Analysis of Map Reduce Performance using Prefetching Mechanism

Abhilasha Singh Chetali Parve Priyanka Tripathi

ABSTRACT

In this massive technological atmosphere the number of information generated is increasing at an awfully high rate. Distributed storage system will be used for storing this immense quantity of information. Massive information will be handled by victimization hadoop and map reduces. The multiple node clusters will discover victimization the hadoop framework. Hadoop comes with default distributed file system that is hadoop distributed file system. The mapped is enforced by map technique and that we can see the performance of map reduce task on the bases of bytes written. The amount of bytes written by map task doesn't increase with the amount of files increasing. The explanation is that once the reduce function perform reduces the map output it simply combines the output of the map function. Thus so as to enhance the performance of map reduce task we are going to setup the cluster of nodes in heterogeneous atmosphere and analyze the behavior of map reduce task victimization the perfecting mechanism.

Keywords

Hadoop, hadoop distributed file system, map reduce, prefetching mechanism.

1. INTRODUCTION

Big data is that the term used for a group of knowledge sets thus massive and complicated that it becomes tough to method using hands on management tools or ancient processing applications. huge knowledge suggests that heaps and much of knowledge i.e. knowledge from terabytes to pet bytes The challenges embrace with huge knowledge is how we are able to capture, create, store, search and share this immense quantity of knowledge. The massive knowledge characteristics are volume, variety, truthfulness and rate.

Table 1:Characteristics of Big Data

Characteristics	Description
volume	Huge amount of data
variety	Different forms of data
velocity	Rate at which data flows
veracity	Inconsistency data due to unavailability of data

Big data client situation includes net and e-tailing, adhere card, health care and natural science, banking and money services and retails.Hadoop is that the best resolution for big data problem [1]. Hadoop could be a framework that leave distributed process of enormous large sets across clusters of trade goods computers employing an easy programming model.

Hadoop is an open supply knowledge management with scale out storage and distributed process. Hadoop core parts area unit are HDFS (hadoop distributed file system) and reduce.Hadoop relies on master slave design. Characteristics of hadoop area unit Reliable as a result of fault tolerance, Economical as a result of trade goods hardware, scalable and versatile.

2. RELATED WORK

To increase the performance degradation caused by data transmission, some work is done. Data transmission overhead can be removing by using data prefetching mechanism. To avoid directly modifying the native Hadoop, a bi-directional processing approach is proposed in HPMR [2]: the processing fetches the data from the beginning of the input split while prefetching fetches the data from the end. While the proposal in this paper fetches data from the beginning of input data to reduce the overhead of data transmission at the maximum.

Some researchers focus on optimizing task scheduling algorithms or data replication policies to improve data locality in Map Reduce [3]. These proposals only improve the probability of data locality in MapReduce and may increase the complexity of achieving load balance. The LATE scheduling algorithm is proposed for Map Reduce in heterogeneous environments [4]. M. Zaharia *et al.*, have proposed a delay scheduling algorithm, which addresses the conflict between locality and fairness in shared Map Reduce cluster [5]. In MTSD [6], computing nodes are classified by computing capability and a modified task scheduling algorithm is studied. X. Zhang et al. have studied scheduling with consideration about data locality in homogeneous cluster [7]. DARE is a distributed adaptive data replication algorithm that is sensitive to the heterogeneity of computing nodes, and the more powerful nodes get more data replications [8].

3. AREA OF STUDY

Big data are often handled by exploitation the Hadoop and Map reduce. The multiple node clusters are often got wind of exploitation the Hadoop and map reduce. Our space of interest during this paper is how map reduce works .The environmental execution of Map Reduce task will be outlined. Map reduce may be a programming model and an associated implementation for process and generating massive knowledge sets with a parallel, distributed algorithmic program on a cluster.

A Map reduce program consists of a Map () procedure that performs filtering and sorting (such as sorting students by name into queues, one queue |for every name) and a Reduce () procedure that performs an outline operation (such as enumeration the quantity of scholars in each queue, yielding name frequencies). The "Map reduce engine" (also known as "infrastructure" or "framework") orchestrates the process by marshalling the distributed servers, running the assorted tasks in parallel, managing all communications and knowledge transfers between the assorted components of the system, and providing for redundancy and fault tolerance.The model is impressed by the map and reduces functions normally utilized in useful programming though their purpose within the Map reduce framework isn't an equivalent as in their original forms. The key contributions of the Map reduce framework don't seem to be the particular map and reduce functions, however the measurability and fault-tolerance is achieved for a range of applications by optimizing the execution engine once.

Map reduce libraries are written in several programming languages, with completely different levels of improvement. A well-liked ASCII text file implementation is Apache Hadoop. The name Map reduce originally spoken the proprietary Google technology, however has since been generalized.

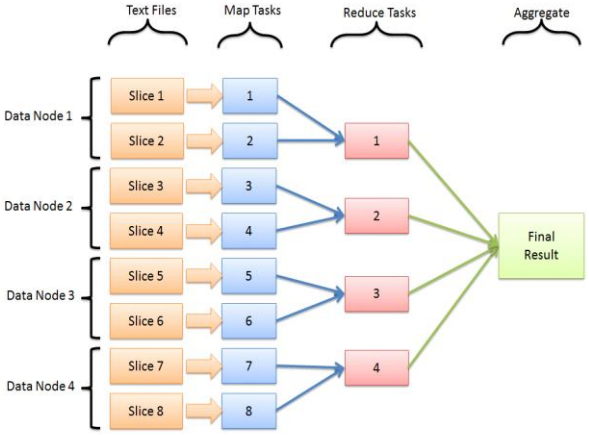

Figure 1: map reduce execution flow

4. RESEARCH DIRECTION

Data locality issues in Hadoop

Map reduce may be a programming model for process giant knowledge sets which are present in hadoop distributed filing system. As we all know that hadoop suffer from data locality downside therefore by using prefetching theme with hadoop hardware we are able to sharply explore for ensuing knowledge blocks to be prefetched thereby avoiding I/O stalls that increased throughout the information interval. With the assistance of prefetching theme we'll be ready to utilize the mainframe processes in knowledge nodes of a hadoop cluster .The hadoop cluster performance are exaggerated simply, and therefore the interaction between the master and slave nodes are exaggerated.

As we all know that once high knowledge intensive applications area unit running on hadoop cluster mainframe and I/O area unit underutilized. To utilize the mainframe and I/O we'll use prefetching theme. Hadoop is mostly used for larger knowledge sets i.e. up to sixty four MB of information or 128 M.B {of knowledge} that the knowledge transfer time of those larger data blocks dominates the I/O interval. That the time interval of this massive knowledge sets area unit get affected owing to underutilized mainframe and I/O stalls. Therefore this motivated North American nation to analyze prefetching technique else I/O performance of HDFS cluster and map reduces.

Data node communicated with the name node so as to work out from wherever the information node got to browse the information or the situation of the information, name node contains the data which data it provides to the corresponding knowledge nodes so knowledge nodes should grasp from wherever he has to browse the information whereas providing the data to the information nodes. The mainframe of the corresponding nodes must look forward to a clear period whereas the nodes area unit act with the master node to accumulate the Meta knowledge.We can in work the prefetching mechanism with {the knowledge|} process procedure in hadoop to stop mainframe from looking forward to master node to deliver Meta knowledge to data nodes.

The basic plan of our prefetching theme is to preload knowledge from native disk and places the information into the native cache of the information node as early as doable with none beginning delay of latest tasks appointed to the information nodes.The original hadoop system will randomly assign tasks to computing nodes and masses the information from native to remote disks whenever the information sets area unit needed.

Prefetching theme

In prefetching mechanism the corresponding computing node that is process {the knowledge|the info|the information} as shown in figure two can prefetch following knowledge before hand on his cache with the assistance of prefetching thread as a result of as that the name node provides data to the information nodes so data nodes know from wherever it's to browse the information ,what is the situation of and from wherever it's to require the information for process therefore before finishing the present process data the information node can fetch the information beforehand on its cache. As presently as doable current knowledge task is finished knowledge node can take following task from its native cache and begin process grasp the information node don’t got to look forward to data that it want from master node .Prefetching mechanism improves the information process tasks and overhead of underutilized mainframe area unit diminished and overall performance of mapped and reducer tasks area unit exaggerated.

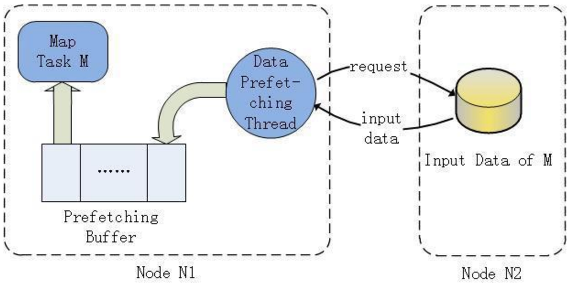

Figure 2: data prefetching mechanism

5. CONCLUSION

Map Reduce is an effective programming model for large-scale data-intensive computing applications. Hadoop, an open-source implementation of Map Reduce, has been widely used. The communication overhead from the big data sets’ transmission affects the performance of Hadoop greatly. To improve the performance of Map Reduce in heterogeneous or shared environments, a data prefetching mechanism is proposed in this paper, which can fetch the data to corresponding compute nodes in advance.We have seen that when prefetching mechanism is applied on mapper function the overall performance of hadoop cluster is increased so in order to futher increase the performance of map reduce we will apply the same prefethcing mechanism in reduce function.

6. REFERENCES

[1] J. Dean and S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters”, Communications of the ACM, vol. 51, no. 1, **(2008**).

[2] S. Seo, I. Jang, K. Woo, I. Kim, J. S. Kim and S. Maeng, “HPMR: Prefetching and Pre-shuffling in Shared Map Reduce Computation Environment”, IEEE International Conference on Cluster Computing and Workshops, **(2009)** August 31-September 4: New Orleans, USA.

[3] S. Khalil, S. A. Salem, S. Nassar and E. M. Saad, “Mapreduce Performance in Heterogeneous

Environments: A Review", International Journal of Scientific & Engineering Research, vol. 4, no. 4, **(2013)**.

[4] J. Xie, S. Yin, X. Ruan, Z. Ding, Y. Tian, J. Majors, A. Manzanares and X. Qin, "Improving MapReduce Performance through Data Placement in Heterogeneous Hadoop Clusters", IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), **(2010)** April 19-23: Arlanta, USA.

[5] M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker and I. Stoica, "Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling", Proceedings of the 5th European conference on Computer systems, **(2010)** April 13-16: Paris, France.

[6] Z. Tang, J. Q. Zhou, K. L. Li and R. X. Li, "MTSD: A task scheduling algorithm for MapReduce base on deadline constraints", IEEE International Symposium on Parallel and Distributed Processing Workshops and PhD Forum (IPDPSW), **(2012)** May 21-25: Shanghai, China.

[7] X. Zhang, Z. Zhong, S. Feng and B. Tu, "Improving Data Locality of MapReduce by Scheduling in Homogeneous Computing Environments", IEEE 9th International Symposium on Parallel and Distributed Processing with Applications (ISPA), **(2011)** May 26-28: Busan, Korea.

[8] C. Abad, Y. Lu and R. Campbell, "DARE: Adaptive Data Replication for Efficient Cluster Scheduling", IEEE International Conference on Cluster Computing (CLUSTER), **(2011)** September 26-30: Austin, USA.