International Journal of Scientific Research and Engineering Development-– Volume 2 Issue 3, May –June 2019

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Analyzing and Improving the Efficiency of Hadoop-Cluster for Big Data Analysis

Deepak Kumar[1], Saurabh Charaya[2]

[1]M.Tech Research Scholar, Department of Computer Science & Engineering, OM Institute of Technology and Management Hisar(Haryana)

[2]Assistant Professor, Head of Department of Computer Science & Engineering, OM Institute of Technology and Management Hisar(Haryana)

Abstract:

The extent of Digitization is continuously increasing by leaps and bounds now a days, resulting in accumulation of large amount of data every second. The data can be a transaction, it can be a social media chat or from any other source. Processing such a Big Data is a very time consuming and tedious task. Though we have advanced systems and techniques to process this data but still there are possibilities of improvements. This paper analysis and explores such possibilities to improve the performance of Hadoop-cluster which is being used to process the big data. In this paper, we first analysis the performance of the cluster and then suggest some method to improve the overall performance of the system.

Keywords: Big Data, Hadoop-cluster, Fault tolerance, MapReduce.

1. INTRODUCTION

Since the inception of computers, our lives have been greatly changed by the growing technologies. Large amount of data is being generated every second and stored in physical storage media. It may be meaningful or may not be but has to be stored indeed. Such a huge storage of data that is exponentially growing too is termed as **Big data**. It is a very tedious task to process such a large amount of data to find out the valuable information. Challenges[1] include examination, get, data curation, look for, sharing, amassing, trade, portrayal, and information assurance. The term consistently suggests just to the usage of perceptive examination or other certain moved systems to expel an impetus from data, and some of the time to a particular size of enlightening gathering. Accuracy in tremendous data may provoke dynamically beyond any doubt essential initiative. Also, better decisions can mean increasingly imperative operational profitability, cost declines and lessened danger.

1.1 Processing Big Data

Google distributed a paper on a methodology called MapReduce[2] in 2004; the MapReduce framework gives a parallel getting ready model and related utilization to process huge proportions of data. With MapReduce, request are part and scattered transversely over parallel center points and dealt with in parallel (the Map step). The

results are then aggregated and passed on (the Reduce step). The structure was particularly successful, so others expected to copy the estimation. Thus, an execution of the MapReduce framework was implemented by an Apache open source adventure named Hadoop [3].

Recent studies exhibit that the usage of an alternate layer configuration is a probability for overseeing colossal data. The Distributed Parallel structure appropriates data over different getting ready units and parallel taking care of units give data significantly faster. This kind of designing installs data into a parallel DBMS, which executes the usage of MapReduce and Hadoop frameworks. This kind of structure wants to make the dealing with power direct to the end customer by using a front end application server.

1.2 Mapreduce structure: Basic engineering

The MapReduce framework[4] is introduced as a fundamental and notable programming model that engages straight forward enhancement of flexible parallel applications to process enormous proportions of data on tremendous gatherings of thing machines [1][2]. In particular, the execution portrayed in the principal paper is predominantly planned to achieve unrivaled on tremendous packs of item PCs[2] One of the basic inclinations of this approach is that it separates the application from the nuances of running a scattered program, for instance, issues on data dissemination, booking, and adjustment to inner disappointment. In this model, the estimation takes a great deal of key/regard sets data and delivers a ton of key/regard consolidates as yield. The MapReduce framework does perform computation using two limits: Map and Reduce. The Map work takes an information consolidate and makes a ton of moderate key/regard sets. The MapReduce framework clusters together all widely appealing regards related with a comparable transitional key I and passes them to the Reduce work. The Reduce work gets a widely appealing key 1 with its game plan of characteristics and solidifies them. Normally just zero or one yield regard is conveyed per Reduce conjuring. The guideline favored point of view of this model is that it empowers extensive figuring's to be viably parallelized and executed to be used as the basic framework for adjustment to interior disappointment. The arrangement of the MapReduce framework has considered going with essential guidelines [5].

—Unreliable Commodity Hardware with low-cost

—A highly Scalable RAIN Cluster.

— Abstracted but highly parallel

— Easy to administer but fault tolerant.

1.3 Fault Tolerance

Fault tolerance[6] is described as, when the system limits suitably with no data lost paying little heed to whether some hardware parts of the structure has failed. It is hard to accomplish penny percent Fault tolerance anyway faults can be persevered up somewhat. HDFS give high throughput to get to data application and proper to have tremendous instructive records as their information [7]. The essential inspiration driving this Fault tolerances to oust occasionally happening frustrations, which happens as a rule and bothers the standard

working of the structure. Single point disillusionment center points happen when a lone center dissatisfaction makes the entire structure crashes. The fundamental commitment of Fault tolerances to empty such center point which bothers the entire commonplace working of the structure [8]. The three standard courses of action which are used to convey adjustment to inside disappointment are data replication, heartbeat messages and checkpoint and recovery.

1.3.1 Data replication

An application can indicate the quantity of reproductions of a document at the time it is made, and this number can be changed whenever after that. [9] The name hub settles on all choices concerning square replication. HDFS utilizes a keen imitation position demonstrate for unwavering quality and execution. Advancing copy position makes HDFS extraordinary from most other dispersed record frameworks, and is encouraged by a rack-mindful imitation arrangement approach that utilizes organize data transmission effectively. A similar duplicate of information is situated on a few distinctive processing hubs so when that information duplicate is required it is given by any of the information hub which isn't occupied in speaking with different hubs. The significant favorable position of utilizing this method is to give moment recuperation from hub and information disappointments. Yet, one fundamental impediment is by utilizing this kind of adaptation to internal failure component high memory is devoured in putting away same information on various hubs. There additionally might be potential outcomes of information irregularity. It is often utilized strategy since this procedure gives fast recuperation of information from disappointments. When the information is put away on the hub in the group same duplicate of the information is imitated in two additional hubs. for example absolutely three duplicates of information is situated on a similar bunch.

1.3.2 Heartbeat messages

The answer for the over two issues are heartbeat messages. Here heartbeat message is a message sent from a creator to the endpoint to recognize whether and when the innovator fizzles or is never again accessible. Heartbeat messages are relentless on an occasional repeating premise from the creator's startup until the designer's shutdown. At the point when the collector recognizes absence of heartbeat messages amid a foreseen entry period, the goal may discover that the designer has fizzled, shutdown, or is commonly no longer accessible.

1.3.3 Checkpoint and recuperation

In this strategy, comparable idea as that of rollback is utilized to endure Faults up to some point. After a settled time span interim the duplicate report has been spared and put away. It only rollbacks to the last spare moment that the fault happens and after that it begins performing exchange once more. In general execution time of framework is expanded, in light of the fact that the rollback tasks need to return and check for the last spared steady stages which increment the time. Likewise there is one noteworthy downside of this strategy is that it is very tedious technique contrasted with first strategy however it requires less extra assets.

2. LITERATURE SURVEY

Expanded development of the information and the need to move the information between server farms, requests, an effective information exchange apparatus which can, exchange the information at higher rates as well as handle the shortcomings happened amid the exchange. Absence of fault tolerance mechanisms, would need to retransmit the entire information, if there should arise an occurrence of any fault amid the exchange. Thus, fault tolerance is an essential part of big data tools.

There is enough enthusiasm for the MapReduce(MR) perspective for generous scale data examination [10]. Regardless of the way that the basic control stream of this structure has existed in parallel DBMS for over 20 years, some have thought about MR as a radically new handling model [2][11]. In this paper[11], depiction and connection of the two perfect models have been made. In addition, the both systems with respect to execution and development have been surveyed. To this end, a benchmark containing a social occasion of endeavors that they continued running on an open source type of MR and furthermore on two parallel DBMSs is portrayed. For each task, every system's execution, for various degrees of parallelism on a cluster of 100 nodes, is assessed. Their results revealed some fascinating tradeoffs. Regardless of the way that the methodology to stack data into and tune the execution of parallel DBMSs took any more extended than the MR structure, the observed performance of these DBMSs was too good. Hypothesis has been made about the explanations behind the thrilling execution qualification and considers use thoughts that future systems should take from the two sorts of structures.

Some investigation has been facilitated to execution and appraisal of execution in Hadoop [2][12][13]. Officer executed MapReduce for shared memory systems. Phoenix give a versatile execution with both multi-focus and conventional symmetric multi-processors. Bingsheng et al. made Mars which is a Mapreduce framework for plans multi-processors [12]. The goal of Mars was to disguise the programming capriciousness of GPU by giving clear MapReduce interface.

There are two sorts of record systems managing broad reports for bundles, specifically, parallel file system and Internet service file system [14]. Hadoop distribution file system (HDFS) [2] is an acclaimed Internet service file system that gives the right reflection to data planning in Mapreduce frameworks.

In recent times, High Performance Computing (HPC) systems have been moving from expensive immensely parallel models to clusters of item PCs to use cost and execution benefits properly. Fault tolerance is creating a lot of stress for such long running systems. In a paper[15] a short review of the rate of failures of HPC systems is given and besides think about the Fault tolerance approaches for HPC systems and issues with these strategies. Rollback-recovery techniques are discussed in light of the way that they are commonly used for long-running applications on HPC structures. Specifically, the component necessities of rollback-recovery are inspected and a logical order is made for in excess of twenty common checkpoint/restart game plans.

A paper[6] by T. Cowsalya and S.R. Mugunthan likewise clarified different fault

tolerance techniques for big data Hadoop bunches. A checkpoint based algorithm[16] has been presented by Peng Hu and Wei Dai to enhance fault toerances and recovery of data using hadoop cluster.

3. CONCLUSION

Big data is vast term and there is an immense scope of improvement in the processing of this data as well.
In this paper, we are analyzing the Hadoop system for big data analysis using map reduce algorithm and purposing just one of those ways to improve the performance of a Hadoop-cluster. The basic idea here is to analyze the hadoop-cluster and suggest the best methodology to increase the efficiency.

REFERENCES

[1] Katal, Avita; Wazid, Mohammad; Goudar, R. H. -- [IEEE 2013 Sixth International Conference on Contemporary Computing (IC3) - Noida, India (2013.08.8-2013.08.10)] 2013 S

[2] MapReduce: Simplifed Data Processing on Large Clusters by Jeffrey Dean and Sanjay Ghemawat

[3] http://hadoop.apache.org

[4] Sakr, S., Liu, A., and Fayoumi, A. G. 2013. The family of mapreduce and large-scale data processing systems. ACM Comput. Surv. 46, 1, Article 11 (October 2013)

[5] H.-c. Yang, A. Dasdan, R.-L. Hsiao, and D. S. Parker. Map-reduce-merge: simplified relational data processing on large clusters.

[6] Hadoop architecture and fault tolerance based hadoop clusters in geographically distributed data center by T. Cowsalya and S.R. Mugunthan, SVS College of Engineering, Coimbatore, Tamil Nadu, India

[7] The Hadoop Distributed File System: Architecture and Design" by Dhruba Borthakur, http://hadoop.apache.org/docs/r0.18.0/hdfs_design.pdf

[8] Selic,B. 2004. Fault tolerance techniques for distributed systems. IBM.

[9] Joey Joblonski. Introduction to Hadoop. A Dell technical white paper. http://i.dell.com/sites/content/business/solutions/whitepapers/en/Documents/hadoop-introduction.pdf

[10] A Comparison of Approaches to Large-Scale Data Analysis (by Andrew Pavlo, Erik Paulson Alexander Rasin et al) 2009.

[11] D. A. Patterson. Technical Perspective: The Data Center is the Computer. *Commun. ACM*, 51(1):105–105, 2008.

[12] B.He, W.Fang, Q.Luo, N.Govindaraju, and T.Wang. Mars: a MapReduce framework on graphics processors. ACM, 2008.

[13] C. Ranger, R. Raghuraman, A. Penmetsa, G. Bradski, and C. Kozyrakis. Evaluating mapreduce for multi-core and multiprocessor systems. High-Performance Computer Architecture, International Symposium on, 0:13–24, 2007.

[14] A scalable, high performance file system. http://lustre.org.

[15] A survey of fault tolerance mechanisms and checkpoint/restart implementations for high performance computing systems (Ifeanyi P. Egwutuoha · David Levy · Bran Selic ·
Shiping Chen)

Deepak kumar is a MTECH student in department of Computer Science & Engineering from OM Institute of Technology and Management, Hisar(Haryana)

Er. Saurabh Charaya is presently working as Head of the Department in the Department of Computer Science & Engineering at Om Institute of Technology and Management, Hisar(Haryana) since August, 2011. He has 13 years of rich experience in the field of Computer Science & Engineering. He did MCA and MTech(CSE) from Ch. Devi Lal University, Sirsa. He did MBA(Information Technology) from Amity University, Noida. He is pursuing Ph.D(CSE) from Bhagwant University, Ajmer. He has published 25 research papers in International and 10 research papers in National Journals. He has presented 5 research papers in International

[16] Enhancing fault tolerance based on hadoop cluster: Peng Hu and Wei Dai (2014)

and 5 research papers in National Conferences.

