International Journal of Engineering & Technology, 7 (2.31) (2018) 19-22

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Optimizing the performance of hadoop clusters through efficient cluster management techniques

K.S. Shraddha Bollamma[1*], S. Manishankar[2], M.V. Vishnu[3]

[1]*Department of Computer Science, Amrita School of Arts and Sciences, Amrita Vishwa Vidyapeetham, Mysore, India.*
[2]*Department of Computer Science, Amrita School of Arts and Sciences, Amrita Vishwa Vidyapeetham, Mysore, India.*
[3]*Department of Computer Science, Amrita School of Arts and Sciences, Amrita Vishwa Vidyapeetham, Mysore, India.*
**Corresponding author E-mail:shraddhasubbaiah@gmail.com*

Abstract

The necessity for processing the huge data has become a critical task in the age of Internet, even though data processing has evolved into a next generation level still data processing and information extraction has many problems to solve. With the increase in data size retrieving useful information with a given span of time is a herculean task. The most optimal solution that has been adopted is usage of distributed computing environment supporting data processing involving suitable model architecture with large complex structure. Although processing has achieved good amount of improvement, efficiency, energy utilization and accuracy has been compromised. The research aims to propose an efficient environment for data processing with optimized energy utilization and increased performance. Hadoop environment common and popular among big data processing platform has been chosen as base for enhancement. Creating a multi node Hadoop cluster architecture on top of which an efficient cluster monitor is setup and an algorithm to manage efficiency of the cluster is formulated. Cluster monitor is incorporated with Zoo keeper, Yarn (Node and resource manager). Zoo keeper does the monitoring of cluster nodes of the distributed system and identifies critical performance problems. Yarn plays a vital role in managing the resources efficiently and controlling the nodes with the help of hybrid scheduler algorithm. Thus this integrated platform helps in monitoring the distributed cluster as well as improving the performance of the overall Big Data processing.

Keywords: *Big data, hadoop, heterogeneous clusters, map reduce, yarn, zookeeper.*

1. Introduction

Big data describes a humongous amount of data. Data is categorized as structured, unstructured, and semi-structured for mining meaningful information.
To extract the value of a data, big data is applied to data analysis methods. The basic characteristics of Big Data are 3v that is velocity, volume and variety. In some cases, there are 4v that includes veracity. The key enablers of big data are increased storage capacity, efficient processing power and data availability. The speed of data arrival, storage, and retrieval is cited at speed. Big data growth has risen from gigabytes in 2005 to Exabyte in 2015. Maintaining large amounts of data in old databases is difficult and requires new mechanisms. Various types of data include traditional, original, structured, and semi-structured types. Sources include e-mail, log files, sensors, social media forums, search indexes and more. Text, audio, video and imagery are available in textual and numerical forms, mainly from banking and insurance services insurance services, followed by manufacturing and social networking sites. It acts as the most popular, open-source software framework that is worn by major portals like Yahoo and Facebook.
Hadoop is an tool which is available as an open source with a architecture and enables a distributed processing of data.. Hadoop is an incorporation of Mapper algorithm Reducer algorithm and combiner algorithm, distributed file system on top of Hadoop and a number of associated projects such as ZooKeeper. MapReduce and Hadoop are the major components of Hadoop. Clusters are made up generally with set of associated data handling machines. The above mentioned system refers to a Hadoop Cluster. It has three main components namely, Master node, slave node and client node.
Yet Another Resource Negotiator (YARN) is in charge for cluster Resource Management, setting up various applications that can run on YARN, MapReduce is just one option. MapReduce on yarn has components like, Client, Resource Manager, NodeManager, MapReduce Application Master and HDFS.

2. Related work

The scholars have done a lot of research in the way of the Optimizing the performance of Hadoop clusters. For example,

A study on performance optimization of hadoop clusters

[1] A Big-data Elasticity Cloud management system is built with a job Aware scheduler, a system that allow a virtual cluster on top of which Map and reduce operation can be done is made to mechanically alter its mass to the workload.[2]Considers an accomplishment of High Availability-Mapping and reducing function in the cloud setting. Initial stage planned a standardized Mapper and Reducer structure for heterogeneous architecture. Next, a co-processor token mechanism for managing the coprocessor scalability and fault tolerance issue. Lastly, a lightweight virtualization based cloud platform for low overhead and simple operation. Reference[3]proposes the rapport amid infrastructure components and power utilization of the cloud computing environment, and propose a optimization technique is

introduced for computing resource in cloud platform.[4] Apache Spark architecture was modeled with memory operation supporting computations. To assist execution and guarantee high performance of Spark-based algorithms in a compound cloud computing environment. In[5],a new YARN scheduler, named HASTE, which reduces the time span of Map Reduce jobs in YARN through provision of requested resources information, resource capacities, and tasks dependencies. Reference[6] takes the heterogeneity of nodes into account, and puts forward a placement strategy according to the proportion of the storage of data. Reference[7]Considers the data generated by the reduce phase that know how to be divided into more partitions, adjusts the partition size to stabilize the load vigorously, although by not considering the influence of nodes' computing power.[8]Proposes a greedy priority algorithm, which adjusts the load by the calculation ability of each node in advance, but it doesn't take the dynamic changes in the computing power of the nodes into account. Reference[9]has proposed two algorithms for job ordering optimization and MapReduce slot configuration optimization. These algorithms aim at increasing the system performance and also improve the resource utilization.

A study on existing techniques in Big data analytics with multiple clusters for heterogeneous data management

[10]data stored digital format is considered and the measurement of the distance is calculated, set that does not apply to the classification facts, and , as there is no accepted distance amid the two classification data centers.[11]Considers the data-intensive applications in succession on the Hadoop cluster, the data residency scheme mechanically balances the sum of information stored in each node to attain enhanced performance of data processing[12] Considers a strategy known as enhanced dynamic slot allocation for Hadoop by retaining the functionalities of the space based model.[13]Proposed a range of methods to meet the various problems that be present on offer by the Mapper and reducer based Framework of Distributed File System on Hadoop. Reference[14]found an effective way to store unstructured data and how to get data. Built a large data application that fetches public data available in twitter, stored in cluster and then invokes data retrieved from HBase through state transfer method for data analysis. In reference[15]a yellow cost function method is used for representing the cluster center of the classification feature, taking into account the mode values. Therefore, there is only one feature as the cluster core. Incorrect allocation of values results in incorrect clustering results. Reference[16]proposed the global expenditure function of the clustering dataset by altering the cost function. In[17],for dividing the data points, the involvement of features of cluster setup is considered by the covariance probability method.[18] Has discussed the plan of naming a cluster core by means of classifying the leading data item and new distance calculation. The entropy based on the alternative Manhattan distance is used. Reference[19]aims at comparing the execution time of Word Count under varying conditions. Single node is setup to execute the Map Reduce Word Count. Effects on the execution time on setting the number of reduced tasks and size in input files are studied. Some of the advanced cluster handling can be done with help of grid based environment[20] and efficient fault tolerance techniques can be adopted to enhance cluster[21]and adaptive approach of scheduling can be integrated in the cluster.

3. Proposed system

The study on existing techniques and literature survey gives a picture that most of the work has been carried out is in improving the speed or processing power of Big data. One of the major areas left out is how efficiently a processing environment can be utilized and monitored. The proposed research work concentrates on modeling a high-performance environment with the help of Hadoop clusters. In general, all the High-performance environments require high configured systems or super computers to support them. The idea here is to setup a powerful computing environment with cluster of minimal configuration machines, as all the machines are of nominal processing power utilizing them efficiently and monitoring them is a highly required task. The environment for the research is shaped with distributed platform of multiple clusters.

The distributed system is back boned with Hadoop clusters each cluster containing Data Node, NameNode, Job tracker and secondary Name Node. A requirement arises here that is how in a distributed environment of multiple data nodes performance can be optimized. Normally in a distributed environment chances of performance degradations are quiet high in-order stabilize it there is a need for a monitor for the nodes. Proposed researches second objective is fulfilled with help of Zoo keeper which acts as a monitor for the Z-nodes present in the Hadoop clusters. Thus, the Hadoop cluster is vigilantly monitored against performance issues. A presence of a monitor only cannot improve or optimize performance it requires a strong scheduling algorithm which works incoherence with the job tracker and resource negotiator. Here Yarn comes into the picture and algorithm is performed on top of Yarn scheduler. The algorithm takes in parameters like number of jobs in a cluster, execution time of jobs, number of incomplete jobs, number of Maps and Reduces, energy of a cluster etc. The algorithm coins an optimization factor with respect to each parameter and with the help of monitor does the setting up of the distributed cluster system. Overall performance of the system can be viewed in terms of decreased execution time, waiting time and increased storage, accuracy and performance. So, the proposed platform can be hence forth called as an optimized one. The structural design in Fig 1 of the proposed system mainly consist a distributed computing environment which basically does the data processing. The user gives inputs, submit jobs and data with the help of an interface made to the cluster. Interface is programmed using java and Hadoop jar packages. The center part of the architecture that is the distributed cluster is set up with minimal configuration machines grouped into cluster.

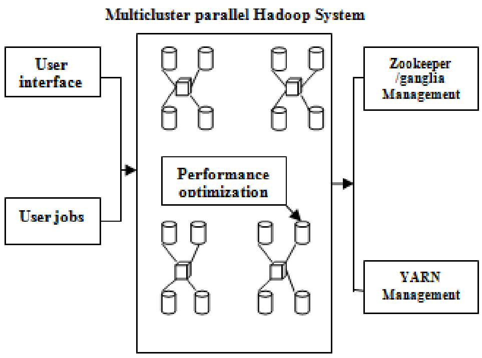

Fig. 1: Architecture of the proposed platform

Hadoop Architecture is imposed in system. So, the cluster consists of multiple nodes with master slave architecture and the master will be configured with a Name node. Then new data node can be added to the slaves as well. With Hadoop service started the cluster can be handled by the Hadoop master and jobs can be executed in the Hadoop cluster. Once the Hadoop cluster is setup the vital roles for a monitor and performance optimizer come. The monitoring task is done by the Zoo keeper and the scheduling and performance optimization is done by Yarn manger as seen in the Fig1.

4. Yarn scheduler algorithm

The proposed system portrays its unique way of improving the resource utilization by intervening the scheduler with help of algorithm which can prevent the performance degradation by

applying threshold to various parameters affecting Yarn's performance.
As in any system of resource scheduling a queue is set up with resources that are ready to be utilized, denoted as Qr
Next important task is to identify the Job set which is put in target and must be executed. Denoted as Jr.
Yarn calculates the running time required by each job as

1. Tj =Tjcurrent-Tjsubmitted. (1)

Hadoop performance monitor gives average number of jobs with which we calculate cluster throughput denoted as Ct

2. Ct= number of jobs in cluster /unit time. (2)

Average cluster through Avg(Ct) for a duration is taken as threshold of cluster execution
Job are categorized as progressive jobs Pj and non-progressive NPj based on the ratio of usage between Map and reduce tasks.

3. Pj → Avg(Tj)<Tj (3)
4. NPj→ Avg(Tj)>Tj (4)

Clusters are categorized as Fast Executing Fc and Slow cluster Sc based on the cluster threshold

5. Fc→Avg(Ct)<ct. (5)
6. Sc→Avg(Ct)>ct. (6)

Scheduler in Yarn checks whether it can execute a job set in a cluster by comparing through put required for the job set and current throughput of the cluster.

7. $\sum AvaiCapacity = \left(\frac{NumberofPj}{NumberofNPj}\right) * (Avg(Ct))$ (7)

Total Available capacity $\sum AvaiCapacity$ of the clusters is put as bench mark for resource requirement
Requesting set of job that run in Qr Rj demand job

7. $\sum Rj < AvailCapacity$ (8)

Request job crosses threshold and total available capacity is a more, so the job set executed.

8. $\sum Rj > AvailCapacity$ (9)

Requested set of jobs need waiting time denoted as Wj
Performance factor FPj is calculated is

9. FPj=$\left(\frac{\sum AvaiCapacity}{Rj*PJ}\right) * Avg(\text{Ct})$ (10)

This factor can be used as watch factor for cluster monitoring
FPj is recorded and passed in time to time duration.
The Yarn monitor keeps an alert with Zoo keeper to monitor the cluster efficiently.

5. Implementation and results

Research idea proposed with optimization algorithm is implemented in a Hadoop cluster platform. Hadoop platform gives flexibility for the researcher to run jobs of various loads.
Jobs are being assigned in to the job queue as mentioned in the algorithm. The task arises here to evaluate the performance of multiple jobs by comparing the schedulers present in YARN(FIFO, Capacity and Fair) in respective manner. The cluster load is then calculated with the help of running time of the job. The average cluster performance is observed each time and variations are recorded. The performance metrics is based on three dimensions, CPU utilization, memory utilization and Load.

To do the performance evaluation and monitoring required for the research ganglia is deployed. Ganglia keep track of the processes happening in the cluster. A cluster named Amrita is setup in Hadoop and it has been configured to ganglia. It monitors the cluster performance which is required for the yarn algorithm to optimize cluster. Tracking of each variation happening with respect to CPU cycles, Memory and cluster load are done. The algorithm takes all this data as input to calculate the performance factor. The jobs given are further classified in to progressive and non-progressive and the resource optimization is done majorly.
Observation in the form of performance result where taken and compared with respect to job types. Three types of job have been prominently taken in to account a word count job, calculation of number sum and industrial case study of a retail store data analysis. All the three jobs are input with high load data and observed the cluster performance.

Fig. 2: Cluster live dash board

The figure depicted above show the cluster Live dash board which gives us over view of the task and performance happening in cluster. Load details are put in and memory, CPU and network load are being monitored with respect to time in scale of hour, day and week. Each report can be downloaded as a csv file which is a source of input to the Yarn scheduler algorithm, peeks, changes and variation with respect to job and time are done by the performance optimizer. Zookeeper is kept as system to observe the data storage management and Znode of amrita cluster analysis.
Three types of jobs were concurrently scheduled, and all performance variations were recorded, before introduction of yarn based scheduler the job running resulted in high variations in the cluster leading to an increased server load as depicted with red color in the below figure 3. Load and memory utilization as well as CPU utilization were recorded high leading to a cluster CPU and memory trash.

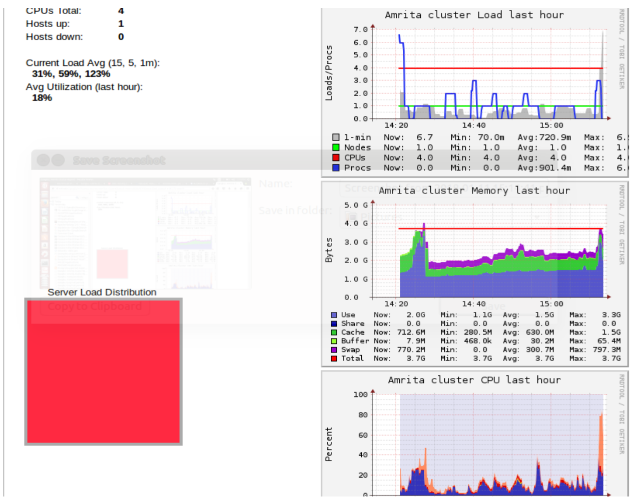

Fig. 3: Performance degradation without yarn optimization

Introduction of the algorithm and performance monitor made a considerable change in the system load. The performance factor calculation and job profiling helped to make the jobs run faster in the cluster, and load of the server is reduced.

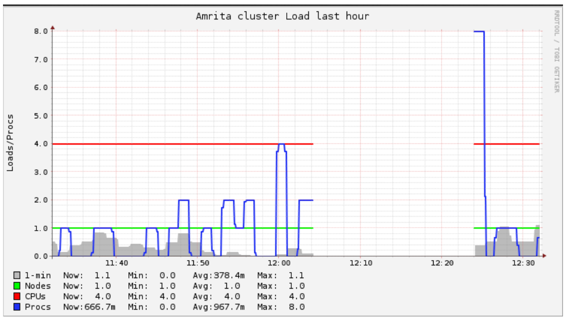

Fig. 4: Load balanced with yarn

The Fig 4 shows the optimized performance of concurrently running task in the presence of optimizer. Load which was observed in the previous instances without the optimizer has come down considerably because the job profiling has helped a lot in taking decision on the job execution with respect cluster load. The schedulers like FIFO, Capacity and Fair are considered in the Yarn, and optimizer chooses the scheduling scheme based on the performance factor and the job type. thus, the scheme of proposed optimizer has helped the system performance improve.

The figure 5 depicts variations and improvements in the CPU utilization of proposed Cluster. Thus, overall reduction of load, improvement of CPU utilization has helped in optimizing the performance.

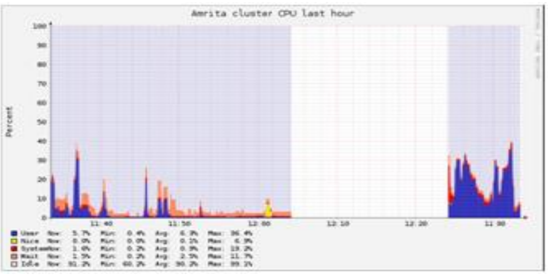

Fig. 5: CPU utilization

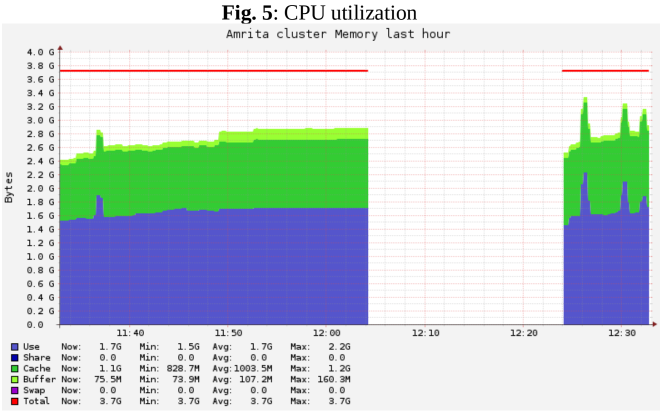

Fig. 6: Utilization of memory in cluster

Finally, Fig 6 depicts the Utilization of Memory in Cluster improvement in the presence of the proposed optimizer. Thus, as observation results depict there has been a considerable improvement in the cluster performance when the optimizer is introduced.

6. Conclusion

The proposed system exhibits a remarkable improvement in performance of Hadoop cluster. System is an integrated combination Hadoop, Zookeeper and Ganglia which makes the system run various types of job concurrently with minimal load. Load and utilization of resource is kept under the observation of the resource manager on top of which the proposed algorithm and approach runs. . There has been a comparison of Yarn scheduler carried out with respect to various types of jobs. The job profiling and optimizing the load has greatly helped in improving the efficiency of the cluster. Thus, overall there is an increase of performance and reduced CPU and memory utilization happening in the proposed cluster. In future the proposed system can be set in to a varied heterogeneous distributed environment and tested scalability and performance with wide-ranging set of jobs.

References

[1] Chalvantzis N, Konstantinou I & Kozyris N, “BBQ: Elastic MapReduce over cloud platforms”, *Proceedings-17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGRID*, (2017), pp. 766–771.

[2] Luo L, Wu W, Di D & Zhang F, “A resource scheduling algorithm of cloud computing based on energy efficient optimization methods”, *IEEE Green Comput. Conf.*, (2012), pp. 1–6.

[3] Cheng D, Rao J, Guo Y, Jiang C & Zhou X, “Improving Performance of Heterogeneous MapReduce Clusters with Adaptive Task Tuning”, *IEEE Trans. Parallel Distrib. Syst.*, Vol.28, No.3, (2017), pp.774–786.

[4] Huang W, Meng L, Zhang D & Zhang W, “In-Memory Parallel Processing of Massive Remotely Sensed Data Using an Apache Spark on Hadoop YARN Model”, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, Vol.10, No.1, (2017), pp.3–19.

[5] Xie J, Yin S, Ruan X, Ding Z, Tian Y, Majors J, Manzanares A & Qin X, “Improving mapreduce performance through data placement in heterogeneous hadoop clusters”, *IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW)*, (2010), pp.1-9.

[6] Slagter K, Hsu CH, Chung YC & Zhang D, “An improved partitioning mechanism for optimizing massive data analysis using Map Reduce”, *J. Supercomput.*, Vol.66, No.1, (2013), pp.539–555.

[7] Tan Y, Wang W, Wu Q & Lin J, “An Implementation of Heterogeneous Architecture Based MapReduce in the Clouds”, *2nd International Conference on Cloud Computing and Internet of Things (CCIOT)*, (2016), pp.16–20.

[8] Gautam JV, Prajapati HB, Dabhi VK & Chaudhary S, “A survey on job scheduling algorithms in Big data processing”, *IEEE Int. Conf. Electr. Comput. Commun. Technol.*, (2015), pp.1–11.

[9] Barbieru C & Pop F, “Soft real-time hadoop scheduler for big data processing in smart cities”, *Proc. - Int. Conf. Adv. Inf. Netw. Appl. AINA*, (2016), pp. 863–870.

[10] Dean J & Ghemawat S, “MapReduce: Simplified Data Processing on Large Clusters”, *Commun. ACM*, Vol.51, No.1, (2008).

[11] Arasanal RM & Rumani DU, “Improving mapreduce performance through complexity and performance based data placement in heterogeneous hadoop clusters”, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, (2013), pp. 115–125.

[12] Deepthy Ashok M, Subramanian K & Prabhu A, “Dynamic Slot Allocation for Map Reduce Clusters”, *International Journal of Control Theory and Applications*, (2017).

[13] Nandimath J, Banerjee E, Patil A, Kakade P, Vaidya S & Chaturvedi D, “Big data analysis using Apache Hadoop”, *IEEE 14th International Conference on Information Reuse & Integration (IRI)*, (2013), pp.700–703.

[14] Das TK & Kumar PM, “BIG Data Analytics: A Framework for Unstructured Data Analysis”, *Int. J. Eng. Sci. Technol.*, Vol.5, No.1, (2013), pp.153–156.

[15] Jain AK, Murty MN & Flynn PJ, “Data clustering: a review”, *ACM Comput. Surv.*, Vol.31, No.3, (1999), pp.264–323.

[16] Ahmad A & Dey L, “A k-mean clustering algorithm for mixed numeric and categorical data”, *Data Knowl. Eng.*, Vol.63, No.2, (2007), pp.503–527.

[17] Velmurugan T, “Evaluation of k-Medoids and Fuzzy C-Means clustering algorithms for clustering telecommunication data”, *Int. Conf. Emerg. Trends Sci. Eng. Technol.*, (2012), pp.115–120.

[18] Kim M & Ramakrishna RS, “Projected clustering for categorical datasets”, *Pattern Recognit. Lett.*, Vol.27, No.12, (2006), pp.1405–1417, 2006.

[19] Subramanian K & Prabhu A, “Simplified Data Analysis of Big Data in Map Reduce”, 2017.

[20] Gokuldev S, Rao A & Karthi, “An EMT scheduling approach with optimum load balancing in computational grid”, *Int. J. Appl. Eng. Res.*, Vol.11, No.8, (2016), pp.5753–5757.

[21] Gokuldev S & Radhakrishnan R, “An adaptive job scheduling with efficient fault tolerance strategy in computational grid”, *Int. J. Eng. Technol.*, Vol.6, No.4, (2014), pp.1793–1798.