IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 1, January 2017

Big Data: Analyzing Map Reduce Auditing Mechanism

K. Shouryadhar[1], C. Gazala Akhtar[2], M. Anantha Lakshmi[3]

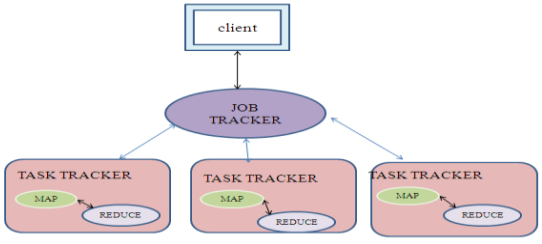
Assistant Professor, Dept of CSE, Ravindra College of Engineering for Women, Kurnool, Andhra Pradesh, India[1, 2,3]

Abstract: Apache hadoop is a distributed system for storing large amount of data and processing the data in parallel. This apache hadoop contains HDFS and MAP REDUCE as a core components .HDFS is as file system that can store very large data sets by scaling out across hosts of clusters .Map Reduce is a huge scalable, parallel processing framework that works in concurrent with HDFS.MAP REDUCE contains two main steps map and reduce. Map collects all the jobs carry out by job tracker and reduce module reduces all the map jobs which is carry out by task tracker. Developers use MapReduce for objects like filtering documents by tags, counting words in documents, and takeout links to related data. This paper describes about all the map reduce audit log events for counting the words that are being carry out during the execution of process. Logs are vital part of any computing system, supporting potential from audits to error management. As logs extends and the number of log origin increases (such as in cloud environment), a scalable system is compulsory to efficiently process logs. This procedure explores processing logs with Apache Hadoop from a distinctive Linux system. AUDITING for mapreduce mechanisms is a major concern, how tracing and logging significant events that could take place during a system run. These auditing mechanisms of map reduce audit logs are efficient, scalable, reliable. Map reduce audit logs have been one of the key enabling feature for security auditing.

Keywords: job tracker audit logs, task tracker audit logs, hadoop counter logs.

I. INTRODUCTION

MAP REDUCE is the heart of hadoop. It is the programming standard that allows for huge scalability across hundreds or thousands of servers in a hadoop cluster. The term map reduce actually refers to two separate and distinct task that hadoop program performs. The first is the map job that takes the set of data and converts it into further set of data, where particular elements are broken down into tuples (key/valve pairs).Second the reduce job takes the output from a map as input and combines those data tuples into a smaller set of tuples. As the sequence of mapreduce, the reduce job is always performed after the map job [1]. Map Reduce is a huge scalable, parallel computing framework that works in tandem with HDFS. With MapReduce and Hadoop, process is perform at the position of the data, instead of moving data to the process of position; data warehouse and computation exists together on the same physical nodes in the cluster. MapReduce processes overreach large amounts of data without being affected by traditional bottlenecks like network bandwidth by taking advantage of this data proximity. [2]



Key Map Reduce Features:

- **Scale-out Architecture** - Add servers to increase processing power
- **Security & Authentication** - Works with HDFS and HBase security to make sure that only approved users can operate against the data in the system
- **Resource Manager** - Employs data locality and server resources to determine optimal computing operations
- **Optimized Scheduling** - Completes jobs according to prioritization
- **Flexibility** – Procedures can be written in virtually any programming language
- **Resiliency & High Availability** - Multiple job and task trackers ensure that jobs fail independently and restart automatically.[3]

II. MOTIVATION

In this paper, we develop map reduce audit logs, which is a standard auditing mechanism. So from the client point of view, he sees the executed process of the map reduce for counting the no of words. Map reduce is a high scalable paradigm which allows massively parallel and distributed execution on large number of computing nodes. Apache Hadoop's daemons such as job tracker, name node, secondary name node, data node, and task tracker all generate logs. These hadoop daemons create logs for each node. Map reduce audit logs have been one of the key enabling feature for security auditing. Job tracker and task

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 1, January 2017

tracker daemons are generated by the map reduce paradigm .Namenode, secondary name node, data node these are generated by the hdfs which implements map reduce on the top of hdfs
In the functioning of map reduce; job tracker orchestrates all the jobs such as job id, job name, job configurations, job submission, job modification, job priority and so on…,creates execution plan, submits jobs to task tracker, manages phases and update status. Job tracker provides the history viewer where all the job related operations of the whole history is stored. This history viewer encounters with the setup of map and reduce, which Hadoop maintains some built-in counters for every job which reports various metrics for the job. The counters include file input and output formats, map reduce frameworks, file system counters. Each of these counters counts the no of bytes in the words, map input records, map output records, reduce shuffle bytes, spilled records, cpu time spent, file bytes read, file bytes written, data storage in physical memory and so on…

In the functioning of task tracker job it executes the job tasks and progress the reports. It contains task tracker history viewer views the reduce output of the map reduce operation. In case of job failure or else task failure, it again generates the jobs. The result of these map reduce audit logs shows of the efficient scalable parallel processing which map and reduce tasks works independently.map reduce audit logs are reliable as it analyze the job success and failures. The performance of map reduce is achieved with analyzing with the best performing map task.

III.RELATED WORK

In its current state, Hadoop map reduce is an extensible framework that allows users to write their own tests/rules for analyzing MapReduce applications. Job Configuration and Job History logs are input for this study, but going ahead we plan to merge the tool with data from Green plum Command Center, a management and tracking principle for both GPDB and GPHD. This will enable hadoop map reduce to incorporate more sources of information from the cluster such as daemon/user logs, audit logs, job queue data and system standard into its survey. It will also authorize real-time job study when running MapReduce jobs. [4]

Parallel to this work, other researchers did a large scale characterization of MapReduce workloads, including some insights on data access patterns. [4] Their work concentrates on interactive query workloads and did not study the batch type of workload that PROD has. Furthermore, the logs they processed were those of the Hadoop scheduler, and for this reason the authors did not have access to information like age of the files in the system, or when a file is deleted. Perhaps the work most similar to ours (in approach) is that of Cherkasova and Gupta [5], who characterized enterprise media server workloads. An analysis of the influence of new files and file life span was made, but they did not possess file creation and deletion time stamps, so a file is considered to be “new” the first time it is accessed, and its lifetime “ends “the last time it is accessed. No analysis on the burstiness of requests was made. Their results have been cited in this paper where appropriate, to enable us to contrast MapReduce workloads with a more traditional workload. [6][7]

IV. PROPOSED SYSTEM

We have proposed an audit logs where a logs shows the entire details of the hadoop and map reduce mechanisms carry out, it automatically create logs per each hadoop daemons. If client perform some operations on files such as editing, modifying, deleting these details are stored in a log4 of history viewer.
Map reduce has master/slave architecture. A map reduce cluster consists of a job tracker, a master server that manages the jobs. In addition, there are a number of task trackers, usually one per node in the cluster, which manages the execution of jobs and reports the progress. These map reduce daemons generates audit logs for each of the daemons. Hadoop contains hadoop logs, hdfs logs, map reduce job tracker logs, map reduce task tracker logs.
Hadoop daemons exists on all machines running at least one daemon. Some of the files end with .log and some with .out. Hdfs logs. HDFS audit logs generates logs by default it is written to the namenode logs.map reduce job history logs generates the logs of job execution process, job completion process etc.map reduce task tracker generates logs of task completion details like success, failure etc. This entire system of hadoop frame work specifies with a logs which are an essential part of any operating system, which holds capabilities from audits to bug management.[11] As logs and the number of log sources increases (such as in cloud domain), is a scalable system for effective computing of logs. This exercise session explores processing logs with Apache Hadoop from a typical Linux system. The existence of these audit logs provides with an efficient processing of data, best performing of map reduce job.

4.1 job tracker audit logs

The Job Tracker is the service within Hadoop that farms out Map reduce tasks to specific nodes in the cluster, preferably the nodes have the information of data, or at least present in the same rack. Client approaches jobs which submits to the Job tracker, the Job Tracker talks to the name node to determine the location of the data. After locating the data, the Job Tracker locates task tracker nodes with available slots at or near the data. The Job Tracker submits the work to the chosen task tracker nodes. The task tracker nodes are observed, if they do not submit heartbeat signals frequently, they are consider to have failed and the task is scheduled on a different task tracker. A task tracker will notice the result of Job Tracker when a task fails, it may resubmit the job somewhere else, it may mark out that specific record of

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 1, January 2017

information as something to neglect, and it may even block the task tracker as untrustworthy. When the work task is finished, the Job Tracker updates its status. The Job Tracker is a point of failure for the Hadoop map reduce service, If it goes down, all running jobs are halted.[8]

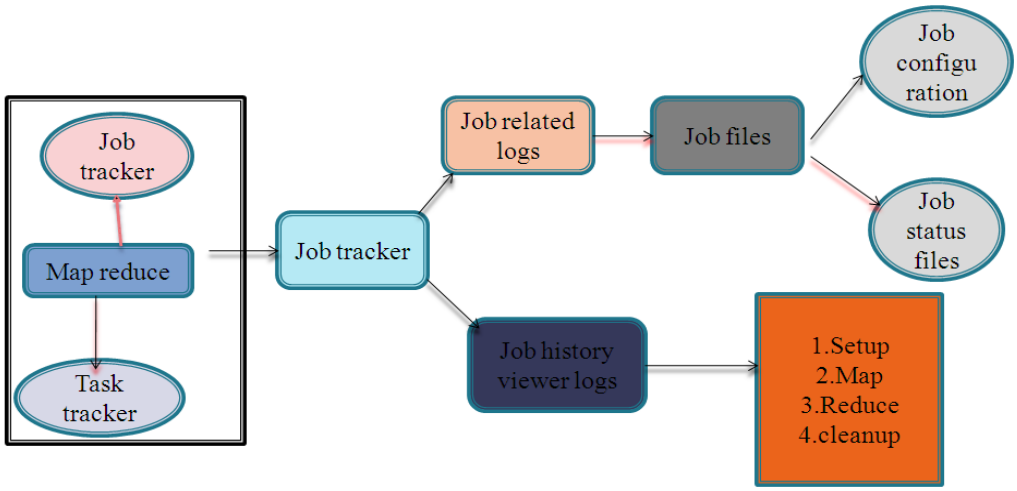


Figure 2 job tracker audit logs

Hadoop comes configured with a single compulsory queue, called 'default'. Job Configuration XML**:** The job configurations XML logs are created by the job tracker. These logs are stored in two places: /user/log/ hadoop and /user/log/hadoop/history.

Table 1 configuration xml

Name	Value
Fs.s3n.impl	Org.apache.hadoop.fs.s3 native.natives3filesystem
Mapred.task.cache.levels	2
Hadoop.tmp.dir	/tmp/hadoop-$(user.name)
Dfs.datnode.address	0.0.0.0.5001

4.2 task tracker audit logs:

The Job Tracker is the master overseeing the overall execution of a MapReduce job and the Task Trackers manage the execution of individual tasks on each slave node. Each Task Tracker is responsible for executing the individual tasks that the Job Tracker assigns. Although there is a single Task Tracker per slave node, each Task Tracker can spawn multiple JVMs to handle many map or reduce tasks in parallel. One responsibility of the Task Tracker is to constantly communicate with the Job Tracker. If the Job Tracker fails to receive a heartbeat from a Task Tracker within a specified amount of time, it will assume the Task Tracker has crashed and will resubmit the corresponding tasks to other nodes in the cluster. [9]

These include the Standard Out and Standard Error logs as well as custom user logs created by code using the logging framework within MapReduce jobs. The location is hardcoded to be ${hadoop.log.dir}/userlogs.

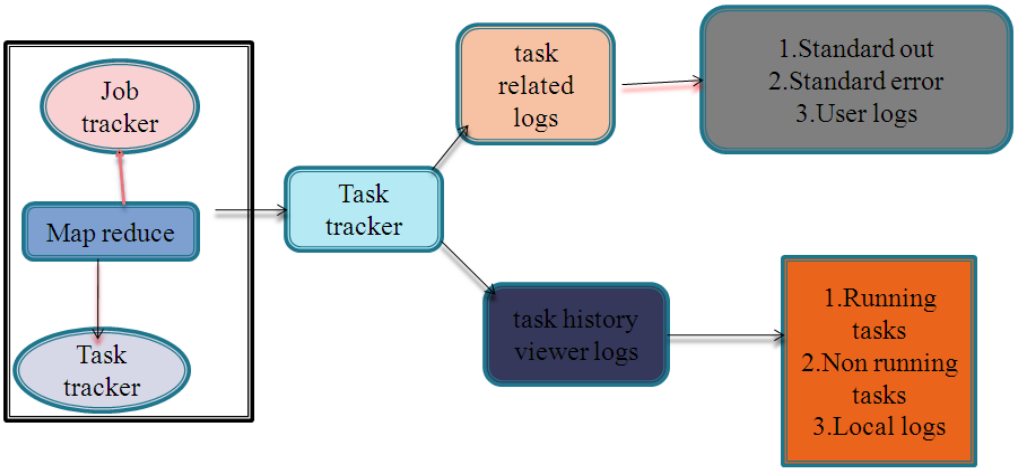


Figure 3 Task tracker audit logs

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 1, January 2017

It is written in the userlogs subdirectory of the directory defined by the HADOOP_LOG_DIR environment variable.these job tracker and task tracker generates the counters,where all the details of the logs of word count is carry out.

4.3 Hadoop counters logs:

Counters are a useful channel for conventional statistics about the job: for quality control or for application level-statistics. They are also useful for problem detection. Hadoop maintains some integral counters for every job which reports various metrics for the job. [10]

Table 2 counters

Group	Counter	description
File input format counters	Bytes read	
	Map output materialized bytes	
	Map input records	
	Reduce shuffle bytes	
	Spilled records	
	Map output bytes	
	Total commited heap usage	
	Cpu time spent	
Map reduce framework	Map input records	No of input records consumed by all jobs
	Split-raw-bytes	The number of records split by all the maps in the job
	Reduce input records	The number of input records consumed by all the reducers in the job
	Reduce input groups	The number of distinct key groups consumed by all the reducers in the job
	Combine ouput records	The number of output records produced by all the combiners (if any) in the job.
	Physical memory(bytes) snapshots	Total physical memory of bytes occupied
	Reduce output records	The number of reduce output records produced by all the maps in the job.
	Virtual memory(bytes)snapshot	Total virtual memory of bytes occupied
	Map output records	The map output records are produced by all the maps in the job
File ouput format counters	Bytes written	Writes all the bytes
File systems counters	File_bytes read	The numbers of bytes read by each filesystem by map and reduce tasks.
	File_bytes written	The number of bytes written by each filesystem by map and reduce tasks.
Job counters	Launched reduce task	The number of reduce tasks that were launched. Includes tasks that were started speculatively.
	SLOTS-MILLIS -MAPS	Slots all maps
	Total time spend by all reduces	Time of the job reduce
	Total time spend by all maps	Time of the map task
	Launched map tasks	The number of map tasks that were started, encompass tasks that were started unpredictable.
	Datalocal map tasks	The number of map tasks that ran on the same node as their input data.
	SLOTS-MILLIS -REDUCES	Slots all reduces

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 1, January 2017

These are all the counter logs which generate the word count logs

V. SIMULATION AND RESULTS

The following map reduce audit logs provides with a detail list of auditing mechanisms. . It contains a key component for security mechanisms.

5.1 Analyze the job logs

We can analyze the working of map reduce job in detail. It mainly shows the time taken by the best performing map task

1. Average time taken by Map tasks: 4sec
2. Worse performing map tasks
3. The last Map task i.e. task_201409031606 _m _000000 finished at (relative to the Job launch time): 3/09 16:08:37 (11sec)

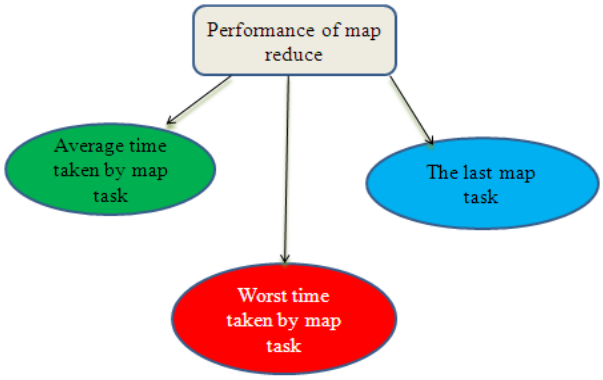


Figure 4 performance module of map reduce

5.2 Execution of word count program

During the execution of map reduce word count program, the following logs are being executed line by line. These a logs simulates the mechanisms of the hadoop.

Table 3 execution logs of map reduce

Logs	Example
1. Job	Job JOB_ID =”201409021524_0001”
2.Job name	JOBNAME=”word count”
3.user	1) USER=”hduser1”
4. Submit time	2) SUBMIT TIME=”140965183261”
5. Job_conf	JOBCONF=”hdfs://localhost:9000/tmp/hadoop-hduser1/mapred/staging/job_201409021524_0001\.xml”
6. View job	VIEW_JOB=”*”
7. Modify job	MODIFY_JOB=”*”
8. Job queue	3) JOB_QUEUE=”default”
9. Job job_id	job JOB_ID=”job_201409021524_0001”
10. Job priority	JOB_PRIORITY=”NORMAL”
11. launch time	LAUNCH_TIME=”1409651832866”
12. Total maps	4) TOTAL MAPS=”1”
13. Total reduces	TOTAL_REDUCES=”1”
14. Job status	JOB_STATUS=”PREP”
15. Task id	task TASK_ID=”Task_201409021524_0001_M_000002”
16. Task type	TASK_TYPE=”SETUP”
17. Start time	START_TIME=”1409651833040”
18. Splits	SPLITS=””
19. Map attempt task time task id	Mapattempt TASK_TYPE=”SETUP”
20. Task attempt id	TASK_ATTEMPT_ID=”attempt_201409021524_0001_m_000002_0”
21. Start time	START_TIME=”1409651833582”
22. Tracker name	TRACKER_NAME=”tracker_akhtar:localhost/127\0.\.0\1:55078”
23. Http port	HTTP_PORT=”50060”
24. Task status	TASK_STATUS=”SUCCESS”
25.Finish time	FINISH_TIME=” 1409651837932”
26.Host name	HOSTNAME=”/default-rack/Akhtar”
27.state	STATE-STRING=”SETUP”
28.Counters	COUNTERS=” {FileSystemCounters}{mapreduceframework}{file formats}”

In this paper, we aim to interact the job tracker and task tracker with audit logs in activity and enforce the execution along expected output.

Whenever a program get executes the logs are being generated for the particular hadoop daemon.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 1, January 2017

VI. CONCLUSION

Logs are a necessary part of any operating system, which supports capabilities from audits to **bug** management. As logs increases, the number of log origin of sources increases (such as in cloud domain), a scalable system is necessary to efficiently process logs. This exercise session identifies processing logs with Apache Hadoop from a typical Linux system. This paper concludes about the auditing of map reduce mechanisms, which keeps the tracks of who accesses what.

REFERENCES

[1] What is map reduce http://www-01.ibm.com/software/ data/ infosphere/hadoop/mapreduce/
[2] Map reduce programming model http://en.wikipedia.org/ wiki/MapReduce
[3] Map reduce key features Cloudera http://www.cloudera.com/ content/cloudera/en/products-and-services/cdh/hdfs-and-mapreduce.html
[4] Performance advisor for hadoop and map reduce jobhttp://blog.pivotal.io/pivotal/products/hadoop-vaidya-performance-advisor-for-hadoop-mapreduce-jobs
[5] L. Cherkasova and M. Gupta, "Analysis of enterprise media server workloads: Access patterns, locality, content evolution, and rates ofchange,"IEEE/ACM Trans. Netw., vol. 12, no. 5, 2004
[6] B. Fan, W. Tantisiriroj, L. Xiao, and G. Gibson, "DiskReduce: RAIDfor data-intensive scalable computing," inProc. PDSW, 2009, pp. 6–10
[7] AStorageCentricAnalysisofMapReduceWorkloads:https://wiki.engr.illinois.edu/download/attachments/194990492/cabad_IISWC_2012.pdf
[8] Working of job tracker http://wiki.apache.org/hadoop/JobTracker
[9] Working of task trackerhttp://www.guruzon.com/6/hadoop-cluster/architecture/what-is-tasktracker-hadoop-cluster-functions-limitations
[10] hadoop counters –hadoop-the definitive guideewww.inkling.com/ read/hadoop-definitive-guide-tom-white-3rd/chapter-8/counters
[11] audit logs for hadoop daemons http://blog.cloudera.com/blog/ 2009/09/apache-hadoop-log-files-where-to-find-them-in-cdh-and-what-info-they-contain/