ISSN:0975-9646

Hadoop Cluster on Linode Using Ambari for ImprovingTask Assignment Scheme Running in the Clouds

Minthu Ram Chiary, R. Anand

Computer Science Of Engineering, Saveetha University
Saveetha Nagar, Thandalam, Chennai – 602117,Tamil Nadu State, India

Abstract: currently, data-intensive problems are so prevailing that varied organizations in varied IT industries are facing in their business process. It's always critical for companies to process the potential of observing huge amount of information in an exceedingly smart and timely manner. Mapreduce‟s code document implementation of Hadoop dramatically simplified the event of parallel information computing applications for traditional users for information intensive parallerly, and thus the mixture of Hadoop and cloud computing created large-scale parallel information computing rather additional accessible and reliable to all or any or any potential users than ever before. Hadoop has become the foremost in popular information management framework for parallel data-in depth computing at intervals the clouds, the Hadoop component is not a perfect match for the cloud environments. During this paper, we have a tendency to discuss the problems faced by Hadoop for task assignment theme associate degreed gift an improved theme for heterogeneous computing environments, Apache Ambari with Parallel quick Fourier remodel. we have a tendency to conducted exhaustive simulation to evaluate the performance of the projected theme compared with the Hadoop theme in 2 varieties of heterogeneous computing environments that are typical on the overall public cloud platforms. The simulation results showed that the projected theme might remarkably shrink the map in half completion time, and it's going to shrink the amount of remote method accustomed a plenty of necessary extent that creates the data method less at risk of every network congestion and disk competition in lesser time.

Keywords: Cloud ,Parallel computing, Mapreduce,Linode Cluster,Apache Ambari.

I. INTRODUCTION

Have you ever thought that how google does their search engine queries in less than a second or how facebook does such a large quantities of data so quickly with their very huge mountains of data. According to re-searchers around 95% of the data was generated last 3-4 years ago, which was generated by smartphones, social networks, multimedias ,online app store and other more sources. In the earlier days when companies would experience very big-data, they would simply give larger checks to data-storage companies for managing their data safely. However companies like Google experienced vast amounts of big data, which were too large for data-storage vendors to manage it. So Google came out with a new algorithm, it was called “map reduce”. Which allowed them to cut their large data calculations into small units and map their data to many computers, and when the calculations were done the data will be brought back together to give out the result in less time. This algorithm was used to develop an open source project which we now know as 'Apache Hadoop' or simply just 'Hadoop', which allowed applications to run with the map-reduce algorithm. To make it more simple, simply assume that “we are processing data in parallel rather than in serial”. Some of the largest users of hadoop are Yahoo, linkedin and facebook etc. Data processing with hadoop is very fast, since we are processing in parallel. Here we have experimented hadoop cluster by using Ambari. Everyone knows, installing & managing a hadoop cluster is not easy, especially in vps env ironments. So the solution is apache ambari. Ambari is a project aimed at making hadoop management much easier by developing software for provisioning, managing, and monitoring hadoop clusters effectively. Ambari provides an instinctive, easy-to-use hadoop management serv ices and web backed by its restful APIs

II. HADOOP FAULT TOLERANCE MECHANISMS

Failures are principally inevitable once Hadoop runs at giant scales. Accordingly, Hadoop is assumed as a fault-tolerance framework which handles various failures with less impact on the standards of serv ice. There are totally three different failure methods, Task failure mode, Task Tracker failure mode, and Job Tracker failure mode. once the Task Tracker detect a task that has failed, it marks the task as failure and free the task slot on that the task that is active, and apprise the Job Tracker of the failure in its message. The JobTracker can then attempt to schedule execution of that task on a special TaskTracker. The entire job can fail, if any task fails a configurable range of times (four times by default), that typically suggests that the user code is buggy. TaskTracker failure happens, once the JobTracker hasn‟t received any heartbeat message from sure TaskTracker for a configurable amount of your time (10 minutes by default). Task Tracker failure may be a way more serious failure mode than task failure, as a result of the intermediate output of all map tasks that antecedently ran and finished on the failing Task Tracker becomes inaccessible. during this case, the JobTracker can rerun all those

completed map tasks, and schedule any tasks current on alternative TaskTrackers. JobTracker failure is that the most serious failure mode, however it's not going to happen because the likelihood that a specific machine fails is low. Within the cases of JobTracker failure, Hadoop prov ides a configuration choice which will plan to recover all jobs that were running at the time the failure occurred.

III. Issues Faced With Hadoop Task Assignment Scheme

Hadoop and MapReduce were originally designed for computer clusters rather than computer clouds. Clusters area unit is largely a similar computing setting, where similar nodes run in similar load conditions, and tasks of an equivalent sort tend to start out and end at roughly shutdown times. However, this matter is totally completely different within the clouds. Cloud service suppliers uses virtualization technology to abstract their physical resources, changes their usage, improve hardware utilization, and supply user remoteness for security functions. though currently virtualization technology will separate hardware and usage of memory effectively, co-located VM‟s still have to be compelled for each network and disk information measure, particularly within the case of Input intensive employment, like the MapReduce jobs. To overcome it we will install hadoop in Ambari and this issue will be solved.

IV. Getting Started With Hadoop In Ambari

At first we'll launch some virtual machines of Redhat Enterprise Linux or CentOS on any IaaS solutions like Amazon Web Services, rackspace, digitalocean, azure or other , or we can even use a local openstack cluster. Then we will use apache AMBARI(which is an easy and faster way to do it) to install Hadoop and all other necessary components, after setting up our hadoop cluster, we can now use it for performing complex calculation.

Steps for installing Hadoop cluster in Ambari

1.Launch a linode with at least 4GB of RAM (lets call it 'linode1')
2.Then deploy a CentOS Distribution to it
3.Now create SSH into the server Since there were no SSH keys added, so you will not find any '.ssh' directory here, you'll have to create them.
4.To create the .ssh directory
mkdir ~/.ssh
5.Change it's permission to 700
chmod 700 ~/.ssh
6.Now to generate the SSH keys
cd ~/.ssh
ssh-keygen -t rsa
save it as id_rsa
7.Rename the public key 'id_rsa.pub' to 'authorized_keys'
mv id_rsa.pub authorized_keys
8.Copy and save the private key in a text file on your local machine, we'll need it later
cat id_rsa
9.Now update the system
yum update
10.Turn off iptables
chkconfig iptables off
service iptables stop
11.Turn on ntpd
chkconfig ntpd on
service ntpd start

Cloning The Servers/Linodes

1.Shutdown the server from your linode control panel
2.Create another linode but do not deploy any distribution yet(lets call it 'linode2')
3.Go back to your old linode(linode1) and click on clone
4.Select the configuration as well as the swap and disk
5.And select the new linode and clone it
6.Do it 5 more times and wait until the cloning gets completed
7.After that, boot all the servers.
so now we have a total of 7 servers/linodes
linode1(main controller node)
linode2(slave node)
linode3(slave node)
linode4(slave node)
linode5(slave node)
linode6(slave node)
linode7(slave node)

Installing Ambari Server

Now, once again SSH into linode1(main controller node)
Now we need to add the Ambari repo, install Ambari & set it up
1.Downlod the repo and set it up
wget http://public-repo-1.hortonworks.com/ambari/centos6/1.x/updates/1.4.3.38/ambari.repo
cp ambari.repo /etc/yum.repos.d
yum repolist
2.Install Ambari
yum install ambari-server
3.Setup Ambari(just accept the default values and continue)
ambari-server setup
4.Start AMBARI server
ambari-server start

Installing The Hadoop Cluster

1.Now open any web browser and open this-
http://$IP-of-main-controller-node$:8080
eg; http://106.185.49.236:8080
2.Login using the default credentials
username- admin
password- admin
3.Give any name to your cluster.
4.Select the hadoop stack version

5.From your linode control panel, copy the public hostnames of all the 6 slave servers(linode2,linode3,linode4,linode5,linode6,linode7).

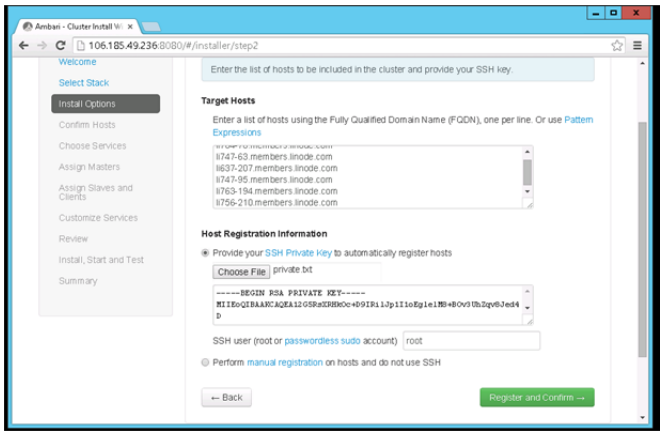
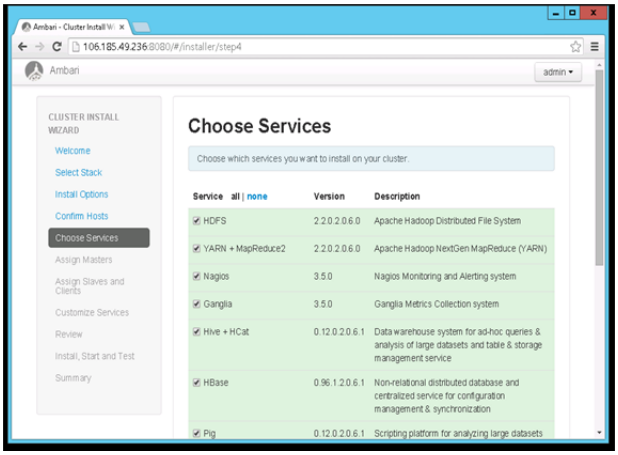


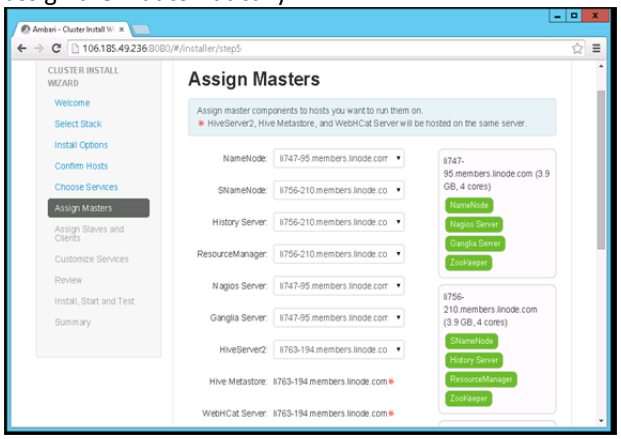
Fig (i). Linode control panel

6.Paste all the hostnames here and then provide the private key that we saved earlier on our local machine
7.Wait until the installation gets completed



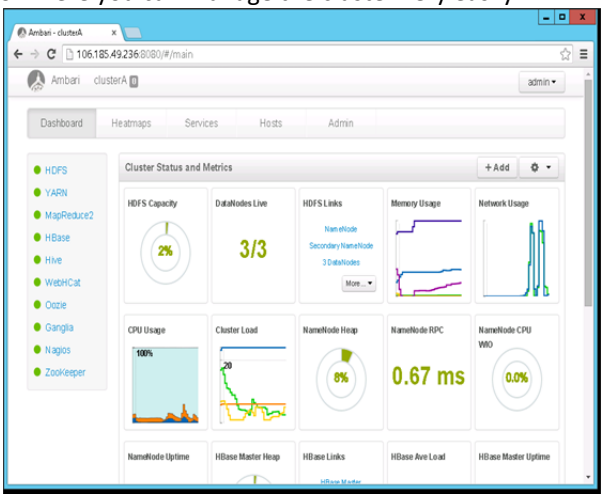
Fig(ii). Installation process

8.Choose services, you can exclude any service(s) that you don't want in your cluster.
9.Assign master servers, you can leave it as default to let it assign them automatically



Fig(iii). Assingning masters and slaves

10.Assign slaves and clients, you can leave it as default too.
11.Setup database password for hive.
12.Setup passwords for oozie and nagios too.
13Confirm everything then hit deploy.
14.Wait for the installation to complete.
15.Now you will be redirected to the ambari dashboard, from here you can manage the cluster very easily.



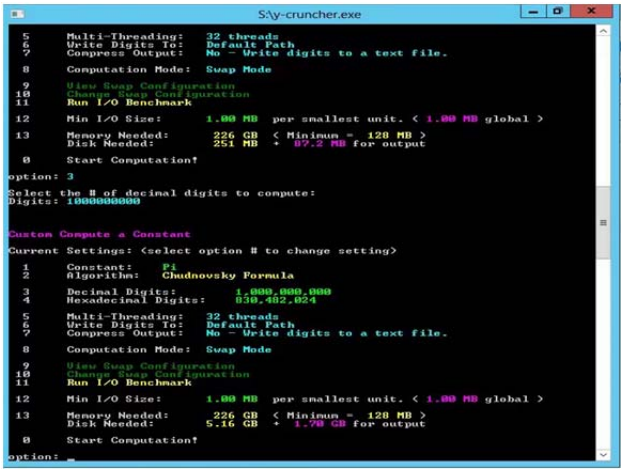
Fig(iv). Ambari Dashboard

Performing Operation

After setting up our hadoop cluster, we can now use it for performing complex calculations.
Fast Fourier Transform (FFT) – the basic rountine behind many algorithms.

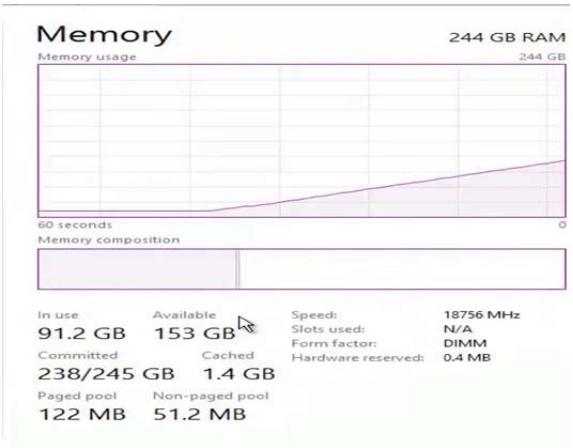
Parallel-FFT Steps

Algorithm
Step 1: I inner DFTs with J-point,
c a(i) = dft(a(i));
Step 2: componentwise shifting, zJI+i def = ζij c a(i)j;
Step 3: transposition, z[j] def = (zJI+(I−1),zJI+(I−2),...,zJI);
Step 4: J outer DFTs with I-point,c z[j] def= dft(z[j]).
After executing the above algorithm it executes very fast and takes very less memory usage also.

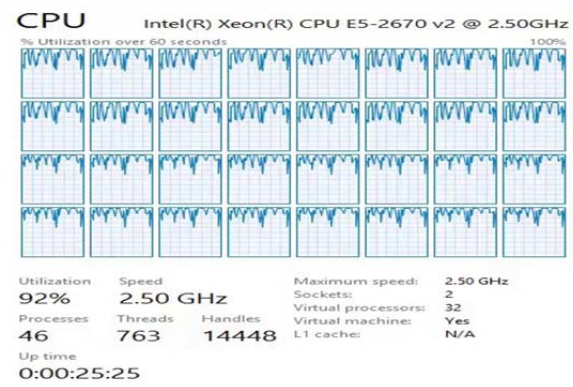


Fig(v). Executing the algorithm

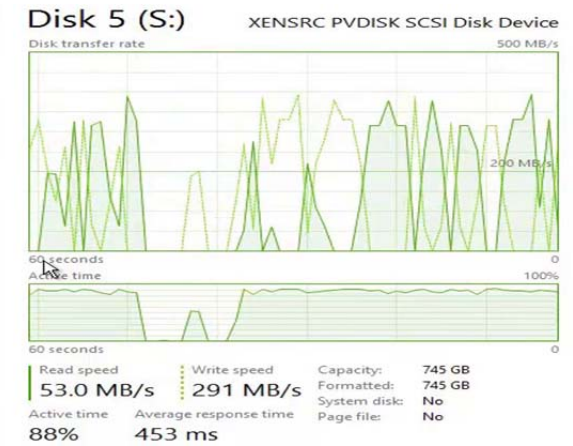
Evaluation



Fig(vi). Memory usage



Fig(vii). CPU UtiliZation



Fig(viii). FFT formula processed result

V. CONCLUSION

In this paper, we have a tendency to mentioned the problems with the Hadoop task assignment theme once Hadoop running within the clouds. we have a tendency to conferred associate improved theme victimization Apache Ambari supported associate optimum formula for a connected planning downside. we have a tendency to additional conducted in depth simulation to judge the performance of Parallel FFT compared with the Hadoop theme. The simulation results confirmed that FFT may considerably outmatch the Hadoop theme with relevance each the completion time of map section and also the quantity of remote process used. In future research, we plan to continue to address the map task assignment drawback based on other related connecting models. Also, we have only focused on the task assignment scheme aspect of Hadoop in this paper. We have planned to address the job scheduling aspectin hadoop, more specifically how to shorten the overall map reduce the phase completion time of the multiple MapReduce jobs, which have different input data block sizes for performing operations and executing big data file.

REFERENCES

[1]. Gantz JF, Chute C, Manfrediz A, Minton S, Reinsel D, Schlichting W, Toncheva A (2008) The Diverse and Exploding Digital Universe: An updated forecast of worldwide information growth through 2011.
[2]. Apache Hadoop(http://hadoop.apache.org) - Apache Ambari(http://ambari.apache.org).
[3]. Ambari Wiki(https://cwiki.apache.org/confluence/display/AMBARI/Ambari.
[4]. Cloning Linodes(https://www.linode.com/docs/migrate-to-linode/disk-images/disk-images-and- configuration-profiles/).
[5]. Official Apache Hadoop Website - http://hadoop.apache.org *webcite*.
[6]. Amazon Elastic Compute Cloud (EC2) - http://aws.amazon.com/ec2/ *webcite*.
[7]. White T (2012) Hadoop: The Definitive Guide. Sebastopol, CA, USA: O'Reilly Media.
[8]. Vijayalakshmi V, Akila A, Nagadivya S (2012) The Survey on MapReduce. Int J Eng Sci Technol
[9]. ElasticHosts, http://www.elastichosts.com/, 2010
[10]. Hadoop wiki. Available: http://wiki.apache.org/hadoop/WordCount.