A framework to audit log files to find operations at the storage level at HDFS

Er. Prachi Jain, Er. Alisha Gupta

Department of Computer Science and Engineering, Haryana Engineering College, Jagadhri, Haryana

E-mail: prachijain1992@gmail.com

Abstract: With enormous data present all over the world, the need of managing the data has also risen. Hadoop is used to maintain and process such large amount of data. Hadoop is an Apache framework which is used to store and process large amount of data. The data is stored in a distributed environment in Hadoop. Hence, Hadoop consists of hadoop distributed file system which is used to store this large amount of data. The data present in such a large scale and complex structure is termed as big data. Map reduce is used for large scale data processing. The data is processed by breaking down it into various jobs which are fed as input to map tasks and reducer processes the data came as output from the mapper. Various scheduling algorithms are proposed to schedule these jobs. This paper covers the simulation study of various scheduling algorithms and audits log files to find operations at storage level. The results show that there is about 15.68% decrease in execution time of word count program when fair scheduler is used as compared to default FIFO scheduler. Also for pcap file of ack packets the execution time is reduced by 14.73% when fair scheduler is used.

Keywords: Hadoop, map reduce, big data, capacity, fair, scheduler, auditing

INTRODUCTION

Due to rapid development of internet applications, the demand of computing power has risen manifolds. Many new technologies like grid computing, cloud computing, distributed computing or parallel computing have emerged to provide enormous computing power. [1] Due to invention of cloud computing, more and more applications are now deployed in the cloud environment enabling people to have access to data at very less rate. The amount of data that we are producing has increased manifolds as a result it has led to the invention of big data.

1. BIG DATA

Big data is basically a terminology that is used for very massive data sets that have a large variation along with complex structure. These are the characteristics that usually add difficulties like storing the data, analyzing it and further applying procedures after which results are to be extracted. [2] Big Data is related to data that surpasses the usual storage, processing power, and computing capacity of traditional databases and data analysis techniques. Moreover, to process such a large amount of data, Big Data requires a large set of tools and methods which can be applied to analyze and extract patterns from large-scale data. Big data can be characterized by three Vs: Volume, Variety and Velocity. Here volume means that there is large amount of data present which can be in the order of terabytes or petabytes. With variety we understand that the data comes from varied sources like text, audio, video, images etc. Velocity defines how the data is kept in motion and how the analysis of streaming data is done.

2. HADOOP

Hadoop is one of the technologies to tame big data. HADOOP is basically a framework provided by Apache which is used to run applications on systems which includes thousands of nodes and data is in the order of terabytes. It handles large amount of data by distributing it among the nodes. [3] It also helps system to work properly even when a node in the network fails. As a result risk of catastrophic system failure is reduced. Apache HADOOP consists of the HADOOP kernel, HADOOP distributed file system (HDFS) and map reduce paradigm. [4]

3. HADOOP DISTRIBUTED FILE SYSTEM

Hadoop solves the problem of storing large amount of data by placing in to number of clusters. There is a fault-tolerant storage system present in HADOOP which is called HADOOP Distributed File System, or HDFS. [5] With HDFS we can store huge amounts of information along with scaling up incrementally and surviving the system failure without threat of losing data. HDFS stores large amount of data across multiple machines. These files are stored in redundant fashion so that whenever any node crashes, the data can be easily recovered.

4. MAP REDUCE

Map reduce is a paradigm used for parallel processing of data using two functions: map and reduce. [6] It provides scheduling, parallelization, replication and failover. Using map and reduce phase map reduce basically encodes data for faster processing. It is a framework that can be used to process large chunks of data in parallel on large clusters of commodity software reliably. The map reduce algorithm comprises of two phases: map and reduce. In map phase input data is taken and converted into some other form. Also individual elements are broken down into elements (key/value pairs). The input to reduce task is the output of map task and reduce phases processes the input. Map reduce has master and workers and the work between them is collaborative in nature.

5. SCHEDULING ALGORITHMS

There are various algorithms proposed for scheduling of applications in cloud environment. Basically, scheduling can be defined as method to select and decide the task which is most appropriate to execute. It is also defined as allocation of machines to tasks so that makespan of workflow is minimized. [6] Algorithms like data aware scheduling algorithms, first come first serve, round robin, minimum completion time, heterogeneous earliest finish time etc can be used for scheduling workflows in a HADOOP environment. FIFO scheduling algorithm is the default algorithm provided by Hadoop architecture. In FIFO scheduling, jobs are executed in first come first serve order. A FIFO queue is maintained by FIFO scheduler that keeps multiple tasks in it. Fair scheduler works on the concept that resources are assigned to job so that every job gets an equal share of the available resources as a result jobs that require more time to execute don't starve. Several queues are created instead of pools in capacity scheduling. Each queue has a configurable number of map and reduce slots and each queue is also assigned a guaranteed capacity. The scheduler monitors the queue and if a queue is not consuming its allocated capacity, the excess capacity can be temporarily allocated to other queues.

LITERATURE SURVEY

Although Hadoop is a new technology but a lot of research has been done in this field by many researchers. A lot of things have been proposed and many new algorithms have been developed to strengthen the features provided by Hadoop framework. Some of the work done by the researchers is listed below:

Laurent Bobelin, Patrick Martineau et al(2016): Big data has revealed itself as a powerful tool for many sectors ranging from science to business. Distributed data-parallel computing is then common nowadays: using a large number of computing and storage resources makes possible data processing of a yet unknown scale. But to develop large-scale distributed big data processing, we have to tackle many challenges. One of the major complexities is scheduling. As it is known to be an optimal online scheduling policy when it comes to minimize the average flowtime, Shortest Processing Time First (SPT) is a classic scheduling policy used in many systems. The author has integrated this policy into Hadoop, a framework for big data processing, and realized an implementation prototype. The paper has described this integration, as well as tests results obtained on test bed. [7]

Xiangming Dai et al.(2016): In this paper, the author has proposed a novel task scheduling algorithm for Hadoop map reduce called dynamic priority multi queue scheduler (DPMQS). DPMQS i) increases the data locality of jobs, and, ii) dynamically increases the priority of jobs that are near to completing their Map phase, to bridge the time gap between the start of the reduce tasks and the execution of the reduce function for these jobs. The author has also discussed the details of DPMQS and its practical implementation, then assessed its performance in a small physical cluster and large-scale simulated clusters and compared it to the other schedulers available in Hadoop. Both real experiments and simulation results show that DPMQS decreases significantly the response time, and demonstrate that DPMQS is insensitive to changes in the cluster geometry. [8]

Divya M, Annappa B(2015): Hadoop map reduce is one of the largely used platforms for large scale data processing. Hadoop cluster has machines with different resources, including memory size, CPU capability and disk space. This introduces challenging research issue of improving Hadoop's performance through proper resource provisioning. The work presented in this paper focuses on optimizing job scheduling in Hadoop. Workload Characteristic and Resource Aware (WCRA) Hadoop scheduler is being proposed by the author which classifies the jobs into CPU bound and Disk I/O bound. Based on the performance, nodes in the cluster are classified as CPU busy and Disk I/O busy. The amount of primary memory available in the node is ensured to be more than 25% before scheduling the job. Performance parameters of Map tasks such as the time required for parsing the data, map, sort and merge the result, and of Reduce task, such as the time to merge, parse and reduce is considered to categorize the job as CPU bound or Disk I/O bound. Tasks are assigned the priority based on their minimum Estimated Completion Time. The jobs are scheduled on a compute node in such a way that jobs already running on it will not be affected. Experimental result has given 30 % improvement in performance compared to Hadoop's FIFO, Fair and Capacity scheduler. [5]

Ping Li et al. (2015): YARN is used to provide resource management and scheduling for large scale map reduce environments. However it faces two major challenges: ability to automatically tailor and control resource allocations to different jobs to achieve their service level agreements and to minimize the energy consumption of the cloud computing system. The author has proposed a scheme which is SLA aware energy efficient scheduling scheme and is used to allocate optimal number of resources to map reduce applications. Job profiling is also performed to obtain the performance characteristics for different phases of a map reduce application which is used during resource provisioning in order to meet the specified SLA. Also, the authors have designed an online user space governor based dynamic voltage and frequency scaling scheme so that the CPU frequency for upcoming tasks can be dynamically changed. Their scheme have achieved better SLA conformance with low resource cost and energy consumption. [9]

PEI Shu-jun et al. (2015): The paper has focused on the insufficient thought for the task locality of FIFO scheduler of the hadoop scheduler. The authors have proposed task locality improvement scheduler. The jobs are set and processed to several job queues according to probability threshold level of task locality. The tasks are executed locally immediately if the local node is idle or they will have to wait until the local node becomes idle for execution. The task locality is improved by 98% and performance is improved by 10.9%. [10]

Qutaibah Althebyan et al. (2014): Map reduce is a paradigm used for parallel processing of data using two functions: map and reduce. Scheduling is the major problem faced by map reduce. The authors have proposed a new scheduling algorithm which is based on multi threading principle. In their scheme the cluster is divided into multi blocks where in each block is scheduled by a special thread and the scheduling is done synchronously. Simulation time and energy consumption is the parameter on which their algorithm is being tested. The results show that the proposed algorithm is 47% better as compared to FIFO algorithm. [11]

PROPOSED WORK

The implementation of hadoop creates a set of pools where the jobs are placed for selection by the scheduler. By default, all pools have equal shares, but configuration is possible to provide more or fewer shares depending upon the job type. The number of jobs active at one time can also be constrained, if desired, to minimize congestion and allow work to finish in a timely manner. Due to increased size of data files, size of log files (files which records the data activity), intrusion detection system faced so many problems and gives inaccurate results for detecting the attackers.

The objective of the study is to perform simulation study of various scheduling algorithms present in hadoop framework. We will also assign a set of shares to each pool to balance resources across jobs in pools. The implementation is set to allow interactivity among Hadoop jobs and to permit greater responsiveness of the Hadoop cluster to the variety of job types submitted. Moreover, the implementation audits log files to find if any, malicious operations are performed or any malicious user is manipulating the data in the nodes.

IMPLEMENTATION

The implementation for the above proposed work is done using Hadoop 0.20.2 along with the net stress and wireshark tool. Netstress is a tool used to measure network performance. It employs bulk data transfer with the help of Layer 3 protocols TCP and UDP. Wireshark is a packet analyzer tool which is used to capture network packets and display the packet data as detailed as possible. The work flow of the proposed work is as follows:

- Flooding is done on data node of the network.
- Live traffic is captured.
- Job is assigned to the job tracker.
- Map and reduce function is applied on the job.
- Result is formulated.

The study revolves around analyzing the results provided by various scheduling algorithms like default FIFO scheduler, capacity scheduler and fair scheduler. For getting the optimal results mapred-site.xml file is configured accordingly.

To audit log records mapred-site.xml file for capacity scheduler is modified and the simulation is done against that file.

We sent the jobs to the data node of hadoop. The respective jobs we sent to the data node are class file of word count program and pcap file of ack packet generated using netstress and wireshark. The action performed on pcap file is that of rate generation. The maximum map and reduce tasks which can be performed by capacity scheduler is 5.

Also, we have built a code which audits log files and records any malicious activity. The code sends the block to the invalid set stating that this particular block can harm the network.

RESULTS

Word count file and pcap file is sent simultaneously as two jobs to the data node and the results are noted. Since, we have taken default, capacity and fair scheduler in consideration in our study, the following graph shows the time taken for execution for both word count program and rate generation program.

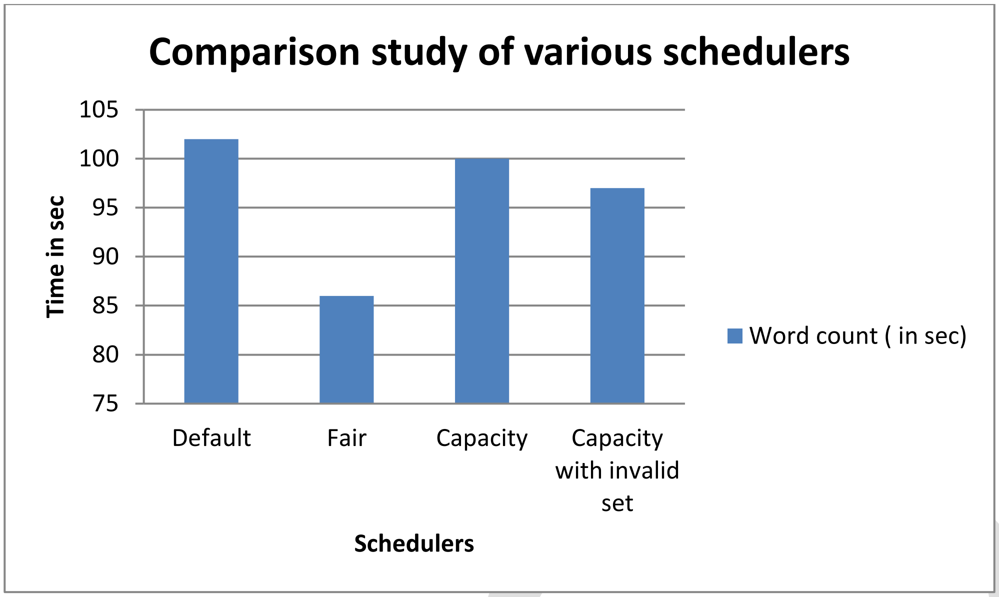


Figure 5: Comparison study of various schedulers on word count program

As we can see from the above graph, the execution time for word count program is maximum for default scheduler i.e. 102 seconds while for fair scheduler it is 86 seconds. The execution time for capacity scheduler is 100 seconds. The time taken by capacity scheduler while sending invalid blocks to invalid set is 97 seconds. We can see clearly that each scheduler is taking less time as compared to the default scheduler provided by hadoop framework. Moreover fair scheduler is taking 15.68% less time as taken by default scheduler.

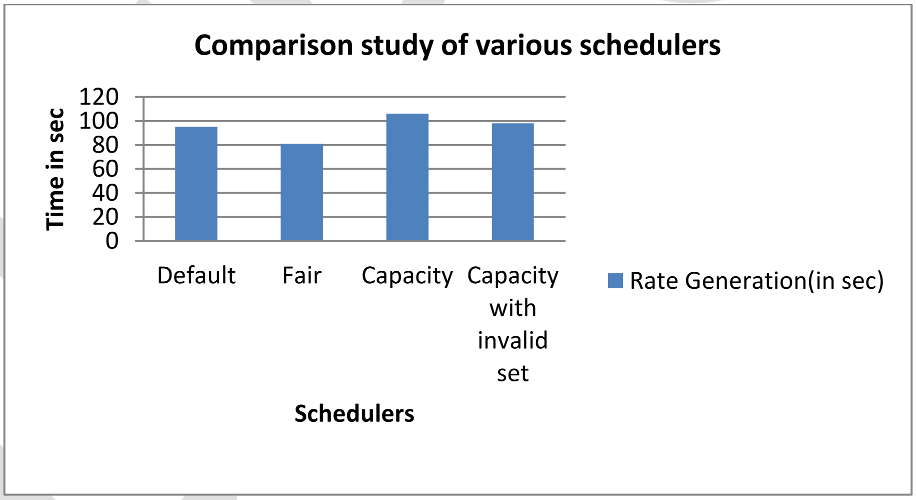
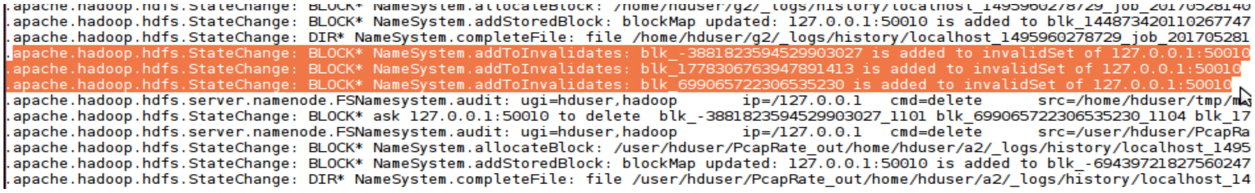


Figure 6: Comparison study of various schedulers on rate generation file

When pcap file is sent to the data node for map and reduce function, the default scheduler takes 95 seconds while fair scheduler takes 81 seconds. The time taken by capacity scheduler and capacity scheduler which sends invalid block to invalid set is 106 seconds and 98 seconds respectively. We can see that there is again a 14.73% decrease in time when fair scheduler is reduced.



```
.apache.hadoop.hdfs.StateChange: BLOCK* NameSystem.allocateBlock: /home/hduser/g2/_logs/history/localhost_1495960278729_job_20170528140
.apache.hadoop.hdfs.StateChange: BLOCK* NameSystem.addStoredBlock: blockMap updated: 127.0.0.1:50010 is added to blk_144873420110267747
.apache.hadoop.hdfs.StateChange: DIR* NameSystem.completeFile: file /home/hduser/g2/_logs/history/localhost_1495960278729_job_201705281
.apache.hadoop.hdfs.StateChange: BLOCK* NameSystem.addToInvalidates: blk_-3881823594529903027 is added to invalidSet of 127.0.0.1:50010
.apache.hadoop.hdfs.StateChange: BLOCK* NameSystem.addToInvalidates: blk_1778306763947891413 is added to invalidSet of 127.0.0.1:50010
.apache.hadoop.hdfs.StateChange: BLOCK* NameSystem.addToInvalidates: blk_699065722306535230 is added to invalidSet of 127.0.0.1:50010
.apache.hadoop.hdfs.server.namenode.FSNamesystem.audit: ugi=hduser,hadoop       ip=/127.0.0.1   cmd=delete      src=/home/hduser/tmp/m
.apache.hadoop.hdfs.StateChange: BLOCK* ask 127.0.0.1:50010 to delete  blk_-3881823594529903027_1101 blk_699065722306535230_1104 blk_17
.apache.hadoop.hdfs.server.namenode.FSNamesystem.audit: ugi=hduser,hadoop       ip=/127.0.0.1   cmd=delete      src=/user/hduser/PcapRa
.apache.hadoop.hdfs.StateChange: BLOCK* NameSystem.allocateBlock: /user/hduser/PcapRate_out/home/hduser/a2/_logs/history/localhost_1495
.apache.hadoop.hdfs.StateChange: BLOCK* NameSystem.addStoredBlock: blockMap updated: 127.0.0.1:50010 is added to blk_-69439721827560247
.apache.hadoop.hdfs.StateChange: DIR* NameSystem.completeFile: file /user/hduser/PcapRate_out/home/hduser/a2/_logs/history/localhost_14
```

Figure 7: Audited log records

As it can be seen from the above figure, some blocks are moved to invalid set of the node. This shows that log files are audited and any malicious activity can be found in the invalid set.

CONCLUSION

Hadoop can be used for processing multiple jobs in a distributed environment. When jobs are sent to the data node, a decrease in the execution time can be seen while using fair scheduler as compared to default scheduler. A decrease of 15.68% is seen in the case of word count program while a decrease of 14.73% is seen in the case of rate generation program. Moreover, some blocks are moved to invalid set as a result of auditing.

REFERENCES:

[1] Kamal Kc and Kemafor Anyanwu, "*Scheduling Hadoop Jobs to Meet Deadlines"*, Proceeding of 2nd IEEE International Conference on Cloud Computing Technology and Science, April, 2010.

[2] Hong Mao, Shengqiu Hu, Zhenzhong Zhang, Limin Xiao and Li Ruan, *"A Load-Driven Task Scheduler with Adaptive DSC for MapReduce"*, Proceeding of 2011 IEEE/ACM International Conference on Green Computing and Communications, March, 2011.

[3] Sutariya Kapil B. and Sowmya Kamath S., "*Resource Aware Scheduling in Hadoop for Heterogeneous Workloads based on Load Estimation"*, Proceeding of 4th ICCCNT – 2013, July 4-6, 2013, Tiruchengode, India.

[4] Anam Alam and Jamil Ahmed, *"Hadoop architecture and its issues"*, Proceeding of 2014 International Conference on Computational Science and Computational Intelligence, 2014.

[5] Divya M. and Annappa B., *"Workload Characteristics and Resource Aware Hadoop Scheduler"*, Proceeding of 2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS), 2015.

[6] Qinghua Lu, Shanshan Li and Weishan Zhang, *"Genetic Algorithm based Job Scheduling for Big Data Analytics"*, Proceeding of 2015 International Conference on Identification, Information, and Knowledge in the Internet of Things, 2015.

[7] Laurent Bobelin, Patrick Martineau, Di Zhao and Haiwu He, *"Shortest Processing Time First Algorithm for Hadoop"*, Proceeding of IEEE 3rd International Conference on Cyber Security and Cloud Computing, June, 2016.

[8] Xiangming Dai and Brahim Bensaou, "*Scheduling for response time in Hadoop MapReduce"*, Proceeding of IEEE ICC 2016 SAC Cloud Communications and Networking, June 2016.

[9] Ping Li, Lei Ju, Zhiping Jia and Zhiwen Sun, *"SLA-Aware Energy-Efficient Scheduling Scheme for Hadoop YARN"*, Proceeding of 2015 IEEE 17th International Conference on High Performance Computing and Communications (HPCC), September, 2015.

[10] PEI Shu-jun, Zheng Xi-min, Hu Da-ming, Lou Shu-hui and Zhang Yuan-xu, *"Optimization and Research of Hadoop Platform Based on FIFO Scheduler"*, Proceeding of 2015 Seventh International Conference on Measuring Technology and Mechatronics Automation, August 2015.

[11] Qutaibah Althebyan , Omar ALQudah, Yaser Jararweh and Qussai Yaseen, *"Multi-Threading Based Map Reduce Tasks Scheduling"*, Proceeding of 2014 5th International Conference on Information and Communication Systems (ICICS), April, 2014.

[12] Xicheng Dong, Ying Wang, Huaming Liao, "Scheduling Mixed Real-time and Non-real-time Applications in MapReduce Environment", 2011 IEEE 17th International Conference on Parallel and Distributed Systems, 2011