International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 4 | Issue 3| ISSN: 2456-3307

An Efficient Management for Map Reduce Using Partition and Aggregation in Software Application

P. Ramya[1],Dr A. Saravanan[2]
[1]Research Scholar,Department of CSC, Sun arts and science college, Thiruvannamalai, Tamil Nadu, India
[2]Professor, Department of CSC, Sun arts and science college, Thiruvannamalai, Tamil Nadu, India

ABSTRACT

In this paper, we study to reduce data traffic and to avoid duplication using a MapReduce technique and data partition scheme. Aggregator problem and large-scale optimization problem of duplication and data traffic were made in online or offline. In these problem we use map reduce technique to clustering the data and use k-nearest algorithm is used to reduce the time and cluster the nearest data to avoid conflict. Then we also use ProMiSH based on random projections and hashing for partition process to avoid data traffic in the search engine. In this, we also using k-nearest algorithm for aggregation to clustering the nearest neighbor data and using ProMiSH based on random projections and hashing for partition process to avoid data traffic in the search engine. Then these we use algorithms to decrease the traffic and conflict while processing the data and improve the speed to access the data faster.

Keywords: MapReduce, NSK, K-nearest, ProMiSH

I. INTRODUCTION

We use the technology of MapReduce concept of clustering to reduce the data traffic and to aggregate the data by using nearest neighbour of k-nearest algorithm then we using the ProMiSH based on random projections and hashing for partition process to avoid data traffic in the search engine. Then these we use algorithms to decrease the traffic and conflict while processing the data and improve the speed to access the data faster.

II. PROBLEM DESCRIPTION

Problem is while we searching a data in any applications or search engine it has having data traffic and unrelated data will be shown. It will create a conflict to the user and also these data traffic will be increase the time of the user. In these problem we use map reduce technique to clustering the data and use k-nearest algorithm is used to reduce the time and cluster the nearest data to avoid conflict. Then we also use ProMiSH based on random projections and hashing for partition process to avoid data traffic in the search engine.

III. EXISTING SYSTEM

In this paper, we study nearest keyword set (referred to as NKS) queries on text-rich multi-dimensional datasets. NKS query is two-dimensional data access and it process the top-k clusters these algorithmsis used to increase size or dimensionality in datasets. In three-based indexes, suggest possible solutions to NKS queries on multidimensional dataset. Then NKS is the user-defined keywords.

IV. DISADVANTAGES OF THE EXISTING SYSTEM

- NKS queries are useful for graph pattern search, where the graphs are in a high dimensional space.
- Nearest neighbor queries usually require coordinate information for queries, which makes it difficult to develop an efficient method to solve NKS queries by existing techniques for nearest neighbor search.

V. PROPOSED SYSTEM

A. Design Considerations

To reduce network traffic within a MapReduce job, we have to consider aggregate data with similar keys before sending them to remote reduce tasks. Even though we have a similar function, called combiner, which has been already adopted by ProMiSH, it operates immediately after a map task for its generated data, failing to exploit the data aggregation opportunities among multiple tasks on different machines. Objective is to minimize the total network traffic by Data partition and aggregation for a MapReduce job.

In this paper, we propose ProMiSH (short for Projection and Multi-Scale Hashing) to enable fast processing for NKS queries. In particular, we develop an exact ProMiSH (referred to as ProMiSH-E) that always retrieves the optimal top-k results, and an approximate ProMiSH (referred to as ProMiSH-A) that is more efficient in terms of time and space, and is able to obtain near-optimal results in practice.

ProMiSH-E uses a set of hash tables and inverted indexes to perform a localized search. Based on this index, we developed ProMiSH-E that finds an optimal subset of points and ProMiSH-A which searches near-optimal results with better efficiency. ProMiSH is faster than state-of-the-art tree-based techniques, with multiple orders of magnitude performance improvement.

B. System Architecture

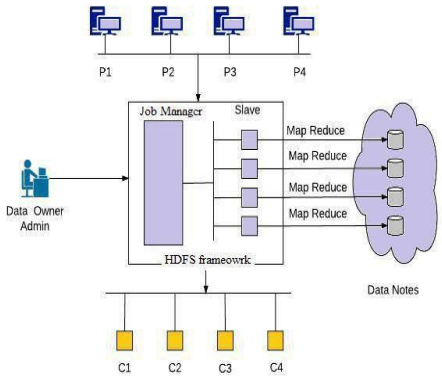

Figure 1. Architecture of Proposed System

The incoming applications from data generators is received by the Job Manager where it is partitioned and Map/Reduce Tasks are carried out. The data is portioned and stored on the nodes by using load balancing techniques to minimize traffic. The Clients ask queries to the system.

Advantages:

1. The performance of ProMiSH on both real and synthetic datasets.
2. We develop efficient search algorithms that work with the multi-scale indexes for fast query processing.

VI. RESULT AND ANALYSIS

Through this project we plan to solve the below two problems:

- Data traffic: by designing a novel intermediate data partition scheme we aim to reduce network traffic cost. And solve the problem of data traffic in search engines and software
- Aggregator placement problem: A decomposition based distributed algorithm is proposed to with the large-scale optimization problem for an software application.

VII. CONCLUSION

In this paper, we proposed solutions to the problem of top-k nearest keyword set search in multi-dimensional datasets. We proposed a novel index called ProMiSH based on random projections and

hashing. Based on this index, we developed ProMiSH-E that finds an optimal subset of points and ProMiSH-A that searches near-optimal results with better efficiency.

VIII. FUTURE WORK

The future scope of the works should be made on complex data partitioning in the database where more intelligent methods have to be employed. This includes analyzing computation cost, skew record etc. So that optimization of data partition is done in map reduce. If the cluster is dynamically growing, the index of the cluster also keep grows.

IX. REFERENCES

[1] C. Long, R. C.-W. Wong, K. Wang, and A. W.-C. Fu, "Collective spatialkeyword queries: a distance owner-driven approach," in SIGMOD, 2013.

[2] D. Zhang, B. C. Ooi, and A. K. H. Tung, "Locating mapped resourcesin web 2.0," in ICDE, 2010, pp. 521–532.

[3] V. Singh, S. Venkatesha, and A. K. Singh, "Geo-clustering of imageswith missing geotags," in GRC, 2010, pp. 420–425.

[4] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi, "Collective spatialkeyword querying," in SIGMOD, 2011.

[5] W. Li and C. X. Chen, "Efficient data modeling and querying systemfor multi-dimensional spatial data," in GIS, 2008, pp. 58:1–58:4.

[6] V. Singh, A. Bhattacharya, and A. K. Singh, "Querying spatial patterns,"in EDBT, 2010, pp. 418–429.

[7] J. Bourgain, "On lipschitz embedding of finite metric spaces in Hilbert space," Israel J. Math., vol. 52, pp. 46–52, 1985.

[8] H. He and A. K. Singh, "Graphrank: Statistical modeling and mining ofsignificantsubgraphs in the feature space," in ICDM, 2006, pp. 885–890.