International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT228369

Analysis of Data Performance that Reduces Resource Utilization Overheads and Increases the Efficiency

Anilkumar Ambore*[1], Udaya Rani V[2]
*[1]Research Scholar, VTU, Department of CSE, REVA ITM, Bangalore, India
[2] Department of CSE, REVA ITM, Bangalore, India

Article Info
Volume 8, Issue 3
Page Number : 191-195

Publication Issue :
May-June-2022

Article History
Accepted: 10 May 2022
Published: 30 May 2022

ABSTRACT

In today, the size of the data is increasing at a random speed. So, this leads to processing of Big data. When we compare this in business applications where the volume of data is huge and at the same time it should be processed in efficient manner. Traditional system fails to process the bigdata because most of the data in bigdata is unstructured. To improve performance in distributed data processing resource utilization plays vital role. There are resource gaps develop while execution occurs. This is more frequent in heterogeneous environment. In the previous techniques there is wastage or not efficient usage of resources. To process data in distributed environment multiple platforms used such as Apache Hadoop, Apache Spark etc. Here we develop new algorithm that reduces the usage of resources and increases the performances. The algorithm implemented in Apache Spark distributed environment. The experimental results indicate efficient utilization of resources and increase in performance.

Keywords: Big Data, Resource Utilization, Spark, Hadoop, Cloud Computing

I. INTRODUCTION

Now the term Bigdata is common in all the domains. The data will be generated from all the sources this intern leads how to analyse and process. Many authors defined bigdata in different ways. As per Wikipedia [1] Huge information alludes to informational collections that are excessively enormous or complex to be managed by customary information handling application programming. Gartner [2] Big data means volume, velocity and variety should be high and it should be process in efficient manner. Author Gueyoung Jung [3] represented big data as Model of Three V. According to McKinsey [4] now in the world surplus of data is generating and to analyse this is termed as bigdata According to O'Reilly [5] data that surpasses the handling limit of ordinary data set framework tools. Microsoft [6] is intended to deal with the ingestion, handling, and examination of information that is excessively huge or complex for conventional data set framework tools. As per IBM [7] data coming in quicker, from variety sources, and in different formats. In paper[8] different data placement strategies were

discussed. Also, in the paper [9] the author outlined the new technique which reduces the resource wastage.

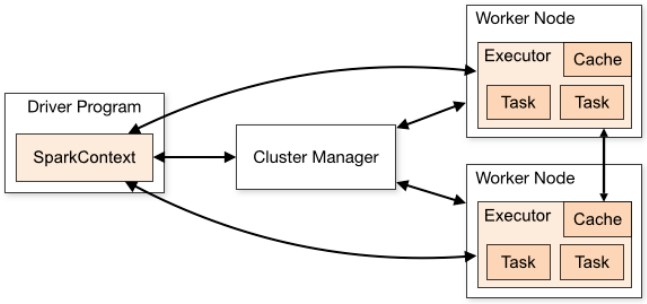

Figure 1. Apache Spark Architecture

Apache Spark architecture mainly contains, Three components viz. Spark Driver, Cluster Manager and Executor. The Spark Driver acts as a Master node which is main entry for Spark shell. Spark diver contains different schedulers also it converts program in different tasks and sends to executor for execution. Cluster manager does get and allocation of jobs. Some of cluster managers are Standalone, Hadoop YARN, Apache Mesos and Kubernetes. Executor is responsible for execution of tasks.

Further, below section II deals with related work, in section III implementation, in section IV Results and discussion and in the section V the conclusion.

II. RELATED WORK

Apache spark [10] is a de facto standard for big data processing in distributed fashion.

TABLE I

SOME PERFORMANCE BOOSTING TECHNIQUES IN SPARK

Technique	Advantages	Disadvantages
Use Join	Uses technique called joining of two tables	Shuffling process will be more internally increases the cost.
Use windows	Window functions can do exactly what we need	It replaces join operation
Use bucketing	Data will be in the organized way. Works for even columns consists of large unique values. avoids multiple shuffles	Same number of buckets on 2 sides of table.
Use cache	Execution and storage memory shared same regions.	Unnecessary caching will reduce the performance
Use checkpoint	Helps to refresh data and performance boosting factor for spark	Execution starts from beginning.
Use User Defined Functions less	Avoid usage of unnecessary UDPs is good idea	Cost increases, spark will forget how the data was distributed before
Use proper file formats	When we use Apache Parquet with spark increases read operation.	Selects only required data

When executing tasks if say task x takes processing time than the expected then it is simply wasting of resource. The time required to complete the assigned task is determined by

S=P/(1-U)

Where the parameters indicate

S means CPU expected service time for the task

P means CPU processing time for the task

U means CPU Utilization time for the task.

Some of the Data reduction methods are Network theory [11], Data Compression [12], Data redundancy

elimination [13], Data pre-processing[14], Dimension reduction[15].

In Network theory primarily data from high dimensional to low dimensional will be done. Data Compression is convenient for reducing data volumes. Reduced size of data is very easy to manage. Data redundancy elimination removes duplicate or repeat data in the dataset. In big data lot of data is duplicated, which internally increases the storage space and processing power. Data pre-processing is the most important phase in big data processing. Dimension reduction where unnecessary columns of dimensions will be removed.

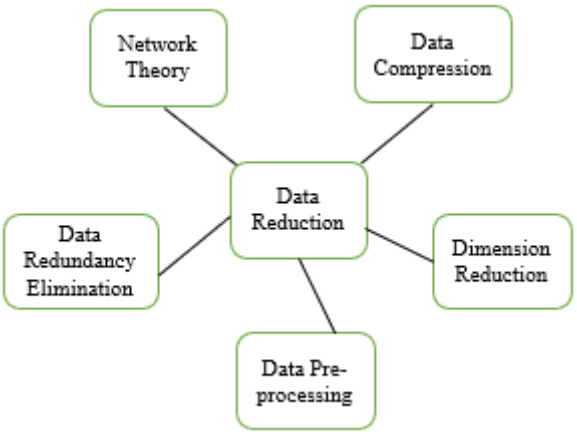

Figure 2. Some data reduction techniques

III.IMPLEMENTATION

Efficient utilization of resources. In this technique, when resource requested it determines and allocates and each server utilized efficiently and reduces wastage in turn increases the performance.
Let servers S {S1, S2…Sn}, Resources R (R1, R2…Rn} so the available resources in the server will be A i.e. A<S. Resource need for the application as N and represented as N= {NR1, NR2…. NRn}

Apache Spark is open distributed computing platform. This also uses Master – Slave architecture. The Proposed technique implemented in Apache Spark, which is an open-source distributed computing framework with the master–slave architecture. Here Two servers used, and the configuration is Server T420-2A, Intel® Xeon® with 2.2 Gz with 16 GB RAM, 2TB HDD and Server T420, Intel® Xeon® with 1.7 Gz with 8GB RAM, 1TB HDD.

IV. RESULTS AND DISCUSSION

.
The experiment performed on Spark pi application which is uses MapReduce framework and executed in Apache Spark platform. For performance evaluation we use Spark Pi application. The application executed in multiple rounds with variation of tasks viz 2, 4, 8, 16, 32.

The result in the figure indicates when resource need increases with available resources then resource utilization is more and improved the performance also the results indicate the performance improvement will not increase even if more resource used.

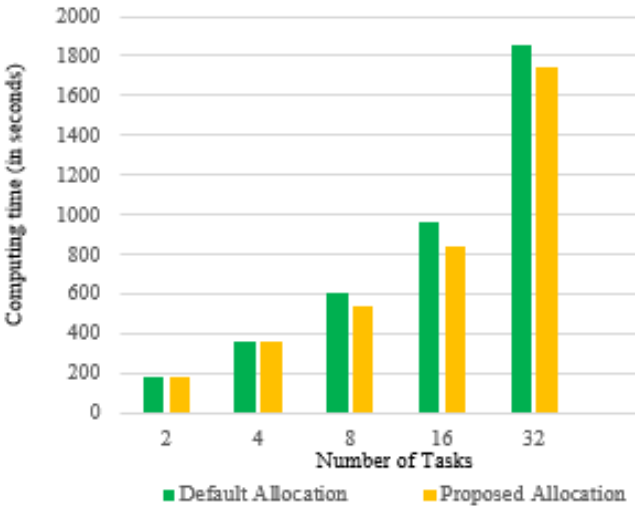

Figure 3. Resource need when 6 CPU cores and 1 GB of RAM used

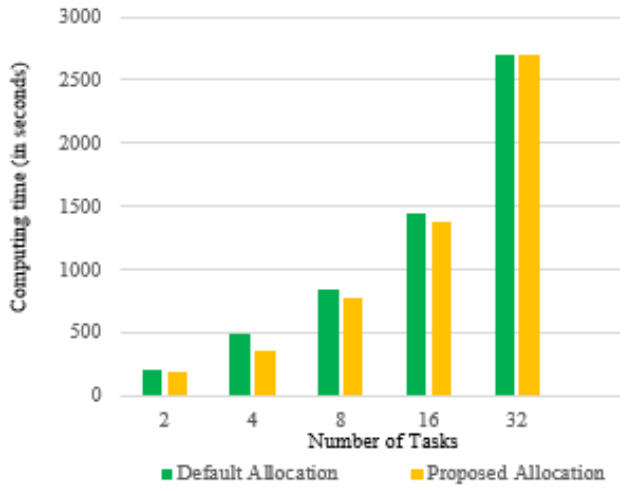

Figure 4. Resource need when 4 CPU cores and 1 GB of RAM used

Figure 5. Resource need when 2 CPU cores and 1 GB of RAM used

Overall, the system performance will be increased with proposed technique. When more tasks are allocated to process the tasks waiting time increases as the available resources will be less. Results indicates resource utilization increased and performance improved.

V. CONCLUSION

Dynamic resource allocation most used in cloud environments. In heterogeneous environment resource allocation affects the performance. When efficiently resources not utilized results in poor performance. To increase the performance the proposed technique is used. The technique is implemented in Apache spark and results indicate increase in the performance in the form of computing time it takes for task execution.

VI. REFERENCES

[1]. www.en.wikipedia.org

[2]. Gartner IT Glossary 2013

[3]. Gueyoung Jung ; Gnanasambandam, N. ; Mukherjee, T. Big Data Analytics2012 International Conference on Communication, Information & Computing Technology (ICCICT), Oct. 19-20, Mumbai, India

[4]. McKinsey Global Institute Big data: The next frontier for innovation, competition, and productivity 2011

[5]. O'Reilly Strata An Introduction to the big data landscape 2012

[6]. Microsoft Enterprise Insights The Big Bang: How the Big Data Explosion Is Changing the World

[7]. IBM Big Data at the Speed of Business 2012

[8]. A. Ambore and U. R. V., "A Survey on Data Placement Strategy in Big Data Heterogeneous Environments," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 439-443, doi: 10.1109/ICOEI.2019.8862676

[9]. Chung, Wu-Chun & Wu, Tsung-Lin & Lee, Yi-Hsuan & Huang, Kuo-Chan & Hsiao, Hung-Chang & Lai, Kuan-Chou. (2020). Minimizing Resource Waste in Heterogeneous Resource Allocation for Data Stream Processing on Clouds. Applied Sciences. 11. 149. 10.3390/app11010149.

[10]. https://towardsdatascience.com/apache-spark-performance-boosting-e072a3ec1179

[11]. Patty JW, Penn EM (2015) Analyzing big data: social choice and measurement. Polit Sci Polit 48(01):95–101

[12]. Yang C et al (2014) A spatiotemporal compression based approach for efficient big data processing on Cloud. J Comput Syst Sci 80(8):1563–1583
[13]. Dong W et al (2011) Tradeoffs in scalable data routing for deduplication clusters. In: FAST
[14]. Xia W et al (2011) SiLo: a similarity-locality based near-exact deduplication scheme with low RAM overhead and high throughput. In: USENIX annual technical conference
[15]. Fan J, Han F, Liu H (2014) Challenges of big data analysis. Nat Sci Rev 1(2):293–314

Cite this article as :

Anilkumar Ambore, Udaya Rani V, "Analysis of Data Performance that Reduces Resource Utilization Overheads and Increases the Efficiency", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 8 Issue 3, pp. 191-195, May-June 2022. Available at doi : https://doi.org/10.32628/CSEIT228369
Journal URL : https://ijsrcseit.com/CSEIT228369