SCHEDULING AND HADOOP

Sunil Kumar

PGT, Computer Science, Jawahar Navodaya Vidyalaya, Kotia, Mahendragarh, Haryana, India

***Email ID**: sunilvats1981@gmail.com*

Received: 04.10.2015 **Accepted**: 18.11.2015

Abstract

Nowadays, Companies need to process Multi Petabyte Datasets efficiently. The Data may not have strict schema for the large system. It has become Expensive to build reliability in each Application for processing petabytes of datasets. If there is a problem of Nodes fail every day, some of the causes of failure may be. Failure is expected, rather than exceptional. The number of nodes in a cluster is not constant. So there is a Need for common infrastructure to have Efficient, reliable, Open Source Apache License. The solution is Hadoop. Hadoop is based on distributed computing having HDFS file system (Hadoop Distributed File System). Hadoop is highly fault-tolerant and can be deployed on low cost hardware. Hadoop is very much suitable for high volume of data and it also provide the high speed access to the data of the application which we want to use. hadoop architecture is cluster based, which is consist of nodes(data note, name node), physically separate to each other, in ideal condition. The performance of hadoop can be increased by proper assignment of the tasks in the default scheduler. In hadoop a program known as map-reduce is used to collect data according to query. As hadoop is used for huge amount of data therefore scheduling in hadoop must be efficient for better performance. The research objective is to study and analyse various scheduling techniques, which are used to increase performance in hadoop.

Keywords

Big data, Hadoop, Map Reduce, Locality, Job Scheduling, Fairness, Synchronization

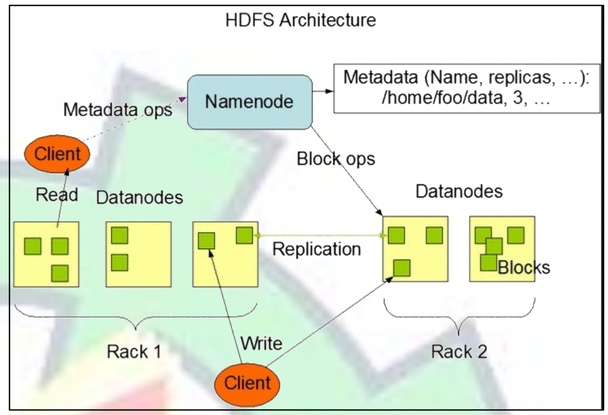
Paper Identification

1. Introduction

Hadoop is a “flexible and available architecture for large scale computation and data processing on a network of commodity hardware”. As Hadoop is an open source framework for processing, storing and analyzing massive amounts of distributed unstructured data. Originally created by Doug Cutting at Yahoo!, Hadoop was inspired by MapReduce, a user-defined function developed by Google in early 2000s for indexing the Web. It was designed to handle petabytes and Exabyte’s of data distributed over multiple nodes in parallel. Hadoop clusters run on inexpensive commodity hardware so projects can scale-out without breaking the bank. Hadoop is now a project of the Apache Software Foundation, where hundreds of contributors continuously improve the core technology. Fundamental concept: Rather than banging away at one, huge block of data with a single machine, Hadoop breaks up Big Data into multiple parts so each part can be processed and analyzed at the same time. Why Hadoop used for searching, log processing, recommendation systems, analytics, video and image analysis, data retention? It is used by the top level apache foundation project, large active user base, mailing lists, users groups, very active development, and strong development teams.

Big Data, now a days this term becomes common in IT industry. As there is lot of data lies in the industry but there is nothing before big data comes into picture.
Why we need the Big Data ?? ? As we know there is lot of data surrounds us but we can't make that data useful to us. Reason ?? Reason is simple, that there is no power tool that can make out analysis or information from this huge amount of data. There is one example, that one team of scientist have some data with them and they want to do some analysis on them, So one well know vendor in market approach them. That vendor will take 15 years to make analysis on that huge data. Now see, after long 15 years, is there any relevance of that data or that data is of any use to that user. In the era where we can't wait for 5 sec to open Google page. How we think of a long time to make the analysis. When we talk about Big Data, the first name comes in mind is "HADOOP" a well know product in the market of big data. Hadoop is Linux based product used by big player of market like Google, Yahoo etc.
Name Node is a type of master node, which is having the information or we can say that meta data about the all data node there is address(use to talk), free space, data they store, active data node , passive data node, task tracker, job tracker and many other configuration such as replication of data..
Data Node is a type of slave node in the hadoop, which is used to save the data and there is task tracker in data node which is use to track on the ongoing job on the data node and the jobs which coming from name node.
Hadoop Architecture, is based on HDFS, which is hadoop distributed file system. In which data is equally (ideally) distributed on each node in the hadoop system. When we (client) want to fetch or add or modify or delete some data from hadoop, then hadoop system collect the data from each node of our interest and do the meaningful actions of our interest.
Scheduler in Hadoop: A scheduler plays a very important role in the big data processing . A fair scheduler, schedules the jobs in such a way that all the resources share by the command or by the system in equal amount without any over loading on one of the part of the system. Like scheduler, will take care of the resources on each data node. It helps to maintain the load on all the data node in the system.
How does scheduling help in the processing of big data?
Take one example , suppose we have ten data node in hadoop cluster, and our scheduler is not fair it can not manage resources in a right manner. So what does he do that, he give work on 5 data node out of 10 data node in the cluster, and suppose it take around x amount of time to complete that command. Now think that our scheduler is fair enough to distribute work on all the data node in our cluster, So according to our assumption it will take around x/2 amount of time to complete the whole process.

HDFS Architecture



Now as we have seen in the above example that, with the help of the fair scheduling we had use the full resources of the system, due to which system which is taking x amount of time previously now taking the just half i.e x/2 just because of fair scheduling.
Comparison of various scheduling technique in hadoop is shown in Table 1.

2. Related Study

Enabling the statically multiplexing, by sharing the hadoop Map Reduce cluster to lower the cost of the execution and allow a number of user to work on the common large data set. As we have seen that the traditional fair scheduler performance is very low because of the two main reasons:
a. Data Locality: Where the data is Reside in the system on which we have to do the computation for our desire result.
b. Dependency between Map and Reduce Task: As Hadoop says that , till all the map will not finish, it will not start the reduce process to aggregate their individual result. To overcome from these problem we follow or use two technique delay scheduling and copy-computing splitting, with the help of which our response time will be improve by factor 2 to 10.
As we have seen that from the last few years, Map Reduce framework is becoming a paradigm for the batch or bulk operations or workload. As adoption of Map Reduce is increasing day by day, this makes the scheduling area very interesting now a days, because

with any good scheduler we are not able to use full and efficient resource of the system or hardware we are using. To achieve this, we are proposing a approach which will maintain harmony between the jobs running on the cluster to reduce their runtime and give us the faster result than before. In our approach scheduler is aware of all the jobs running on the cluster. Now when any incoming task comes to scheduler, then it is going to allocate that task on that node of the cluster, on which the new task will not affect the already running jobs on that node. Our scheduler will take one of the task from the pending task list of the system, which is most compatible on running of that node, and will not effect the already running job on that system. This approach is saving runtime around 21% in heuristic approach and 27 % if we compare to Yahoo's capacity scheduler.

As we know that Map Reduce is an emerging paradigm, now a days for a data intensive processing. As the Map Reduce provides the distributed data intensive work, with very high fault tolerance, easy scalable and low cost hardware according to the need of the individual. Many application now a days using MapReduce like Web data processing and high performance computing etc. The problem caused by the Map Reduce scheduler is mostly caused by the locality and synchronization overhead. For resolving the synchronization overhead , we have two categories of study: asynchronous processing and speculative execution for fairness constraints with locality improvement. Map Reduce is one of the platform for large scale data processing and for low cost hardware. To achieve the good scheduler comparative to hadoop default FIFO scheduler, our match making scheduler work with default FIFO scheduler of the hadoop and with also existing data locality enhancement technique. Experiment show that our technique make gives the highest data locality rate and lowest response time.

In this we are talking about the scheduling of the concurrent jobs on cluster where application data is stored on the computing nodes .In this type of data intensive computing, it is important to having data near to the computation nodes for the crucial performance of the system. In this we are introducing the Quincy scheduler. We evaluate Quincy on few hundreds nodes using sharing workload of data and CPU intensive jobs. Quincy is better when we talk about the fairness, when we tested Quincy data transfer volume is reduce by factor of 3.9% and through put increases. Jisha S Manjaly & Chinnu Edwin A of MG University has studied different default and extended task scheduler for improving the performance of the system. They observed that with the proper assignment of the task in the default scheduler in hadoop we can improve the performance of the hadoop scheduler. As hadoop framework is based on distributed file system in which data is distributed on the number of nodes in the cluster based architecture. The Hadoop is used for big data processing like on the database of the email, calls in the networks, and many more. In hadoop Mapreduce is the program which used to collect and aggregate the data according to our query. As hadoop is working on huge amount of data his scheduling should be efficient for better performance of the overall system.

3. Issues of Scheduling In Map Reduce

Issues in map reduce scheduling are listed as:

Locality: Locality is one of the main issue of map-reduce scheduling. Locality is defined as the distance

Proposed Technique	**Implementation**	**Advantages**
Asynchronous Processing	Process at the global/Local level individually	Reduce extra processing at global reducer end No waiting for others maps to complete to start one level reducer
Speculative Execution	Monitoring of the straggles tasks Restart start the straggle Task	No Limitation of application Robust at node
Job Awareness	New Job does not effect the current running job Pending job to node , not effecting current running	Reducing overall runtime of job Reducing jobs overhead
Delay Scheduling	Pending job in queue till job end at node level	No overhead for any type of calculation. Simple technique
Copy Compute Splitting	Split the work into small chunks Consider data locality and give work to each node	Workload divided on all Node

Table 1: Comparison of various scheduling techniques in Hadoop

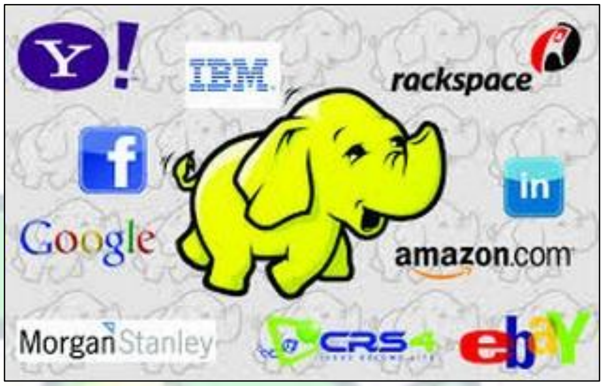
between the input data node and task -assigned node. When input data node is nearer to the computation node it takes less data transfer cost. Locality is consider as a basic approach in scheduling jobs with other scheduling constraints. Locality is a very critical issue which affects the performance in a shared cluster environment, due to limited bisection bandwidth of network. High locality increases the throughput of tasks. The processing of a task on a node holding the data, called node locality is the efficient case of locality. Whenever node locality is not possible to achieve then executing a task/job on the same rack, known as rack locality, is preferred. If the requirement of locality is not fulfilled, data transferring I/O costs can seriously affect the performance because of the shared bandwidth of network. Most methods of scheduling of map reduce jobs follow a policy of attempting to assign tasks to a place near the input data to save cost of network.

Synchronization: Synchronization is the process of transferring the intermediate output of the map processes to the reduce processes as input is also consider as a factor which affects the performance. Mappers have to wait until all the map processes are finished to initiate sending intermediate output. Due to dependency between the map and reduce phases of processing, a single node can slow down the whole process, causing the other nodes to wait until it is finished. There are various factors which results in performance degradation in the synchronization step ,few of them are as heterogeneity of the cluster, node failures miss-configuration, and serious overhead of the I/O cost.

Fairness: Various map-reduce jobs are performed in a shared data warehouses of enterprises like facebook, Amazon, Google and Yahoo. A map-reduce job with a heavy workload may dominate utilization of the shared clusters, so some short computation jobs may not have the desired response time. The demands of the workload can be elastic, so fair workload to each job sharing cluster should be considered. Fairness constraints have tradeoffs between the locality and dependency between the map and reduce phases. When each map-reduce job has roughly an equal share of the nodes and the input files are spread in distributed file system, some map processes have to load data from the networks. This causes a great degradation in throughput and response time. Synchronization overhead could affect the fairness. For example reduce processes have to wait for the completion of map processes which leads to idle nodes and starvation of other jobs. Due to this problem a poor utilization situation occurs.

Hadoop Users

The following company are the users of hadoop Adobe, Alibaba, Amazon, AOL, Facebook, Google, IBM.



Major Contributors

The following companies are the major contributors of Hadoop. They are Apache, Cloudera and Yahoo.

4. Conclusion

Big Data (Hadoop) is in huge demand in the market now a days. As there huge amount of data is lying in the industry but there is not tool to handle it and hadoop can implemented on low cost hardware and can be used by large set of audience on large number of dataset. In hadoop map reduce is the most important component in hadoop. In this paper we have studied many technique for making the efficient scheduler for the map reduce so that we can speed up our system or data retrieval Technique like quincy, Asynchronous Processing, Speculative Execution, Job Awareness, Delay Scheduling, Copy Compute Splitting etc had made the scheduler effective for the faster processing.

5. References

[1] Job Scheduling for Multi-User MapReduce Clusters, Matei Zaharia ,hruba Borthakur,Joydeep Sen Sarma,Khaled Elmeleegy,Scott Shenker,Ion Stoica ; Electrical Engineering and Computer Sciences, University of California at Berkeley

[2] Job Aware Scheduling Algorithm for MapReduce Framework by Radheshyam Nanduri, Nitesh aheshwari, Reddy Raja, Vasudeva Varma in 3rd IEEE International

Conference on Cloud Computing Technology and Science Athens, Greece.

[3] A Comparative review of job scheduling for MapReduce, Dongjin Yoo, Kwang Mong Sim Multi-Agent and Cloud Computing Systems Laboratory, School of Information and Communication, Gwangju Institute of Science and Technology (GIST), Gwangju, Republic of Korea

[4] Matchmaking: A New MapReduce Scheduling Technique Chen He Ying Lu David Swanson Department of Computer Science and Engineering University of Nebraska-Lincoln Lincoln, U.S.

[5] Survey on Improved Scheduling in Hadoop MapReduce in Cloud Environments B.Thirumala Rao Associate Professor Dept. of CSE Lakireddy Bali Reddy College of Engineering Dr. L.S.S.Reddy Professor & Director Dept. of CSE Lakireddy Bali Reddy College of Engineering

[6] Quincy: Fair Scheduling for Distributed Computing Clusters ,Michael Isard, Vijayan Prabhakaran, Jon Currey, Udi Wieder, Kunal Talwar and Andrew Goldberg Microsoft Research, Silicon Valley — Mountain View, CA, USA

[7] Apache Hadoop: http://Hadoop.apache.org

[8] Hadoop Tutorial: http://developer.yahoo.com/hadoop/tutorial/module1.html

[9] Tom white, Hadoop Definitive Guide,Third Edition,2012

[10] Research on Job Scheduling Algorithm in Hadoop Yang XIA†, Lei WANG1, Qiang ZHAO1, Gongxuan ZHANG

[11] Apache Hadoop!(hadoop.apache.org)

[12] Hadoop on Wikipedia (http://en.wikipedia.org/wiki/Hadoop)

[13] Free Search by Doug Cutting (http://cutting.wordpress.com)

[14] Hadoop and Distributed Computing at Yahoo! (http://developer.yahoo.com/hadoop)

[15] Apache Hadoop for the Enterprise (http://www.cloudera.com)

[16] Siones, M. Tim (6 December 2011). "Scheduling in Hadoop". ibm.com. IBM. Retrieved 20 November 2013.

[17] Jones, M. Tim (6 December 2011). "Scheduling in Hadoop". ibm.com. IBM. Retrieved 20 November 2013.

[18] Murthy, Arun (2012-08-15). "Apache Hadoop YARN – Concepts and Applications". hortonworks.com. Hortonworks. Retrieved 2014-09-30.

[19] "Version 2.0 provides for manual failover and they are working on automatic failover:" Hadoop.apache.org. Retrieved 30 July 2013.

[20] "Amazon Elastic MapReduce Now Supports Spot Instances". Amazon.com. 2011-08-18. Retrieved 2013-10-17. "Apache Accumulo User Manual: Security". apache.org. Apache Software Foundation. Retrieved 2014-12-03.

[21] "Refactor the scheduler out of the JobTracker". Hadoop Common. Apache Software Foundation. Retrieved 9 June 2012.

[22] www.tutorialspoint.com/hadoop/

[23] Andrew S. Tanenbaum Computer Networks Prentice Hall PTR New Jersey 2003

[24] Ramjee Prasad An Introduction to OFDM John Wiley & Sons New York 2001