ISSN 2395-1621

Dynamic Slot Allocation Optimization Framework for MapReduce Clusters

[#1]Prof.Riyaz Jamadar, [#2]Karishma Shinde, [#3]Pooja Waghmare, [#4]Shreya Sengupta, [#5]Ganesh Yewate

[#1]Assistant Professor, Dept. of Information Technology, AISSMS IOIT, near RTO, Pune, India
[#2345]UG Student, Dept. of Information Technology, AISSMS IOIT, near RTO, Pune, India

ABSTRACT

The Hadoop MRv1 uses the slot-based resource model with the static configuration of map/reduce slots. There is a strict utility constrain that map tasks can only run on map slots and reduce tasks can only use reduce slots. Due to the rigid execution order between map and reduce tasks in a MapReduce environment, slots can be severely under-utilized, which significantly degrades the performance .In contrast to YARN that gives up the slot-based resource model and propose a container-based model to maximize the resource utilization via unawareness of the types of map/reduce tasks, we keep the slot-based model and propose a dynamic slot utilization optimization system called DynamicMR to improve the performance of Hadoop by maximizing the slots utilization as well as slot utilization efficiency while guaranteeing the fairness across pools. We proposed dynamic slots can be severely under-utilized, which significantly degrades the performance. To address proposes a dynamic slot allocation and scheduling system called DynamicMR consisting of three levels of schedulers, i.e., Dynamic Hadoop Fair Scheduler (DHFS), Dynamic Speculative Task Scheduler (DSTS), and Data Locality Maximization Scheduler (DLMS).The experimental results show that our DynamicMR can improve the system performance significantly while guaranteeing the fairness, by up to 32% 40% for a single job and 19% 28% for MapReduce workloads with multiple jobs, in comparison to Hadoop.

ARTICLE INFO

Article History

Received :13th March 2016

Received in revised form :

15th March 2016

Accepted : 18th March 2016

Published online :

21st March 2016

Keywords— **Slot Allocation , delay scheduler , dynamicMR , MapReduce , hadoop fair sceduler , slot preScheduling .**

I. INTRODUCTION

We live in the age of data , and it's a Big Data .It is not easy to store ,process or access such kind of big data .the .IDC(International Data Corporation) estimates the growth by 2011 to 1.8 zettabytes .A zettabyte is one thousand Exabyte ,one million petabytes ,one billion terabytes. Such big data generates in huge amount for example, the New York Stock Exchange generates about one terabytes of new trade data per day. Facebook generates approximately one petabytes data etc. . big data takes a lot of time for gathering and analysing data. In recent years, MapReduce has become a popular high performance computing paradigm for large-scale data processing in clusters and data centres. In MapReduce job are of a to types, first is map and another one is reduce tasks, where reduce tasks are performed after the map tasks. What Hadoop actually provides?? The answer is it provides a reliable shared storage and analysis .the storage is provided by HDFS and analysis provided by mapreduce. Hadoop(version1) , Firstly, the compute resources (e.g., CPU cores) are abstracted into map and reduce slots, which are basic compute units and statically configured by administrator in advance. A MapReduce job execution has two unique features: 1) the slot allocation constraint assumption that map slots can only be allocated to map tasks and reduce slots. can only be allocated to reduce tasks, and 2) the general execution constraint that map tasks are executed before reduce tasks. Due to these features, we have

two observations: (I). there are significantly different performance and system utilization for a MapReduce workload under different slot configurations, and (II) even under the optimal map/reduce slot configuration, there can be many idle reduce (or map) slots while map (or reduce) slots are not enough during the computation, which adversely affects the system utilization and performance.

II. LITERATURE SURVEY

1) Healthcare Data Analysis using Dynamic Slot Allocation in Hadoop

The potential of big data is to transform the way healthcare providers use sophisticated technologies to gain knowledge from their clinical and other data sources and make good decisions [2]. In the near future we will see rapid and widespread implementation of big data analytics in health care industry. Big data analytics guarantees privacy and security. The applications of big data analytics are still at nascent stage of development and its implementation in the health care industry will surely help its organizations. This paper proposes a framework which is aiming that it will improve the performance of MapReduce workloads and at the same time will maintain the fairness. DHSA [1] the technology about which we have mentioned above focuses on the maximum utilization of slots by allocating map (or reduce) slots to map and reduce tasks dynamically. There are two types of DHSA namely, PI –DHSA and the other is PD-DHSA they differ in the levels of fairness and user can choose any among them according to their requirements.

2) DynamicMR: A Dynamic Slot Allocation Framework for MapReduce Clusters in Big Data Management using DHSA and SEPB

This paper proposes a DynamicMR Technique can be used to enhance the execution of MapReduce workloads while keeping up the fairness. It comprises of three methods, in particular, DHSA, SEPB and Slot PreScheduling, all of which concentrate on the slot use optimization for MapReduce group from alternate points of view. DHSA concentrates on the slot use expansion by distributing map or reduce slots to map and reduce tasks alterably. Especially, it doesn't have any presumption or require any earlier learning and can be utilized for any sorts of MapReduce jobs (e.g., autonomous or subordinate ones). Two sorts of DHSA are introduced, in particular, PI-DHSA and PD-DHSA, in view of distinctive levels of fairness. Client can pick both of them likewise. Rather than DHSA, SEPB and Slot PreScheduling consider the effectiveness advancement for a given slot usage. SEPB recognizes the slot unused issue of speculative execution. It can adjust the execution tradeoff between a single job and a batch of job alterably. Slot PreScheduling enhances the proficiency of slot use by expanding its data locality.

3) Optimal Resource Allocation and Job Scheduling to Minimize the Computation Time under Hadoop Environment

In this project work, the study of map reduce algorithm is performed under Apache Hadoop framework. It deals with workload classification and minimizing the computation time of entire jobs. The Hadoop cluster is formed and jobs are allocated to the specific pools. Data node, Namenode, Job tracker and Task tracker are the Hadoop cluster components which does their tasks in complete manner. If client sends dataset to the Hadoop distributed file system, it separates the tasks to master node and slave node and performs the job using mapreduce concepts taken under map stage and reduce stage. By using Johnson's algorithm, the optimal solution for individual jobs for different disks are been calculated. Further, the efficiency of the computation task can be computed by the datasets taken and number of nodes that is generated in Hadoop distributed file system.

III. PROPOSED SYSTEM

In our proposed system, we trying to overcome the limitation of Hadoop (version 1) i.e. mentioned above. We proposed the Dynamic approach towards the mapreduce. Dynamic mapreduce has to major functions. they are slot utilization optimization and utilization efficiency optimization. The DynamicMR technique has the three slot allocation techniques they are

1.Dynamic Hadoop Slot Allocation (DHSA)
2. Speculative Execution Performance Balancing (SEPB)
3.Slot Prescheduling.

1.Dynamic Hadoop Slot Allocation (DHSA) : In Dynamic Hadoop Slot Allocation we dynamically allocate the slot for map and reduce function without any constrains like slots allocated for map function first than reduce. Also the map and reduce slots are different. such constraint overcome in DHSA. slot utilization improves here.

2. Speculative Execution Performance Balancing (SEPB):
In Speculative Execution Performance Balancing (SEPB) fairness while allocating the slots. Map slots should not use all slots and also the reduce slots. Distribution balanced here. Improve the efficiency of slot utilization.

3.Presheduling: In Prescheduling we improve the performance of slot utilization with data locality.

IV. OVERVIEW OF THE TECHNICAL AREA:

i. Big Data

Big data means really a big data, it is a collection of large datasets that cannot be processed using traditional computing techniques. Big data is not merely a data, rather it has become a complete subject, which involves various tools, techniques and frameworks.

ii. Big Data Technologies

Big data technologies are important in providing more accurate analysis, which may lead to more concrete decision-making resulting in greater operational efficiencies, cost reductions, and reduced risks for the business.To harness the power of big data, you would require an infrastructure that can manage and process huge volumes of structured and unstructured data in realtime and can protect data privacy and security. There are various technologies in

the market from different vendors including Amazon, IBM, Microsoft, etc., to handle big data.

iii. Analytical Big Data

This includes systems like Massively Parallel Processing (MPP) database systems and MapReduce that provide analytical capabilities for retrospective and complex analysis that may touch most or all of the data.

MapReduce provides a new method of analyzing data that is complementary to the capabilities provided by SQL, and a system based on MapReduce that can be scaled up from single servers to thousands of high and low end machines.

iv. Hadoop Distributed File System

Hadoop File System was developed using distributed file system design. It is run on commodity hardware. Unlike other distributed systems, HDFS is highly fault tolerant and designed using low-cost hardware.

HDFS holds very large amount of data and provides easier access. To store such huge data, the files are stored across multiple machines. These files are stored in redundant fashion to rescue the system from possible data losses in case of failure. HDFS also makes applications available to parallel processing.

v. MapReduce

MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes. Under the MapReduce model, the data processing primitives are called mappers and reducers. Decomposing a data processing application into mappers and reducers is sometimes nontrivial. But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration change. This simple scalability is what has attracted many programmers to use the MapReduce model.

vi) The MapReduce Algorithm:

- Generally, MapReduce paradigm is based on sending the computer to where the data resides!
- MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.
  - **Map stage**: The map or mapper's job is to process the input data. Generally, the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.
  - **Reduce stage**: This stage is the combination of the **Shuffle** stage and the **Reduce** stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.
  - During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster.
- The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.
- Most of the computing takes place on nodes with data on local disks that reduces the network traffic.

After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server

V. WORKING OF PROPOSED SYSTEM

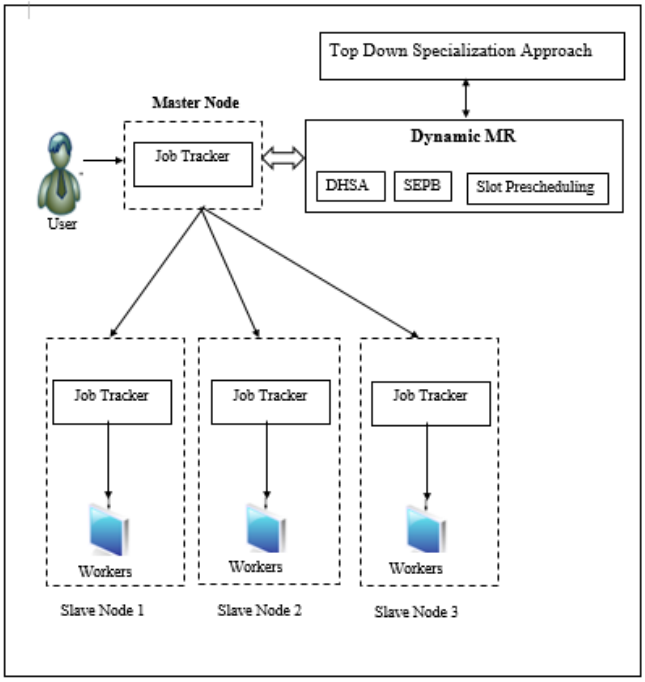

Fig.1

VI. CONCLUSION

MapReduce is a popular parallel computing paradigm for large-scale data processing in clusters and data centers. However, due to the pre-allocation of *slots* among map and reduce tasks, and the rigid execution order between map and reduce tasks in a MapReduce environment, slots can be severely under-utilized, which significantly degrades the performance. To address this problem, this paper proposes a dynamic slot allocation and scheduling system called *DynamicMR* consisting of three levels of schedulers, i.e., Dynamic Hadoop Fair Scheduler (DHFS), Dynamic Speculative Task Scheduler (DSTS), and Data Locality Maximization Scheduler (DLMS). DHFS improves the slot utilization by relaxing the slot allocation constraint to allow slots to be reallocated to either map or reduce tasks depending on their needs. DSTS and DLMS focus on the efficiency issue of slot utilization. DSTS balances the

performance tradeoff in speculative execution, considering that speculative execution can improve the performance for a single job at the expense of slot utilization efficiency for other jobs. DLMS improves slot utilization from the perspective of data locality. We have integrated DynamicMR into Hadoop. The experimental results show that our DynamicMR can improve the system performance significantly while guaranteeing the fairness, by up to 32% ~ 40% for a single job and 19% ~ 28% for MapReduce workloads with multiple jobs, in comparison to Hadoop.

DynamicMR framework aiming to improve the performance of MapReduce workloads while guaranteeing the fairness. It consists of three schedulers, namely, DHFS, DSTS and DLMS, all of which focus on the slot utilization optimization for MapReduce cluster from different perspectives. DHFS focuses on the slot utilization maximization by allocating map (or reduce) slots to map and reduce tasks dynamically. Particularly, it does not base on any assumption or require any prior-knowledge and can be used for any kinds of MapReduce jobs (e.g., independent or dependent ones).Two types of DHFS are presented, namely, PI-DHFS and PDDHFS, based on different levels of fairness. User can choose either of them accordingly. In contrast to DHFS, DSTS and DLMS consider the efficiency optimization for a given slot utilization. DSTS identifies the slot inefficiency problem of speculative execution. It can balance the performance tradeoff between a single job and a batch of jobs dynamically. DLMS, consisting of Delay Scheduler and our proposed PreScheduler, tries to improve the efficiency of slot utilization by maximizing its data locality. By enabling the above three schedulers to make them work cooperatively, the experimental results show that our proposed DynamicMR can improve the performance of the Hadoop system significantly.

REFRENCES

[1] G. Ananthanarayanan, S. Kandula, A. Greenberg, I. Stoica, Y. Lu, B. Saha, and E.Harris, Reining in the outliers in map-reduce clusters using mantri, in OSDI'10,pp. 1-16, 2010.

[2] Hadoop. http://hadoop.apache.org

[3] M. Zaharia, D. Borthakur, J. Sarma, K.Elmeleegy,S.Schenker,I. Stoica, Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling. In EuroSys'10, pp. 265-278, 2010.

[4] Y. C. Kwon, M. Balazinska, B. Howe, and J. Rolia.SkewTune: mitigating skew in mapreduce applications. In SIGMOD'12. pp. 25-36, 2012

[5] M. A. Rodriguez, R. Buyya. Deadline based Resource Provisioning and Scheduling

[6] Q. Chen, C. Liu, Z. Xiao, Improving MapReduce Performance Using Smart Speculative Execution Strategy. IEEE Transactions on Computer, 2013

[7] Z.H. Guo, G. Fox, M. Zhou, Y. Ruan. Improving Resource Utilization in MapReduce. In IEEE Cluster'12. pp. 402-410, 2012

[8] http://www.tutorialspoint.com/hadoop/index.htm