Volume 4, Issue 5, May 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Combining Hadoop on Demand with High Performance Computing Resources

Swati Swami*, Gunjan Gupta
CSE & Kurukshetra University
India

Abstract— ***Hadoop on demand is a setup to create virtual hadoop clusters over large physical clusters. It allocates nodes using torque resource manager and starts hadoop map/reduce and HDFS daemons. HPC resource management systems and hadoop both have different job submission system,so it is quite difficult to combine hadoop with HPC resources. We describe a framework so that users can develop their hadoop codes on hpc resources. we are configuring processor limit to keep our hadoop clusters running strong and performing optimum***

Keywords— mapreduce,hadoop,high performance computing,HDFS

I. INTRODUCTION

. Apache Hadoop is an open-source software framework for large-scale processing and storage of data sets on clusters of commodity hardware. It provides a distributed file system and a mapreduce paradigm[6] for the analysis and transformation of very large data sets. A very important feature of Hadoop is the partitioning of data and computation across many number of hosts, and execution of application computations in parallel close to their data. The Apache Hadoop framework is composed of the following modules:

- Hadoop Common- contains libraries and utilities needed by other Hadoop modules
- Hadoop Distributed File System (HDFS) – a distributed file-system which is for storing data on commodity machines and it provides high aggregate bandwidth across the cluster
- Hadoop YARN – a resource-management platform responsible for managing compute resources in clusters and using them for scheduling of users' applications.
- Hadoop MapReduce – a programming model for large scale data processing.

A small Hadoop cluster is composed of a single master and many worker nodes. Further the master node consists of a NameNode, DataNode, JobTracker and TaskTracker,. A worker node acts as both a DataNode and TaskTracker, as it is possible to have compute-only worker nodes and data-only worker nodes. As shown in figure 1 the HDFS is managed through a dedicated NameNode server to host the file system index. HDFS also has a secondary NameNode that can generate copies of the namenode's memory structures to prevent file-system corruption and to reduce loss of data. In same way, standalone JobTracker server can manage job scheduling. HDFS separately stores file system metadata and application data. Like PVFS, Lustre and GFS (other distributed systems) , HDFS stores metadata on a dedicated server, called the NameNode and it stores application data on other servers called DataNodes. Servers are fully connected and communicate through TCP-based protocols.

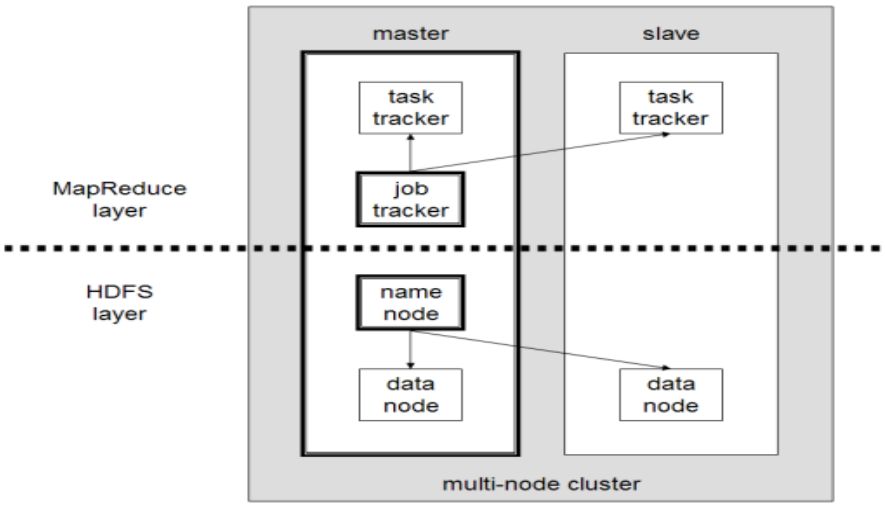

Figure 1: A Multinode Hadoop Cluster

© 2014, IJARCSSE All Rights Reserved

II. HADOOP ON DEMAND AND HPC RESOURCES

Hadoop On Demand (HOD)[2], is a setup to create virtual Hadoop clusters over a large physical cluster. It allocates the nodes using Torque resource manager and starts Hadoop Map/Reduce and HDFS daemons on the allocated nodes. It generates the appropriate configuration files like hadoop-site.xml for the Hadoop daemons and client automatically. It is also capable to distribute Hadoop to the nodes in the virtual cluster that it allocates. Simply HOD makes it easy for administrators and users to quickly setup and use Hadoop. It is also a useful tool for Hadoop developers and testers who need to share a physical cluster for testing their own Hadoop versions.

TORQUE [5] (also known by its historical name Portable Batch System - PBS), or the Sun Grid Engine[4] a distributed resource management system is supported for job submissions by traditional high performance computing (HPC) resources , such as those available on the TeraGrid [9]. System administrators put these systems on these resources to enable tracking, management of batched, submission, non-interactive jobs, so that it maximizes the overall utilization of the system, and that it enables sharing of the resources among many users. Users typically do not have a choice of batch systems to use on a particular resource - they simply use the interfaces provided by the batch systems that are made available on those resources.

MapReduce tools such as Apache Hadoop are being used for the growing number of codes in scientific domains such as Bioinformatics [8] and Geosciences. Users find it difficult to run their Hadoop codes on traditional HPC systems that they have access to. Hadoop manages its own job and provides its own scheduling and task submissions and tracking so it is hard for Hadoop to co-exist with existing HPC resource management systems. This is because both systems are designed to have complete control over the resources that they manage, it is a challenge to make Hadoop to co-exist with traditional batch systems such that users may run Hadoop jobs on these resources. In addition, Hadoop uses a shared-nothing architecture [10] , whereas traditional HPC resources typically use a shared disk architecture, using high performance parallel file systems. Because to these challenges, HPC users have been left with no option other than to obtain a physical cluster and manage and maintain their own Hadoop instances. Hadoop jobs are also being run on new resources such as Amazon's Elastic MapReduce [1] by some users. we present a simple framework, myhadoop for Hadoop on-demand on traditional HPC resources, using standard batch processing systems such as TORQUE or SGE. With the help of myHadoop, users do not need dedicated clusters to run their jobs - instead, they can configure Hadoop clusters on-demand by requesting resources via TORQUE or SGE, and then configuring the Hadoop environment based on the set of resources provided. It is an open source, and available for download via SourceForge [3].

III. ARCHITECTURE

There is a difference in the system architecture for shared-nothing frameworks such as Apache Hadoop from that of the traditional shared HPC resources. Most HPC resources consists of a set of powerful compute nodes with minimal local storage. The compute nodes are themselves connected to each other using a high-speed network interconnect. On the other hand,Shared-nothing frameworks, are designed for use in large clusters of commodity PCs connected together with switched commodity networking hardware, with storage directly attached to the individual machines. Figure 2 describes the workflow of myhadoop Thus, every machine is both a data and a compute node. The main challenges in enabling Hadoop jobs to run on shared HPC resources are:

(i) To make the resource management functions of Hadoop to co-exist with the native batch resource managers, and

(ii) Implementation of shared-nothing infrastructure on top of the typical shared-disk systems such as traditional HPC architectures

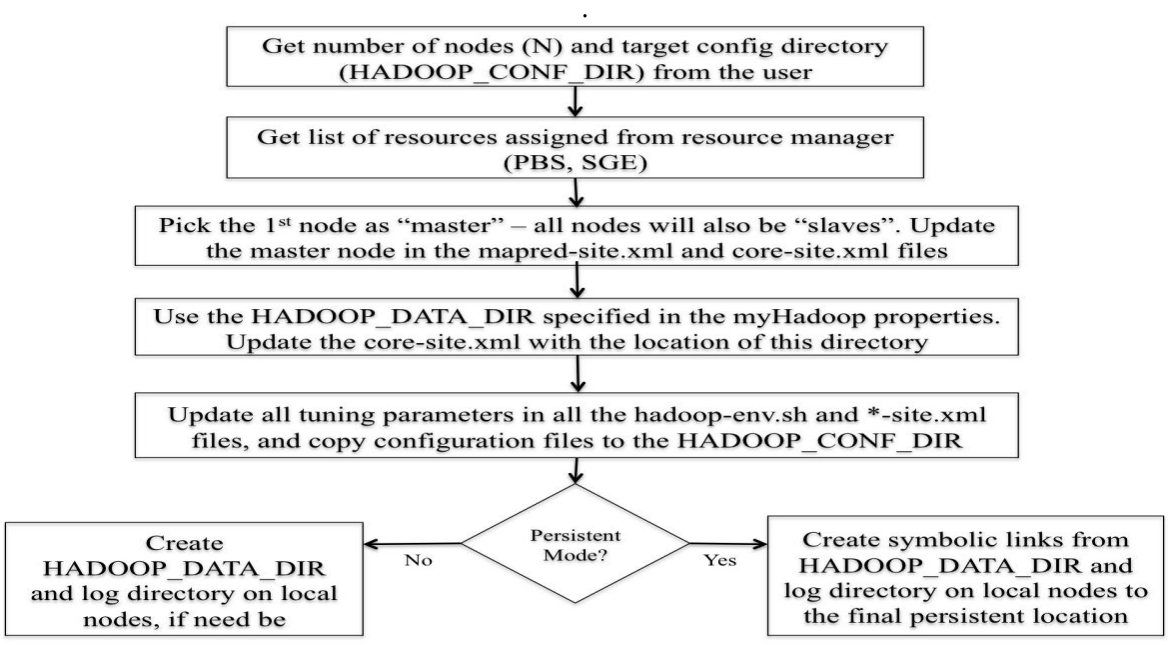

Fig 2: myhadoop configuration workflow

© 2014, IJARCSSE All Rights Reserved

IV. RELATED STUDY

Jeffrey dean et.al [11] introduces about MapReduce which is a programming model for processing and generating large datasets. Users use a map and a reduce function to specify the computation, and the runtime system parallelizes the computation across large-scale clusters of machines, schedules inter-machine communication, and handles machine failures to make efficient usage of the disks and networks. Most of the programmers find the system easy to use:over the past four years more than ten thousand different MapReduce programs have been implemented internally at Google, and almost one hundred thousand MapReduce jobs are executed on Google's clusters every day and processes a total of more than twenty petabytes of data every day.

Shadi Ibrahim et.al [12] proposes about the poor performance of existing MapReduce framework in virtualized environment suffers , because of the heavy overhead of I/O virtualization, and difficulty in management for storage and computation. To solve these problems, they propose a novel MapReduce framework,ClOUDLET on virtual machines . The main aim of designing it is to reduce the overhead of VM while benefiting of the other features of VM (i.e. management and reliability issues).

Konstantin Shvachko et.al [13] said that The Hadoop Distributed File System (HDFS) is used to store very large data sets reliably, and to flow those data sets at high bandwidth to user applications. Thousands of servers in large clusters both host directly attached storage and execute user application tasks. With the distribution of storage and computation across multiple servers, the resource can grow with demand while remaining economical at every size. It describes the architecture of HDFS and report on experience using HDFS to manage 25 petabytes of enterprise data at Yahoo!.

Thomas sandholam et.al [14] introduces the Dynamic Priority (DP) parallel task scheduler. It allows users to control their allocated capacity with the adjustment of their spending over time. The mechanism is very simple which allows the scheduler to make more efficient decisions about which jobs and users to prioritize and provides users the required tool to optimize and customize their allocations to fit the importance and requirements of their jobs. In addition, it gives users the incentive to scale back their jobs when demand is high, because the running cost of slot is then also more expensive.

Chen Zhang et.al [15] said that with the growing popularity of MapReduce in data processing applications, the demand for Hadoop clusters also grows. However, Hadoop is not compatible with existing cluster batch job queuing systems. It requires a dedicated cluster under its own control. It also lacks support for user access control, batch job processing facilities, accounting, fine-grain performance and monitoring comparable to existing cluster job queuing systems that makes dedicated Hadoop clusters less amenable for administrators and users alike with hybrid computing needs which involves both MapReduce and legacy applications. To get a properly suited and sized Hadoop cluster has not been easy in organizations with existing clusters. To solve this problem it presents CloudBATCH, a prototype solution making Hadoop to function as a traditional batch job queuing system with effective functionality for cluster resource management. With CloudBATCH, it becomes feasible to complete shift Hadoop for managing whole cluster to cater for hybrid computing needs.

Although there has been a lot of user interest in recent times for running Hadoop jobs on traditional HPC resources while there are very few efforts that have been sufficiently documented, and publicly available. Blog article [7],provides the description about a similar concept myhadoop in which the author describes the process of getting Hadoop to run as a batch job using PBS. myHadoop is freely available for download via SourceForge. It is a more general and configurable framework, to provide support for other schedulers as well. But there has not been a performance evaluation of this effort that has been published. The Apache Hadoop on Demand (HOD - [2]) is a setup to create virtual Hadoop clusters over a large physical cluster. It allocates the nodes using Torque resource manager and starts Hadoop Map/Reduce and HDFS daemons on the allocated nodes. It generates the appropriate configuration files like hadoop-site.xml for the Hadoop daemons and client automatically. It is also capable to distribute Hadoop to the nodes in the virtual cluster that it allocates. HOD differs from myHadoop in the following ways.
First thing is HOD requires the use of an external HDFS that is statically configured which means that the MapReduce jobs can't take advantage of any data locality, since the data is not collocated with the map and reduce tasks. The non-persistent mode of myHadoop gives ability for data locality, with the HDFS being dynamically configured to use the same number of nodes, with the exception that the data does have to be prepared in and out before and after the execution. The implementation of HDFS in the persistent mode of myHadoop is same with the concept of the statically configured external HDFS used by HOD. However, in the persistent mode, the HDFS in myHadoop may be thought of as pseudo-local because there is a 1-1 mapping between every node that runs the MapReduce daemons and its corresponding external HDFS data directory. Second difference is that HOD is based on the TORQUE (PBS) resource manager - while myHadoop in addition to TORQUE can also use different resource manager. Finally, HOD has some documented problems [7] in setting up many concurrent Hadoop instances. While myHadoop is designed to support concurrent own instances of Hadoop for multiple users. Finally, Amazon's Elastic MapReduce [1] makes the allocation of Hadoop clusters on the fly with the help of a Web service API on Amazon's cloud resources. It is similar to myHadoop in the sense that Hadoop clusters are allocated on-demand. However, there is a difference that it uses Amazon's cloud resources, instead of using traditional HPC resources via batch scheduling systems. Both systems

© 2014, IJARCSSE All Rights Reserved

enable users to run Hadoop jobs with minimal overhead and configuration however, it is more efficient for end-users to run their Hadoop jobs on the same HPC resources on which their data resides, as it is opposed to staging all the data over to Amazon's cloud before running their Hadoop jobs, and also pay for their usage as they go for both data transfers and computation.

V. CONCLUSION

myHadoop is an open source tool for running hadoop jobs on hpc resources without the need of root level access. It uses the standard batch scheduling systems such as TORQUE (also known by its historical name Portable Batch System - PBS), or the Sun Grid Engine a distributed resource management system which is supported for job submissions by traditional high performance computing (HPC) resources. Without any interference , it allows many users to execute their hadoop jobs on shared resources. It coexists with traditional batch systems and allows 'persistent' and 'non-persistent' modes to save HDFS state across runs. Configuring processor limits on hadoop clusters make the performance optimum.

REFERENCES

[1] Amazon Elastic MapReduce. 2011. Available : http://aws.amazon.com/elasticmapreduce/.
[2] Apache Hadoop on Demand (HOD). 2011. Available: http://hadoop.apache.org/common/docs/ r0.20.2/hod user guide.html.
[3] myHadoop on SourceForge. 2011. Available: http://sourceforge.net/projects/myhadoop/.
[4] The Sun Grid Engine (SGE), 2011 Available: http://wikis.sun.com/display/GridEngine/Home.
[5] TORQUE Resource Manager, 2011 http://www.clusterresources.com/products/torqueresource-manager.php.
[6] Apache Software Foundation. Hadoop MapReduc .2011.available : http://hadoop.apache.org/mapreduce.
[7] J. Ekanayake. Hadoop as a Batch Job using PBS. 2008. http://jaliyacgl.blogspot.com/2008/08/hadoopas-batch-job-using-pbs.html.
[8] T. Gunarathne, T. Wu, J. Qiu, and G. Fox. *"Cloud computing paradigms for pleasingly parallel biomedical applications* "In 19th ACM Intl Symp on High Perf Dist Comp, pages 460–469. ACM, 2010.
[9] C. Catlett. "*The philosophy of TeraGrid: building an open, extensible, distributed TeraScale facility*" In 2nd IEEE/ACM Intl Symp on Clust Comp & the Grid, 2005.
[10] M. Stonebraker."*The case for shared nothing. Database Engineering Bulletin*" 9(1):4–9, 1986.
[11] J. Dean and S. Ghemawat. "*Mapreduce: Simplified Data Processing on Large Clusters*." *Commun. ACM*, 51:107–113, 2008.
[12] S.Ibrahim, H.Jin, B.Cheng, Haijun Cao, S.Wu "*Cloudlet: Towards MapReduce Implementation on Virtual Machines*"ieee 2009.
[13] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler "*The Hadoop Distributed File System*" ieee 2010.
[14] Thomas sandholm,Kevin lai "*Dynamic Proportional Share Scheduling in Hadoop*" springer 2010
[15] Chen Zhang,Hans De Stark " *CloudBATCH: A Batch Job Queuing System on Clouds with Hadoop and HBase*" ieee 2010

© 2014, IJARCSSE All Rights Reserved