms in a pay-as-you-go structure [3]. The adaptation from relational to NoSQL databases in a cloud environment results in significant improvement in data allocation and cost savings with respect to machine usage. Still the lack of optimum data retrieval techniques demands extensive research in the field of query processing [10]. Zehmakan et al. [11] discusses the resource allocation in the cloud environment using FFD strategy to resolve the issue of data allocation. Zheng et al. [12] presents a survey on the recent advancements in data allocation and query process. They focus on the challenges during the development of data computing applications across homogeneously distributed machines but failed to address indexing. Gil et al. [13] suggest a model for data load optimization during storage and query computation in machines. In this approach, the amount of data transferred between the machines, structure of communication, memory requirements, I/O activity, hard disk, affinity of relationships, sequential index mechanisms etc. are considered. The main drawbacks of the above model is that, replication and relationships are not considered and the use of sequential index causes time overhead during the search. Zehmakan et al. [14] discuss how feature selection is done in text categorization using indexing. To upgrade the performance of text classification, they present a new algorithm depended on Ant Colony Optimization and hash-based index. This integrated algorithm supports replication of data and better query performance compared to other heuristic approaches with index.

Bin Packing Problem (BPP) [15], a problem in which a group of fragments of data of various sizes needed to be packed into less number of machines has been studying for a long time with an aim to develop a fast heuristic solution for data allocation. But BPP did not take replication into consideration. Traditional methods of BPP are solved using heuristic based algorithms. Some of the heuristic based algorithms in Bin Packing Problem are discussed below:

1. First Fit (FF): FF deals with first come first allocate strategy. An approximation factor of 2 is calculated for FF by Zheng et al. [12].
2. First Fit Decreasing (FFD): The fragments in FFD are arranged in decreasing order and further processed similar to FF algorithm. Brenda S Baker paper proved that for storage FFD utilizes not greater than $11/9OPT + 3$ bins [14], where OPT indicate the number of bins obtained from the optimal solution computed. A research by Dosa claimed the tighter bound of heuristic FFD is, $FFD(I) \leq 11/9OPT(I) + 6/9$ [16], where I indicate every instance taken for storage.
3. Best Fit Decreasing (BFD): Similar to FFD, BFD arrange items in decreasing order of weights and choose the bin in such way that minimum empty space will be left after the items get packed [11]. Dosa and Sgall offered a tight upper bound for the best-fit decreasing strategy, showing that it never require more than $17/10OPT$ bins for any input. Zehmakan et al. proved BFD to be a linear time 3/2-approximation algorithm [14]. It was also proved that both FFD and BFD in BPP has approximation factor of 2 [11].

Social data analytics for efficiently storing terabytes or petabytes of data based on relations between them and replication in a distributed environment should be computationally modeled. The optimal store model needs to support query

Please cite this article as: A.B. Mathew et al., Efficient query retrieval in Neo4jHA using metaheuristic social data allocation scheme, Computers and Electrical Engineering (2017), https://doi.org/10.1016/j.compeleceng.2017.12.005