ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 7, Special Issue 6, May 2018

2nd National Conference on "RECENT TRENDS IN COMPUTER SCIENCE & INFORMATION TECHNOLOGY (RTCSIT'18)"

13th-14th March 2018

Organized by

Departments of Computer Science & Information Science Engineering, Sri Krishna Institute of Technology, Bangalore, India

Distributed Hadoop Mapreduce on the Grid

Harshitha.B.S[1] , Apeksha.T.Alva[2] , G.V.Kavya[3], Dr K.S Jagadeesh Gowda [4]

U.G. Students, Department of Computer Science & Engineering, Sri Krishna Institute of Technology, Bangalore, India [123]

Professor, Department of Computer Science &Engineering, Sri Krishna Institute of Technology, Bangalore, India [4]

ABSTRACT: MapReduce is an intense information preparing stage for business and scholarly applications. In this paper, we fabricate a novel HadoopMapReduce system executed on the Open Science Grid which traverses different foundations over the Joined States – HadoopOn the Grid (HOG). It is unique from past MapReduce stages that keep running on committed conditions like bunches or mists. Hoard gives a free, flexible, what's more, dynamic MapReduce condition on the crafty assets of the framework. In HOG, we enhance Hadoop's blame resilience for wide territory information examination by mapping server farms over the U.S. to virtual racks and making multi-foundation disappointment areas. Our adjustments to the Hadoop structure are straightforward to existing HadoopMapReduce applications. In the assessment, we effectively stretch out HOG to 1100 hubs on the framework. Moreover, we assess HOG with a recreated Facebook HadoopMapReduce workload. We presume that HOG's fast versatility can give equivalent execution to a devoted Hadoop bunch.

KEYWORDS: Distributed Computing, Clusters, Mapreduce, Grid Computing

I. INTRODUCTION

MapReduce [1] is a structure spearheaded by Google for handling a lot of information in an appropriated domain. Hadoop [2] is the open source execution of the MapReduce structure. Because of the straightforwardness of its programming show and the run-time resistance for hub disappointments, MapReduce is generally utilized by organizations, for example, Facebook [3], the New York Times [4], and so forth. Futher more, researchers likewise utilize Hadoop to obtain versatile and dependable investigation and capacity administrations. The University of Nebraska-Lincoln developed a 1.6PB Hadoop Distributed File System to store Compact Muon Solenoid information from the Large Hadron Collider [5], and also information for the Open Science Grid (OSG) [6]. In the University of Maryland, specialists created blastreduce in view of HadoopMapReduce to investigate DNA successions [7]. As HadoopMapReduce wound up prominent, the number and scale of MapReduce programs turned out to be progressively expansive. To use HadoopMapReduce, clients require a Hadoop plat- shape which keeps running on a committed domain like a group or cloud. In this paper, we develop a novel Hadoop stage, Hadoop on the Grid (HOG), in light of the OSG [6] which can give adaptable and for nothing out of pocket administrations for clients who plan to utilize HadoopMapReduce. It can be transplanted to other expansive scale conveyed framework frameworks with minor adjustments. The OSG, which is the HOG's physical condition, is made out of around 60,000 CPU centers and traverses 109 locales in the United States. In this across the country appropriated framework, hub disappointment is a typical event [8]. At the point when running on the OSG, clients from foundations that don't claim assets run astutely and can be seized at any time. A seizure on the remote OSG site can be caused by the handling work running over designated time, or if the proprietor of the machine has a requirement for the assets. Acquisition is controlled by the remote site's arrangements which are outside the control of the OSG client. In this way, high hub disappointment rate is the biggest obstruction that HOG addresses. Hadoop's adaptation to internal failure centers around two disappointment levels and utilizes replication to keep away from information misfortune. The primary level is the hub level which implies a hub disappointment ought not influence the information trustworthiness of the group. The second level is the rack level which implies the information is sheltered if an entire rack of hubs come up short. In HOG, we present another level which is the site disappointment level. Since Hoard keeps running on different destinations inside the OSG. It is conceivable that an entire site could come up short. Hoard's information arrangement and replication arrangement

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 7, Special Issue 6, May 2018

2nd National Conference on "RECENT TRENDS IN COMPUTER SCIENCE & INFORMATION TECHNOLOGY (RTCSIT'18)"

13th-14th March 2018

Organized by

Departments of Computer Science & Information Science Engineering, Sri Krishna Institute of Technology, Bangalore, India

considers the site disappointment when it places information pieces. The augmentation to a third disappointment level will likewise bring information area benefits which we will clarify in Section III. Whatever is left of this paper is sorted out as takes after. Segment II gives the foundation data about Hadoop and the OSG. We portray the engineering of HOG in segment III. In segment IV, we demonstrate our assessment of HOG with an outstanding workload. Segment V quickly depicts related work, and VI talks about conceivable future research. At long last, we condense our conclusions in Section VII.

II.RELATED WORK

Our stage is unique in relation to running HadoopOn Demand (HOD) [21] on the Grid. HOD will make a transitory Hadoop stage on the hubs acquired from the Grid Scheduler and close down Hadoop after the MapReduce work wraps up. For visit MapReduce asks for, HOD has high recreation overhead, settled hub number, and a haphazardly picked head hub. Contrasted with HOD, HOG does not have recreation time, has an adaptable size, and has a static devotedhead hub which has the JobTracker and the Namenode. In spite of the fact that Hoard loses hubs, and needs to rebalance information as often as possible, HOD reproduces the whole Hadoop bunch while HOG as it were redistributes information from hubs that are lost by seizure.
MOON [22] makes a MapReduce structure on opportunistic assets. Be that as it may, they require devoted hubs to give a grapple to supplement the shrewd assets.
Also, the versatility of MOON is restricted by the size of the committed grapple bunch. Hoard stores all information in the network assets and utilizations replication plans to ensure information accessibility while MOON requires one copy of every datum piece put away on the grapple bunch. The steady focal server in HOG resembles a grapple, however the versatility of HDFS isn't constrained by the extent of plate on the focal server, rather on the circle on the numerous hubs on the matrix.

III.ARCHITECTURE

The design of HOG is contained three segments. The first is the matrix accommodation and execution part. In this part, the Hadoop specialist hubs demands are conveyed to the framework and their execution is overseen. The second major component is the Hadoop appropriated record framework (HDFS) that runs over the matrix. What's more, the third segment is the MapReduce structure that executes the MapReduce applications over the matrix.

A. Matrix Submission and Execution

Matrix accommodation and execution is overseen by Condor also, GlideinWMS, which are non specific systems planned for asset portion and administration. Condor is utilized to deal with the accommodation and execution of the Hadoop specialist hubs. GlideinWMS is utilized to dispense hubs on remote locales straightforwardly to the client.
At the point when the HOG begins, a client can ask for Hadoop specialist hubs to keep running on the network. The quantity of hubs can develop and recoil flexibly by submitting and expelling the laborer hub occupations. The Hadooplaborer hub incorporates both the datanode and the tasktracker forms. The Condor accommodation record is appeared in Listing 1. The accommodation record indicates properties of the occupations that will run the Hadoop specialist hubs. The prerequisites line implements an arrangement that the Hadoop specialist hubs should just keep running at these locales. We limited our analyses just to these locales since they give freely reachable IP addresses on the laborer hubs. Hadoop specialist hubs must have the capacity to convey to each other specifically for information exchanging and informing. In this way, we limited execution to 5 destinations. FNAL_FERMIGRID and USCMS-FNAL-WC1 are groups at Fermi National Accelerator Laboratory. UCSDT2 is the US CMS Tier 2 facilitated at the University of California – San Diego. AGLT2 is the US Map book Great Lakes Tier 2 facilitated at the University of Michigan.

MIT_CMS is the US CMS Tier 2 facilitated at the Massachusetts Foundation of Technology. The executable indicated in the condor submit document is a basic shell wrapper content that will introduce the Hadoop specialist hub condition. The wrapper content takes after these ventures keeping in mind the end goal to begin the Hadooplaborer hub:

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 7, Special Issue 6, May 2018

2nd National Conference on "RECENT TRENDS IN COMPUTER SCIENCE & INFORMATION TECHNOLOGY (RTCSIT'18)"

13th-14th March 2018

Organized by

Departments of Computer Science & Information Science Engineering, Sri Krishna Institute of Technology, Bangalore, India

1) Initialize the OSG working condition
2) Download the Hadoop specialist hub executables
3) Extract the specialist hub executables and set late authoritative setups
4) Start the Hadoop daemons
5) When the daemons close down, tidy up the working index.

Introducing the OSG working condition sets the required condition factors for appropriate activity on an OSG specialist hub. For instance, it can set the double inquiry way to incorporate matrix document exchange devices that might be introduced in non-standard areas.

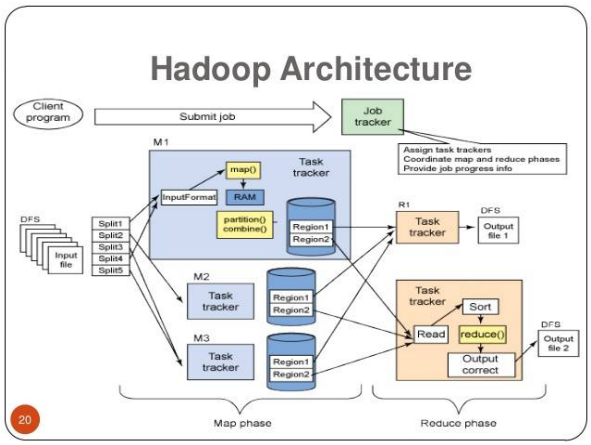

Fig 1.Hadoop Architecture

In the assessment the Hadoopexecutables bundle was compacted to 75MB, which is sufficiently little to exchange to laborer hubs. This bundle incorporates the setup document, the organization contents, and the Hadoop jugs. It can be downloaded from a focal vault facilitated on a web server. Decompression of the bundle takes a paltry measure of time on the specialist hubs, and isn't considered in the assessment. The wrapper must set arrangement alternatives for the Hadooplaborer hubs at runtime since nature isn't known until the point that it achieves the hub where it will execute. The Hadoop design must be powerfully changed to point the estimation of mapred.local.dir to a laborer hub nearby catalog. In the event that the Hadoop working registry is on shared storage, for example, a system document framework, it will moderate calculation because of document get to time and we will lose information territory on the laborer hubs. We abstain from utilizing shared capacity by using GlideinWMS's components for beginning an occupation in a specialist hub's nearby index.

B. HadoopOn The Grid

The Hadoop occurrence that is running on the matrix has two significant parts, Hadoop Distributed File System (HDFS) what's more, MapReduce, appeared in Figure 3. The ace servers, Namenode for HDFS and JobTracker for MapReduce, are single purposes of disappointment for the HOG framework, in this manner they dwell on a steady focal server. In the event that the ace server moves toward becoming inaccessible, execution of MapReduce employments will stop and the HDFS document framework will end up inaccessible (however no information will be lost). At the point when the matrix employments begin, the slave servers will answer to the single ace server. Hadoop requires laborer hubs to be reachable by each other. A few groups in the Open Science Grid are intended to be behind at least one machines that give Network Address Interpretation (NAT) access to the Internet. Hadoop can't converse with hubs behind a remote NAT in light of the fact that NAT squares coordinate get to. Consequently, we are constrained to locales in the OSG that have an open IPs on their specialist hubs. Disappointments on the framework are extremely regular because of the site un- accessibility and seizure. It is essential that HOG reacts rapidly to recuperate lost hubs by redistributing information and professional Accessing to outstanding hubs, and asking for more hubs from the

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 7, Special Issue 6, May 2018

2nd National Conference on "RECENT TRENDS IN COMPUTER SCIENCE & INFORMATION TECHNOLOGY (RTCSIT'18)"

13th-14th March 2018

Organized by

Departments of Computer Science & Information Science Engineering, Sri Krishna Institute of Technology, Bangalore, India

framework. The Hadoop ace hub gets pulse messages from the specialist hubs occasionally revealing their wellbeing. In Hoard, we diminished the time between pulse messages furthermore, diminished the timeout time for the specialist hubs. On the off chance that the specialist hubs don't report like clockwork, at that point the hub is checked dead for both the namenode and jobtracker. The customary incentive for the heartbeat.recheck.interval is 15 minutes for a hub before pronouncing the hub dead.

1) HDFS On The Grid: Creating and keeping up a distributed file system on a divergent arrangement of assets can be testing. Hadoop is suited for this test as it is intended for visit disappointments. In customary Hadoop, the datanode will contact the namenode and report its status including data on the size of the plate on the remote hub and what amount is accessible for Hadoop to store. The namenode will figure out what information documents ought to be put away on the hub by the area of the hub utilizing rack mindfulness and by the percent of the space that is utilized by Hadoop. Rack mindfulness gives both load adjusting and moved forward adaptation to non-critical failure for the document framework. Rack mindfulness is composed to isolate hubs into physical disappointment areas and to stack balance. It expect that data transfer capacity inside a rack is substantially bigger than the transmission capacity between racks, in this manner the name node will utilize the rack attention to put information nearer to the source. For adaptation to non-critical failure, the namenode utilizes rack attention to put information on the source rack and one other rack to watch against entire rack disappointment. A whole rack could fall flat if a organize part comes up short, or if control is hindered to the rack control supply unit. Be that as it may, rack mindfulness requires information of the physical format of the bunch. On the network, clients are probably not going to know about the physical format of the group, in this way customary rack mindfulness would be Infeasible. Rather, rack mindfulness in HOG is reached out to site mindfulness. We separate hubs in view of their locales. Locales are basic disappointment areas, thusly fitting great into the current rack mindfulness display. Locales in the OSG are generally one or a couple of bunches in the same authoritative area. There can be numerous disappointments that can cause a whole site to go down, for example, a center system part disappointment, or an extensive power blackout. These are the mistakes that rack mindfulness was intended to moderate.

Likewise, locales more often than not have high transmission capacity between their specialist hubs, and lower transfer speed to the outside world. This is synonymous with HDFS's presumptions about the rack, that the data transmission inside the rack is considerably bigger than the transmission capacity between racks. Locales are identified and isolated by the announced host- names of the laborer hubs. Since the specialist hubs require to be freely addressable, they will probably have DNS names. For instance, the DNS names will be separated into workername.site.edu. The laborer hubs will be isolated relying upon the last two gatherings, the site.edu. All laborer hubs with a similar last two gatherings will be resolved to be in a similar site. The identification and partition is finished by a site mindfulness content, characterized in the Hadoop design as topology.script.file.name. It is executed each time another hub is found by the namenode and the jobtracker. Notwithstanding site-mindfulness, we expanded the default replication factor for all records in HDFS to 10 copies from the customary replication factor for Hadoop of 3. Concurrent acquisitions on a site is normal in the OSG since higher need clients may submit numerous employments, appropriating a significant number of our HDFS occurrences. With a specific end goal to address synchronous preemptions, both site mindfulness and expanded replication are utilized.

Additionally, expanded replication will make preparations for acquisitions happening speedier than the namenode can recreate missing information pieces. An excessive number of imitations would force additional replication overhead for the namenode. Excessively few would cause visit information disappointments in the dynamic HOG condition. 10 imitations was the test number which worked for our assessment.

2) MapReduce On The Grid: The objective of our implementation is to give a Hadoop stage equivalent to that of a committed group for clients to keep running on the matrix. They ought not need to change their MapReduce code keeping in mind the end goal to keep running on our adjustment of Hadoop. Thusly, we rolled out no API improvements to MapReduce, just hidden changes with a specific end goal to better fit the lattice use demonstrate. At the point when the network work starts, it begins the tasktracker on the telecommuter hub. The tasktracker is accountable for overseeing the execution of Map and Reduce errands on the specialist hub. When it starts, it contacts the jobtracker on the focal server which denotes the hub accessible for handling. The tasktrackers report their status to the jobtracker and acknowledge errand assignments from it. In the present rendition of HOG we take after Apache Hadoop's FIFO work booking approach with theoretical execution empowered. Whenever, an undertaking has at most two duplicates of execution in the framework.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 7, Special Issue 6, May 2018

2nd National Conference on "RECENT TRENDS IN COMPUTER SCIENCE & INFORMATION TECHNOLOGY (RTCSIT'18)"

13th-14th March 2018

Organized by

Departments of Computer Science & Information Science Engineering, Sri Krishna Institute of Technology, Bangalore, India

The correspondence between the tasktracker and the activity tracker depends on HTTP. In the HOG framework, the HTTP solicitations and reactions are over the WAN which has high dormancy and long transmission time contrasted and the LAN of a group. Due to this expanded correspondence dormancy, it is normal that the startup and information exchange starts will be expanded. Similarly as site mindfulness influences information arrangement, it likewise influences the arrangement of Map occupations for handling. The default Hadoop scheduler will endeavor to plan Map undertakings on hubs that have the info information. On the off chance that it can't discover an information nearby hub, it will endeavor to plan the Map undertaking in an indistinguishable site from the information. Once more, Hadoop accept the data transfer capacity inside a site is more noteworthy than the data transfer capacity between destinations.

IV. EXPERIMENTAL RESULTS

As we specified in our presentation, HOG is adaptable. In the event that clients need to build the quantity of hubs in the HOG, they can submit more Condor occupations for additional hubs. They can utilize the HDFS balancer to adjust the information appropriation. In our experiments, we flexibly stretch out our framework from 132 to 1101 hubs and run the workload to confirm HOG's execution. . As a rule, we can get shorter reaction time if there are more hubs in the HOG framework. Be that as it may, there are drawbacks on the off chance that we continue expanding the size of HOG. Above all else, the information development and replication can't be disregarded on the grounds that this procedure will force additional overhead on the Namenode and Datanode. The execution of HOG will debase if replication occurs amid the execution of a workload. Also, the likelihood of losing a hub ascends with the expanding of centers in HOG. As should be obvious in the Table IV,
the more hub change, the more drawn out reaction we will get for a given workload. The execution may debase as hub disappointment increments. We adapted a few lessons amid the development of HOG. We show them in this segment.
1) Abandoned Data Nodes: In our first emphasis of HOG, the Hadoop daemons were begun with the Hadoop-given startup contents. These contents play out a "twofold fork", where the daemons will leave the procedure tree of the content and run autonomously in their own particular procedure tree. Sadly, numerous site asset administrators can't seize a daemon that has twofold forked, since they just take a gander at the procedure tree of the underlying executable. At the point when a site wishes to seize a HOG hub, it will to begin with murder the procedure tree, at that point expel the activity's working indexes. Since HOG daemons were outside of the procedure tree, they proceeded after the site slaughtered the framework work. At the point when the site erased the activity's working indexes, the datanode would bomb, however the tasktracker would keep working. At the point when the tasktracker acknowledged a guide or diminish work, it would fall flat instantly as it was not able spare the info information to plate. With a specific end goal to take care of the deserted daemon issue, we actualized two fixes. To start with, we adjusted the Hadoop source code to empower it to intermittently check the working catalog for writability and decipherability by composing a little record and understanding it back. We changed the Datanode.java class in Hadoop. The first usage of Datanode.java checks the circle accessibility when it begins. We include the circle accessibility check in benefit code and do the check at regular intervals. On the off chance that these tests bomb, at that point the daemons will close themselves down.
Second, we changed how we began the daemons to keep them in a similar procedure tree. In this execution, the datanode furthermore, errand tracker would not be lost to the framework as it was a coordinate kid procedure of the wrapper content. 2) Disk Overflow: During our tests, we took note numerous disappointments amid the execution of a MapReduce test work process. The blunders showed that the specialist hubs were coming up short on circle. Our replication factor and the high idleness between a few hubs on the matrix caused the circle floods. It is additionally important that Hadoop won't erase delineate information until the point that the whole occupation is finished. The high replication factor for HOG takes into account great information territory. With the information on an indistinguishable hub from the guide execution, perusing in the information is speedy. Be that as it may, each diminish necessities to get information from every mapper. Since the diminish stages keep running on different locales, the information should be exchanged over the WAN, which will be considerably more gradually than the guide errands. As guide errands finish, the guide assignments for resulting employments are executed. In the interim, lessen undertakings are finishing much slower. This prompts a development of transitional guide yield on the specialist hubs, making the hubs bomb because of absence of plate space. These disappointments are accounted for to the jobtrackeras a specialist hubs out of circle blunder.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 7, Special Issue 6, May 2018

2nd National Conference on "RECENT TRENDS IN COMPUTER SCIENCE & INFORMATION TECHNOLOGY (RTCSIT'18)"

13th-14th March 2018

Organized by

Departments of Computer Science & Information Science Engineering, Sri Krishna Institute of Technology, Bangalore, India

v .CONCLUSION

In this paper, we made a Hadoop framework based on the Open Science Grid. Our commitment incorporates the identification and determination of the zombie data node issue, site mindfulness, and an information accessibility answer for HOG. Through the assessment, we found that the inconsistency of the lattice makes Hadoop on the network testing. The HOG framework employments the Open Science Grid, and is consequently free for analysts. We found that the HOG framework, however hard to create, can dependably accomplish proportionate execution with a committed bunch. Furthermore, we demonstrated that HOG can scale to 1101 hubs with potential adaptability at bigger numbers. We will chip away at security and empowering various duplicates of guide and decrease assignments execution later on.

REFERENCES

[1] Grace NilaRamamoorthy:K-Means Clustering Using HadoopMapReduce. Published by UCD School of Computer Science and Informatics.
[2] Chen He, Derek Weitzel, David Swanson, Ying Lu. HOG: Distributed HadoopMapReduce on the Grid Published by 2012 SC Companion: High Performance Computing, Networking Storage and Analysis
[3] XuanWang,Clustering in the cloud:Clustering Algorithms to Hadoop Map/Reduce Framework" (2010),Published by Technical Reports-Computer Science by Texas State University.
[4] Apache Software Foundation, Hdfs user guide http://hadoop.apache.org/hdfs/docs/current/hdfsuserguide
[5] MapReduce tutorial Apache Hadoop 1.2.1 documentation by Hadoop wiki.
[6] Eucalyptus Systems, Inc.Eucalyptus 3.3.1, Eucalyptus Administration Guide (2.0), 2010.
[7] J.Dean and S.Ghemawat. MapReduce: Simplified Data Processing on Large Clusters.Proceedingsof 6th Symposium on Operating SystemDesign and Implementation,Published by Communications of ACM,Volume 51 Issue 1,January 2008.
[8] Cloudera. Cloudera's distribution including Apache hadoop.
[9] A.T. Velte,T.J. Velte,andR.Elsenpeter. Cloud ComputingA Practical Approach,Published by The McGraw-Hill Companies, 2010.
[10] Tom White Hadoop- The Definitive Guide,Published by O'Reilly Media/Yahoo Press, 2nd edition, 2010.
[11] Huan Liu and Dan Orban, Cloud MapReduce: a MapReduce Implementation on top of a Cloud Operating System. Published in Cluster, Cloud and Grid Computing(CCGrid) 2011,11th IEEE/ACM International Symposium.
[12] WeizhongZhao,HuifangMa,QingHe,Parallel K-Means clustering based on MapReduce Published by The Key Laboratory of Intelligent Information Processing, Institute of Computing Technology,ChineseAccademy of Sciences.
[13] The Apache HadoopEcosystem,University of Cloudera, OnlineResources.
[14] Amazon. Amazon elastic block storage (ebs),aws documentation by Amazon Elastic Compute Cloud User Guide.
[15] Blaise Barney, Introduction to Parallel Computing, Published By Lawrence Livermore National Laboratory.