DYNAMIC PROCESSING SLOTS SCHEDULING FOR I/O INTENSIVE JOBS OF HADOOP ON PATHOLOGY DATA

K.H Naz Mufeeda , Anisha Rodrigues

Abstract - **The increasing use of computing resource in our daily lives leads to data generation at an astonishing rate. The computing industry is being repeatedly questioned for its ability to accommodate the unpredictable growth rate of data.It has encouraged the development. Hadoop consists of Hadoop Mapreduce and Hadoop Distributed File System (HDFS), is a platform for large scale data and processing. Distributed processing has become common as the number of data has been increasing rapidly worldwide and the scale of processes has become larger, so that Hadoop has attracted many cloud computing enterprises and technology enthusiasts. Hadoop users are expanding under this situation. My studies are to develop the faster of executing jobs Originated by Hadoop and Hive. Our Proposed work is to set dynamic processing slots scheduling for I/O intensive jobs of Hadoop MapReduce focusing on I/O wait during execution of the pathology data efficiently on the Hadoop cluster or clouds. Assigning more tasks to added free slots when CPU resources with the high rate of I/O wait have been detected on each active Task Tracker node leads to the improvement of 30% of CPU performance.**

Index Terms— **Hadoop, Hive, MapReduce, Slots Scheduling.**

I. INTRODUCTION

Every year data increases in size very rapidly. It's not easy to measure the total volume of data stored electronically, but estimate put the size of the "digital universe" at 0.18 zettabytes in 2006, and is forecasting a tenfold growth by 2011 to 1.8 zettabytes. A zettabyte is 1021 bytes, or equivalently one thousand Exabyte's, one million petabytes, or one billion terabytes. That's roughly the same order of magnitude as one disk drive for every person in the world. We have taken the measure to improve overall performance. It is called distributed computing, which have cost-effective and general-purpose computers cooperate with each other to handle distributing data or processes all over the cluster consists of them. However, programming for distributed computing is highly complicated. Hadoop could provide us a good solution, Hadoop in general, designed to be deployed on low-cost hardware. It provides high throughput access to application data and is suitable for applications that have large data sets. Even if hundreds or thousands of CPU cores are placed on a single machine, it would not be possible to deliver input data to these cores fast enough for processing. Individual hard drives can only sustain read speeds between 60-100 MB/second. These speeds have been increasing over time, but not at the same incase of processors Hadoop cluster consists of a jobtracker and tasktracker. there is possibility that the physical distance between the node assigned a task to and another node maintaining the pathological data which may requires longer time in proportion to scale the cluster, resulting in communication delay. which makes task I/O wait time longer, hence CPU resources cannot be used effectively. Since maximum slot is already executed, Hadoop scheduler does nothing, except waiting for the slot to be free.

In this paper, we propose dynamic processing slots scheduling for I/O pathological data in Hadoop MapReduce and Hive to use CPU resources effectively by assigning more tasks to added free slots when it detects CPU resources with the high rate of I/O wait on each active TaskTracker node.

II. RESEARCH BACKGROUND

Hadoop is a platform for parallel and distributed large-scale data processing and one of the open-source projects by the Apache Software Foundation[2], The Hadoop system consist of MapReduce and HDFS. MapReduce programming model was developed at Google . MapReduce is a computing model that decomposes large data manipulation jobs into individual tasks that can be executed in parallel across a cluster of servers. The results of the tasks can be joined together to compute the final results. The term MapReduce comes from the two fundamental data transformation operations used, map and reduce. In MapReduce, records are processed by tasks called Mappers. The output that is generated from the Mappers is brought together into a second set of tasks called Reducers; here the results from different Mappers are merged

 All Rights Reserved © 2014 IJARCET

together.

A map operation converts the elements of a collection from one form to another. In this case, input <key><value> pairs are converted to zero-to-many output <key><value> pairs, where the input and output keys might be completely different and the input and output values might be completely different. Map Reduce programming model requires a successfully configured Hadoop environment to run the Map Reduce programs. Large volumes of data are computed in a parallel fashion using Map Reduce programs. In MapReduce the data elements cannot be updated i.e. if in the mapping task try to change the input pairs (key, value) it will not get reflected in the input files used. [1], [4].

Original Hadoop provides the three job schedulers: Job Queue Task Scheduler, Fair Scheduler and Capacity Task Scheduler. Users can select which job scheduler among them. Job Queue Task Scheduler, which is the base of other job schedulers, is default job scheduler based on First In First Out FIFO queue. Tasks are assigned to nodes which maintain their pathological data split based on priority, or other nodes nearby such nodes which maintain their input split with second priority.

Hive is a data warehouse system for Hadoop that facilitates ad-hoc queries and the analysis of large datasets stored in Hadoop. Hive provides a SQL-like language called HiveQL. Due its SQL-like interface, Hive is increasingly becoming the technology of choice for using Hadoop. Hive data is organized into: Database, Tables, Partitions, Clusters

Hive is become a common interface in mapping and reducing the data based on Hadoop Mapreduce . hive also contribute to the CPU performance.

III. General Description of Pathology data Processing

Pathological science, as defined by Langmuir, is a psychological process in which a scientist, originally conforming to the scientific method, unconsciously veers from that method, and begins a pathological process of wishful data interpretation.

The four aspects of a disease process that form the core of pathology : Etiology: causes of the disease. Pathogenesis: the mechanisms of its development. Morphologic changes: the structural alteration induced in the cells and organs of the body. Clinical significance: the functional consequences of the morphologic changes. Different types of pathology data: General pathology: concerned with the basic reaction of cells and tissues to abnormal stimuli that underlie all diseases. Systemic pathology: describe the specific responses of specialized organs and tissues to defined stimuli. The pathological data analysis results in rating the death rate and also helps in the treatment for the various diseases based on setting the priority to cure the critical disease first so that the patient survive for long ago.

IV. Dynamic Processing Slot Scheduling for Pathology Data

The goal of our studies is to work with pathological data set and reduce the whole execution time of Hadoop jobs by using each CPU resource effectively on slave nodes consist in Hadoop cluster. Pathological data are retrieved from the hive warehouse. In dynamic processing slots scheduling for I/O intensive jobs of Hadoop MapReduce use CPU resources effectively by assigning more tasks to added free slots when it detects CPU resource with high rate of I/O wait on each active TaskTracker node

A. Approach

At first, each active TaskTrackers on slave nodes consist in the cluster monitor the state of the CPU. We use I/O wait time information among CPU. If this I/O wait percentage is greater than the predetermined threshold value, one Map slot is added (the number of Map slots is incremented) and which CPU has caused adding Map slot is recorded by TaskTracker. If it is less than threshold value, one added Map slot is eliminated (the number of Map slots is decremented) . Note that the Map slots are up to the initial number of Map slots added the total number of CPUs of the slave node.

The implementation method on Hadoop 1.2.1 as follows:

B. Initialization

When MapReduce nodes start, each TaskTracker reads the required values from the configuration file mapred-site.xml and initialize them. The initial number of Map slots predetermined as the property mapred.tasktracker.map.tasks.maximum is stored with the variable maxMapSlots and simultaneously with the variable maxMapSlots first as the initial value in order to change maxMapSlots in our method.

After intilization then pathological data is given to hive warehouse, which processes tha data based on the hadoop job tracker and tasktracker. The assigned slots reduces the CPU performance

V. Performance Evaluation

We have implemented on Hadoop 1.2.1 and Hive 0.9.0 and evaluated performance of pathological data based on number of different slot.

 All Rights Reserved © 2014 IJARCET

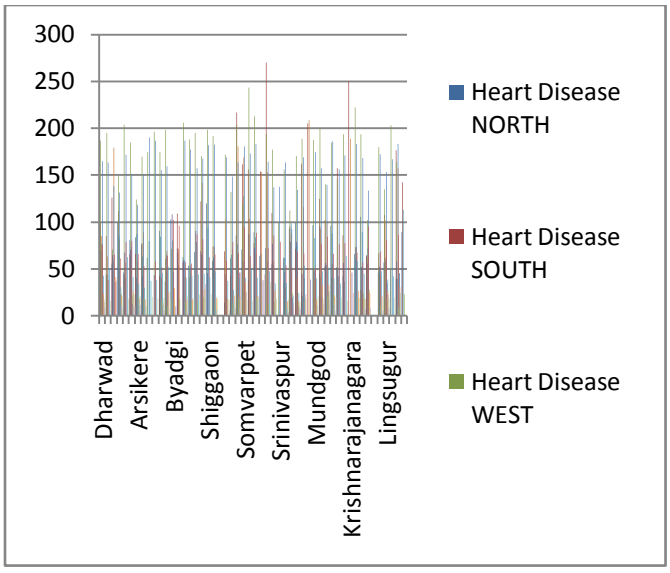

Figure 1 Pathological data analysis

In the figure 1, the pathological data is given to the HDFS, which forwards to jobtracker dynamically allots to the input slot and processes the data with 30% less CPU time and also can find the dynamically allotted slot and the maximum allotted slot, as shown in figure 2 the performance of CPU.

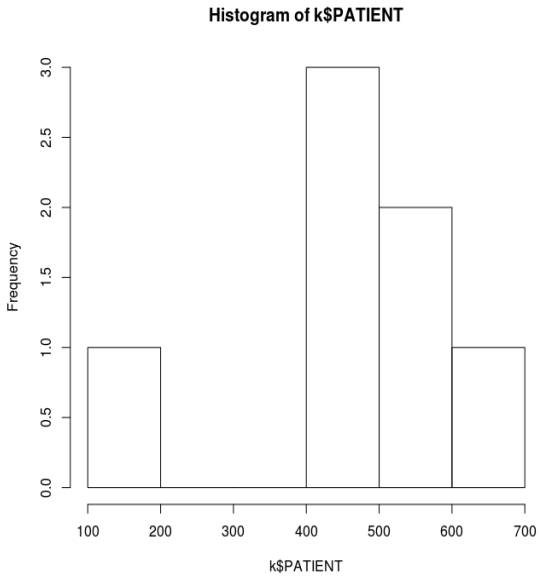

Figure 2 Hadoop job's CPU performance

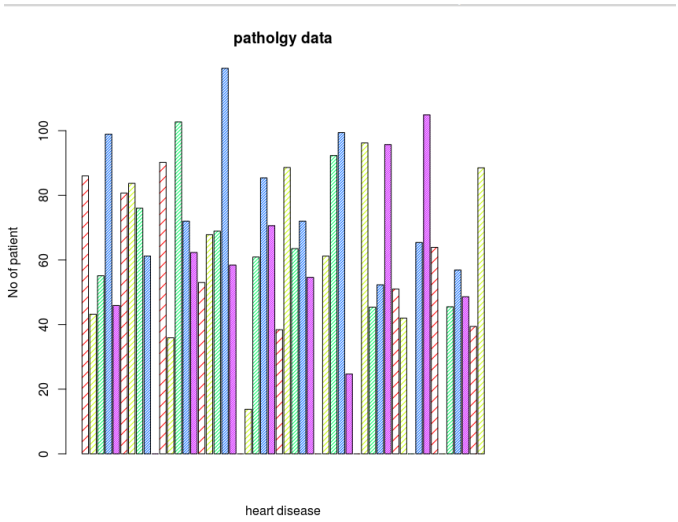

Figure 3 heart disease analysis

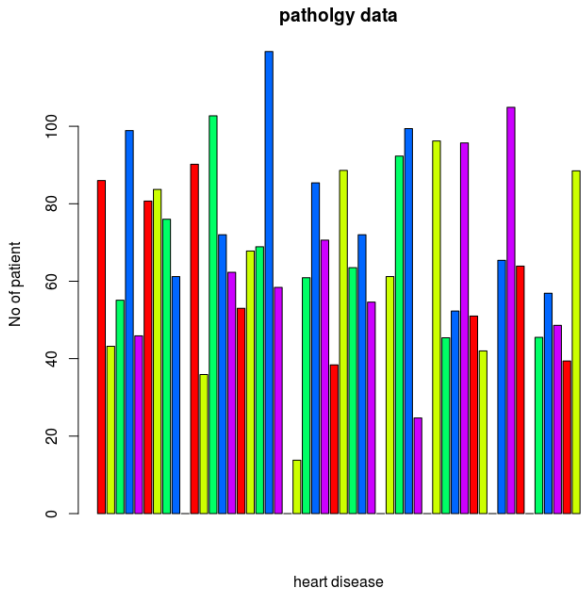

Figure 4 stroke disease analysis

The graph represents the pathological data like stroke, cancer and heart disease. The graphs are based on the various places, pathological death rate studies with respect to individual cause and overall cause. These pathological data is given to test the CPU time performance reducing by 30 %.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed dynamic processing slots scheduling for I/O intensive job of Hadoop MapReduce to use the CPU resources with low usage effectively caused by I/O wait related to task execution which appears during executing job on Hadoop cluster and Hive. In order to examine the

 All Rights Reserved © 2014 IJARCET

effectiveness of the proposal, the proposed method is implemented on Hadoop 1.2.1 and hive 0.9.0, evaluated it by executing jobs using pathological data. Modified Hadoop was enabled to control the number of Map slots dynamically in comparison with default static assignment. The performance of the system is presented using R language, the execution time was improved up to about 23% compared with default Hadoop and Hive.

There are few future works that can be carried out on Hadoop System. one of these is switching how to control the number of Map slots according to the change of MapReduce tasks in the job queues, adding Map slots is actually pointless

REFERENCES

[1] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler, "Hadoop Distributed File System", 2010

[2] Chen Zhang, Hans De Sterck, "CloudBATCH: A Batch Job Queuing System on Clouds with Hadoop and HBase", 2nd IEEE International Conference on Cloud Computing Technology and Science, 2010

[3] T. Kyriacos Talattinis, Aikaterini Sidiropoulou, Konstantinos Chalkias, and George Stephanides, "Parallel Collection of Live Data Using Hadoop", in 14th Panhellenic Conference on Informatics, 2010

[4] Jeffrey Dean.and Sanjay Ghemawat, "MapReduce:Simplified Data Processing on large Clusters , appears in OSDI , 2004

[5] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, "The Google File System", 2003

[6] Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Nin Zhang, Suresh Antony, Hao Liu and Raghotham Murthy "Hive – A Petabyte Scale Data Warehouse Using Hadoop", ICDE Conference, 2010

[7] S. Chandra Mouliswaran, Shyam Sathyan ,"Study on replica management and high availability in hadoop distributed file system " Journal of Science, 2012.

[8] Shiori Kurazumi,Tomoaki Tsumura,Shoichi Saito,Hiroshi Matsuo,"Dynamic processing slots scheduling for I/O intensive job of Hadoop MapReduce, 3rd IEEE International Conference on Networking Computing , 2012.

[9] Ross Ihaka,Robert Gentleman,"R: A Language for Data analysis and graphics", Journal Computational and Graphics Statics,1996

.

 All Rights Reserved © 2014 IJARCET