STUDY OF VARIOUS CONFIGURATION PARAMETERS OF APACHE HADOOP

[1]Bhavika A Tailor, [2]Prof Hetal A Joshiara
[1]ME Student, [2]Assistant Professor
[1]Department of Computer Engineering, [2]Department of Computer Engineering
[1]L.D College of Engineering, Gujarat Technological University, Ahmedabad, India
[2]L.D College of Engineering, Gujarat Technological University, Ahmedabad, India

Abstract: Big data is emerging technology. Processing and storing these huge data on the distributed hardware clusters needs a powerful computation model like Hadoop. Running Hadoop jobs in cloud environment are beneficial because it provides flexible, scalable, elastic, multitenant and on demand infrastructure. Despite of its potential benefits running Hadoop jobs inside cloud environment is challenging task due to diversification of model, technologies. Performance of Hadoop cluster in cloud environment can be optimized by tuning its default configuration parameters values. In this paper, we study and analysis various Hadoop configuration parameters. Performance of Hadoop cluster can be improved by tuning the value of configuration parameters according to the characteristics of job and virtual machine state and type of application.

IndexTerms: *Hadoop, Map reduce, Apache Hadoop YARN, Configuration Parameters.*

I. INTRODUCTION

Big data is massive volume of data that is so huge like terabytes, petabytes of data generated through various kind of social media and networking sites like facebook, twitter, Sensor devices, Scientific instruments, mobile devices and mobile networks, web sites clicks and various kind of transactions likes banking, e-commerce web site transaction. Velocity of big data generation is fast and required fast processing like online data analytics. Big data is in variety of forms like structured, semi-structured and unstructured data like text, images, pictures, numbers, audio, video, sensor data, social media data, static data, streaming data. Processing and storing these huge data on the distributed hardware clusters needs a powerful computation model like Hadoop [2].

Hadoop is open source, scalable, fault tolerant framework written in java. It is used for processing and storing big data. It performs automatic parallelization and distribution of work. It consists of main three key component: HDFS (Hadoop distributed file system) is used for storing data, Map reduce is used for data processing and YARN (Yet Another Resource Negotiation) is used for resource management.

Hadoop is basically designed for cluster environment. On the downside, the cluster can be used by a single big data application, resources cannot be scaled independently. It is challenging to manage, scale and upgrade the cluster.So, Hadoop jobs with huge computing requirement are preferred to be executed on cloud due to the benefits like easy provision of infrastructure, scalability, pay-per-use, flexibility, elasticity. Deployment of cluster on cloud is effected by virtualization and multi tenancy factor which affects the performance of the job. Schedulers that are provided by Hadoop consider either time of arrival or the maximum capacity guaranteed. All the schedulers are not consider parameters related to machine performance and job characteristics [5].

Resource scheduling and execution time of Hadoop job running on cloud can be optimized and make it suitable for cloud environment by customizing the value of configuration parameters like virtual machine states like the load on physical machine, amount of memory free, no of task currently running on virtual machine, job running on the VM(is it CPU intensive or memory intensive), CPU utilization and job characteristics like CPU usage each task takes.no of task remaining to the complete the job, is job is CPU intensive or Memory intensive, priority of job[5].

II. APACHE HADOOP

Hadoop is open source framework that allows distributed processing large data sets across the cluster of commodity hardware. Hadoop is scalable and fault tolerant framework written in java.
It consists of main three component:

a) HDFS: It is used for storing data.
b) Map Reduce: It is used for data processing.
c) YARN: It is used for resource management.

Shown in fig 1 Hadoop follows a master slave architecture design for data storage and distributed data processing using HDFS and MapReduce respectively. The master node for data storage is Hadoop HDFS is the NameNode and the master node for parallel processing of data using Hadoop MapReduce is the Job Tracker. The slave nodes in the Hadoop architecture are the other machines in the Hadoop cluster which store data and perform complex computations. Every slave node has a Task Tracker daemon and a

DataNode that synchronizes the processes with the Job Tracker and NameNode respectively [8]. Master nodes uses mapreduce process to assign task to slave node which is executed in parallel on each node. After slave nodes complete their tasks return result to the master node. Reduce process regroup data and assemble the result and this result return to the user.

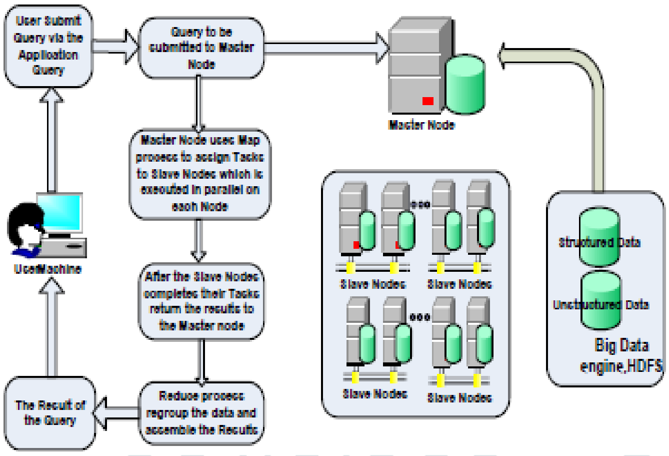

Figure 1 Sequence in Hadoop framework [6]

III. HADOOP CONFIGURATION PARAMETERS

Performance of job is affected by many factors which are differ from environment to environment. Hadoop is basically design for cluster environment. If it is deployed on cloud then job is affected by many factors due to diversification of tools and technologies. Parameters play a major role when Hadoop job is executed on cloud. Hadoop provides various parameters which are categorized into four parts 1) CPU 2) Memory 3) Disk I/O and 4) Network. Performance of the Hadoop job on cloud can be optimized by tuning the default values of Hadoop parameters according the job characteristics, virtual machine state, JVM etc. There are two type of Hadoop configuration files:

1) Read-only default configuration core-default.xml, hdfs-default.xml, yarn-default.xml and mapred-default.xml.
2) Site-specific configuration - core-site.xml, hdfs-site.xml, yarn-site.xml and mapred-site.xml which can be configured by user if required.

3.1 CPU related Hadoop Configuration parameters

Table 3.1: CPU related Hadoop configuration parameters [10]

Parameter Name	Default Value	Description
mapreduce.job.maps	2	How many map tasks per job will execute?
mapreduce.job.reduces	1	How many reduce tasks per job will execute?
mapreduce.tasktracker.map.tasks.maximum	2	The maximum number of map tasks that will be run simultaneously by a task tracker.
mapreduce.tasktracker.reduce.tasks.maximum	2	The maximum number of reduce tasks that will be run simultaneously by a task tracker.
mapreduce.map.speculative	True	If true, How many multiple instances of map tasks may be executed in parallel.
mapreduce.reduce.speculative	TRUE	If true, How many multiple instances of reduce tasks may be executed in parallel.
mapreduce.map.cpu.vcores	1	The number of virtual cores to request from the scheduler for each map task.
mapreduce.reduce.cpu.vcores	1	The number of virtual cores to request from the scheduler for each map task.

3.2 Memory Related Hadoop Configuration Parameters

Table 3.2: Memory related Hadoop configuration parameters [10]

Parameter Name	Default Value	Description
mapreduce.map.memory.mb	1024	Maximum memory request from the scheduler for each map task.
mapreduce.reduce.memory.mb	1024	Maximum memory request from the scheduler for each map task.
mapreduce.map.java.opts	-Xmx1024m	Amount of heap-size memory required for each child jvms of map task.
mapreduce.reduce.java.opts	-Xmx1024M	Amount of heap-size memory required for each child jvms of reduce task.
mapreduce.task.io.sort.mb	100	The total amount of buffer memory to use while sorting files.
mapreduce.task.io.sort.factor	10	The number of streams to merge at once while sorting files.
mapred.child.java.opts	-Xmx200m	Amount of required Java opts for the task processes.
mapreduce.map.sort.spill.percent	0.80	Limit of percent of spill for sort.
mapreduce.reduce.merge.inmem.threshold	1000	The number of files required for the in-memory merge process.
mapreduce.reduce.input.buffer.percent	0.0	The percentage of memory- relative to the maximum heap size- to retain map outputs during the reduce.
mapreduce.reduce.shuffle.input.buffer.percent	0.70	The percentage of memory to be allocated from the maximum heap size to storing map outputs during the shuffle.
mapreduce.reduce.shuffle.merge.percent	0.66	The percentage of the total memory allocated to storing in-memory map outputs.

3.3 I/O Related Hadoop Configuration Parameters

Table 3.3: I/O related Hadoop configuration parameters [8,10]

Parameter Name	Default Value	Description
dfs.blocksize	134217728	The default block size for new files, in bytes.
dfs.replication	3	Default block replication.
dfs.datanode.data.dir	file://${hadoop.tmp.dir}/dfs/data	Location of data directories on each datanode.
fs.defaultFS	file:///	Location of Hadoop Distributed File system URL.
mapreduce.output.fileoutputformat.compress	false	Decide whether job output compressed or not?
mapreduce.output.fileoutputformat.compress.type	RECORD	Compression type of job output: NONE, RECORD or BLOCK.
mapreduce.output.fileoutputformat.compress.codec	org.apache.hadoop.io.compress.DefaultCodec	If the job outputs are compressed, how should they be compressed?
mapreduce.map.output.compress	False	Decide whether job output maps compressed or not?
mapreduce.map.output.compress.codec	org.apache.hadoop.io.compress.DefaultCodec	Decide compression codec when the map job outputs are compressed.
mapreduce.task.merge.progress.records	10000	The number of records to process during merge.
dfs.namenode.replication.interval	3	The periodicity in seconds with which the namenode computes replication work for datanodes.
io.file.buffer.size	4096	The size of buffer for use in sequence files.

3.4 Network Related Hadoop Configuration Parameters

Table 3.4: Network related Hadoop configuration parameters [10]

Parameter Name	Default Value	Description
net.topology.script.file.name	-	User-defined script to determine rack-host mapping to configure rack awareness.
mapreduce.reduce.shuffle.parallelcopies	5	The number of parallel transfers run by reduce during the copy(shuffle) phase.

3.5 YARN Configuration Parameters

Table 3.5: YARN configuration parameters [11]

Parameter Name	**Default Value**	**Description**
yarn.nodemanager.resource.cpu-vcores	-1	How many vcores that can be allocated for containers?
yarn.nodemanager.resource.memory-mb	4096	How much amount of physical memory, in MB, that can be allocated for containers?
yarn.scheduler.minimum-allocation-mb	1024	The minimum allocation for every container in MBs.
yarn.scheduler.maximum-allocation-mb	8192	The maximum allocation for every container in MBs.
yarn.scheduler.minimum-allocation-vcores	1	The minimum allocation for every container request in terms of virtual CPU cores
Yarn.scheduler.maximum-allocation-vcores	4	The maximum allocation for every container request in terms of virtual CPU cores
yarn.nodemanager.vmem-pmem-ratio	2.1	Ratio between virtual memory to physical memory when setting memory limits for containers.

IV CONCLUSION AND FUTURE WORK

We studied different parameters of Hadoop like CPU, I/O, memory, network, yarn configuration. They have default values which is not suitable for all type of application. Performance of Hadoop job on cloud environment is effected by virtualization. We found that Performance of Hadoop job on cloud environment can be improved by proper tuning of Hadoop parameters. In future we plan to improve performance of Hadoop cluster on private cloud by tuning parameter and making scheduling decision based on virtual machine state and job characteristics.

REFERENCES

[1] Lin Cai, Yong Qi, and Jingwei Li. 2017 A recommendation-based parameter tuning approach for Hadoop.IEEE 7th International Symposium on Cloud and Service Computing:223-230.

[2] Bhavin J. Mathiya and Vinodkumar L. Desai. Feb. 25 – 27, 2015 .Apache Hadoop Yarn Parameter Configuration Challenges and Optimization.IEEE International Conference on Soft-Computing and Network Security (ICSNS -2015).

[3] Ailton S Bonifacio, Andre Menolli and Fabiano Silva.2014.Hadoop MapReduce Configuration Parameters and System Performance: a Systematic Review.

[4] Janardhanan PS and Philip Samuel.2017.Study of Execution Parallelism by Resource Partitioning in Hadoop YARN.IEEE:1170-1175.

[5] A. Sree Lakshm and Dr. M. BalRaju. 2016.Towards Optimization of Hadoop Map Reduce Jobs on Cloud. International Conference on Computing, Analytics and Security Trends (CAST)College of Engineering Pune, India:255-260.

[6] Ehab Mohamed and Zheng Hong.2016.Hadoop -Map Reduce job scheduling algorithm survey. IEEE 7th International Conference on Cloud Computing and Big Data:237-242.

[7] http://hadoop.apache.org/

[8] http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/hdfs-defult.xml

[9] http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/core- --default.xml

[10] http://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/mapred-default.xml

[11] http://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-common/yarn-default.xml