ABSTRACT

In this paper, the authors present storage structures, PK-map and Tuple-index-map, to improve the performance of query execution and inter-node communication in Cloud Data Warehouses. Cloud Data Warehouses require Read-Optimized databases because large amount of historical data are integrated on a regular basis to facilitate analytical applications for report generation, future analysis, and decision-making. This frequent data integration can grow the data size rapidly and hence there is a need to allocate resource dynamically on demand. As resource is scaled-out in the cloud environment, the number of nodes involved in the execution of a query increases. This in turn increases the number of inter-node communications. In queries, join operation between two different tables are most common. To perform the join operation of a query in the cloud environment, data need to be transferred among different nodes. This becomes critical when there is a huge amount of data (in Terabytes or Petabytes) stored across a large number of nodes. With the increase in number of nodes and amount of data, the size of the communication messages also increases, resulting in even increased bandwidth usage and performance degradation. In this paper, the authors show through extensive experiments using PlanetLab Cloud that their proposed storage structures PK-map and Tuple-index-map, and query execution algorithms improve the performance of join queries, decrease inter-node communication and workload in Cloud Data Warehouses.

Keywords: Cloud Storage, Communication Cost, Data Warehouse, Multi-join Query, Query Optimization, Read-Optimized Database

DOI: 10.4018/ijghpc.2013100108
1. INTRODUCTION

Historical Data from one or more heterogeneous data sources are extracted using ETL (Extract Transform and Load) technique and stored in a central repository called Data Warehouse or Data Marts (smaller version of Data warehouse), so that the data can be easily accessed for analysis. Data Warehouse is an organized collection of databases containing such historical data, which may represent the business history of an organization, or biological data of diseases, or agricultural data. Analysts use this data for online analysis and report generation, which require quick responses to iterative complex analytical queries. There are several good commercial OLAP (Online Application Processing) applications for data warehouse analysis such as EMC’s Greenplum, IBM’s InfoSphere, Microstrategy, Oracle BI, SqlServer, and Vertica.

In the past several decades, read-optimized databases have gained popularity in read intensive analytical workloads such as data warehousing and business intelligence applications. In these databases, data are partitioned and organized in many different ways such that it can be accessed quickly. Organization of data may be row-oriented, column-oriented or hybrid, depending on the type of application, usage or queries. Row-oriented databases store rows of a table sequentially on physical storage, whereas column-oriented databases store attribute values of one or more column (called projection) sequentially on physical storage (MacNicol, 2004; Stonebraker, 2005). To achieve high level of parallelism, these tables or projections can be further horizontally divided and distributed over multiple nodes/systems in a network.

Cloud computing and storage has gained attention from researchers and consumers in recent years. It is designed to provide dynamically scalable resources to consumers, eliminating the hassle of investment and maintenance. Many commercial products from Amazon, Microsoft, EMC and IBM provide cost effective solutions like hourly or monthly or yearly billing (Amazon EC2, 2012). Cloud provides an environment where the end user can perform tasks as if the data is stored locally when it is actually stored in remote systems. Providing such an environment needs powerful computing, fast execution strategies for tasks and high-speed communication network.

In the cloud architecture, depending on the availability of resources such as nature of application, storage space, and CPU cycles, data is distributed to different nodes. In addition, the physical location of the data may dynamically change from one node to another.

In traditional distributed databases, to reduce the inter-node communication during the execution of a query, tables are horizontally partitioned on the join attributes and related partitions are stored on the same physical system. In cloud environment, it is not possible to ensure that these related partitions are always stored on the same physical system. Thus, execution of queries in cloud data warehouses becomes more complicated when queries contain multiple joins between partitions of different tables stored in different nodes. These joins need back and forth communication among the query execution nodes to find the correct result. This heavy communication among the nodes will have adverse affects on the performance of the query and increase network traffic.

A long stream of research work has been done to improve the join operation in distributed read-optimized databases and network attached storages. Some of them are Abadi (2009), Abouzeid (2009), Akinde (2002), Candea (2009), Holloway (2008), Ivanova (2008), Kalnis (2003), Stonebraker (2005), Wang (2011).

Efficient ways to concurrently execute multiple queries in order to increase the throughput is discussed in Candea (2009) and Kalnis (2003). In Stonebraker (2005), authors present vertical storage structure and indexing to handle record-oriented query. This vertical storage structure has improved the performance of data warehouse applications in the order of magnitude and has gained lot of attention. Paper (Abadi, Madden, & Hachem, 2008) compares the actual difference between column and row-oriented DBMS in both physical and
A Hybrid Dynamic Load Balancing Algorithm for Distributed Systems Using Genetic Algorithms
[www.igi-global.com/article/a-hybrid-dynamic-load-balancing-algorithm-for-distributed-systems-using-genetic-algorithms/117167?camid=4v1a](www.igi-global.com/article/a-hybrid-dynamic-load-balancing-algorithm-for-distributed-systems-using-genetic-algorithms/117167?camid=4v1a)

Applications of Visual Algorithm Simulation
[www.igi-global.com/chapter/applications-visual-algorithm-simulation/38264?camid=4v1a](www.igi-global.com/chapter/applications-visual-algorithm-simulation/38264?camid=4v1a)