ISSN:2320-0790

An International Journal of Advanced Computer Technology

EFFECT OF COUNTERS IN PERFORMANCE OF HADOOP

Mrs. Preeti Jain, Ms. Juhi Kanungo

Department of Computer Science and Engineering
Acropolis Institute of Technology and Research, Indore

ABSTRACT: Recent technological advancements have led to an overflow of data from distinctive domains (e.g., health care and scientific sensors, user-generated data, Internet and financial companies, and supply chain systems) over the past two decades [1]. Big data is commonly unstructured, huge in volume and require more real-time analysis. This paper discusses a Big Data problem from NCDC for huge volume of weather data collected from various parts of world. We had generated map () and reduce () function for solving this problem and experimental results of these applications on a Hadoop cluster are being discussed. In this paper, performance of above application has been shown with respect to some counters available.

KEYWORDS: Big Data, Hadoop, Map Reduce, Hadoop Distributed File System (HDFS), CPU time spent, Size of data, number of blocks, Heap Size, HDFS Bytes Read, Spilled Records.

I. INTRODUCTION

Big data analytics is the area where advanced analytic Techniques operate on big data sets. It is really about two things, Big data and analytics and how the two have teamed up to Create one of the most profound trends in business intelligence [2].

Map reduce is capable for analyzing large Distributed data sets; but due to the heterogeneity, velocity and volume of big data, it is a challenge for traditional data analysis and management tools..Here we are going to show the analysis of experiments done on the Hadoop cluster. This study is done to find effect of various counters on the execution time with different data sizes. This study may also help to understand the tuning of Hadoop cluster for better performance [5].

II. PERFORMANCE ANALYSIS OF HADOOP

Here we are going to show the analysis of experiments done on the Hadoop cluster. This study is done to find effect of following counters on the execution time with different data sizes. This study may also help to understand the tuning of Hadoop cluster for better performance.

The counters that we had worked on in this study are:

1. Size of Data
2. Number of Blocks
3. Heap Size
4. HDFS Bytes Read
5. Spilled Records
6. CPU Time Spent

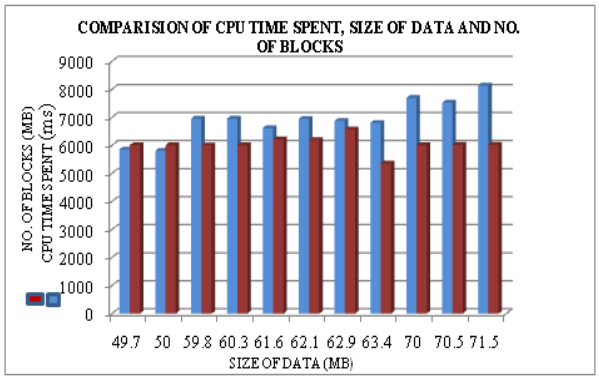
The values obtained from this analysis are as follows:

Table 1: Comparison of values of various counters on different data size

S. no	Size of data (MB)	No of blocks	Heap Size (MB)	HDFS bytes read	Spilled records	CPU time spent
1	49.7	6013	55.19	50257	10926	5860
2	50	6013	55.19	50550	10948	5810
3	59.8	6005	55.19	60481	13122	6960
4	60.3	6013	55.25	62018	13170	6960
5	61.6	6215	55.12	62331	13164	6620
6	62.1	6198	55.12	62758	13108	6950
7	62.9	6578	37.25	63572	13130	6880
8	63.4	5360	53.64	64122	13130	6810
9	70	6012	55.25	70870	15072	7700
10	70.5	6023	55.25	71293	15300	7530
11	71.5	6031	37.25	72331	15252	8140

III. COMPARISION OF CPU TIME SPENT, SIZE OF DATA AND NO. OF BLOCKS

In this experiment we are changing the size of data used to find the change in execution time of the cluster. In case of Hadoop bigger is the data block size, it is easier for HDFS to store information at NameNode and the communication with NameNode also becomes less [7].



Graph can be created on the above values as follows:

Figure 1: Comparision between CPU time Spent, no. of blocks and size of data

In Figure 1 we had compared the values of CPU Time Spent, Size of Data and No. of Blocks. with variation in Size of Data. Study of this comparison shows that if number of blocks increases than parallel execution increases and as a result efficiency of the system increases.

a. CPU TIME SPENT VS HEAP SIZE

Map and Reduce task are java processes. JVM heap size causes effect on the performance of Hadoop. As studied in Figure 2,if the heap size is small, then it may results in task failure because of "out of memoy" error, or due to exccess garbage collection, it may run slowly. If the size of heap is large system resources may be wasted.

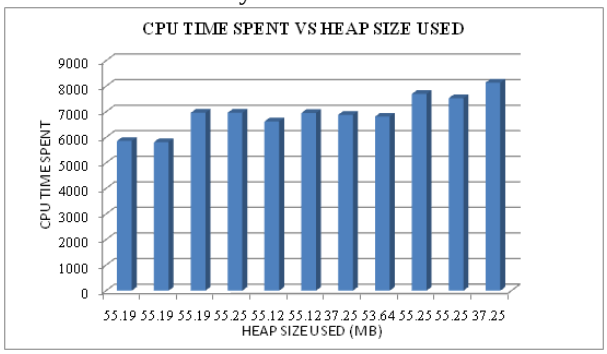


Figure 2: Graph between CPU time Spent Vs Heap size

b. CPU TIME SPENT VS HDFS BYTES READ

HDFS bytes read is the count of bytes read by the mappers, when the HDFS job starts. This value contains content of source file as well as the metadata about splits.Figure 3 shows the difference in CPU time spent with change in HDFS bytes read.

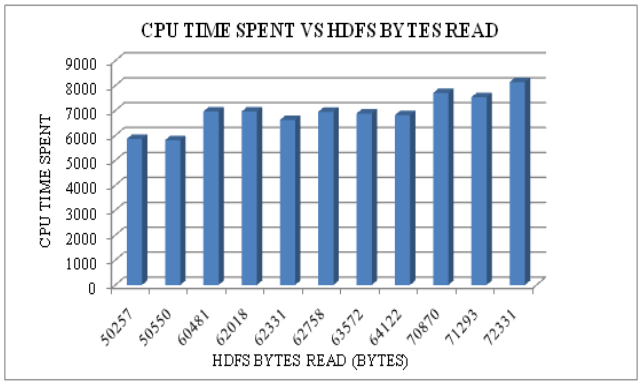


Figure 3: Graph between CPU time Spent Vs Heap size

c. CPU TIME SPENT VS SPILLED RECOREDS:

When output of map function is produced it is not directly written to the disk. Map task have a circular memory buffer where output is temperoraily saved. Buffer size is 100MB by default and can be changed by editing the io.sort.mb property [4]. When the buffer gets filled upto a certain thershold size, another thread gets started and it begins to spill the contents to disk. If limitation regarding size of heap occurs, then number of spills should be minimized by changing io.sort.record.percent parameter values [6]. Figure 4 shows graph that results in increase in number of spilled records,with the increase in the execution time.

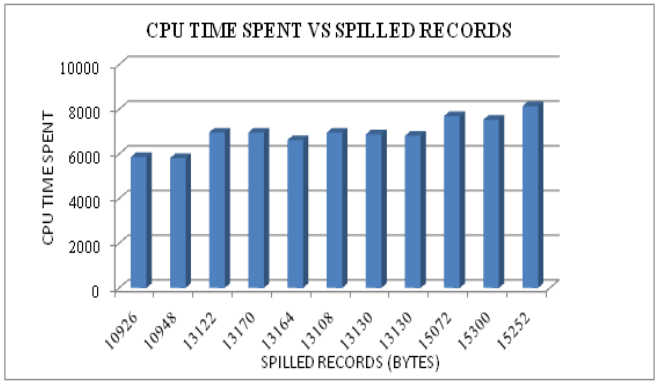


Figure 4: Graph between CPU time Spent Vs Spiled Records

IV. CONCLUSION AND FUTURE SCOPE

Big data offers a prospect to dig out enormous opportunities in new and emerging types of data. Hadoop and other improved technologies make us capable of storing, handling and analyzing structured, unstructured and semi structured data, without much effort [3].The HDFS has been designed to use commodity hardware. this system is highly fault tolerant and is designed with an aim to provide deployment on low price hardware. Thus it provides high throughput access to application data and is best suitable for applications with large datasets [8].

Under the conclusion of this work, we can say that performance of Hadoop for Big data analytics problem does not vary much with the size of data. With the results of implementation, the graph drawn represents, the variation in time required to perform the searching task over the data with assorted size along with other counters like spilled records, HDFS bytes read, Heap size, and number of blocks . By observing the results of the implementation and the graph based on the implementation, we can conclude that Hadoop is an efficient system for handling and analyzing big data.

FUTURE SCOPE

Though HDFS resolves the Big Data problem efficiently but there is always scope for improvements. Hadoop provides the Replication of data to avoid data loss, thereby making the system "Fault Tolerant". The more number of copies are created, the more the system becomes fault tolerant. If a DataNode fails then map and reduce process of that node is shifted to the other nodes carrying the replicated data. But we have only one NameNodes which contains the metadata of all the DataNodes, and if this node gets crashes, then the processing cannot takes place further. This weakness of Hadoop is being resolved in Hadoop 2, by providing multiple namespaces and multiple NameNodes in it.

REFRENCES

[1] Maurya and Mahajan, S., "Performance analysis of MapReduce programs on Hadoop cluster", Information and Communication Technologies (WICT), 2012 World Congress, ISBN:978-1-4673-4806-5, Oct. 30 2012.

[2] A. Alexandrov, R. Bergmann, S. Ewen, J.C. Freytag, F. Hueske, A. Heise, O. Kao, M. Leich, U. Leser, V. M. Felix Naumann, M. Peters, A. Rheinländer, M. J. Sax and S. Schelter, "The Stratosphere Plateform for Big Data Analytics", The VLDB Journal, Springer-Verlag Berlin Heidelberg, July 2013.

[3] Shilpa and Manjit Kaur, "Big Data Visualization tool with Advancement of Chalange" International Journal of Advanced Research in Computer and Software Engineering", ISSN: 2277 128X, Volume 4, Issue 3, March 2014.

[4] D. Borthakur, "The Hadoop Distributed File System: Architecture and Design".

[5] Han Hu, Yonggang Wen , Tat-Seng Chua and Xuelong Li, "Toward Scalable Systems for Big Data Analytics: A Technology Tutorial", ISSN :2169-3536, INSPEC Accession Number: 14429402

[6] P. Kumar and Dr. V. S. Rathore, "Efficient Capabilities of Processing of Big Data using Hadoop Map Reduce", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 3, Issue 6, June 2014.

[7] Shilpa and Manjit Kaur, "Big Data Visualization tool with Advancement of Chalange" International Journal of Advanced Research in Computer and Software Engineering", ISSN: 2277 128X, Volume 4, Issue 3, March 2014.

[8] X. Zhang, "Simple example to demonstrate how does the map reduce work", June 2013.