Cairo University

Faculty of Computers & Information

Computer Science Department

Indexing For Improving Big Data Analysis

By

Hussien Shahata Abdel Aziz

A Thesis Submitted to the

Faculty of Computers & Information

Cairo University

In Partial Fulfillment of the

Requirements for the Degree of

(Master of Science)
In
Computer Science

Under the Supervision of

Prof. Fatma A. Omara

Computer Science Department

Faculty of Computers & Information

Cairo University

Dr.Mohamed H.Khafagy

Computer Science Department

Faculty of Computers & Information

Fayoum University

Faculty of Computers & Information

Cairo University

Jan. 2016

Abstract

Today Big Data analysis has become one of the most complicated computing tasks today because data growing heavily in size, dimensionality, unstructured formats. Recent studies expect data growing into zettabytes which gathered from different sources like sensors, social networks, machines Logs, etc. Parallel Relational Database Processing Systems can't fit into these large data sizes because of data indexing and relational constraints that will slow down data storing / retrieval performance.

MapReduce has become an effective framework for processing and analysis huge data size in large systems. On the Other Hand, Hadoop represents one of the core frameworks build on Map/Reduce for big data analysis and processing. Also Hive is Database like software built on top of Hadoop. It is act as database engine without data relations or data indexing also it's only a translator from SQL queries into Hadoop map/reduce tasks. but HIVE join query execution pipeline (which based on Hadoop map/reduce tasks) is complex pipeline, execution time consuming, temporary storage consuming and java heap memory crashes done because the vast amount of data needs to be hold in memory or hard disk through execution pipeline. Database star schema model almost uses join query to gather required information/reports for decision maker and this will be somehow very difficult to run over HIVE pipeline.

Many approaches have been tried to index Hadoop data on HDFS by injecting (online/offline, static/dynamic) data index to improve data retrieve. But it is still suffer from Slow Join query execution although data been indexed on HDFS because of the complex execution pipeline in HIVE itself.

In this thesis we have introduced an enhancement of HDFS and Hadoop MapReduce that dramatically improves the runtimes of join operations of HIVE when translated into MapReduce jobs. Two different execution pipelines for HIVE based on static/offline data index built at data loading step for the first time of loading star schema data into HIVE. This schema is joined into a table and be in a pre-join state all the time to be ready for querying. This pre-joined sate saves memory/storage/time through the execution pipeline. Interesting features of both pipelines that don't affect HIVE framework anywhere since only add pre/post layers for HIVE framework. Also, we typically create a win-win situation.

We improve both temporary data stored in HDFS and the runtime of the actual Hadoop join MapReduce job. Join Once Use Many (JOUM) and Keys/Facts indexing methodologies have been introduced and the evaluation of them has been done using TPC-H benchmark data/queries. In terms of query execution time; the two methodologies outperforms HIVE execution pipeline on join query by (٥٣.3%, 49.8%) respectively for execution time, (٢٨%,٢٧ %) for temporary storage, (2٧%, 1٩%) overhead for permanent storage and small number of memory crashes. Generally, JOUM and Keys/Facts indexing methodologies are suitable methodologies for Big Data analysis. However, minimum overhead in permanent storage is produced because of index, but it is small compared to the large size saved by temporary storage.