

# Hadoop On Demand

## Table of contents

1 Introduction .....	2
2 Feature List .....	2
2.1 Simplified Interface for Provisioning Hadoop Clusters .....	2
2.2 Automatic installation of Hadoop .....	2
2.3 Configuring Hadoop .....	2
2.4 Auto-cleanup of Unused Clusters .....	2
2.5 Log Services .....	2
3 HOD Components .....	3
3.1 HOD Client .....	3
3.2 RingMaster .....	3
3.3 HodRing .....	3
3.4 Hodrc / HOD configuration file .....	3
3.5 Submit Nodes and Compute Nodes .....	3
4 Getting Started with HOD .....	4
4.1 Pre-Requisites .....	4
4.2 Setting up HOD.....	5
5 Running HOD.....	6
5.1 Overview.....	6
5.2 Command Line Options.....	7
6 HOD Configuration .....	9
6.1 Introduction to HOD Configuration .....	9
6.2 Categories / Sections in HOD Configuration .....	9
6.3 Important and Commonly Used Configuration Options .....	9

## 1. Introduction

The Hadoop On Demand (HOD) project is a system for provisioning and managing independent Hadoop MapReduce instances on a shared cluster of nodes. HOD uses a resource manager for allocation. At present it supports [Torque](#) out of the box.

## 2. Feature List

### 2.1. Simplified Interface for Provisioning Hadoop Clusters

By far, the biggest advantage of HOD is to quickly setup a Hadoop cluster. The user interacts with the cluster through a simple command line interface, the HOD client. HOD brings up a virtual MapReduce cluster with the required number of nodes, which the user can use for running Hadoop jobs. When done, HOD will automatically clean up the resources and make the nodes available again.

### 2.2. Automatic installation of Hadoop

With HOD, Hadoop does not need to be even installed on the cluster. The user can provide a Hadoop tarball that HOD will automatically distribute to all the nodes in the cluster.

### 2.3. Configuring Hadoop

Dynamic parameters of Hadoop configuration, such as the NameNode and JobTracker addresses and ports, and file system temporary directories are generated and distributed by HOD automatically to all nodes in the cluster. In addition, HOD allows the user to configure Hadoop parameters at both the server (for e.g. JobTracker) and client (for e.g. JobClient) level, including 'final' parameters, that were introduced with Hadoop 0.15.

### 2.4. Auto-cleanup of Unused Clusters

HOD has an automatic timeout so that users cannot misuse resources they aren't using. The timeout applies only when there is no MapReduce job running.

### 2.5. Log Services

HOD can be used to collect all MapReduce logs to a central location for archiving and inspection after the job is completed.

### 3. HOD Components

This is a brief overview of the various components of HOD and how they interact to provision Hadoop.

#### 3.1. HOD Client

The HOD client is a Unix command that users use to allocate Hadoop MapReduce clusters. The command provides other options to list allocated clusters and deallocate them. The HOD client generates the *hadoop-site.xml* in a user specified directory. The user can point to this configuration file while running Map/Reduce jobs on the allocated cluster.

The nodes from where the HOD Client is run are called *submit nodes* because jobs are submitted to the resource manager system for allocating and running clusters from these nodes.

#### 3.2. RingMaster

The RingMaster is a HOD process that is started on one node per every allocated cluster. It is submitted as a 'job' to the resource manager by the HOD client. It controls which Hadoop daemons start on which nodes. It provides this information to other HOD processes, such as the HOD client, so users can also determine this information. The RingMaster is responsible for hosting and distributing the Hadoop tarball to all nodes in the cluster. It also automatically cleans up unused clusters.

#### 3.3. HodRing

The HodRing is a HOD process that runs on every allocated node in the cluster. These processes are run by the RingMaster through the resource manager, using a facility of parallel execution. The HodRings are responsible for launching Hadoop commands on the nodes to bring up the Hadoop daemons. They get the command to launch from the RingMaster.

#### 3.4. Hodrc / HOD configuration file

An INI style configuration file where the users configure various options for the HOD system, including install locations of different software, resource manager parameters, log and temp file directories, parameters for their MapReduce jobs, etc.

#### 3.5. Submit Nodes and Compute Nodes

The nodes from where the *HOD Client* is run are referred as *submit nodes* because jobs are submitted to the resource manager system for allocating and running clusters from these nodes.

The nodes where the *Ringmaster* and *HodRings* run are called the Compute nodes. These are the nodes that get allocated by a resource manager, and on which the Hadoop daemons are provisioned and started.

## 4. Getting Started with HOD

### 4.1. Pre-Requisites

#### 4.1.1. Hardware

HOD requires a minimum of 3 nodes configured through a resource manager.

#### 4.1.2. Software

The following components are assumed to be installed before using HOD:

- *Torque*: Currently HOD supports Torque out of the box. We assume that you are familiar with configuring Torque. You can get information about this from [here](#).
- *Python*: We require version 2.5.1, which can be downloaded from [here](#).

The following components can be optionally installed for getting better functionality from HOD:

- *Twisted Python*: This can be used for improving the scalability of HOD. Twisted Python is available [here](#).
- *Hadoop*: HOD can automatically distribute Hadoop to all nodes in the cluster. However, it can also use a pre-installed version of Hadoop, if it is available on all nodes in the cluster. HOD currently supports Hadoop 0.15 and above.

HOD configuration requires the location of installs of these components to be the same on all nodes in the cluster. It will also make the configuration simpler to have the same location on the submit nodes.

#### 4.1.3. Resource Manager Configuration Pre-requisites

For using HOD with Torque:

- Install Torque components: `pbs_server` on a head node, `pbs_moms` on all compute nodes, and PBS client tools on all compute nodes and submit nodes.

- Create a queue for submitting jobs on the pbs\_server.
- Specify a name for all nodes in the cluster, by setting a 'node property' to all the nodes. This can be done by using the 'qmgr' command. For example: `qmgr -c "set node node properties=cluster-name"`
- Ensure that jobs can be submitted to the nodes. This can be done by using the 'qsub' command. For example: `echo "sleep 30" | qsub -l nodes=3`

More information about setting up Torque can be found by referring to the documentation [here](#).

## 4.2. Setting up HOD

- HOD is available in the 'contrib' section of Hadoop under the root directory 'hod'. Distribute the files under this directory to all the nodes in the cluster.
- On the node from where you want to run hod, edit the file hodrc which can be found in the `install dir/conf` directory. This file contains the minimal set of values required for running hod.
- Specify values suitable to your environment for the following variables defined in the configuration file. Note that some of these variables are defined at more than one place in the file.

Variable Name	Meaning
<code>\${JAVA_HOME}</code>	Location of Java for Hadoop. Hadoop supports Sun JDK 1.5.x
<code>\${CLUSTER_NAME}</code>	Name of the cluster which is specified in the 'node property' as mentioned in resource manager configuration.
<code>\${HADOOP_HOME}</code>	Location of Hadoop installation on the compute and submit nodes.
<code>\${RM_QUEUE}</code>	Queue configured for submitting jobs in the resource manager configuration.
<code>\${RM_HOME}</code>	Location of the resource manager installation on the compute and submit nodes.

- The following environment variables *may* need to be set depending on your environment. These variables must be defined where you run the HOD client, and also be specified in the HOD configuration file as the value of the key `resource_manager.env-vars`. Multiple variables can be specified as a comma separated list of key=value pairs.

Variable Name	Meaning
---------------	---------

HOD_PYTHON_HOME	If you install python to a non-default location of the compute nodes, or submit nodes, then, this variable must be defined to point to the python executable in the non-standard location.
-----------------	--

You can also review other configuration options in the file and modify them to suit your needs. Refer to the the section on configuration below for information about the HOD configuration.

## 5. Running HOD

### 5.1. Overview

A typical session of HOD will involve atleast three steps: allocate, run hadoop jobs, deallocate.

#### 5.1.1. Operation allocate

The allocate operation is used to allocate a set of nodes and install and provision Hadoop on them. It has the following syntax:

```
hod -c config_file -t hadoop_tarball_location -o "allocate cluster_dir number_of_nodes"
```

The `hadoop_tarball_location` must be a location on a shared file system accesible from all nodes in the cluster. Note, the `cluster_dir` must exist before running the command. If the command completes successfully then `cluster_dir/hadoop-site.xml` will be generated and will contain information about the allocated cluster's JobTracker and NameNode.

For example, the following command uses a `hodrc` file in `~/hod-config/hodrc` and allocates Hadoop (provided by the tarball `~/share/hadoop.tar.gz`) on 10 nodes, storing the generated Hadoop configuration in a directory named `~/hadoop-cluster`:

```
$ hod -c ~/hod-config/hodrc -t ~/share/hadoop.tar.gz -o "allocate ~/hadoop-cluster 10"
```

HOD also supports an environment variable called `HOD_CONF_DIR`. If this is defined, HOD will look for a default `hodrc` file at `$HOD_CONF_DIR/hodrc`. Defining this allows the above command to also be run as follows:

```
$ export HOD_CONF_DIR=~/hod-config
$ hod -t ~/share/hadoop.tar.gz -o "allocate ~/hadoop-cluster 10"
```

#### 5.1.2. Running Hadoop jobs using the allocated cluster

Now, one can run Hadoop jobs using the allocated cluster in the usual manner:

```
hadoop --config cluster_dir hadoop_command hadoop_command_args
```

Continuing our example, the following command will run a wordcount example on the allocated cluster:

```
$ hadoop --config ~/hadoop-cluster jar /path/to/hadoop/hadoop-examples.jar wordcount  
/path/to/input /path/to/output
```

### 5.1.3. Operation deallocate

The deallocate operation is used to release an allocated cluster. When finished with a cluster, deallocate must be run so that the nodes become free for others to use. The deallocate operation has the following syntax:

```
hod -o "deallocate cluster_dir"
```

Continuing our example, the following command will deallocate the cluster:

```
$ hod -o "deallocate ~/hadoop-cluster"
```

## 5.2. Command Line Options

This section covers the major command line options available via the hod command:

*--help*

Prints out the help message to see the basic options.

*--verbose-help*

All configuration options provided in the hodrc file can be passed on the command line, using the syntax `--section_name.option_name[=value]`. When provided this way, the value provided on command line overrides the option provided in hodrc. The `verbose-help` command lists all the available options in the hodrc file. This is also a nice way to see the meaning of the configuration options.

*-c config\_file*

Provides the configuration file to use. Can be used with all other options of HOD. Alternatively, the `HOD_CONF_DIR` environment variable can be defined to specify a directory that contains a file named `hodrc`, alleviating the need to specify the configuration file in each HOD command.

*-b 1/2/3/4*

Enables the given debug level. Can be used with all other options of HOD. 4 is most verbose.

*-o "help"*

Lists the operations available in the operation mode.

*-o "allocate cluster\_dir number\_of\_nodes"*

Allocates a cluster on the given number of cluster nodes, and store the allocation information in cluster\_dir for use with subsequent hadoop commands. Note that the cluster\_dir must exist before running the command.

*-o "list"*

Lists the clusters allocated by this user. Information provided includes the Torque job id corresponding to the cluster, the cluster directory where the allocation information is stored, and whether the Map/Reduce daemon is still active or not.

*-o "info cluster\_dir"*

Lists information about the cluster whose allocation information is stored in the specified cluster directory.

*-o "deallocate cluster\_dir"*

Deallocates the cluster whose allocation information is stored in the specified cluster directory.

*-t hadoop\_tarball*

Provisions Hadoop from the given tar.gz file. This option is only applicable to the allocate operation. For better distribution performance it is recommended that the Hadoop tarball contain only the libraries and binaries, and not the source or documentation.

*-Mkey1=value1 -Mkey2=value2*

Provides configuration parameters for the provisioned Map/Reduce daemons (JobTracker and TaskTrackers). A hadoop-site.xml is generated with these values on the cluster nodes

*-Hkey1=value1 -Hkey2=value2*

Provides configuration parameters for the provisioned HDFS daemons (NameNode and DataNodes). A hadoop-site.xml is generated with these values on the cluster nodes

*-Ckey1=value1 -Ckey2=value2*

Provides configuration parameters for the client from where jobs can be submitted. A

hadoop-site.xml is generated with these values on the submit node.

## 6. HOD Configuration

### 6.1. Introduction to HOD Configuration

Configuration options for HOD are organized as sections and options within them. They can be specified in two ways: a configuration file in the INI format, and as command line options to the HOD shell, specified in the format `--section.option[=value]`. If the same option is specified in both places, the value specified on the command line overrides the value in the configuration file.

To get a simple description of all configuration options, you can type `hod --verbose-help`

This section explains some of the most important or commonly used configuration options in some more detail.

### 6.2. Categories / Sections in HOD Configuration

The following are the various sections in the HOD configuration:

Section Name	Description
hod	Options for the HOD client
resource_manager	Options for specifying which resource manager to use, and other parameters for using that resource manager
ringmaster	Options for the RingMaster process
hodring	Options for the HodRing process
gridservice-mapred	Options for the MapReduce daemons
gridservice-hdfs	Options for the HDFS daemons

### 6.3. Important and Commonly Used Configuration Options

#### 6.3.1. Common configuration options

Certain configuration options are defined in most of the sections of the HOD configuration. Options defined in a section, are used by the process for which that section applies. These options have the same meaning, but can have different values in each section.

Option Name	Description
temp-dir	Temporary directory for usage by the HOD processes. Make sure that the users who will run hod have rights to create directories under the directory specified here.
debug	A numeric value from 1-4. 4 produces the most log information, and 1 the least.
log-dir	Directory where log files are stored. By default, this is <i>install-location/logs/</i> . The restrictions and notes for the temp-dir variable apply here too.
xrs-port-range	A range of ports, among which an available port shall be picked for use to run any XML-RPC based server daemon processes of HOD.
http-port-range	A range of ports, among which an available port shall be picked for use to run any HTTP based server daemon processes of HOD.

### 6.3.2. hod options

Option Name	Description
cluster	A descriptive name given to the cluster. For Torque, this is specified as a 'Node property' for every node in the cluster. HOD uses this value to compute the number of available nodes.
client-params	A comma-separated list of hadoop config parameters specified as key-value pairs. These will be used to generate a hadoop-site.xml on the submit node that should be used for running MapReduce jobs.

### 6.3.3. resource\_manager options

Option Name	Description
queue	Name of the queue configured in the resource manager to which jobs are to be submitted.
batch-home	Install directory to which 'bin' is appended and under which the executables of the resource manager can be found.
env-vars	This is a comma separated list of key-value

	<p>pairs, expressed as key=value, which would be passed to the jobs launched on the compute nodes. For example, if the python installation is in a non-standard location, one can set the environment variable 'HOD_PYTHON_HOME' to the path to the python executable. The HOD processes launched on the compute nodes can then use this variable.</p>
--	--

### 6.3.4. ringmaster options

Option Name	Description
work-dirs	<p>These are a list of comma separated paths that will serve as the root for directories that HOD generates and passes to Hadoop for use to store DFS / MapReduce data. For e.g. this is where DFS data blocks will be stored. Typically, as many paths are specified as there are disks available to ensure all disks are being utilized. The restrictions and notes for the temp-dir variable apply here too.</p>

### 6.3.5. gridservice-hdfs options

Option Name	Description
external	<p>If false, this indicates that a HDFS cluster must be bought up by the HOD system, on the nodes which it allocates via the allocate command. Note that in that case, when the cluster is de-allocated, it will bring down the HDFS cluster, and all the data will be lost. If true, it will try and connect to an externally configured HDFS system.</p> <p>Typically, because input for jobs are placed into HDFS before jobs are run, and also the output from jobs in HDFS is required to be persistent, an internal HDFS cluster is of little value in a production system. However, it allows for quick testing.</p>
host	<p>Hostname of the externally configured NameNode, if any.</p>
fs_port	<p>Port to which NameNode RPC server is bound.</p>
info_port	<p>Port to which the NameNode web UI server is bound.</p>

pkgs	Installation directory, under which bin/hadoop executable is located. This can be used to use a pre-installed version of Hadoop on the cluster.
server-params	A comma-separated list of hadoop config parameters specified key-value pairs. These will be used to generate a hadoop-site.xml that will be used by the NameNode and DataNodes.
final-server-params	Same as above, except they will be marked final.

### 6.3.6. gridservice-mapred options

Option Name	Description
external	If false, this indicates that a MapReduce cluster must be bought up by the HOD system on the nodes which it allocates via the allocate command. If true, it will try and connect to an externally configured MapReduce system.
host	Hostname of the externally configured JobTracker, if any.
tracker_port	Port to which the JobTracker RPC server is bound.
info_port	Port to which the JobTracker web UI server is bound.
pkgs	Installation directory, under which bin/hadoop executable is located. This can be used to use a pre-installed version of Hadoop on the cluster.
server-params	A comma-separated list of hadoop config parameters specified key-value pairs. These will be used to generate a hadoop-site.xml that will be used by the JobTracker and TaskTrackers.
final-server-params	Same as above, except they will be marked final.