Project-8A

Integration of Virtualization with Hadoop Tools

Name	Roll Number	Contact	e-mail
Aparna Raj K	MT2011019	9902422888	aparnaraj.k@iiitb.org
Kamaldeep Kaur	MT2011061	7259197082	Kamaldeep.Kaur@iiitb.org
Prasad Pai	MT2011106	9663269499	Prasad.Pai@iiitb.org
Uddipan Dutta	MT2011162	8880430536	Uddipan.Dutta@iiitb.org
V Venkat Sandeep	MT2011167	8095088171	Sandeep.VV@iiitb.org

Team Leader : Aparna Raj K

Contents

1 *Introduction*

With significant advances in information technology, the amount of data being processed in the computing world, is increasing day by day. As a result, the need of a new kind of infrastructure and systems that can handle large amount of data is on high demand. Google designed a programming model for cloud computing, called MapReduce[1], as a possible solution for managing large amounts of data sets. Later Hadoop, from the Apache project[1], came up with a more popular open source implementation for MapReduce. Hadoop MapReduce is a programming model and software framework for writing applications that rapidly process vast amounts of data in parallel on large clusters of compute nodes. As the amount of data processed is huge, a lot of resources has to be quickly coordinated and allocated properly.

Though it is true that Hadoop gives good results on physical machines, service providers find it difficult to manage incoming requests dynamically. Virtualization can solve the management problems for Hadoop and this project aims at enhancing the hadoop environment with dynamic resource allocation capabilities.

2 *Project Description*

A brief description of an insight to the project, just to understand and appreciate the importance and magnitude of this project.

2.1 Objective

Integration of Hadoop tools with virtualization using hadoop MapReduce framework.

2.2 Motivation

Though Hadoop works on physical machines and provides good result, there are some difficulties for service providers in terms of management. This is because there are no ways to manage requests dynamically. Virtualization can solve those management problems for Hadoop as it does for other software solutions, giving dynamic resource allocation capabilities.[2]

As each individual machine has more and more cores/cpus, it makes sense to partition each machine into multiple virtual machines. That gives us a number of benefits :

i **Scheduling:** Taking advantage of unused capacity in existing virtual infrastructures during periods of low usage (for example, overnight) to run batch jobs.

ii **Resource utilization:** Co-locating Hadoop VMs and other kinds of VMs on the same hosts. This often allows better overall utilization by consolidating applications that use different kinds of resources.

iii **Storage models:** Although Hadoop was developed with local storage in mind, it can just as easily use shared storage for all data or a hybrid model in which temporary data is kept on local disk and HDFS is hosted on a SAN. With either of these configurations, the unused shared storage capacity and bandwidth within the virtual infrastructure can be given to Hadoop jobs.

iv **Datacenter efficiency:** Virtualizing Hadoop can increase datacenter efficiency by increasing the types of workloads that can be run on a virtualized infrastructure.

v **Deployment:** Virtualization tools ranging from simple cloning to sophisticated products like VMware vCloud Director can speed up the deployment of Hadoop nodes.

vi **Performance:** Virtualization enables the flexible configuration of hardware resources.

2.3 Description

Hadoop MapReduce computes large amounts of data in a reasonable time, due to its high scalability. This is achieved by dividing the the job in small tasks that can be split through a large collection of computers.

As part of the MapReduce implementation, Hadoop has an internal scheduler for managing these incoming requests. The capabilities of the scheduler are to be enhanced by providing virtual infrastructure to Hadoop Environment. To achieve this, EMOTIVE cloud[3], (Elastic Management of Tasks in Virtualized Environments) is to be used. It is a virtualized environment manager. By coupling the dynamic resource virtualization of EMOTIVE with Hadoop Mapreduce, resource allocation and manipulation of Hadoop environments is to be enhanced.

3 *Related Works*

A similar effort for virtualizing the Hadoop environment can be seen in **Support for managing dynamically Hadoop clusters**[4], a final master project in Barcelona supercomputing centre.This project focuses on the virtualization of Hadoop environments and the design of a piece of software for automatizing the configuration of the shared resources for Hadoop environments. In addition, the project uses a modified internal Hadoop scheduler that adapts the available resources in the cluster for running jobs according to time restrictions.

4 *Gap Analysis*

i. Though Hadoop Mapreduce works well on Physical systems, dynamic provisioning of resources is a problem. As the computations involved are large, a lot of resources should be quickly coordinated and properly allocated. If there are a large number of requests at a particular time, and if the hadoop environment requires more resources to process these requests, adding and configuring a new physical machine would take a considerable amount of time.

ii. For a large job, adding more tasktrackers to the cluster will help in faster computations, but there is no flexibility in adding or removing nodes from a hadoop cluster setup entirely on physical machines.

iii. The Jobtracker, which acts as the master node in the cluster, is a single point of failure.

The project aims at enhancing the hadoop environment with more virtualized infrastructure so that the hadoop cluster can be dynamically configured. The enhanced hadoop internal scheduler will evaluate the current number of resources and required number of resources to finish the jobs in the required time. It will determine whether or not more virtual machines are required to finish the jobs. On request, the EMOTIVE Cloud VM manager will create a new VM. This will be created form a predefined machine image with the required hadoop configuration and software . As it is already configured according to the requirements, it can be added to the hadoop cluster easily. This will solve the problem of dynamic resource provisioning on Hadoop. With multiple virtual machines running, the overall utilization of the clusters will be improved as multiple tasktrackers are scheduled on the same machine.

5 *Architecture*

The basic components involved are :

i. MapReduce framework

ii. HDFS (Hadoop Distributed File System) which provides the data storage required by the MapReduce framework

iii. EMOTIVE cloud framework which provides the necessary virtualization infrastructure

5.1 MapReduce framework

MapReduce is a programming model that processes large amount of data, such as web logs, crawled documents, etc. Though the computations involved in MapReduce are more or less straightforward, the input will be very large. So it needs to be distributed across a large amount of machines in order to finish the tasks within a reasonable amount of time.

The nodes in the cluster are categorized as :

i. Master Node or JobTracker

ii. Slave Node or TaskTracker

The Master Node or the JobTracker, receives input and splits the job into smaller tasks and assign each split to Slave Nodes or TaskTrackers for computation.

MapReduce consists of two phases :

i. Map phase

ii. Reduce phase

Each phase has key-value pairs as input and output, the types of which may be chosen by the programmer. The programmers specify an appropriate map function and reduce function in these phases. First the file is partitioned into parts called splits that will be distributed to process across the cluster nodes.

Map phase

i. The input to the map function is raw data.

ii. The map function is just a data preparation phase, setting up the data in such a way that the reduce function can do its work on it.

iii. For a given key-value pair as input, the map function filters raw data.

iv. The map function is also a good place to drop bad records, here we filter out records that are missing, suspect, or erroneous.

v. Once all the map tasks are finished, it means that the whole data to process was completely read, and reordered into key-value pairs.

vi. The output from the map function is stored temporarily on local disk which is then sent to MapReduce framework.

vii. The output from the map function is processed by the MapReduce framework before being sent to the reduce function.

viii. MapReduce framework sorts and groups the key-value pairs by key.

Reduce phase

i. In Reduce step, the nodes performing reduce task, collects all the results from the map function running in different nodes and, if necessary process the results to obtain the desired output.

5.2 HDFS - The data storage for MapReduce

Hadoop includes its own distributed file system called HDFS, that stands for Hadoop Distributed File System. Instead of using another independent dedicated storage, MapReduce uses the same machines used for the computation for data storage also.

HDFS also follows the client-server architecture just like MapReduce. The Master Node is called NameNode and the Slave Nodes are termed as Datanodes. The NameNode stores all the meta-information of the file system and coordinates it. The DataNodes are spread among the cluster and are responsible of storing the data.

5.3 EMOTIVE CLOUD - The Virtualization Platform

EMOTIVE (Elastic Management of Tasks in Virtualized Environments) is a middleware used to provide virtual machines in a very easy way, where the user can run common applications, such as Tomcat, with a user-friendly GUI. The EMOTIVE Cloud framework consists of three layers :

i. Virtualization Fabrics

ii. Virtual Machine Manager

iii. Virtual Machine Scheduler

The EMOTIVE Cloud VM Manager, is capable of creating or deleting virtual machines.

5.4 Schematic Diagram

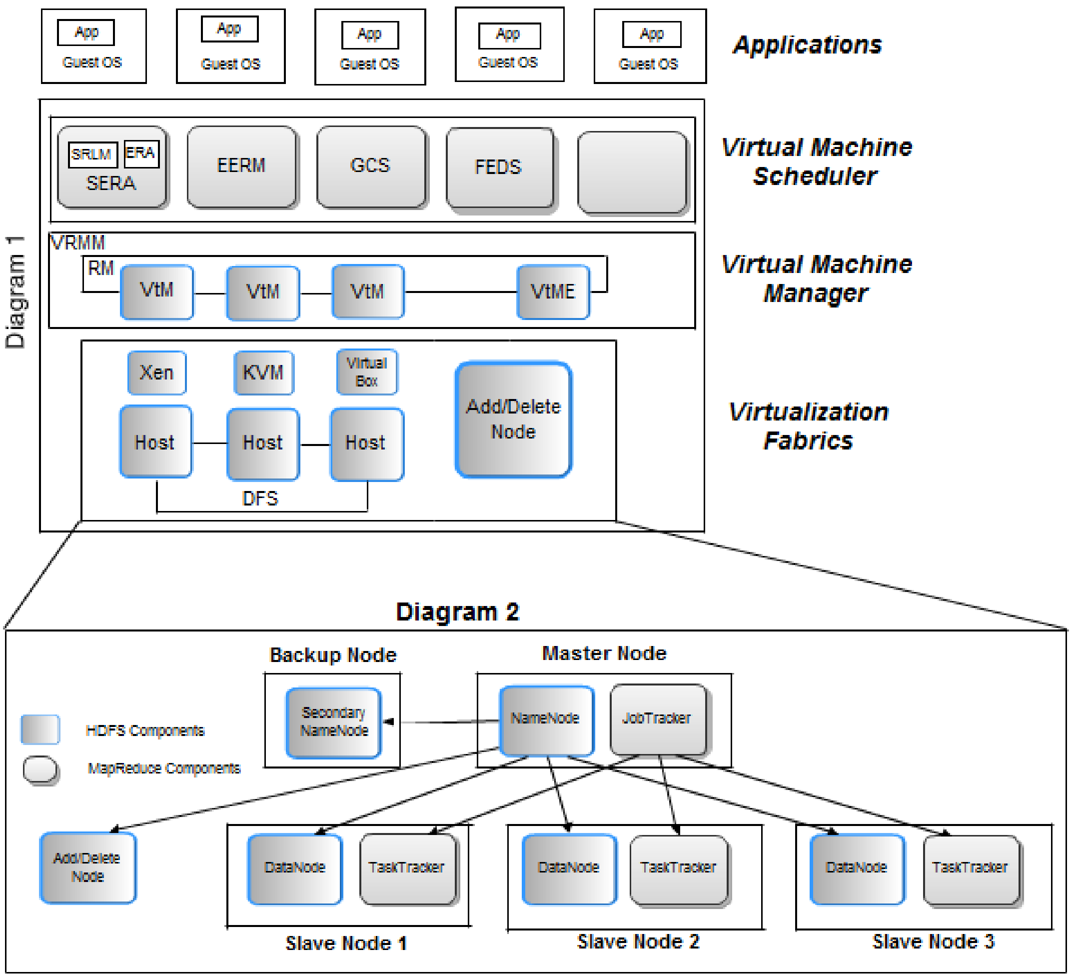

Figure 1: Basic Architecture using [5]

The figure above illustrates the virtualized Hadoop architecture. The basic Hadoop MapReduce framework is coupled with EMOTIVE Cloud framework.

Diagram 1

The topmost layer is the Virtual Machine Scheduler. This Scheduler decides where a VM will be executed and manages its location during the execution. The Scheduler tries to consolidate the VMs in the providers physical resources, so as to optimize their use.

The layer below it, is the Virtual Machine Manager layer. It is implemented by means of the Virtualized Resource Management and Monitoring (VRMM), which includes several subcomponents. Each physical host has a Virtualization Manager (VtM) that is in charge of creating and maintaining the whole virtual machine life cycle. In this layer VMs are created whenever there is some application requirement for both hardware and software. Once the VMs are created, the users can work directly with them using a remote connection like ssh, or they can also use the EMOTIVE API[6] to execute tasks in them.

The bottom layer is the Virtualization Fabrics layer comprising of the physical resources on which the VMs will run. This layer consists of all the virtualized resources, which will be provided to the upper layers as and when required. The data infrastructure is based in a Distributed File System (DFS) that makes the data available through all the machines in the network. Here it is the HDFS. The HDFS caches virtual machines in each physical machine to make virtual machine creation process faster and makes virtualization capabilities such as migration and checkpoint possible.

Diagram 2

A detailed view of the Hadoop framework is shown in Diagram 2. The Hadoop MapReduce framework is setup with some of the nodes as the VMs created by the EMOTIVE. The Hadoop Scheduler communicates with the Scheduler in the EMOTIVE, and it sends a request to add or delete a node. The add/delete node in the Diagram 2, represents the VM provided by the EMOTIVE, which is added to (or removed from) the Hadoop framework.

By virtualizing the Hadoop environment, we aim at achieving :

i. **Dynamic Resource Allocation**

   Communication is to be setup between the Hadoop internal Scheduler and the EMOTIVE Cloud Scheduler. On receiving a request for a new VM, the EMOTIVE Cloud VM Manager, creates a new VM. EMOTIVE VM Manager does this from a predefined virtual machine image with a distribution of Hadoop already installed and configured. Virtual machine images are stored in HDFS in each physical machine to make virtual machine creation process faster. Now this new node can be included as a TaskTracker in the Hadoop cluster. Hadoop Scheduler can request deletion of a node also, if required.

ii. **Improving Fault Tolerance**

The JobTracker is a single point of failure in the cluster. Adding the virtual machine images of the JobTracker node, will make the retrieval faster in case of a failure. These Master Nodes can be restored to the previous state by creating a new VM, from the predefined virtual image of their configurations. This will improve fault tolerance of the system.

6 *Implementation Plan*

In order to virtualize the Hadoop environment, the steps to be followed are :

Phase 1 : Setting up of HDFS

First, the basic HDFS framework is setup.

i. Download the latest release of Hadoop from one of the Apache Download Mirrors.

ii. Setup a Hadoop cluster and learn the working of MapReduce.

Phase 2 : Installing Xen and EMOTIVE Cloud

These are the components that gives the necessary support for building the infrastructure required for virtualizating the Hadoop environment.

i. Download latest Open Source Xen binary file for appropriate platform and its source from [7]. Configure the system with all the required dependencies.

ii. Install Xen monitor to manage Xen.

iii. After installing the necessary dependencies for EMOTIVE Cloud, setup the VtM[3], the Virtualization Manager of EMOTIVE Cloud.

Phase 3 : Virtualizing Hadoop using EMOTIVE

Here we actually couple the virtual infrastructure created, with the Hadoop environment. This gives the enhanced Hadoop environment with dynamic resource allocation capabilities.

i. Once the Virtualization Manager (VtM) is setup, create the basic files required to create the first Virtual Machine.

ii. Create a Scheduler that will run on top of EMOTIVE Cloud, which will interact with the Hadoop built-in JobTracker Scheduler. Ensure communication is possible between the two Schedulers.

iii. Create a Virtual Machine image of a node in the Hadoop cluster. Create a VM using the EMOTIVE from this predefined image.

iv. Once the new VM is created, add it to the Hadoop cluster and assign tasks to it.

v. Make sure whether VM can be removed from the Hadoop cluster.

vi. Ensure that dynamic addition and deletion of nodes into the Hadoop cluster is done on request from the Hadoop Schedule.

7 *Milestones*

Milestone	**Goal**	**Date**
Phase 1	Study Hadoop Architecture. Understand the workflow of Mapreduce.	22-01-2012
Phase 2	Setup svn. Final Draft of Goal Statement.	27-01-2012
Phase 3	SetUp hadoop Environment. Do sample MapReduce function.	03-02-2012
Phase 4	Install Xen. Presentation on project architecture and plans.	07-02-2012
Phase 5	SetUp VtM environment and Emotive Cloud. Learn in detail about scheduler in master node.	11-02-2012
Phase 6	Create virtual image of a node in Hadoop cluster and create a new Virtual Machine using EMOTIVE. Alpha version due.	01-03-2012
Phase 7	Establish communication between Hadoop scheduler and the scheduler in EMOTIVE.	17-03-2012
Phase 8	Implementing dynamic addition and deletion of nodes on request.	28-03-2012
Phase 9	Beta release. Implement rough draft of final document.	03-04-2012
Phase 10	Final submission.	19-04-2012

References

[1] “Hadoop,” The Apache Software Foundation, Dec. 2011. [Online]. Available: http://hadoop.apache.org/

[2] J. Buell, “A Benchmarking Case Study of Virtualized Hadoop Performance on VMware vSphere 5,” Technical White Paper, VMware, Oct. 2011. [Online]. Available: http://www.vmware.com/files/pdf/techpaper/VMW-Hadoop-Performance-vSphere5.pdf

[3] “EMOTIVE Cloud,” Barcelona, Dec. 2009. [Online]. Available: http://www.emotivecloud.net/

[4] D. de Nadal Bou, “Support for managing dynamically Hadoop clusters,” in *Master in Information Technology - MTI*, Sep. 2010, Project Director : Yolanda Becerra. [Online]. Available: http://hdl.handle.net/2099.1/9920

[5] “gliffy,” online Diagram Software. [Online]. Available: http://www.gliffy.com/

[6] “libvirt Virtualization API,” Dec. 2005. [Online]. Available: http://libvirt.org/java.html

[7] “Xen,” University of Cambridge, 2002. [Online]. Available: http://xen.org/

[8] “the Hadooper in me,” Research,Labs & IT stuff, Buenos Aires, Nov. 2010. [Online]. Available: http://hadooper.blogspot.com/

[9] J. J. Hierro, L. M. Vaquero, and J. Caceres, “EMOTIVE CLOUD : The BSC’s IaaS Open Source Solution for Cloud Computing,” in *Open Source Cloud Computing Systems : Practices and Paradigms*. United States of America: Information Science Reference, pp. 44–60.

[10] T. White, *Hadoop : The Definitive Guide*. Sebastapol,CA 95472: O’Reilly Media, June 2009.