International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Survey on Schedulers Optimization to Handle Multiple Jobs in Hadoop Cluster

Shivaraj B. G.[1], Nagaraj Naik[2]

[1]PG Student, Department of CS&E, MITE, Moodabidri, Mangalore, India

[2]Sr. Assistant Professor, Department of CS&E. MITE, Moodabidri, Mangalore, India

Abstract: ***An apache Hadoop project is a good platform that supports cost-effective such as commodity hardware implementation, with scalable infrastructure called Hadoop Distributed File System HDFS with parallel processing mechanism called MapReduce. Hadoop is well known for Big Data analytics requires more resources for collecting; storing, processing petabytes of large data requires one meaningful resource management. Hadoop handles jobs in batch modes, allocating resources, scheduling to these modes is an important issue in terms of Network Bandwidth, CPU time and Memory. Resources handled by MapReduce schedulers assigning resources in the form of MapReduceTasks. The MapReduceTasks are carefully handled by MapReduce schedulers by setting some benchmarks for individual namely MapCombineReducer and different Block Sizes, which ensures schedulers are optimized to achieve maximum efficiency in storage capacity, time and cost for handling multiple batch jobs in multiple cluster while guaranteeing the effectiveness of individual scheduler in terms of job execution.***

Keywords: Big Data, Hadoop, HDFS, MapReduce, Schedulers optimization

1. Introduction

Big Data The term "big data" is when the size of the data itself becomes part of a problem and refers to large datasets. Generated more or less bigger organizations including IT and Non-IT, an option includes Telecommunications, Facebook, Amazon, and Logistics a lot more. Big data is a combination of transactional and interactive data and we are currently

Living in the big data era and need to work on day-to-day basis. Big Data has challenges to collect, store and process those large data sets because of complexity that traditional tools like relational database management systems (RDBMS) unable to store and process those large datasets. Big data referred to datasets of size (TB or PB) of different orders of magnitude and sometimes beyond the expectations of normal views. Data volumes are classified into ***volume***-dataset size, ***velocity***-data arrival and processing, ***variety***-different data structures (images, audio, sensor dataset and other types).Over the past couple of years the data and information that are being stored in some electronic format. The collection of all that data has really taken place at high rate. According to research studies the amount of information around world-wide doubles every twenty (20) months. Cloud computing refers or defined to Internet based development and utilization of computer information technology (CIT). the National Institute of Standard and Technology defines Cloud-Computing as "Is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources such as networks, servers, storage application and services that can be rapidly provisioned and released within minimal management effort or service provider interaction with the act of "big data" the volume, variety and velocity of available information exponentially increase the complexity of information that companies need to manage. And we expect at 2025 the world generated data is about Exabyte and Zettabyte. Big data represents opportunity as a products and services for broad range of business and social benefit along with hard risk finding and maintaining a balance between the new innovations and also determines serious risk. By observing all these related issues we can also say why only big data why not other data. After all practical applications observed and suggest many organizations and institutions have been struggling to figure out how to manage their data and one right choice for solving big data problem with storage and processing techniques uses tools such as Hadoop or Hadoop clusters, Bloom filters, and analysis tools such as WebUI and is used by Amazon, Facebook, Yahoo, Cloudera, Hortonworks, IBM, Intel, Microsoft, MapR Technologies, Teredata, Pivotal Software. Below table shows different measures for big data.

1000Gigabytes	= 1 Terabyte (TB)
1000 Terabytes	= 1Petabyte (PB)
1000 Petabyte	= 1Exabyte (EB)
1000 Exabyte	= 1Zettabyte (ZB)
1000 Zettabyte	= 1Yottabyte (YB)

Tools for managing big data: Big data requires exceptional technologies to efficiently process large quantities of data within tolerable elapsed times and these products are available from large to small vendors and scattered those products to handle social media analytics, NoSQL (Not only SQL) databases with visualization and capacity to analyze them. Following figure 1 shows some examples and supported tools to implement very large data sets.

Table 1: Tools and Vendors.

Hadoop distributions	Cloudera, hortonworks, pivatalHD.
Hadoop SQL interface	Apache hive, ClouderaImpla, EMC HAWQ.
MapReduce alternatives	Spark, Nokia Disco.
NoSQL database	IBM Netezza, HP Vertica, Aster Teradata, EMC Greenplum.

Integration tools	Talend, Informatica.
Datasets	Infochimps
Management Systems	Cloudant, marklogic, couchbase.

So combining all these cases, a Big Data may be: *Structured* - E.g. Relational database. Semi-*Structured* – E.g. XML, Transaction data, Graphs. *Un-Structured* – E.g. Character and Binary data.

2. Literature Survey

FIFO scheduler it is the default scheduler for almost all Hadoop applications. The Job queue is processed in First in First out fashion. With the scheduler, the main drawback is that only after finishing the previous job, next jobs in the job queue will be assigned. The scheduler implementation is simple and efficient. Fair scheduler is introduced by Facebook. Here, fair sharing of resources is possible. Main advantage of the scheduler is that whenever slot becomes free, shorter jobs can be assigned. Unlike FIFO scheduler, the smaller jobs need not wait in order to complete a big job. Main drawback is that tasks will be allocated to all the slots in the cluster with maximum slot capacity. In fair scheduling, pool of jobs will be there. Fair scheduler follows three concepts. Jobs placed in a pool of jobs, each pool is having a guaranteed capacity, fair sharing of resources. [1] [2]. Capacity scheduler is introduced by Yahoo. Capacity scheduler is very effective for large scale applications. Other than job pool, here multiple job queues are allocated. There is a guaranteed capacity for each queue. When the queue capacity exceeds the job will be allocated to other queues. Here, higher priority jobs can access resources with faster than lower priority jobs. In the job queue, Capacity scheduler follows FIFO scheduling with priority. But there is a limit on percentage of running tasks per user. So that users can share the cluster setup equally. [1] [2]. According to the survey of different schedulers we can come to know that local data processing takes lesser times as compared to moving data across network. So to improve the performance of jobs most of the algorithms work to improve the data locality. So to meet our expectations scheduling algorithms must use some of the prediction methods based on the small or large volume of data to be processed with underlying software or hardware.

3. Problem Statement

Hadoop's default scheduler falls into the first category. It manages a FIFO queue for all submitted jobs based on submission time or user-specified priority levels. All the pending jobs are sorted according to their priorities, and then by their submission time. Once a task slot becomes available (e.g. a running job finishes all its map tasks and starts reduce tasks), it will be assigned a task of the first waiting job in the pending queue. This approach allows one job to take all task slots within the cluster, i.e., no other jobs can utilize the cluster until the current one completes. Consequently, jobs that arrive at a later time or with a lower priority will be blocked by those ahead in the queue4. Given the total number of jobs is large, there will be a significant delay caused by FIFO scheduler.

4. HADOOP

Thanks to Google, Doug and Yahoo because it all started with Google, which in 2003 and 2004 released two academic papers describing Google technology: the Google File System (GFS) and MapReduce The two together provided a platform for processing data on a very large scale in a highly efficient manner. Doug started work on the implementations of these Google systems, and Hadoop was born, and became top-level project within the Apache open source foundation. At its core, Hadoop is an open source platform that provides implementations of both the MapReduce and GFS technologies and allows the processing of very large data sets across clusters of low-cost commodity hardware. In addition to often publicizing some of the largest Hadoop deployments in the world, it has contributed some of its own internally developed Hadoop improvements and extensions remains a major Hadoop contributor.the new Hadoop is nothing less than the apache foundations [3] [4] attempts to create a whole new general framework for the way big data can be stored, mined and processed and is a open source software framework for operating system of sorts for the data vertical used to store and process large volumes of data with the help of disparate computing system instead of few super computers. The hardware is cheap and the software is open-source leveling the field for all players and emerged as more of an operating system that manages resources secures data and performs other functions and applications written on top of it help process data and cross-pollinate data points. Hadoop is an apache top-level project Where several million pieces of content is a lot and cannot be easily be analyzed within any excel spreadsheet. A traditional ETL (Extract, Transform, Load) process extracts data from multiple sources, then cleanses, formats and loads it into a data warehouse for analysis when the source data sets are large, fast, and unstructured, traditional ETL can become the bottleneck because it is too complex to develop too expensive to operate and takes too long to execute. Various criticisms of Hadoop have resolved around its scaling limitations but the biggest constraint on scale has been its job scheduling in Hadoop are run as batch processes through a single daemon called JobTracker which creates a scalability and processing speed. Thus Apache Hadoop is an open source distributed software platform for storing and processing data written in java it runs on clusters of industry standards servers configured with direct attached storage using Hadoop can store petabytes of data reliably on tens of thousands of servers while scaling performance cost-effective by merely adding inexpensive nodes to the clusters because of its batch processing Hadoop should be deployed in situations such as index building, pattern recognitions, creating recommendation engines and sentiment analysis. All situations where data is collected in ever escalating sheer volumes with variable semantic contexts, stored in Hadoop, queried at later using MapReduce functions.

Common building blocks

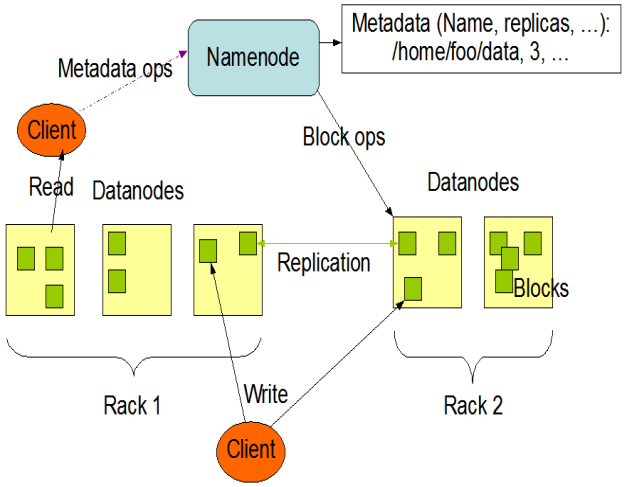
- Both HDFS and MapReduce exhibit important architectural principles
- Designed to run on clusters of commodity (that is, low-to-medium specification) servers

- Both scale their capacity by adding more servers (scale-out) with different mechanisms for identifying and working around failures
- Provide many of their services transparently, allowing the user to concentrate on the problem at hand
- Both have an architecture where a software cluster sits on the physical servers and controls all aspects of system execution

4.1 Hadoop Distributed File System (HDFS)

HDFS [5] the only building block in the Hadoop stack. Hadoop is a file system capable of storing bytes of data, if we take a file system view of the world then users want a common distributed file system and HDFS is a perfectly reasonable alternative. It has one main difference between others file system is "Write-Once-Read-Many" times' model support concurrency control, high fault-tolerance, and self-healing and high throughput access. Unique attribute of HDFS [6] is to locate pre-processing logic near the data rather than moving the data to the application space by Storing petabytes of web data, logs, and web snapshots runs on Commodity X86 servers, storage(SAN), Gigabit Ethernet LAN with Open source software that Keep track per-node costs down to afford more nodes computation in individual server.
HDFS consists of following daemons: HDFS master "NameNode", worker "DataNode", Secondary-NameNode.

HDFS Architecture



NameNode-A master server manages the file system namespace and regulates access to files by clients using Meta-data in memory, the entire metadata is in main memory with Meta-data types, lists of blocks for each file, datanodes for each block and file attributes.

DataNode-One-per node in the cluster manages storage attached to the nodes that they run on, by block server stores data in the local file system in terms of (/ext3) format and Serves data and metadata to clients with block reports and sends existing blocks to the NameNode

Secondary NameNode-Not used as host, stand-by or mirror node. Backup NameNode periodically wakes up and processes check point and update the NameNode. And Directory is same as NameNode except it keeps previous checkpoint versions in addition to current. Used to restore failed NameNode.

4.2 MAPREDUCE

Model for processing large amounts of data in parallel on commodity hardware and Lots of nodes Derived from functional programming known Map and reduce functions can be implemented in multiple languages Java, C++, Ruby, Python etc, imposing key-value input/output. Defines map and reduce functions map: (K1, V1) gives the list (K2, V2) and reduce: (K2, list (V2)) gives the list (K3, V3). Map function is applied to every input key-value pair. Map function generates intermediate key-value pairs, intermediate key-values are sorted and grouped by key, and Reduce is applied to sorted and grouped intermediate key-values. Reduce emits result key-values. MapReduce framework takes care ofdistributed processing and coordination in terms of Scheduling Jobs is broken down into smaller chunks called tasks These tasks are scheduled with respect to Localization with Data and Framework strives to place tasks on the nodes that host the segment of data to be processed by that specific task and where code is moved to the data is located with Error Handling without failures expectations so that tasks are automatically re-tried on other machines with Data Synchronization, Shuffle and Sort re-arranges and moves data between machines and Input and output are coordinated by the framework.

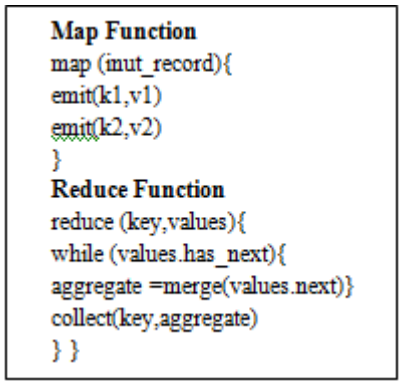
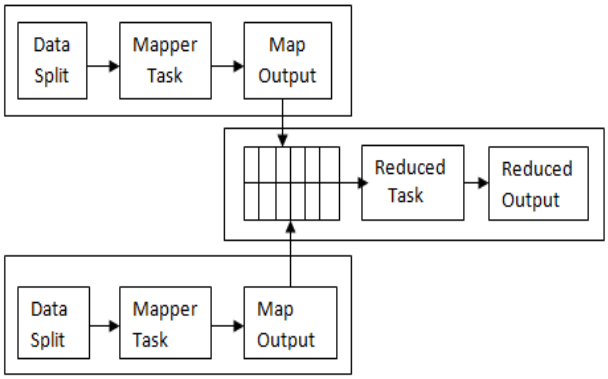


Figure 2: MapReduce Data Flow



5. Mapreduce Schedulers Optimization

Cluster compute capacity, the aggregate of worker node CPU, memory, network bandwidth, and disk IO, is a finite resource. Today, Hadoop represents these resources

primarily in the form of map and reduce task slots Task assignment is one of the important issues in the processing of big data in Hadoop. Schedulers are more responsible for doing task assignment. There are two types of nodes called JobTracker handled by the NameNode and tasktracker handled by datanode are taking part in the job execution process. Jobtracker acts as a master coordinator of all jobs executing in the master machine while the TaskTracker executes the tasks and sends a continuous report to the jobtracker.

When a MapReduce job is executed by the client, a list of input splits is first computed. Each split is a range of the data set to process, as determined by the input format. The jobtracker creates a map task for each input split and based on the job configuration, adds these tasks to a designated queue. The scheduler decides what tasks, from what queues, should be processed on what tasktrackers, in what order. Because as per the survey there are different scheduling algorithms, each with their own benefits and optimizations, there are multiple schedulers' choices but one can choose from when configuring a cluster. Only one scheduler at a time may be configured.

5.1 Default FIFO Scheduler

The first in, first out (FIFO) scheduler is the default scheduler in Hadoop. As the name implies, it uses a simple "first come, first served" algorithm for scheduling tasks. Supports five levels of job prioritization, from lowest to highest: very low, low, normal, high, very high. Each priority is actually implemented as a separate FIFO queue. Tasks are then scheduled left to right top to bottom. This means that all very high priority tasks are processed before any normal priority tasks, and soon. Beyond prioritized task scheduling, the FIFO scheduler does not offer much in the way of additional features. For small, experimental, or development clusters, the FIFO scheduler can be adequate.

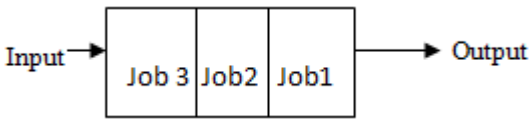


Figure 3: A single FIFO Queue order.

5.2 Fair Scheduler

Is an alternative scheduler to the default FIFO scheduler [1] [2]it was developed by Facebook to solve some of the problems that arise when using the FIFO scheduler in high traffic multitenant environments with the issues of resource starvation.In Fair Scheduler Jobs are submitted to queues by creating pools. Eachpool is assigned a number of task slots based on totalcluster capacity defined. The total number tasks in pool have minimum slot guarantees, and available slot capacity but pools may optionally have minimum slot guarantees.This scheduler uses some set of rules to decide how resources to slots are assigned for each created pools. Every time a tasktracker will communicates to the jobtracker for every three minute and it can also set by default as per the user demand called Heartbeat and it generates reports on datanodes to ensure if they are alive or dead and also for available remaining slots, the rules are evaluated and queued tasks are assigned for execution.

5.3 Capacity Scheduler

Developed by the Hadoop team at Yahoo. The Capacity Scheduler [1] [2] supports a feature where job initialization is performed lazily; one of the most significant features of the Capacity Scheduler is the ability to controlallocation based on physical machine resources. But it is different for FIFO and Fair schedulersthey work exclusively in terms of slots, but the Capacity Scheduler additionally understandsscheduling tasks based on user defined memory consumption of a job's tasks as well.When properly configured, the scheduler uses information collected by the tasktracker for scheduling decision. Configuring the Capacity Scheduler needs to set the value of mapred.jobtracker.taskscheduler to its class name.Overall, this has the effect of enforcing cluster capacity sharing among users, rather than among jobs, but different in the case in the Fair Scheduler and FIFO scheduler.

6. Methodology for Scheduler Optimization

After initial setup of hardware and software components the cluster are verified and determined to be operating as well. The default numbers of MapReduce slots, as well as the default Java heap size are normally not sufficient for most Hadoop workloads. Therefore, the first step while establishing the performance is taken care and potentially adjust these configuration parameters. The goal is to maximize the hardware resource utilization like CPU of the cluster, The Java heap size requirements, the number of MapReduce task, the number of CPU cores, the IO bandwidth, as well as the amount of available RAM impact the baseline setup. Our proposed system uses MapCombineReducer and different Block-Size approaches for above mentioned three schedulers.

6.1 MAPCOMBINEREDUCE

MapCombineReduce processing model is used for performance optimization of Reduce phase. Consist of three phases i.e. Map phase, Combine phase and Reduce phase. Figure 4 shows the MapCombineReduce work flow. Combiner function to be run after the map function and before the reduce function. It helps to decrease the amount of data shuffled between map tasks and reduce tasks. It's used to optimize bandwidth usage of MapReduce and to reduce the reduce phase work.

map (k1,v1) → list (k2,v2)
combine (k2,v2) → list (k3.v3)
reduce (k3,v3) → list (k4,v4)

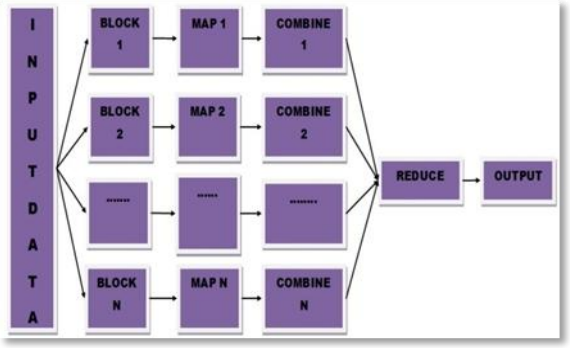


Figure 4: MapCombineReduce work flow

6.2 Different Block Sizes

Vary the block sizes of the data that is used by the HDFS file system for data distribution. Block sizes have the following effects on the performance metrics. Higher the block sizes it is easier for the HDFS to manage the information in the Namenode and lesser is the communication to the Namenode. Furthermore, when multiple set of data blocks for different applications are placed on a data node, accesses to the I/O (throughI/O scheduling) by different applications will have an impact on the completion times. On the other hand, increasing the block size reduces the parallelism that can be exploited across the clusters. When the cluster size is large, this will have a huge impact on the overall execution time.

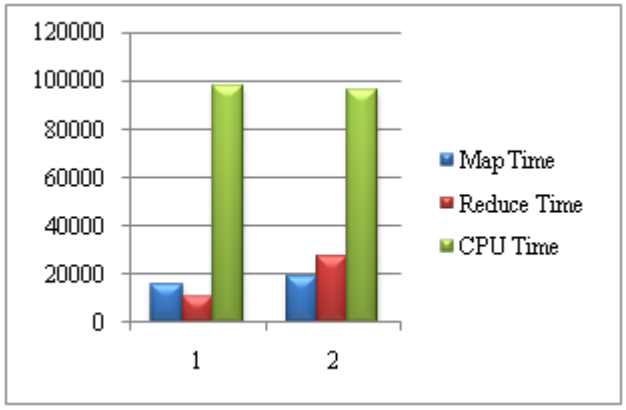
7. Experimental Setup

To evaluate the performance of our system and scheduling algorithm on a Hadoop (Hadoop version 1.0.3) with a single master node in Ubuntu 12.04 LTS with system configuration like 64-bit operating system, x64 based processor with 4.00GB RAM. Now the results are taken only for default internal scheduler that is FIFO scheduler with the approaches mentioned above of input data size. Job1 (195 MB) and Job2 (160 MB) of web log data as an example of WordCount.

8. Results

The below table and graph display the input and output values considered for execution of WordCount program using MapReduce framework. Here integer 1 and 2 in the graph indicates Job1 and Job2 with Default Block size of 64MB with FIFO scheduler.

Data Set Name	Size in MB	Map Time	Reduce Time	CPU Time
Web_log	195 MB	15697	10741	97910
Web_log	160 MB	18915	27429	96430



9. Conclusion and Future Work

This survey paper gives a brief idea about different task schedulers in Hadoop MapReduce. It compares the properties of various task schedulers. According to the survey of different schedulers we can come to know that local data processing takes lesser times as compared to moving data across network. So to improve the performance of jobs most of the algorithms work to improve the data locality. So to meet our expectations scheduling algorithms must use some of the prediction methods based on the small or large volume of data to be processed with underlying software or hardware. So as a future work going to optimize the three basic schedulers that are used in Hadoop environment namely First in First out (FIFO), Fair and Capacity scheduler by setting some benchmarks for each individual scheduler. In this project we kept two approaches for scheduler optimization namely MapCombineReducer and Different Block Sizes. For example the default block size is 64MB and by changing it to 128MB and so on for optimizing the scheduler for handling homogeneous and heterogeneous Hadoop jobs in multiple clusters which can schedule the jobs efficiently.

References

[1] Hadoopjob scheduling:http://blog.cloudera.com/blog/2008/11/job-scheduling-in-hadoop
[2] Hadoopscheduling: http://www.ibm.com/developerworks//library/os-hadoop-scheduling/
[3] Apache Hadoop. http://hadoop.apache.org.
[4] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. OSDI '04, pages 137–150, 2004.
[5] Sanjay Ghemawat, Howard Gobioff, and Shun-TakLeung.The Google file system. In 19th Symposium on Operating Systems Principles, pages 29–43, Lake George, New York, 2003.
[6] Hadoop Distributed File System, http://hadoop.apache.org/hdfs.
[7] Apache Hadoop: http://Hadoop.apache.org.
[8] Intel, Optimizing Hadoop Deployments, Intel White Paper, 2010, http://www.intel.in/content/dam/doc/white-paper/cloud-computing-optimizing-hadoop-deployments-paper.pdf.
[9] Chunk Lam, *Hadoop in Action*, Second Edition.

Authors Profile

Mr. Shivaraj B G completed the bachelor's degree in Information Science and Engineering from STJIT, Ranebennur, and Pursuing M.Tech in Computer Network Engineering at MITE, Mangalore under VTU, Belgaum.

Mr. Nagaraj Naik senior assistant professor MITE, Mangalore. Completed his M.Tech in computer science and engineering having 8.5 years of academic experience and his areas of interest are java programming, operating systems.