The FutureGrid Testbed for Big Data

Gregor von Laszewsi, Geoffrey C. Fox

laszewski@gmail.com

Abstract

In this chapter we will be introducing you to FutureGrid that provides a testbed to conduct research for Cloud, Grid, and High Performance Computing. Although FutureGrid has only a relatively small number of compute cores (about 4500 regu-lar cores and 14000 GPU cores) it provides an ideal playground to test out various frameworks that may be useful for users to consider as part of their big data analy-sis pipelines.

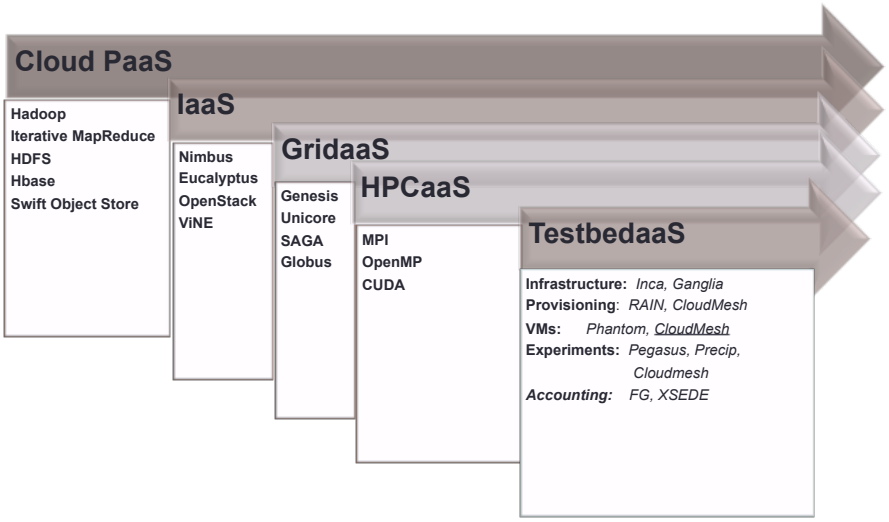
The chapter is structured as follows. First we will provide the reader with an in-troduction to Future Grid. We will list a number of projects that use Futuregrid to conduct data analysis and introduce some of them to the reader. We will tell you about which services and Hardware exists. Next we will analyze which services are preinstalled and are available for big data analysis. As services that users may need for their work we point out how such a testbed can be utilized not only while provisioning virtual machines, but also on bare metal.

We conclude the chapter with our observation cast throught three years of op-erating FutureGrid and provide an outlook for the next steps.

1 Introduction

FutureGrid [2, 1]is a project led by Indiana University and funded by the Na-tional Science Foundation (NSF) to develop a high-performance grid test bed that will allow scientists to collaboratively develop and test innovative approaches to parallel, grid, and cloud computing. FutureGrid will provide the infrastructure to researchers that allows them to perform their own computational experiments us-ing distributed systems. The goal is to make it easier for scientists to conduct such experiments in a transparent manner. FutureGrid users will be able to deploy their own hardware and software con-figurations on a public/private cloud, and run their experiments. They will be able to save their configurations and execute their experiments using the provided tools. The FutureGrid test bed is composed of a high-speed network connecting distributed clusters of high performance computers. FutureGrid employs virtual-ization technology that will allow the test bed to support a wide range of operating systems.

Figure 1: FutureGrid High Level User Services.

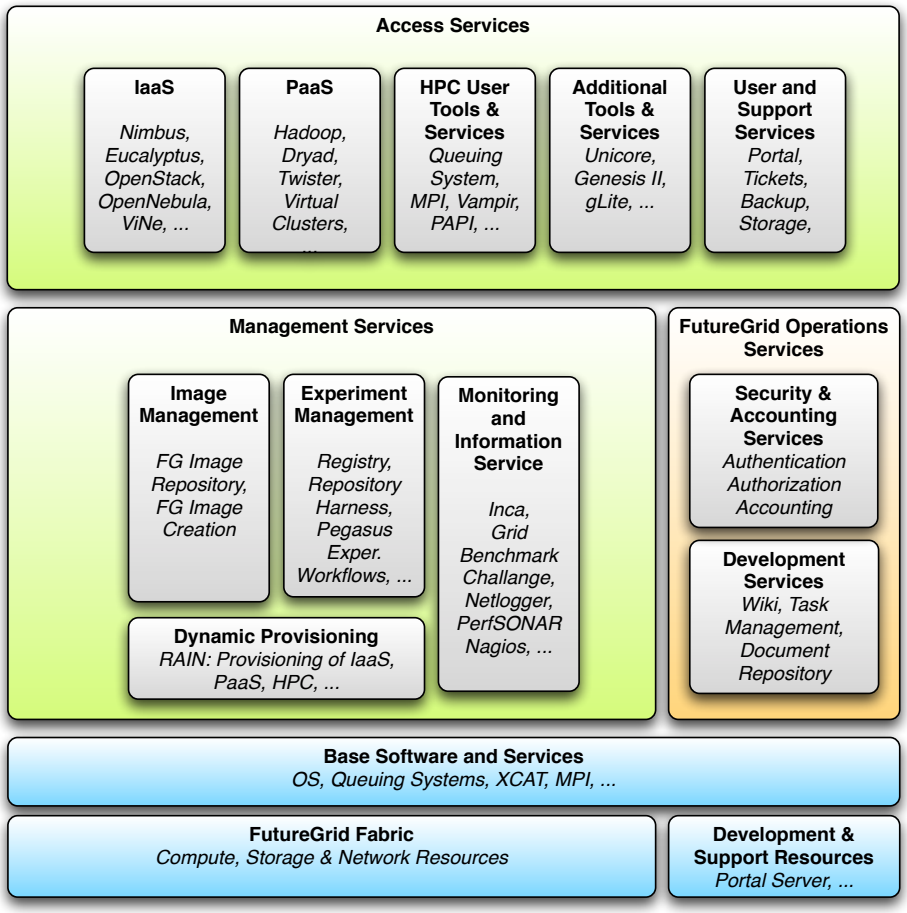


2 Overview of FutureGrid for Big Data

2.1 Service Overview

According to the manual FutureGrid provides a number of different services. These services include: OpenStack which includes a collection of open source components to deliver public and private clouds. These components currently include OpenStack Com-pute) OpenStack Object Storage, and OpenStack Image Service. OpenStack has received considerable momentum due to its openness and the support of compa-nies. Nimbus which is an open-source service package that allows users to run vir-tual machines on FutureGrid hardware. Just as in Openstack users can upload their own virtual machine images or customize existing once. Nimbusnext to Eucalyp-tus is one of the earlier frameworks that make managing virtual machines easier. Eucalyptus is an open-source software platform that implements IaaS-style cloud computing. Eucalyptus provides an Amazon Web Services (AWS) compli-ant EC2-based web service interface for interacting with the Cloud service. Euca-lyptus has been previously the dominant alternative to AWS in academia. How-ever, based on usage patterns in FutureGrid we believe it is replaced by OpenStack. High Performance Computing can be defined as the application of super-computing techniques to solve computational problems that are too large for standard computers or would take too much time. This is one of the more im-portant features that the scientific community needs to achieve their projects. Nat-urally using HPC resouces and services is also useful in the area of Big Data. Sometimes big data needs big

Figure 2: FutureGrid High Level User Services.



machines. Thus, using HPC may be an ovious choice. Map Reduce . TBD Storage on FutureGrid has moderate size storage capability that will satisfy the users demand to compare and test someof the previously outlined services. Information Services gather the information of the different elements that make up FutureGrid to provide accurate and complete knowledge of the computa-tional environment. This information is presented using different web portals.

2.2 Hardware Overview

According to the manual, FutureGrid is build out of a number of clusters of different type and size that are interconected with up to a 10GB Ethernet among its sites. The sites include Indiana University, University of Chicago, San Diego Su-percomputing Center, Texas Advanced Computing Center, and University

Figure 3: FutureGrid High Level User Services.

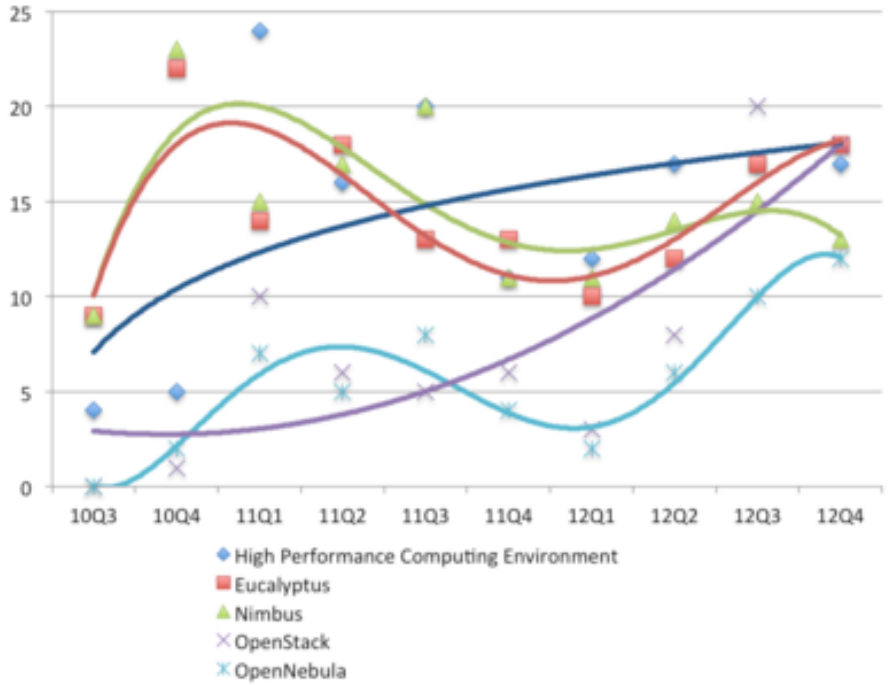


Figure 4: FutureGrid High Level User Services.

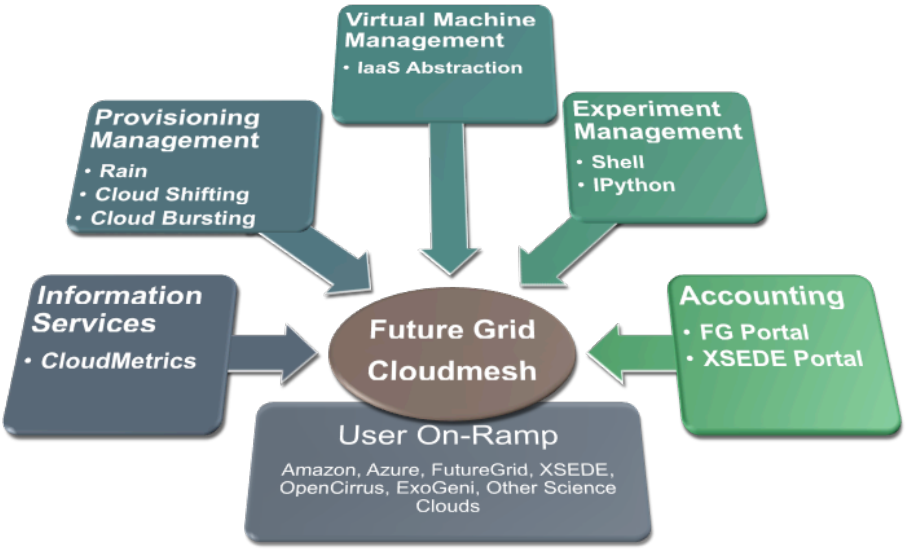
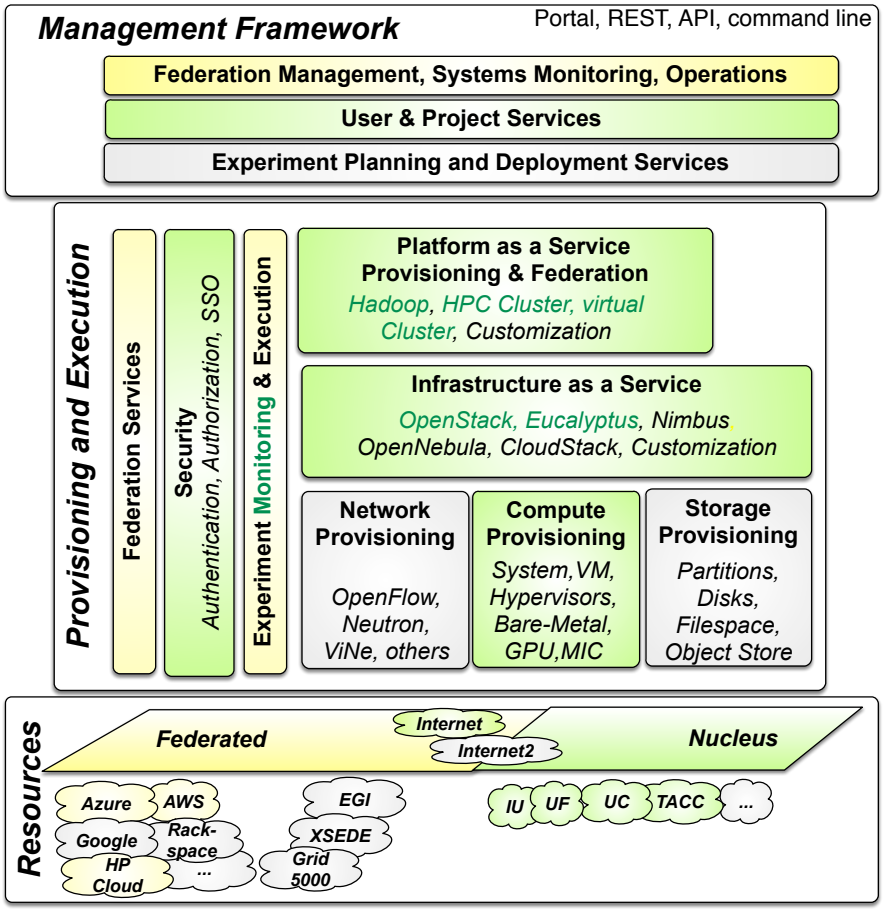


Figure 5: FutureGrid High Level User Services.



of Flor-ida.

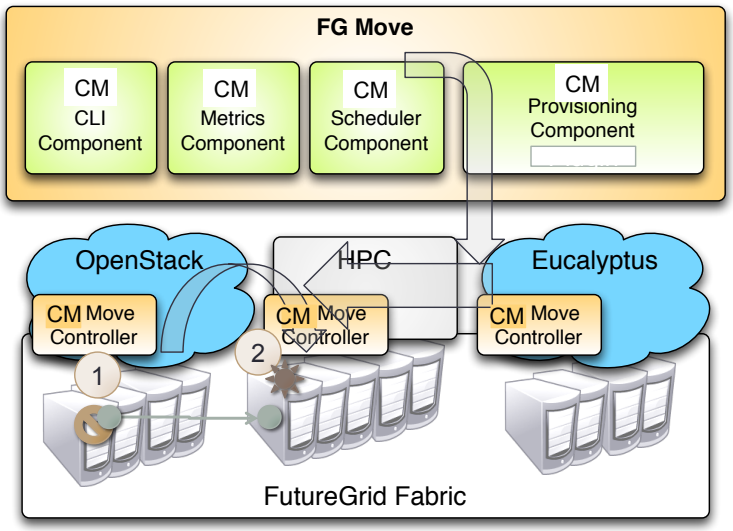
Overview of the Clusters

3 Dev Ops

Cobbler

xcat

Figure 6: FutureGrid High Level User Services.



4 Services and Tools for Big Data

4.1 Support for Educational Services

manual

reprovisioning

reconfiguration

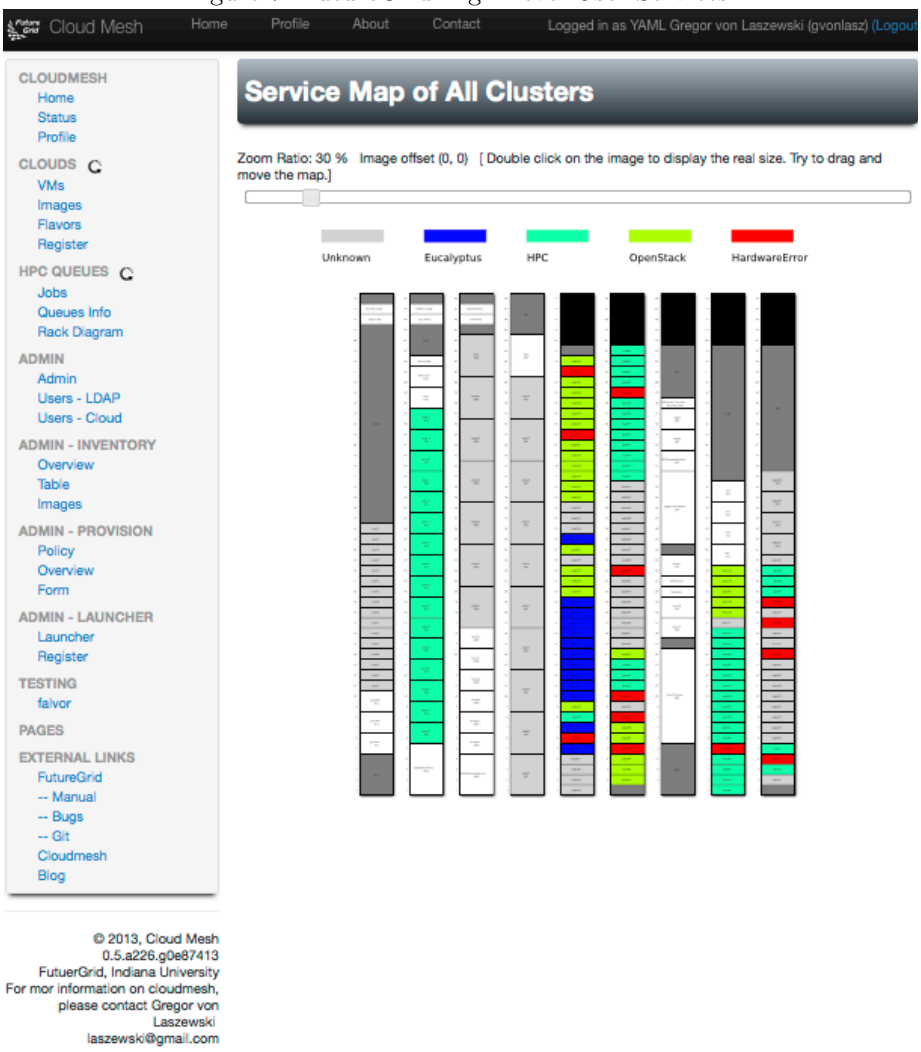
4.2 High Performance Computing

from sriram:

1. Introduction

Traditional HPC environments typically support batch job submissions using resource management systems such as the TORQUE Resource Manager (also known as the Portable Batch System PBS) or the Sun Grid Engine (SGE). On the other hand, Hadoop provides it own scheduling, and manages its own job and task submissions, and tracking. Since both systems are designed to have complete control over the resources that they manage, the challenge is how to enable users to run Hadoop jobs in a typical HPC environment using a scheduler such as PBS or SGE. In this release, we support Hadoop job submissions via PBS and SGE. However, this approach is equally feasible for other schedulers such as Condor, as well. Our approach is to configure Hadoop clusters on-demand by first requesting resources for an N-node Hadoop cluster via PBS. Once the resources are received, the Hadoop configurations and environments are set up based on the set of resources provided by PBS. The Hadoop Distributed File

Figure 7: FutureGrid High Level User Services.



System (HDFS) can be configured in one of two ways in 1) transient (non-persistent) or 2) persistent modes. In the non-persistent mode, the HDFS is set up to use local storage. In the persistent mode, the HDFS is set to symbolically link to an external location that will be persistent i.e. data from Hadoop runs will continue to persist even after the Hadoop runs are complete. More details are as follows.

2. Details

The pre-requisite for myHadoop is a valid Hadoop installation we recommend that you use Hadoop version 0.20.2 since that is the only version of Hadoop

Table 1: FutureGrid Compute Resources

Name	System Type	Nodes	CPUS	Cores	TFLOPS	RAM (GB)	Storage (TB)	Site
india	IBM iDataplex	128	256	1024	11	3072	335	IU
hotel	IBM iDataplex	84	168	672	7	2016	120	UC
sierra	IBM iDataplex	84	168	672	7	2688	96	SDSC
foxtrot	IBM iDataplex	32	64	256	3	768	0	UF
alamo	Dell Poweredge	96	192	768	8	1152	30	TACC
xray	Cray XT5m	1	166	664	6	1328	5.4	IU
bravo	HP Proliant	16	32	128	1.7	3072	128	IU
	SuperMicro							
delta	GPU Cluster	16	32	192		1333	144	IU
lima	Aeon Eclipse64	8	16	128	1.3	512	3.8	SDSC
	SuperMicro							
echo	ScaleMP Cluster	16	32	192	2	6144	192	IU

that this package has been tested with. Henceforth, we will refer to the location of the Hadoop installation as HADOOP_HOME. We will refer to the location of the myHadoop installation (i.e. this package) as MY_HADOOP_HOME. The $MY_HADOOP_HOME/pbs-example.sh shows an example of how to use myHadoop with PBS. A similar script for SGE can be found in $MY_HADOOP_HOME/sge-example.sh. A step-by-step process for using myHadoop is as follows.

2.1. Initial Configuration

Ensure that the environment variables inside $MY_HADOOP_HOME/bin/setenv.sh are set correctly. You can set your HADOOP_HOME, and the locations for your HDFS data and log directories using this script. You will need to update this script before you can proceed further. All the tuning parameters for Hadoop can be found in the $MY_HADOOP_HOME/etc directory. There is no need to edit any of the parameters, especially if you are not an expert Hadoop user. If you are familiar with the various Hadoop parameters, you may edit the parameters that fall outside the DO NOT EDIT sections.

2.2. Request N nodes from the Scheduler

Once the environment variables have been set correctly, we are ready to use myHadoop using a regular PBS or SGE submission script. Your PBS script should contain the following lines to initialize PBS as follows:

```
#!/bin/bash
#PBS -q <queue_name>
#PBS -N <job_name>
#PBS -l nodes=4:ppn=1
```

```
#PBS -o <output file>
#PBS -e <error_file>
#PBS -A <allocation>
#PBS -V
#PBS -M <user email>
#PBS -m abe
```

In the above case, we are requesting 4 nodes. Note that you must set the processors per node (ppn) to 1. Your SGE script should contain the following lines to initialize SGE:

```
#!/bin/bash
#$ -V -cwd
#$ -N <job_name>
#$ -pe <queue_name> 4
#$ -o <output file>
#$ -e <error file>
#$ -S /bin/bash
```

For SGE, there is one important rule to remember. The queue name specified above should be pre-configured with an allocation_rule set to 1 (one). This ensures that the Hadoop cluster is set up such that multiple instances of the Hadoop daemons are not scheduled on the same node.

2.3. Set the myHadoop Environment

Run the $MY_HADOOP_HOME/bin/setenv.sh script (that you modified in Section 2.1) to set all the environment variables required by myHadoop. . $MY_HADOOP_HOME/bin/setenv.sh Set the HADOOP_CONF_DIR to the directory where Hadoop configs should be generated all configuration files for the Hadoop run will be picked up from here. Ensure that this directory is accessible to all nodes and a way to do this is to make sure that this directory is on a shared file system such as NFS or Lustre. export HADOOP_CONF_DIR=¡configuration directory¿

2.4. Configure the myHadoop Cluster

You can initialize and configure the Hadoop cluster by using the $MY_HADOOP_HOME/bin/pbs-configure.sh (or sge-configure.sh) script. You may create a transient or persistent myHadoop cluster by changing the command-line arguments as follows. For a transient myHadoop cluster, configure it as follows (replace 4 with the total number of nodes requested): $MY_HADOOP_HOME/bin/pbs-configure.sh -n 4 -c $HADOOP_CONF_DIR In this mode, you will have to copy all of your data into the myHadoop cluster after it is configured, and copy out the results after the job is complete. All data will be inaccessible from HDFS once the PBS job is complete. Alternatively, you may set up a persistent myHadoop cluster by using the p option, and setting the BASE_DIR for HDFS as follows: $MY_HADOOP_HOME/bin/pbs-configure.sh -n 4 -c $HADOOP_CONF_DIR -p -d ¡HDFS BASE_DIR¿ The BASE_DIR should be on a directory accessible to all nodes, to ensure that the data will not be cleaned up after job completion.

For instance, the BASE_DIR could be on a Lustre file system. Note that, if N-node cluster is being created, then the BASE_DIR should have directories named 1, 2, , N. The configuration script sets up symbolic links from node I to the BASE_DIR/I directory. When this mode is used, there is no need to copy data back and forth from HDFS to another file system between runs.

2.5. Format HDFS (if need be) If myHadoop is being used in transient mode, or if it is being used for the first time in persistent mode, then you will have to format the HDFS as follows: $HADOOP_HOME/bin/hadoop –config $HADOOP_CONF_DIR namenode format

2.6. Run Hadoop Jobs You are now all set to start all the Hadoop daemons as follows: $HADOOP_HOME/bin/start-all.sh Once the daemons are all started up, you can start using Hadoop as usual. You may also stage data in and out from HDFS, as required.

2.7. Clean up Although, PBS or SGE may be set up to automatically clean up after your Hadoop job is complete, it is always a good idea to stop all the Hadoop daemons, and use the clean-up script to clean up after yourself. $HADOOP_HOME/bin/stop-all.sh $MY_HADOOP_HOME/bin/pbs-cleanup.sh -n 4 OR $MY_HADOOP_HOME/bin/sge-cleanup.sh -n 4

4.3 Hadoop

My Hadoop

We have various platforms that support Hadoop on FutureGrid. MyHadoop is probably the easiest solution offered for you. It provides the advantage that it is integrated into the queuing system and allows hadoop jobs to be run as batch job. This is of especial interest for classes that may run quickly out of resources if every students wants to run their hadoop application at the same time.

MapReduce is a programming model developed by Google. Their definition of MapReduce is as follows: MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key. For more information about MapReduce, please see the Google paper here.

The Apache Hadoop Project provides an open source implementation of MapReduce and HDFS (Hadoop Distributed File System).

This tutorial illustrates how to run Apache Hadoop thru the batch systems on FutureGrid using the MyHadoop tool.

1.1.1. myHadoop on FutureGrid MyHadoop is a set of scripts that configure and instantiate Hadoop as a batch job.

myHadoop 0.20.2 is currently installed on Alamo, Hotel, India, and Sierra FutureGrid systems.

5 Cloudmesh

Acknowledgement

Some of the text published in this chapter is available form the FutureGrid por-tal. The FutureGrid project is funded by the National Science Foundation (NSF) and is led by Indiana University with University of Chicago, University of Florida, San Diego Supercomputing Center, Texas Advanced Computing Center, Universi-ty of Virginia, University of Tennessee, University of Southern California, Dres-den, Purdue University, and Grid 5000 as partner sites. This material is based up-on work supported in part by the National Science Foundation under Grant No. 0910812. If you use FutureGrid, we ask you to include the reference [2, 1].

References

[1] G. C. Fox, G. von Laszewski, J. Diaz, K. Keahey, J. Fortes, R. Figueiredo, S. Smallen, W. Smith, and A. Grimshaw, *Contemporary HPC Architectures*, draft ed., 2012, ch. FutureGrid - a reconfigurable testbed for Cloud, HPC and Grid Computing. [Online]. Available: http://cyberaide.googlecode.com/svn/trunk/papers/pdf/vonLaszewski-12-fg-bookchapter.pdf

[2] G. von Laszewski, G. C. Fox, F. Wang, A. J. Younge, Kulshrestha, G. G. Pike, W. Smith, J. Voeckler, R. J. Figueiredo, J. Fortes, K. Keahey, and E. Deelman, "Design of the FutureGrid Experiment Management Framework," in *Proceedings of Gateway Computing Environments 2010 (GCE2010) at SC10*. New Orleans, LA: IEEE, Nov. 2010. [Online]. Available: http://cyberaide.googlecode.com/svn/trunk/papers/10-FG-exp-GCE10/vonLaszewski-10-FG-exp-GCE10.pdf