Performance Optimization of Big Data Processing using Clustering Technique in Map Reduces Programming Model

Ravindra Singh Raghuwanshi
Samrat Ashok
Technological Institute
VIDISHA,M.P India

Deepak Sain
Samrat Ashok
Technological Institute
VIDISHA, M.P India

ABSTRACT
The generation of technology and requirement fulfill the demand of digital universe data. Day to day the digital universe data are exploded in terms of megabyte and petabyte. The exploding rate of data demands the new generation of technology such as big data processing. In this paper optimized the performance of map reduce programming model for the enhancement of data processing. The modified model of programming used clustering technique. the clustering technique incorporate the process of map data in terms of task group. The task group of map data correlated with different index of data for the processing of data node. The proposed model implemented in Hadoop framework and programmed in java. For the evaluation of performance used three standard datasets and measure the processing time and count value of file.

Keywords
Big Data, Hadoop, MapReduce, Clustering, Optimization

1. INTRODUCTION
The increasing rate of digital data faced a problem of processing, speed and analysis. The normal file system and framework cannot support the concept of Nosql. The concept of Nosql precedes the unstructured and unformatted data for the analysis and processing. For the processing of big data used map reduces function process. The map reduces function process basically based on java programming model[1,2]. The java programming model generated the value of key task for the processing of data. In this dissertation proposed prototype model for the processing of data. The prototype mode based on the concept of data mining. The process of data mining gives and precedes the various algorithms for the processing of data. The data mining technique provide association rule mining technique for the processing of relation data[3,4]. Here mining clustering technique are used for the improvement of map reduces file processing in HADOOP data analysis tool. the modified model of Map reduces simulated in Hadoop framework. The processing of map reduces function based on two attribute data one is cluster of data and other is key value. The value of key generates the processing of group of data for the process of analysis[11]. The modified map reduces function component encapsulate the processing of DB scale clustering technique. the DB scale clustering technique define the value of range of group data for the processing of map cerates block. More than 30% improvement has been observed in some cases of applications which are quite impressive from computational perspective. It has also been observed that, the time for clustering becomes almost stationary for higher number of nodes even the input volume of data has been increased from 7 million to 10 million[12]. Thus, DB scale being very useful clustering technique, using it in cloud environment for processing Big Data has some inherent advantages and may be used for various application. For processing and analysis of datasets many tools are available and the most popular and widely used is Apache Hadoop [7]. Hadoop can handle all types of data such as structured, unstructured, log files, pictures etc. Hadoop supports redundancy, scalability, parallel processing, and distributed architecture [7]. In general, distributed computing [8] is a field of computer science that involves multiple computers, located remotely from each other. Each computer has a common shared role in a computation problem and coordinates their actions by message passing. Scheduling problem is also faced in other computing systems. The work in [9] addresses scheduling in general-purpose distributed computing environment.Rest of this paper is organized as follows in Section II discusses MapReduce programming Model, Section III proposed algorithm IV. Experimental result analysis Finally, concluded in section V.

2. MAPREDUCE PROGRAMMING MODEL
MapReduce is a programming model which process large amount data in parallel way on large clusters of machines [14]. MapReduce program mainly consists of two functions i.e. map function and reduce function as described in Fig. 1. Map function takes value as input and generates key: value pair. When all the values get a key, this programming model groups all the values together according to their keys. This is the job of combiner module. The output of the combiner module becomes the input of reduce function. Reduce function takes a key and list of values as an input. Reduce function processes on the input and generates output as per requirement. Users define map and reduce function and the runtime architecture automatically distribute the task, take care of machine failures, handles communication among different nodes, balance the load among different nodes. Hadoop provides MapReduce runtime system along with a distributed file system which provides high fault tolerance and scalability. Hadoop distributed file system replicate the data across the node which increases availability of data. The file system uses TCP/IP for communication. There are five kinds of server available in hadoop as shown in Fig.4.2. 2. Name node, data node, secondary name node handle data storage, retrieval and fault tolerance. Job tracker and task tracker handle map reduce computational part.

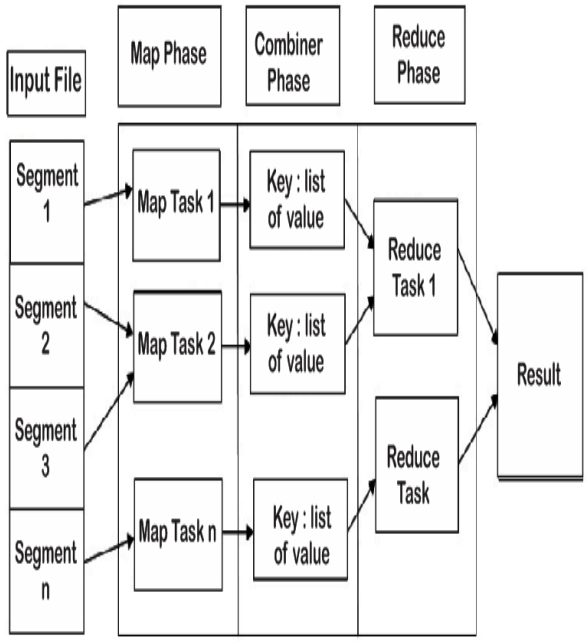

Figure 1 shows that processing of Map Reduces data segment for the process of analysis.

3. PROPOSED ALGORITHM

The proposed model describe into two different section one is the grouping of data for the processing of clustering task and other is reduces process for Hadoop framework.

In this section, we have described about clustering process of Map reduce framework for big data analysis. the proposed Map reduce framework based data analysis system, which consists of three important functions: Map, Intermediate system and Reduce. The overall operation of proposed architecture is given by

$$DS \rightarrow M \rightarrow IMS \rightarrow R \rightarrow Fianl\ value \quad (1)$$

Where, DS is the dataset, M is the mapper, IMS is the intermediate system

MAPPER OPERATION

A big data dataset DS, it is firstly divided into number of subsets. Subset contains many attributes.

$$DS_i = DS_1 + DS_2 + \cdots \ldots + DS_n\ , (0 < i < m) \quad (2)$$

where, $DS_1,\ DS_2\ and\ DS_n$ are the subsets.

Normally, map is written by the user, takes an input pair and generates a set of intermediate key/value pairs. In map reduce architecture figure 2, for each data, we associate a map operation. The first step is to partition the input dataset, typically stored in a distributed file system, among the computers that execute the map functionality. From the logic perspective, all data is treated as a Key (K), Value (V) pair. Each attributes in the input dataset is represented as a <key1, value1>. In the second step, each mapper applies the map function on each single attribute to generate a list on the form$(< key2, value2 >) *$, where $() *$ represents lists of length zero or more.

$$Map : < key1, value1 > \rightarrow (< key2, value2 >) \quad (3)$$

In this context, firstly initializes the necessary structures, primarily input key and value. For this purpose, we have utilized the firefly and naïve bayes classifier.

Firefly algorithm based feature selection process is explained below:

Firstly, we have developed a modified dataset from the training dataset for this fitness selection purpose. The modified dataset contains only identified attributes (*'1's*). This is created based on the initial firefly. Then this modified dataset is classified using naive bayes classifier, we obtain mean and variance.

$$Mean\ (\mu) = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (4)$$

$$Varience\ (\sigma^2) = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu) \quad (5)$$

Where, x_i is the i^{th} attribute n is the number of attribute

3.1 Reducer Operation

The third step is to shuffle the output of the mappers into the systems that execute the reduce functionality. A reduce operation takes all values represented by the same key in the intermediate list and processes them accordingly, emitting a final new list. Here, once the best feature space is identified through firefly algorithm, the big data analysis is done using the naive bayes classifier. Output from all Map nodes, <key1, and <value1> entries, are grouped by key1values before being distributed to Reduce operation. It is the turn of Reduce operation to combine value1 values according to a specific key1. Product of Reduce operation may be in format of a list, <value2>or just a single value, value2.

$$DS(< key1, value1 >) \rightarrow M\ (< key2, value2 >) \rightarrow R((< key2, value2 >) \rightarrow value2 \quad (7)$$

Analysis using DB Scale

Validation of each in coming input data is attained by tokenizing the attribute and using the pre-calculated attribute probability of each feature to classify the incoming value as reduced output data using following naïve bayes expression. Firstly, calculate the mean and variance equation (4) and (5) using naive bayes classifier, and then analysis process is followed by

For the analysis as data the posterior

$$posterior(RO\ f_1, f_2, \ldots f_i) = \frac{P(RO \prod_{i=1}^{n} P(f_i o))}{evidence} \quad (8)$$

For the analysis as used data the posterior is given by

$$posterior(RO - O\ f_1, f_2, \ldots f_i) = \frac{P(RO \prod_{i=1}^{n} P(f_i | RO - O))}{evidence} \quad (9)$$

Where,

$$evidence = P(O)\prod_{i=1}^{n} P(f_i|O) + P(RO - O)\, p(f_i|RO - O)$$

The evidence (also termed normalizing constant) may be calculated since the sum of the posterior probabilities must equal one.

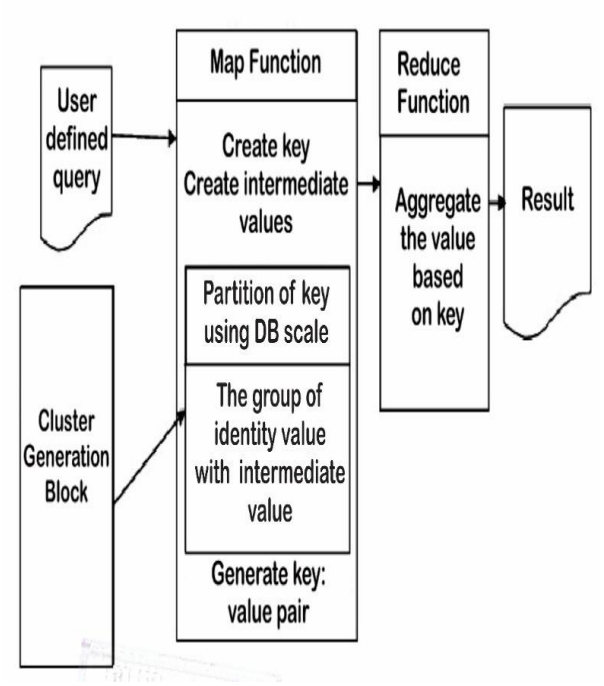

Figure 2 sows that processing of Map reduces file system using DB scale clustering technique

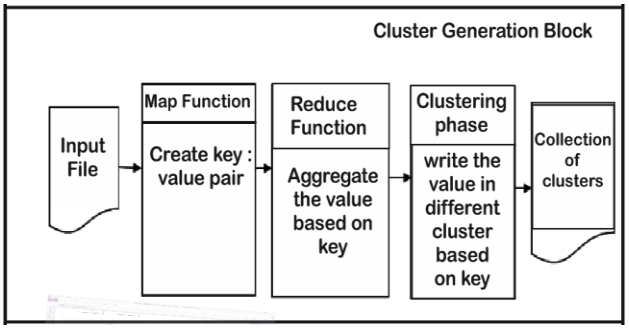

Figure 3 sows that processing of Map reduces for the generation of data based on cluster node.

4. EXPERIMENTAL ANALYSIS

The proposed algorithm imlemented in Hadoop tools.the hadoop tool is open source linuex based freamwork. The hadoop freamwork proced the MapReduce function for the analysis of data. The propgramming model of MapReduces in JAVA JDK compliation tool. For the evlaution of proposed model used word count program for the analysis of different dataset.

Table 1: Shows that the performance evaluation of number of word, number of count, mapreduce and input mapreduce using batch-tweet dataset.

Dataset	No. of word	No. of count	Mapreduce	Improved mapreduce
Batch-tweet	10500	5010	2700	3200
Batch-tweet	11000	6000	3200	3350
Batch-tweet	10350	4500	4000	4200
Batch-tweet	12450	4700	3700	4000

Table 2: Shows that the performance evaluation of number of word, number of count, mapreduce and input mapreduce using nrt-tweet dataset.

Dataset	No. of word	No. of count	Mapreduce	Improved mapreduce
nrt-tweet	10500	4500	3200	3450
nrt-tweet	11000	4700	4000	4500
nrt-tweet	10350	6500	3700	3800
nrt-tweet	12450	5000	3000	4200
nrt-tweet	11750	5010	2700	3200

Table 3: Shows that the performance evaluation of number of word, number of count, mapreduce and input mapreduce using tpcds-setup dataset.

Dataset	No. of word	No. of count	Mapreduce	Improved mapreduce
tpcds-setup	10500	4700	2800	3100
tpcds-setup	10000	4800	2400	2600
tpcds-setup	10500	5000	3600	4550
tpcds-setup	11000	5510	3550	4000
tpcds-setup	10350	6000	3700	3900

Table 4: Shows that the performance evaluation of number of word, number of count, mapreduce and input mapreduce using zipcode-setup dataset.

Dataset	No. of word	No. of count	Mapreduce	Improved mapreduce
zipcode-setup	12000	5010	3000	3800
zipcode-setup	8500	4000	2700	3250
zipcode-setup	14000	6000	3200	3900
zipcode-setup	11350	4500	4000	4350
zipcode-setup	12450	4700	3700	4000

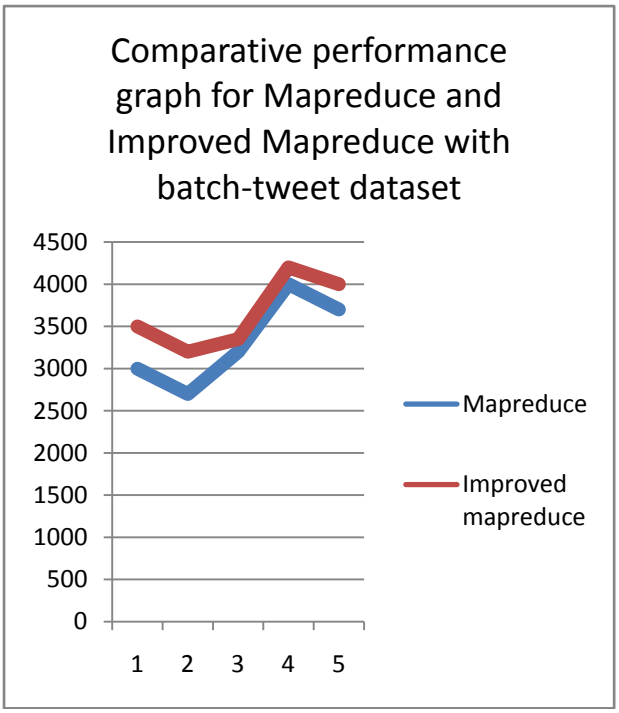

Figure 4: Shows that the comparative performance evaluation graphs using Mapreduce and Improved mapreduce with batch-tweet dataset.

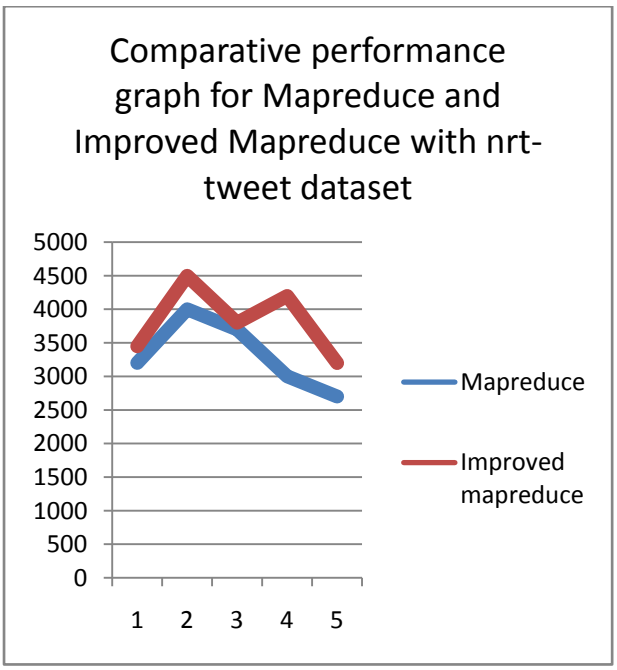

Figure 5: Shows that the comparative performance evaluation graphs using Mapreduce and Improved mapreduce with nrt-tweet dataset.

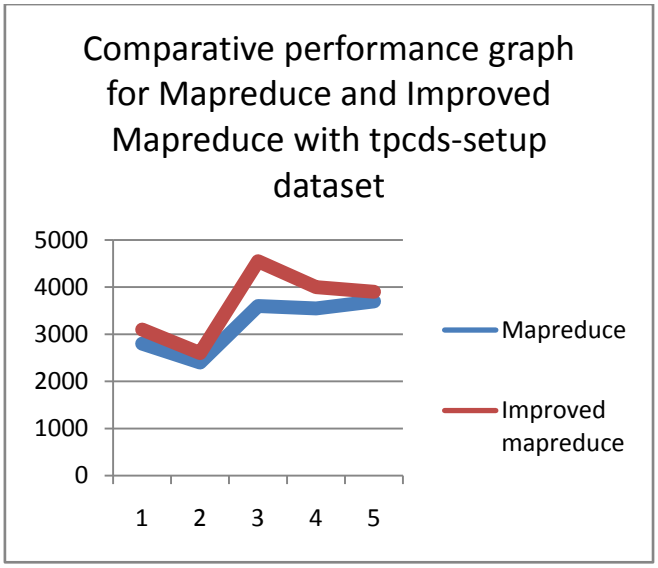

Figure 6: Shows that the comparative performance evaluation graphs using Mapreduce and Improved mapreduce with tpcds-setup dataset.

5. CONCLUSION & FUTURE SCOPE

In this paper modified the map reduces programming model using DB scale clustering technique. the modified model of Map reduces simulated in Hadoop framework. The processing of map reduces function based on two attribute data one is cluster of data and other is key value. The value of key generates the processing of group of data for the process of analysis. The modified map reduces function component encapsulate the processing of DB scale clustering technique. the DB scale clustering technique define the value of range of group data for the processing of map cerates block .More than 30% improvement has been observed in some cases of applications which are quite impressive from computational perspective. It has also been observed that, the time for clustering becomes almost stationary for higher number of nodes even the input volume of data has been increased from 7 million to 10 million. Thus, DB scale being very useful clustering technique, using it in cloud environment for processing Big Data has some inherent advantages and may be used for various application. For the modification of map reduces programming model used DB scale clustering technique. the DB scale clustering technique perform very well in terms of limited data. But the processing of data change in petabyte the grouping rule and policy is suffered for the creation of data node. In future the processing of petabyte data used some time based optimization technique.

6. REFERENCES

[1] Carson Kai-Sang Leung, Richard Kyle MacKinnon and Fan Jiang "Reducing the Search Space for Big Data Mining for Interesting Patterns from Uncertain Data", IEEE, 2014, Pp 315-322.

[2] Rama Satish K. V. and Dr. N. P. Kavya "Big Data Processing with harnessing Hadoop - MapReduce for Optimizing Analytical Workloads", IEEE, 2014, Pp 49-54.

[3] Seungwoo Jeon, Bonghee Hong and Byungsoo Kim "Big Data Processing for Prediction of Traffic Time based on Vertical Data Arrangement", IEEE, 2014, Pp 327-333.

[4] Rajiv Ranjan "Modeling and Simulation in Performance Optimization of Big Data Processing Frameworks", IEEE, 2014, Pp 14-19.

[5] Muhammad MazharUllahRathore, Anand Paul, Awais Ahmad, Bo-Wei Chen, Bormin Huang, and Wen Ji "Real-Time Big Data Analytical Architecture for Remote Sensing Application", IEEE, 2015, Pp 1-12.

[6] Jyoti V Gautam, Harshadkumar B Prajapati, Vipul K Dabhi and Sanjay Chaudhary "A Survey on Job Scheduling Algorithms in Big Data Processing", IEEE, 2015, Pp 1-11.

[7] Alfred Daniel, Anand Paul and Awais Ahmad "Near Real-Time Big Data Analysis on Vehicular Networks", International Conference on Soft-Computing and Network Security, 2015, Pp 1-7.

[8] Chun-Wei Tsai, Chin-Feng Lai, Ming-Chao Chiang and Laurence T. Yang "Data Mining for Internet of Things: A Survey", IEEE, 2014, Pp 77-97.

[9] Albert Bifet "Mining Big Data in Real Time", Informatica, 2013, Pp 15-20.

[10] GwangbumPyun ,Unil Yun and Keun Ho Ryu "Efficient frequent pattern mining based on Linear Prefix tree", Elsevier, 2013, Pp 125-139.

[11] Unil Yun and Keun Ho Ryu "Approximate weighted frequent pattern mining with/without noisy environments", Elsevier, 2010, Pp 73-82.

[12] Zhi-Hua Zhou, Nitesh V. Chawla, YaochuJin and Graham J. Williams "Big Data Opportunities and Challenges: Discussions from Data Analytics Perspectives", IEEE, 2011, Pp 1-20

[13] Boris Novikov, Natalia Vassilieva and Anna Yarygina "Querying Big Data", International Conference on Computer Systems and Technologies, 2012, Pp 1-10.

[14] Liwen Sun, Reynold Cheng, David W. Cheung and Jiefeng Cheng "Mining Uncertain Data with Probabilistic Guarantees", ACM, 2010, Pp 273-282.

[15] Yuxuan Li, James Bailey, Lars Kulik and Jian Pei "Mining Probabilistic Frequent Spatio-Temporal Sequential Patterns with Gap Constraints from Uncertain Databases", Pp 1-10.

[16] Carson Kai-Sang Leung and Fan Jiang "Frequent Itemset Mining of Uncertain Data Streams Using the Damped Window Model", ACM, 2011, Pp 950-955.