



**Moab Adaptive HPC Suite™**

**Administrator's Guide**

version 6.1

# Moab Adaptive HPC Suite™ Administrator's Guide

version 6.1

Copyright © 1999-2011 Adaptive Computing Enterprises, Inc.  
Distribution of this document for commercial purposes in either hard or soft copy form is strictly prohibited without prior written consent from Adaptive Computing Enterprises, Inc.  
[Release notes](#)

The Moab Adaptive HPC Suite uses Moab Cluster Suite® to dynamically change a node's OS to meet workload needs. As a meta-scheduler, Moab optimally determines when the OS mix should be modified based on defined policies and service level agreements as well as current and projected workload. When the specified conditions are met, Moab triggers the OS change using a site's preferred OS-modification technology, such as dual boot or provisioning.

The Moab Adaptive HPC Suite can be applied to both new and existing clusters to help yield maximum hardware utilization and ROI. The hybrid cluster model also consolidates administration and centralizes job submission across both platforms. Administrators can easily manage the policies and workload for both OS environments—Linux and Windows—from Moab's unified console. Moab can also make the dual-OS nature of the cluster transparent to end-users by applying application and workload information that ensures jobs run on the correct OS without the user needing to specify it. Moab can manipulate, grow, and shrink allocated resources to meet service level targets.

## Administrative Topics

- [1.0 Deploying the Operating Systems](#)
  - [1.1 Installing and Configuring Microsoft Windows HPC Server 2008](#)
    - [1.1.1 Automating the Home Directory and SSH Key Creation](#)
    - [1.1.2 Installing the MSMHPC Services](#)
    - [1.1.3 Reinstalling the HPC Pack](#)
  - [1.2 Configuring Linux](#)
    - [1.2.1 Configuring CentOS](#)
    - [1.2.2 Configuring SUSE Linux Enterprise Server \(SLES\)](#)
  - [1.3 Installing MSMHPC Tools](#)
  - [1.4 Deploying Compute Nodes](#)
  - [1.5 Installing TORQUE](#)
  - [1.6 Test Your Configuration](#)
  - [1.7 Configuring SSL for Adaptive HPC](#)
    - [1.7.1 Windows Head Node Setup](#)
    - [1.7.2 Linux Head Node Setup](#)
    - [1.7.3 Troubleshooting](#)
- [2.0 Preparing the Linux Head Node](#)
  - [2.1 Installing Moab](#)
  - [2.2 Configuring Samba](#)
- [3.0 PXE Booting](#)
  - [3.1 Configuring a DHCP Server](#)
  - [3.2 TFTP Server and PXE Booting](#)
    - [3.2.1 Switching Operating Systems with xCAT](#)
- [4.0 Moab Configuration](#)
  - [4.1 Scheduler Configuration](#)
  - [4.2 On-Demand Provisioning](#)
  - [4.3 Provisioning & Load Balancing](#)
    - [4.3.1 Switching from Dual to Single OS Provisioning](#)
    - [4.3.2 Configuring Multiple Operating Systems in Windows](#)
  - [4.4 Resource Manager Configuration](#)
    - [4.4.1 Linux Resource Manager Configuration](#)
      - [4.4.1.1 TORQUE Setup](#)
      - [4.4.1.2 Sun Grid Engine Configuration](#)

- 4.4.2 Windows Resource Manager Configuration
- 5.0 Submitting Jobs through Moab
  - 5.1 Caching the User Credentials
  - 5.2 Submitting Jobs with msub
  - 5.3 Verifying the Job has Migrated to HPC 2008 R2
  - 5.4 Verifying the Job Runs Correctly
- 6.0 Configuring Moab for Dual-Boot Systems
  - 6.1 Node Setup
- 7.0 Creating SSH Shared Keys for Cluster Users
- 8.0 Centralizing Authentication with Active Directory
- 9.0 Reservation Tracking
- Appendix A: GRUB Dual-Boot OS Switching
- Appendix B: Troubleshooting Common Problems

# Moab Adaptive HPC Suite Administrator's Guide

Copyright © 1999-2011 Adaptive Computing Enterprises Inc. All Rights Reserved  
Distribution of this document for commercial purposes in either  
hard or soft copy form is strictly prohibited  
without prior written consent from Adaptive Computing.

This document provides an overview of the changes made in a minor release of Moab Adaptive HPC Suite. A minor release includes significant new functionality and may break compatibility with previous versions.

## Requirements

Moab Adaptive HPC requires the following software versions:

- Adaptive HPC Suite 6.1 or 6.0.5
- MSMHPC-6.1
- MSMHPC\_tools-6.1
- Moab 6.1 or 6.0.5
- Torque 2.5.6

## Fixes

- Documentation now describes managing multiple (more than two) OSes.
- SMHPC tools can force a recache of nodes.
- Create active directory user script now adds the user to the configured HPC users group automatically (HPCUsers by default).
- Job failure information is now more descriptive and no longer tells the user to look in the logs for most failure reasons.
- The os.switch.pl script allows dual boot configuration with xCAT.
- MSMHPC Manater now allows you to change the secured service port from the default.
- Secured and unsecured service ports can be enabled or disabled. The firewall ports will not be opened for them if both are disabled.
- The import.node.xml.pl script allows you to deploy a large number of nodes by enabling the ability to consume nodes.xml files directly from Linux.
- The initial configuration file saves by default. It is not necessary to save the configuration the first time MSMHPC is installed.
- The installer now uses the DNS host name for the spool share instead of the IP address of the first network adapter on the system.
- The MSMHPC web service now supports SSL for communication with the Linux head node. Configuration is explained in the documentation.
- MSMHPC service no longer times out on the first restart in the MSMHPC Manger.
- The MSMHPC cache service log file no longer contains warnings when a node is deleted.
- Moab immediately shows a Removed state when jobs are canceled in Windows.
- Documentation is now clearer on configuring the domain between env.hpc and the moab.cfg files.

## Upgrading

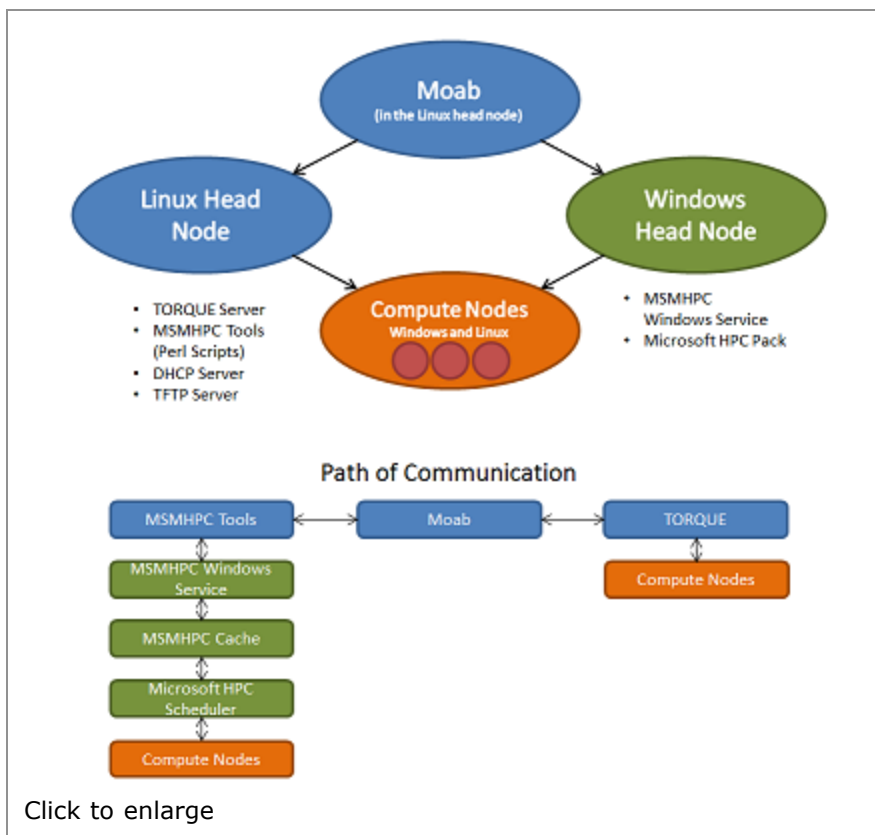
- To upgrade from MSMHPC 6.0.1, uninstall it, then install the newest version. Open the MSMHPC Manager, save the configuration, and restart the service. Your previous configuration should be preserved.

# 1.0 Deploying the Operating Systems

Software versions required:

- CentOS 5.4 (x64) or Suse Linux Enterprise Server 11 (x64)
- Moab Workload Manager (official 6.1 release)
- MSMHPC 6.1
- MSMHPC\_tools-6.1
- Windows HPC Server 2008 R2
- Windows HPC Pack 2008 R2 SP1

Moab, physically located in the Linux head node, communicates with the compute nodes through both the Linux and Windows resource managers. If TORQUE is configured, Moab communicates with the compute nodes via the TORQUE server. On Windows, Moab communicates with the MSMHPC tools (or the Perl scripts) on the Linux head node, which communicate with the MSMHPC cache on the Windows head node. The MSMHPC cache is updated with information from the Microsoft HPC Scheduler. The Microsoft HPC Scheduler communicates directly with the compute nodes.



- 1.1 Installing and Configuring Microsoft Windows HPC Server 2008
  - 1.1.1 Installing the MSMHPC Services
  - 1.1.2 Reinstalling the HPC Pack
- 1.2 Configuring Linux
  - 1.2.1 Configuring CentOS
  - 1.2.2 Configuring SUSE Linux Enterprise Server (SLES)
- 1.3 Installing MSMHPC Tools
- 1.4 Deploying Compute Nodes
- 1.5 Installing TORQUE
- 1.6 Test Your Configuration
- 1.7 Configuring SSL for Adaptive HPC
  - 1.7.1 Windows Head Node Setup

- [1.7.2 Linux Head Node Setup](#)
- [1.7.3 Troubleshooting](#)

# 1.1 Microsoft Windows HPC Server 2008 R2

- [1.1.1 Automating the Home Directory and SSH Key Creation](#)
- [1.1.2 Installing MSMHPC Services](#)
- [1.1.3 Reinstalling the HPC Pack](#)

HPC Pack 2008 R2 SP 1 is required for MSMHPC and Moab 6.1. Follow instructions in the [Windows HPC Server 2008 Reviewers Guide](#) to install and configure Microsoft Windows HPC Server 2008 R2.

1. Deploy the Windows head node (Microsoft Windows HPC 2008 R2), configure static IP addresses, disable IPV6, and add AD Server role if no Enterprise AD is present.



Turn off IPv6 support, both on the head node and compute nodes, if you are not going to use it as it might conflict with your setup.

If you deploy nodes automatically, consider modifying the Windows `diskpart.txt` script to leave free space at the end of the drive. This script is located at `C:\Program Files\Microsoft HPC Pack 2008\Data\InstallShare\Config\diskpart.txt`. Note that the partition size is in **MB**.



Verify the NTFS partition does not fill the entire drive when you install HPC 2008 R2 on the compute nodes.

```
select disk 0
clean
create partition primary size=150000
select partition 1
assign letter=c
format FS=NTFS LABEL="Node" QUICK OVERRIDE
active
exit
```

Because the DHCP server requests are answered by an external DHCP server by default, disable the DHCP Server Role in Windows.

2. Install HPC Pack 2008 R2 on the Windows head node.
3. Configure HPC Pack. Create the Default ComputeNode Template including the OS installation steps.
4. Remove the DHCP Server Role from the Windows head node, otherwise the WDS DHCP server will interfere with the DHCP server running on Linux.
5. Install MSMHPC Service on the Windows head node.



When installing MSMHPC, MSMHPC Manager prompts to create a default configuration file if one does not exist on startup. It then prompts to restart the service.



The MSMHPC installer makes necessary registry changes to the WDS during installation so that they do not interfere with the DHCP server running on Linux. If the HPC Pack is reinstalled, the administrator must ensure that the DHCP Server Role is removed and that the HPC settings are reset using the MSMHPC Manager.

## 1.1.1 Automating the Home Directory and SSH Key Creation

1. Share `/home` through NFS

```
#cat /etc/exports
/home *(rw, sync)
/data/network-install *(ro, sync)
```

2. Add NFS share to `etc/fstab`:

```
# for i in node01 node02 node03 node04 ; do ssh ${i} "echo x36-
```

```
lhn:/home /home nfs defaults 0 0 >> /etc/fstab"; done
```

3. Mount NFS share:

```
# for i in node01 node02 node03 node04 ; do ssh ${i} mount /home ;  
done
```

4. Create home directories for all the AD users:

```
# for i in `wbinfo -u` ; do su ${i} -l -c echo ; done
```

5. Generate new keys.

```
# for i in `wbinfo -u` ; do su ${i} -l -c "ssh-keygen -t rsa -q -f  
~${i}/.ssh/id_rsa -N \"\" \"\" ; done  
# for i in `wbinfo -u` ; do su ${i} -l -c "cp .ssh/id_rsa.pub  
.ssh/authorized_keys" ; done
```

## 1.1.2 Installing the MSMHPC Services

To set up Windows to enable Moab and HPC 2008 R2 integration, do the following:

1. Obtain the latest MSMHPC Server package (MSMHPC\_Installer-<version>.msi). It can be found at:

<http://www.clusterresources.com/product/hybrid/snapshots>



MSMHPC requires .NET 3.5 SP1.

2. Install the MSMHPC Server package.

- A. Run the setup package to launch the MSMHPC Setup Wizard.
- B. Read the License Agreement. If you accept the terms in the license, click **Next**.
- C. On the **Select Installation Folder** page, verify the installation path is correct, specify whether you want to make the installation available across all users or just for your user, and click **Next**.



Typically, you will want to select the default option ("Just me") as regular users will not be able to read/write to the MSMHPC installation directory.

- D. When the **Installation Complete** page opens, click **Close** to exit the wizard.
3. Launch MSMHPC Manager. (Depending on selected preferences, you might be able to double-click the MSMHPC Manager shortcut on your desktop to launch MSMHPC Manager.)
  4. Click **Configure** and make adjustments in the MSMHPC Manager Configuration forms according to your requirements. The following offers explanations for some of the fields:

A. **Security** tab

1. **Server Port**: Used to configure an unsecured service port.
2. **Secure Port**: Used to configure a secure service port.
3. **Private Key**: This is used to authenticate privileged actions required by the MSMHPC Tools to the MSMHPC Service.

You can specify a unique key or click the **Generate** button. Clicking **Generate** creates a random key using the number of bytes specified in the **Key Size** option box.

4. **Public Key**: This is used to authenticate unprivileged actions required by the MSMHPC Tools to the MSMHPC Service.

You can specify a unique key or click the **Generate** button. Clicking **Generate** creates a random key using the number of bytes specified in the **Key Size** option box.

The Disable Unsecured Service Connections and Disable Secured Service Connections



options disable the non-SSL port and the SSL secured port configured at the top of the window. You cannot disable both secured and unsecured ports for the web service at once.

The screenshot shows the 'Moab Services for HPC 2008 Manager Configuration' window. The title bar includes the Moab logo and the text 'Moab Services for HPC 2008 Manager Configuration'. The main window is titled 'MSMHPC Service Parameters:'. At the top right is the 'Adaptive COMPUTING' logo. Below the title, there are input fields for 'Cluster Hostname' (localhost), 'Server Port' (5343), and 'Secure Port' (5345). A tabbed interface is visible with tabs for 'Logging', 'Database', 'Diskless', 'Security' (selected), 'Preferences', 'Miscellaneous', 'Client', and 'Node Features'. The 'Security Settings' section contains: 'Private Key' and 'Public Key' fields with 'Generate' buttons, a 'Key Size' dropdown set to 32, and two checkboxes: 'Disable Unsecured Service Connections' and 'Disable Secured Service Connections'. At the bottom are buttons for 'Flush DBs', 'Reset HPC Settings', and 'Save Settings'.

## B. Database tab:

1. **Type:** This is used to change the database type from the default SQL Compact Edition.



Moab Adaptive HPC currently supports Microsoft SQL Compact Edition, SQL Server, and SQL Server Express. SQL Compact Edition is the recommended database for small installations of fewer than 64 nodes. For large scale clusters above 64 nodes, SQL Server and SQL Server Express are recommended.

2. **Connection String:** This is used to modify the connection string. If you switch the database type, you must change to a valid connection string.
3. **Generate Default String:** This is used to overwrite the current connection string with a valid one if you do not know which to use with the selected database.




As long as the correct permissions are in place, the schemas and databases generate automatically. SQL Compact Edition and SQL Server/Express and their schema are automatically added to the database if the correct credentials are present. To create a new SQL Compact Edition database, verify that the connection string points to a .sdf file that does not yet exist. MSMHPC creates the file and the schema at startup.


## C. Preferences tab

1. **Cron Timeout:** If you want MSMHPC to be more responsive, you can lower the **Cron Timeout** setting to 10000 milliseconds (10 seconds), but doing so consumes more CPU.
2. **Exclude Nodes:** Exclude certain nodes from being reported to Moab by entering a comma-delimited list or regular expression.

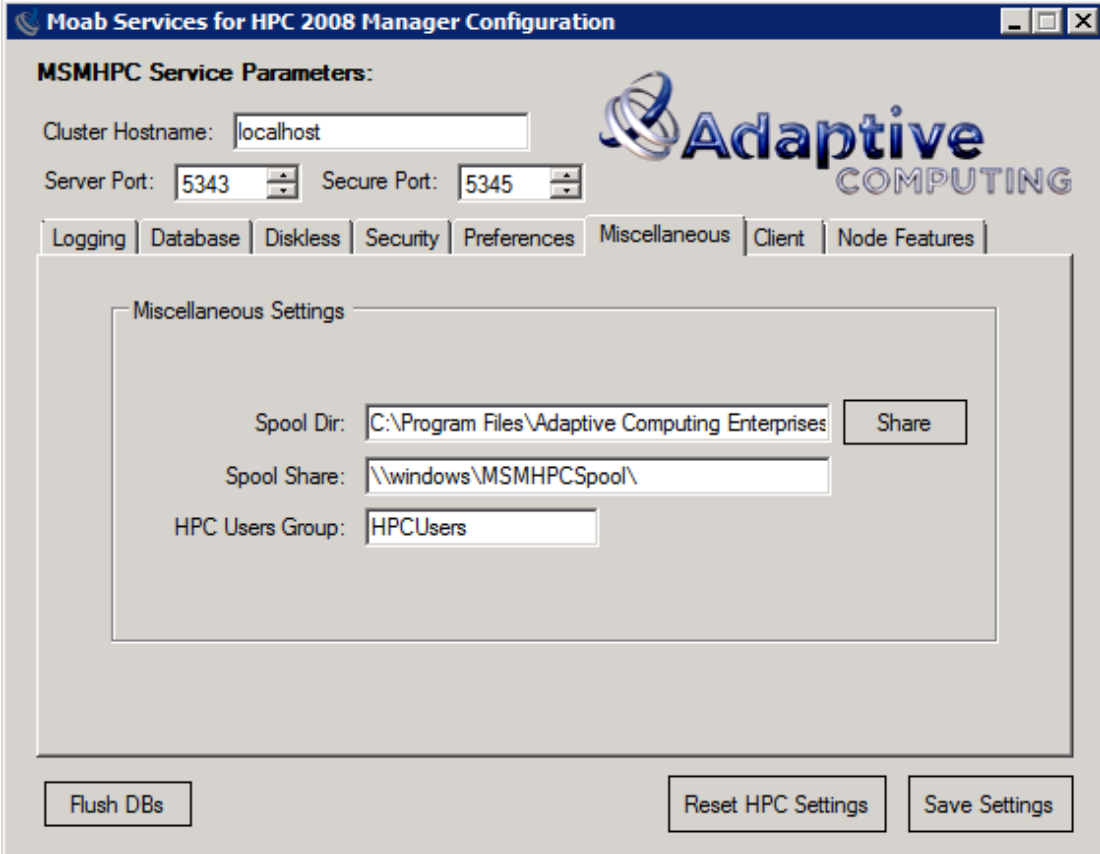
#### D. Miscellaneous tab

1. **Spool Dir**: Change to a local directory that is shared by the network path specified by the **Spool Share** parameter.
2. **Spool Share**: Change this to a network share reachable by all the compute nodes. This share should point to the local directory specified in **Spool Dir**.

 The default MSMHPC share is created by the DNS host name of the head node. For instance, the default location of the spool share for head node windows.domain.com is \\windows\MSMHPCSpool\

 The default values for both **Spool Dir** and **Spool Share** should be acceptable because the setup application automatically creates a share.


3. **HPC Users Group**: The default group in which users are placed when the create.ad.account.pl script is used.



The screenshot shows the 'Moab Services for HPC 2008 Manager Configuration' window. The 'Miscellaneous' tab is selected. The 'MSMHPC Service Parameters' section includes: Cluster Hostname: localhost; Server Port: 5343; Secure Port: 5345. The 'Miscellaneous Settings' section includes: Spool Dir: C:\Program Files\Adaptive Computing Enterprises; Spool Share: \\windows\MSMHPCSpool\; HPC Users Group: HPCUsers. Buttons for 'Share', 'Flush DBs', 'Reset HPC Settings', and 'Save Settings' are visible.

- E. **Client** tab: Click **Generate Client Config** to create Moab configuration files. This will save two files to your desktop: moab.cfg and moab-private.cfg. You will use these two files later to configure Moab (see [Moab Configuration](#)).

5. Click **Save Settings**. Then, when prompted, click **Yes** to restart the MSMHPC Service.

 Most of the settings should be automatically set up during the installation process. After installing MSMHPC, the administrator should click **Save Settings** (even if no changes were made) to create a pair of shared keys.

6. Exit the Manager Configuration and return to the MSMHPC Manager. Click **Start or stop** to identify whether the MSMHPC Service is running. If it is not running, click **Start**.
7. On MSMHPC Manager, click **Check**.

- A. Click **Check Service** to again confirm that the MSMHPC Service is running.
  - B. Specify the location of the Web Service in the **Web Service URL** field. The suggested URL should be correct.
  - C. Click **Check Web** to confirm that the Public Web Service is running.
8. On MSMHPC Manager, click **Test** to open the MSMHPC Manager Test Form.
- A. Click **Get Nodes**.  
If nodes are reported, then MSMHPC is communicating correctly with the Windows HPC job scheduler.
9. For Linux users to submit jobs to the HPC cluster, they must exist as users on the Windows domain. To add users to the domain, do the following:
- A. Open the User Administration GUI. (**Start**→**Administrative Tools**→**Active Directory Users and Computers**)
  - B. Click **Users** in the left navigation window.
  - C. On the toolbar, click **Create Users**.
  - D. Fill in the user's information.
  - E. Click **Next**.
  - F. Create a password for the user. You may allow the user to create a password, but the password must be the same for both the Windows and Linux environments for write access to the home directory on both systems. Ensure that the user is not required to change the password at the next login. Click **Finish**.
  - G. Right-click the newly created user name and make any additional changes needed, such as adding the user to additional groups, for example.



Users must be cached before they can submit jobs. See [Cache User Credentials](#).

To submit jobs from Moab, all Linux users must have the same user credentials as the credentials on the Windows system. If the head node is a compute node as well as the domain controller, a policy disables users from logging in to that node. Consider only using the head node as a compute node if that machine is not the domain controller for the cluster.

### 1.1.3 Reinstalling the HPC Pack

If you encounter a situation in which you need to reinstall the HPC Pack, take the following into consideration:

- If you want to deploy HPC nodes automatically using Microsoft Windows Deployment Services but rely on an external DHCP server to reply to PXE requests, then you need to remove the DHCP Server Role.
- You need to reset the HPC Scheduler settings so that MSMHPC can have control over it. To do so, run the MSMHPC Manager from your desktop, go to **Configure** and click **Reset HPC Settings**.
- You need to cache all the user's credentials. You can do this by running the `user.cache.pl` script from your Linux head node, using the MSMHPC Manager (**Test** → **Service Operations**) or by using the provided MSMHPCCacheCredentialsCLI or MSMHPCCacheCredentialsGUI applications inside the directory where you installed the MSMHPC Services. For more information, see [Cache User Credentials](#).

## 1.2 Configuring Linux

- [1.2.1 CentOS](#)
- [1.2.2 SUSE Linux Enterprise Server](#)

The following explains how to set up the Linux side of a Windows HPC2008/Moab cluster. Before setting up the Linux portion of the cluster, verify the Windows side is configured and can pass the Windows HPC Cluster Diagnostic tests using the HPC Cluster Manager, and synchronize the Linux and Windows clocks by using the `ntpdate` command on the Linux system.

```
ntpdate windows.headnode.ip.address
```

Deploy the Linux head node, configure static IP addresses, and disable IPV6.



Turn off IPv6 support, both on the head node and compute nodes, if you are not going to use it as it might conflict with your setup.



You must select the development libraries during the OS installation to prevent the Perl module builds from failing.

### 1.2.1 CentOS

1. Create the necessary files and directories for kickstart deployment.

```
mkdir -p /data/network-install/ISO
mkdir -p /data/network-install/kickstart
mkdir -p /data/network-install/RPM
```

2. [Download the CentOS ISO](#) to `/data/network-install/ISO`.
3. Copy or mount the ISO to `/data/network-install/RPM`. Copying the ISO is recommended, because this must be available each time you deploy a new Linux node.

```
mount -o loop /data/network-install/ISO/CentOS-5.3-x86_64-bin-DVD.iso
/data/network-install/RPM
```

4. Choose your compute node's disk partitioning scheme. The `diskpart.txt` (located in the Windows head node), `ks.cfg`, and `tftpboot` files are set up differently depending on the number of disks used.
5. Create the kickstart file at `/data/network-install/kickstart/ks.cfg`. Edit the path to reflect your setup. The following sample assumes there is a single disk and that any existing Linux partitions should be eliminated. It creates three new partitions: one for `/boot`, one for `/`, and one for `swap`.

#### Sample for single disk systems:

`/data/network-install/kickstart/ks.cfg`:

```
install
url --url http://192.168.3.149/network-install/RPM
lang en_US.UTF-8
keyboard us
network --device eth0 --bootproto dhcp
rootpw --iscrypted $1$9pyYDTug$cUEbn4xP1Tdjxs7nj2IFl0
firewall --disabled
authconfig --enableshadow --enablemd5
selinux --enforcing
timezone --utc America/Denver
clearpart --linux
bootloader --location=partition
part /boot --fstype ext3 --size=100 --ondisk=sda --asprimary
part / --fstype ext3 --size=1024 --grow --ondisk=sda --asprimary
part swap --size=128 --grow --ondisk=sda --asprimary
reboot
```

```

%packages
@editors
@text-internet
@legacy-network-server
@dns-server
@core
@base
@ftp-server
@network-server
@server-cfg
device-mapper-multipath

```

### Sample for dual disk systems:

/data/network-install/kickstart/ks.cfg:

```

install
url --url http://192.168.3.149/network-install/RPM
lang en_US.UTF-8
keyboard us
network --device eth0 --bootproto dhcp
rootpw --iscrypted $1$9pyYDTug$cUEbn4xP1Tdjxs7nj2IFl0
firewall --disabled
authconfig --enableshadow --enablemd5
selinux --enforcing
timezone --utc America/Denver
ignoredisk --drives=sda
clearpart --all --drives=sdb --initlabel
bootloader --location=partition
part /boot --fstype ext3 --size=100 --ondisk=sdb --asprimary
part / --fstype ext3 --size=1024 --grow --ondisk=sdb --asprimary
part swap --size=128 --grow --size=256 --ondisk=sdb --asprimary
reboot
%packages
@editors
@text-internet
@legacy-network-server
@dns-server
@core
@base
@ftp-server
@network-server
@server-cfg
device-mapper-multipath

```

6. Configure Apache to share /data/network-install/.

```

...
DocumentRoot "/data/"
<Directory /data/network-install>
  Options +Indexes
  AllowOverride AuthConfig
  order allow,deny
  allow from all
</Directory>
...

```

7. Copy /data/network-install/RPM/isolinux/vmlinuz to /tftpboot/vmlinuz.
8. Copy /data/network-install/RPM/isolinux/initrd.img to /tftpboot/initrd.img.
9. Create the appropriate tftpboot kickstart boot configuration file. Direct the http address to the head node's internal IP address.

/tftpboot/pxelinux.cfg/kickstart:

---

```
DEFAULT ks
PROMPT 0
TIMEOUT 30
LABEL ks
  kernel vmlinuz
  append text initrd=initrd.img ramdisk_size=8192 ip=dhcp
ks=http://10.0.0.200/network-install/kickstart/ks.cfg ksdevice=eth0
```

## 1.2.2 SUSE Linux Enterprise Server

1. Create the necessary files and directories for autoyast deployment.

```
mkdir -p /data/network-install/ISO
mkdir -p /data/network-install/autoyast
mkdir -p /data/network-install/RPM
```

2. [Download the SUSE ISO](#) to /data/network-install/ISO. Only the first ISO is necessary for a network installation.
3. Copy or mount the ISO to /data/network-install/RPM. Because this must be available each time you deploy a new Linux node, copying the ISO is recommended.

```
mount -o loop /data/network-install/ISO/SLES-11-DVD-x86_64-GM-DVD1.iso /data/network-install/RPM
```

4. Choose your compute node's disk partitioning scheme. The diskpart.txt (located in the Windows head node), ay.cfg, and tftpboot files are set up differently depending on the number of disks used.
5. Create the autoyast file in /data/network-install/autoyast/autoyast.xml. Edit the path to reflect your setup. The following sample assumes there is a single disk and that any existing Linux partitions should be eliminated. It creates three new partitions: one for /boot, one for /, and one for swap.

### Sample for single disk systems:

/data/network-install/autoyast/autoyast.xml:

```
<?xml version="1.0"?>
<!DOCTYPE profile SYSTEM "/usr/share/autoinstall/dtd/profile.dtd">
<profile xmlns="http://www.suse.com/1.0/yast2ns"
  xmlns:config="http://www.suse.com/1.0/configns">
  <configure>
    <networking>
      <dns>
        <dhcp_hostname config:type="boolean">true</dhcp_hostname>
        <dhcp_resolv config:type="boolean">true</dhcp_resolv>
      </dns>
      <interfaces config:type="list">
        <interface>
          <bootproto>dhcp</bootproto>
          <device>eth0</device>
          <startmode>onboot</startmode>
        </interface>
      </interfaces>
      <modules config:type="list">
        <module_entry>
          <device>static-0</device>
          <module></module>
          <options></options>
        </module_entry>
      </modules>
    </networking>
  </configure>
  <install>
    <bootloader>
      <global>
```

6. Configure Apache to share /data/network-install/.

```
...
DocumentRoot "/data/"
<Directory /data/network-install>
  Options +Indexes
  AllowOverride AuthConfig
  order allow,deny
  allow from all
</Directory>
...
```

7. Copy /data/network-install/RPM/boot/x86\_64/loader/linux to /tftpboot/vmlinuz.
8. Copy /data/network-install/RPM/boot/x86\_64/loader/initrd to /tftpboot/initrd.img.
9. Create the appropriate tftpboot autoyast boot configuration file. Direct the http address to the head node's internal IP address.

/tftpboot/pxelinux.cfg/autoyast:

```
DEFAULT ay
PROMPT 0
TIMEOUT 30
LABEL ay
  kernel vmlinuz
  append text initrd=initrd.img ramdisk_size=8192 ip=dhcp
install=http://10.0.0.200/network-install/rpm/
autoyast=http://10.0.0.200/network-install/autoyast/autoyast.xml
```

## 1.3 Installing MSMHPC Tools



In order to use the scripts provided by MSMHPC Tools, you must have the following Perl modules installed (on the same node where Moab is located):

- LWP
- Crypt::SSLeay
- SOAP::Lite (Required for communicating with the MSMHPC Web Service)

To install the MSMHPC tools, do the following:

1. Obtain the latest MSMHPC\_tools tarball (MSMHPC\_tools.r<version>.zip). The tarballs are located at:

<http://www.clusterresources.com/product/hybrid/snapshots>

Contact your Adaptive Computing account representative if your clusterresources.com account does not have permission to access the download package.

2. Untar the MSMHPC\_tools tarball into your Moab tools directory, which by default is `/opt/moab/tools`.

```
> tar -xvzf <file name>
```

3. Verify you have **at least** the following files in the Moab tools directory:

```
tools/  
cluster.query.hpc.pl  
env.hpc.example  
import.node.xml.hpc.pl  
job.cancel.hpc.pl  
job.start.hpc.pl  
job.submit.hpc.pl  
job.requeue.hpc.pl  
os.switch.pl.grub //If using the GRUB bootloader, rename this file  
to os.switch.pl  
os.switch.pl.pxe //If using the PXE bootloader, rename this file to  
os.switch.pl  
os.switch.xcat.pl  
recache.nodes.hpc.pl  
user.cache.pl  
user.cache.secure.pl  
workload.query.hpc.pl  
  
tools/grub/ //Only necessary if using GRUB bootloading  
bootccs.bat  
bootccs.sh  
boothpc.bat  
boothpc.sh  
bootrhel.bat  
switch.grub.pl  
  
tools/Moab/  
MSMHPC.pm  
Tools.pm
```

4. Copy the `chain.c32` and `pxelinux.0` files from `msmhpc-linux/moab_tools/pxe/tftpboot/` to `/tftpboot`. For more information, see [TFTP Server and PXE Booting](#).
5. Configure Moab. (See [Moab Configuration](#) for more information.)
6. Append the configuration files (from the Windows head node) created by MSMHPC Manager to `moab.cfg` and `moab-private.cfg` respectively.

7. When the MSMHPC Service is running, start Moab.
8. Run **mdiag -R -v** and verify that the resource manager shows up and has no errors.

## 1.4 Deploying Compute Nodes

1. Create the Linux and Windows tftpboot configuration files.

### Sample files for single disk systems:

Windows:

```
DEFAULT windows
PROMPT 0
TIMEOUT 30
LABEL windows
  KERNEL chain.c32
  APPEND hd0 1
LABEL linux
  KERNEL chain.c32
  APPEND hd0 2
```

Linux:

```
DEFAULT linux
PROMPT 0
TIMEOUT 30
LABEL windows
  KERNEL chain.c32
  APPEND hd0 1
LABEL linux
  KERNEL chain.c32
  APPEND hd0 2
```

### Sample files for dual disk systems:

Windows:

```
DEFAULT windows
PROMPT 0
TIMEOUT 30
LABEL windows
  KERNEL chain.c32
  APPEND hd0 1
LABEL linux
  KERNEL chain.c32
  APPEND hd1 1
```

Linux:

```
DEFAULT linux
PROMPT 0
TIMEOUT 30
LABEL windows
  KERNEL chain.c32
  APPEND hd0 1
LABEL linux
  KERNEL chain.c32
  APPEND hd1 1
```

2. Set up the `env.hpc` shipped with the `MSMHPC_tools` to reflect your environment.

`env.hpc`

```
export RMNAME=MSMHPC
export PUBKEY=user
export DOMAIN=SGE
export PROXY=http://WINDOWSHEADNODE:5343/MSMHPC
export PRIVKEYFILE=/root/moab-private.cfg
```



The DOMAIN variable does not need to be the full domain name, but should match the login domain, such as DOMAIN\user.

moab-private.cfg:

```
CLIENTCFG[RM:MSMHPC] KEY=moab
```

3. Run the `deploy.hpc.node.pl` script to deploy a new node.

```
> . ./env.hpc
> /root/msmhpc_tools/msmhpc/scripts/deploy.hpc.node.pl
USAGE: ./deploy.hpc.node.pl <NODENAME> <MACADDRESS> <NODE TEMPLATE>
<DOMAIN NAME>
ie: ./deploy.hpc.node.pl node01 AA:BB:CC:DD:EE:FF "Default
ComputeNode Template" MYDOMAIN at
/root/msmhpc_tools/msmhpc/scripts/deploy.hpc.node.pl line 26.
> /root/msmhpc_tools/msmhpc/scripts/deploy.hpc.node.pl node01
00:50:56:35:bb:cc "Default ComputeNode Template" SGE
```

4. Check in the HPC Cluster Manager to see if the node is ready to be provisioned.
5. Start the compute node, ensuring that it is set up to PXE boot. The compute node starts deploying Windows.
6. When Windows finishes deploying, change the `dhcpd.conf` file so that the Linux deployment can start.

```
host node01 {
fixed-address      10.0.0.101;
hardware ethernet 00:50:56:35:bb:cc;
option host-name  "node01";
next-server       10.0.0.200;
filename         "pxelinux.0";
#next-server     10.0.0.100;
#filename        "Boot\\x64\\WdsNbp.com";
#option domain-name-servers 10.0.0.100;
}
```

7. Restart the DHCPD service.
8. Create a symlink for the compute node's address. Point it to a file with the same name as the compute node, and point that file to the kickstart or autoyast tftp boot file.

```
ln -sf NODE01 01-00-50-56-35-bb-cc
ln -sf kickstart NODE01
```

```
ln -sf NODE01 01-00-50-56-35-bb-cc
ln -sf autoyast NODE01
```



Note that the 01 at the beginning of the mac address is needed for PXE booting.

9. Reboot the compute node to start the Linux deployment. You can immediately change the configuration so that the node boots to Linux when the deployment finishes.

```
ln -sf linux NODE01
```



To enable centralized authentication, give the Linux host a different name from the Windows host name. This allows both operating systems to be joined simultaneously in the domain. To do this:

1. Name the Windows node `node01` and the Linux node `node011`.
2. Create the necessary aliases in `/etc/hosts` directory of the machine that runs the `pbs_server`

daemon.

3. Include only the Windows name in the `/var/spool/torque/server_priv/nodes` file so that TORQUE reports the same node name and Moab can map the cluster state on all the resource managers.

## 1.5 Installing TORQUE

For instructions on how to install TORQUE, point your browser to the following URL:

<http://www.adaptivecomputing.com/resources/docs/torque/a.torquequickstart.php>

Ensure that the resource managers on both operating systems are set to start on bootup. For example, make sure the pbs\_mom init script is installed and that it has been added to the default run level. It is also helpful to set the polling interval on polling resource managers fairly low. The more responsive the resource managers are, the more responsive Moab can be.



Setting the polling intervals low increases the CPU load. To minimize the problem, set the value between 15 and 20 seconds.

## 1.6 Test Your Configuration

1. Before a user can submit jobs, the user's credentials must be cached on the HPC head node, which allows the MSMHPC Service to authenticate as the user when it submits the job to HPC 2008 R2. Credential caching only needs to be done once for each user. Then the user can submit jobs normally with the **msub** command. To cache user credentials, use the `user.cache.pl` script provided with the MSMHPC tools. The syntax is `./user.cache.pl <username> <password>`. Note that if you do not have SSL enabled, these credentials will be sent in plain text over the network.

End users can also use the `user.cache.secure.pl` script, the MSMHPC Cache Credentials command-line client, or the MSMHPC Cache Credentials GUI application to cache their own credentials. Provide users with the **Public Shared Key** for authentication to MSMHPC. Under Linux, the `user.cache.secure.pl` script is the recommended method as it does not show the user's credentials in plain text in the output of `ps -a` on the head node.

2. Users can use the **msub** command normally to submit jobs. Submit a test job to test the system. Note that all executables called must exist on the Windows compute nodes or the script will fail. For example:

```
> echo ping -n 300 localhost | msub -l walltime=300,os=windows
```

3. Now verify that the job successfully migrated to HPC 2008 R2. On the HPC 2008 R2 head node, open HPC Cluster Manager (found in **Start**→**All Programs**→**Microsoft HPC Pack**→**HPC Job Manager**). You should see the newly submitted job.
4. Verify that Moab starts the job and that the job's state changes to **Running**. RDP (remote desktop) into the node that is running the job (which you can determine with **checkjob**) and open task manager. Verify that the job is running as the correct user. When the job finishes, the stdout and stderr files should be staged back to the user's home directory (if Samba is configured) or the the shared directory on the HPC head node. Verify that these files are present and have the correct contents.



You can also check where the job is running, and under which user, by going to the HPC Cluster Manager and checking the job properties for a job.

# 1.7 Configuring SSL for Adaptive HPC

- [1.7.1 Windows Head Node Setup](#)
- [1.7.2 Linux Head Node Setup](#)
- [1.7.3 Troubleshooting](#)

## 1.7.1 Windows Head Node Setup

In order to use a certificate, you must install it using the netsh tool in a Windows command prompt with administrator privileges.

### 1.7.1.1 Installing a Certificate from a Known Authority

If you use a certificate from a known certificate authority, it can be used directly by doing the following:

1. Open the Certificates MMC Snap-in.
  1. Click on Start, then Run...
  2. Type "mmc" and click OK. The MMC console window will appear.
  3. Click on File, then Add/Remove Snap-in...
  4. Select Certificates, click the Add button, and choose Computer account. Click Next and then Finish. Click OK on the Add or Remove Snap-ins page.
2. Using the Certificates MMC Console, find the certificate and open it, then click the Details tab.
  1. Copy the Thumbprint value and remove the spaces. This is the certificate hash.
3. Select the proper folder or store location for the certificate. Right click on it, go to All Tasks, and click Import.
4. Follow the wizard to import the certificate.
5. Skip to step 4 below and run the netsh command with the certificate hash provided in the Certificates details view.

### 1.7.1.2 Installing a Self-signed Certificate



In order to run the `makecert` and `certutil` commands, you must [download Windows SDK](#) version 6.1 or later and install them. Use the SDK Command Prompt or CMD Shell to run the tools. To do so, click Start, hover over All Programs, then Microsoft Windows SDK, and click SDK Command Prompt or CMD Shell.

If you generate and use a self-signed certificate, you must do the following:

1. First run the following the command and enter a private key to generate the root certificate authority (CA):

```
makecert -n "CN=RootCA" -r -sv RootCA.pvk RootCA.cer
```

RootCA is a unique, arbitrary name.

2. Install or import the RootCA.cer certificate into the Trusted Root Authorities certificate store by following the directions above for the Certificates MMC console.
3. Run the following commands to generate a self-signed certificate from the certificate authority.

```
makecert -sk machinekey -iv RootCA.pvk -n "CN=machine" -ic RootCA.cer  
-sr localmachine -ss my -sky exchange -pe  
certutil -store my machine
```

machinekey is a unique, arbitrary key name, and machine is the DNS name of the Windows head

node.

4. Copy the hash key from the output of certutil and remove the spaces. This will be the certhash parameter for the following netsh command. The appid parameter is unique to MSMHPC and must be copied exactly.

```
netsh http add sslcert ipport=0.0.0.0:5345
certhash=8e853e4e2fcd9c70e35f38fb1659c55941d43e9c appid={c7263768-
9bba-4efc-b851-07b1ea218b1e}
```



The port specified in the above example must match the configured server port, or SSL will not work correctly.

## 1.7.2 Linux Head Node Setup

Once the certificate is installed on the Windows head node correctly, the setup on the Linux side is minimal. If you use a common certificate signed by a real certificate authority (CA), it should work on its own. If you use a self-signed certificate or custom CA, you must install the CA certificate into the certificate store on the Linux head node by doing the following:

1. Convert the certificate from the DER file format (cer or crt files) that makecert uses into a PEM file format using the following command:

```
openssl x509 -in RootCA.crt -inform DER -out RootCA.pem -outform PEM
```



The openssl library is required for SSL configuration.

2. Copy the RootCA.pem file into the correct location for the Linux distribution.



The location is different for each platform. For CentOS it's at `/etc/pki/tls/certs`.

3. In MSMHPC tools, remove the comment brackets from the lines setting the environment settings for openssl (they should be in the first block of code) in Moab/MSMHPC.pm:

```
# Set to the correct root CA PEM file if using a self-signed
certificate
$ENV{HTTPS_CA_FILE} = 'certs/RootCA.pem';
$ENV{HTTPS_CA_DIR} = 'certs/';
```

4. Verify that the RootCA.pem file location is set correctly. The file name `RootCA` is arbitrary.

## 1.7.3 Troubleshooting

If you encounter problems using the certificate on the Linux side, uncomment the `$ENV` line in Moab/MSMHPC.pm in MSMHPC tools:

```
# Use for debugging HTTPS connections (openssl)
#$ENV{HTTPS_DEBUG} = 1;
```

This will show what is occurring with openssl to allow troubleshooting.

## 2.0 Preparing the Linux Head Node

The following explains how to set up the Linux side of a Windows HPC 2008 R2/Moab cluster. Before setting up the Linux portion of the cluster, verify the Windows side is configured, and synchronize the Linux and Windows clocks by using the `ntpdate` command on the Linux system.

```
ntpdate windows.headnode.ip.address
```

- [2.1 Installing Moab](#)
- [2.2 Configure Samba](#)

## 2.1 Installing Moab

A Moab 6.1 hybrid build or later is required for setting up the hybrid system. Moab installation packages are readily available for all major architectures and operating systems. Full or evaluation licenses will work with any build.

Versions that work together include:

- For Windows HPC Server 2008 R1: [Moab 5.4.1](#), [MSMHPC 5.4.0.3](#), and [MSMHPC tools-5.4.0.131](#)



Windows HPC Server 2008 R1 is not being developed and is not compatible with Moab 6.0 or higher.

- For Windows HPC Server 2008 R2 SP1: [Moab 6.1](#), [MSMHPC tools-6.1](#), and [MSMHPC 6.1](#)

A provisioning-capable hybrid Moab license is also required in either case.

The following example shows the commands to do a basic installation from the command line for Moab. In this case, the install package for Moab 6.1 needs to be in the current directory.

```
> tar xvzf moab-6.1-linux-x86_64-torque.tar.gz
> cd Moab
> ./configure
> make install
> chmod 1777 /opt/moab/log
```

By default, Moab installs everything to the `/opt/moab` directory (pre-Moab 6.0 versions install binaries to `/usr/local`). The following example shows a sample `ls -l` output in `/opt/moab`.

```
drwxr-xr-x 2 root root 4096 2010-01-22 12:26 etc
drwxr-xr-x 2 root root 4096 2010-01-22 12:26 log
drwxrwxrwt 2 root root 4096 2010-01-22 12:42 spool
drwxr-xr-x 2 root root 4096 2010-01-22 12:26 stats
drwxr-xr-x 2 root root 4096 2010-01-22 12:26 tools
```

Note that the installation creates a default `moab.cfg` file in the `etc/` folder. This file contains the global configuration for Moab that is loaded each time Moab is started. The definitions for users, groups, nodes, resource managers, quality of services, and standing reservations are placed in this file. The default `moab.cfg` file provided with an installation is simple. The installation process defines several important default values, but the majority of configuration needs to be done by the administrator, either through directly editing the file or using one of the provided administrative tools such as Viewpoint.

The following shows the default `moab.cfg` file without comments:

```
SCHEDCFG[MyMoab]  SERVER=master:42559
ADMINCFG[1]      USERS=root
RMCFG[Torque]    TYPE=PBS
```

The first line of the configuration file defines a new scheduler named `MyMoab`. In this case, it is located on a host named `master` and listening on port `42559` for client commands. These values are added by the installation process, and should not be modified in most cases.

The second line, however, requires some editing by the administrator. The example on the second line specifies what users on the system have level 1 administrative rights—users who have global access to information and unlimited control over scheduling operations in Moab. Additional users can be added in a comma-separated list. Moab attempts to run as the first level 1 administrator, so `root` should not be removed from its position at the beginning of the list. Moab must run as `root` to submit jobs to the resource managers as the original owner. There are five default administrative levels defined by Moab, each of which is fully customizable.

The final line in this example is the configuration for the default resource manager. This particular distribution is for the TORQUE resource manager. Because TORQUE follows the PBS style of job handling, the resource

manager is given a type of PBS. To differentiate it from other resource managers that may be added in the future, it is also given the name Torque.



To increase system security, configure a security key that authenticates client commands to Moab. To do so, create a `.moab.key` file inside your Moab home directory (`/opt/moab` by default). Run `chmod 400 .moab.key` and start Moab. For more information, see [these detailed instructions](#).

This constitutes the basic installation of Moab. Many additional parameters can be added to the `moab.cfg` file to fully adapt Moab to your needs. A more detailed installation guide is available at the following URL:

<http://www.adaptivecomputing.com/resources/docs/mwm/2.0installation.php>

## 2.2 Configuring Samba

- [2.2.1 Configuring Moab to Store STDERR/STDOUT Files in a Samba Share](#)

Configuring Samba is optional. Moab will send  $\$HOME/dir$  as the user's working directory to MSMHPC. MSMHPC will look up the user's home directory in Active Directory and translate  $\$HOME$  to that directory.

MSMHPC will attempt to start the job in the same directory where it was submitted on the Linux head node (usually the user's home directory). If you want, you can specify another directory via **msub** at job submission time. To run the job correctly, you must export your home directory and other common directories to the HPC cluster.

Alternatively, you can mirror your directory structure on the HPC head node. Note, though, that if you follow this process, files you may need to run jobs might not be available. To mirror the directory structure, create a tree of directories on the HPC head node mirroring your home directory and share the new directory with the name of the root directory.

For example, suppose there is a user named *test*. In *test*'s home directory there is a directory named *jobs* that *test* uses to submit jobs. The full path of the directory must be created on the Windows head node. Create a folder named *home* at the root, `C:\home`, for example. Within the *home* folder, create another folder named *test* and another folder inside that one named *jobs*. Doing so yields the full path `\home\test\jobs`, the same as on Linux. The *home* directory would then need to be shared as *home*.

To configure Samba, do the following on the head node:

1. Install a Samba server, configure the service to run on startup, and change the Samba configuration file that is typically located at `/etc/samba/smb.conf`. For Red Hat and SLES, do the following to configure the service to run on startup:

```
chkconfig --level 3 5 smb on
```

2. Open the `smb.conf` file and ensure that the following line is included (where you replace `<ACTIVE DIRECTORY DOMAIN NAME>` with your active directory name):

```
workgroup = <ACTIVE DIRECTORY DOMAIN NAME>
```

3. Add a "home" share so that the Windows nodes can create output files at the user's home directory; to do so, add the following to the end of the `smb.conf` file:

```
[home]
comment = home
path = /home
browseable = yes
read only = no
```

4. Start Samba and set up users on all of the Windows and Linux systems so that they all have the same user ID, group ID, password, and home directory. To add users to Samba, issue the following command and type in the same password for the Windows and Linux systems:

```
smbpasswd -a username
```

### 2.2.1 Configuring Moab to Store STDERR/STDOUT Files in a Samba Share

An administrator can configure Moab to store STDERR/STDOUT files in a Samba share, instead of on the compute node. To do so, follow these steps:

1. Click **Start**, **Administrative Tools**, and then **Active Directory Users and Computers**
2. Double-click the user you want to modify.
3. Click the **Profile** tab.
4. Change the **Home folder** settings to a network drive associated as the user's home directory (by pointing to the user's remote shared address).

**test1 Properties** [?] [X]

Member Of	Dial-in	Environment	Sessions		
Remote control	Terminal Services Profile		COM+		
General	Address	Account	Profile	Telephones	Organization

User profile

Profile path:

Logon script:

Home folder

Local path:

Connect:  I.o:

OK Cancel Apply Help

## 3.0 PXE Booting

Instructions for automating dual-boot operating system switching assume that you are using a PXE boot setup, including a DHCP server and TFTP server on the Linux head node. It is easiest to set up dual-booting via PXE if the cluster has its own network segment and subnet (to avoid potential DHCP conflicts).



You should set up dual booting only for compute nodes.

To automate dual-boot operating system switching during the Linux installation, do the following:

- Install the operating systems on each compute node in this order:
  1. Windows
  2. Linux
- [3.1 Configuring a DHCP Server](#)
- [3.2 TFTP Server and PXE Booting](#)
- [3.3 Provisioning OSES with xCAT](#)
  - [3.2.1 Switching Operating Systems with xCAT](#)

## 3.1 Configuring a DHCP Server

Install a DHCP server on the Linux head node. Then, to configure the server, do the following:

1. Configure the service to run on startup, and change the DHCP Server configuration file (typically located at `/etc/dhcp3/dhcp.conf` or `/etc/dhcp.conf`). For Red Hat and SLES, run the following command to configure the service to run on startup:

```
> chkconfig --level 3 5 dhcpd on
```

2. Open the `dhcp.conf` file and configure preferred options.
3. Add a "host" configuration section for each host, or for larger ranges of machines or IP address pools, consult the documentation for the DHCP server (`man dhcpd`).

The following is a sample DHCP Server configuration file:

```
default-lease-time 600;
max-lease-time 7200;
ddns-update-style ad-hoc;

subnet 192.168.0.0 netmask 255.255.255.0 {
interface eth0;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.0.255;
option routers 192.168.0.254;
option domain-name-servers 192.168.0.1;
option domain-name-servers 192.168.0.1, 192.168.0.2;
option domain-name "server.example.com";
range 192.168.0.100 192.168.0.200;
}

host node001 {
#hardware MAC address
hardware ethernet 00:03:47:43:3F:73;

# this is the unused IP address we will assign temporarily to the
PXE client
fixed-address 192.168.0.90;

#TFTP server address (should be same as Linux head node)
next-server 192.168.0.1;

# path of the bootloader file, relative to tftpd's root (usu.
/tftpboot)
```



If your DNS configuration is pointing anywhere other than the Windows head node, you will see unexplained errors in your HPC/WMI configuration. If you intend to use any other DNS server (Windows or Linux), you must set up forwarding zones so lookups to the Windows Domain Controller are answered correctly.

## 3.2 TFTP Server and PXE Booting

- 3.2.1 Switching Operating Systems with xCAT

Install a TFTP server. Then, to set up PXE booting, do the following:

1. Download the SYSLINUX binaries from the following location:  
<http://www.kernel.org/pub/linux/utils/boot/syslinux/>



These binaries are also included with MSMHPC Tools. See [Installing the MSMHPC Tools](#) for more information.

2. Unpack at least chain.c32, pxelinux.0, and pxelinux.cfg/default in the TFTP server root.



chain.c32 will not work with a mismatching version of pxelinux.0. If you are using automated deployment tools, replace pxelinux.0 with the latest version or find the matching version of chain.c32 for your version of SYSLINUX.

The default PXE boot configuration file is pxelinux.cfg/default, a sample of which follows:

```
DEFAULT windows
PROMPT 0
TIMEOUT 100
LABEL windows
    KERNEL chain.c32
    APPEND hd0 1
LABEL linux
    KERNEL chain.c32
    APPEND hd0 0
```

3. The "Linux" label boots the MBR bootloader, which should be set to boot into Linux by default. The "Windows" label boots the OS on the first partition. The previous example defaults to Windows. Reboot the machine and test PXE booting to ensure proper function. Then, modify the partitions as needed to reflect your setup.
4. After confirming that you can PXE boot both partitions by switching the DEFAULT, create two files in pxelinux.cfg/: "windows" and "linux". Make a list of MAC addresses for the compute nodes. Create a symlink from each node to one of the OSes it will boot:

```
ln -fs windows NODE001
ln -fs linux NODE002
ln -fs linux NODEXXX
```

5. Create a symlink from each MAC address (that you will boot) to its hostname (note: prepend the MAC address with 01- and use lowercase A-F characters):

```
ln -fs NODE001 01-00-03-47-43-3f-73
ln -fs NODEXXX 01-00-03-47-43-4f-92
```

6. Add a provisioning resource manager to the Moab configuration file (moab.cfg):

```
#####
#
# OS Switching Resource Manager
#
#####
RMCFG[prov] TYPE=NATIVE RESOURCETYPE=PROV
# This signifies the resource manager is a provisioning RM.
RMCFG[prov]
```

```

ENV=OSSTRING=windows;RMNAME=HPC;PUBKEY=moabpublickey;DOMAIN=HPCDOMAIN;P
# RMNAME refers to the RM configured in 4.0.
# More information for this line is found in the moab.cfg file
generated in section 1.1.1.

RMCFG[prov] PROVDURATION=5:00
# This tells Moab how long it takes the node to reboot.

RMCFG[prov] NODEMODIFYURL=exec://$HOME/scripts/os.switch.pl
# This is the script Moab calls to switch the operating system.

```

Example os.switch.pl:

```

#!/usr/bin/perl
#
# Copyright (c) 2007-2010 Adaptive Computing Enterprises, Inc.
#
use strict;
use warnings;

use FindBin;

use lib "$FindBin::Bin";

use Moab::MSMHPC;

my ($host, undef, $newos, undef) = @ARGV;
my $os;
my $pxe_cfg_dir = '/tftpboot/pxelinux.cfg';

if (defined $newos)
{
    (undef, $os) = split '=', $newos;
}

die "Usage: os.switch.pl --set OS=\n"
    . "    os: 'linux' or 'windows'"
    unless defined $host and defined $os;

$host = uc($host);

```

The script is called when Moab needs to change the operating system on a node. It changes the operating system running on a node by taking the destination operating system and compute node name as shell parameters.

Modify the examples supplied in this section as needed.

- After you install MSMHPC on the Windows head node and the MSMHPC Tools Perl scripts in the "scripts" directory of your Moab installation, verify PXE booting by running `os.switch.pl NODENAME -set OS=windows` (or `linux`) for each node.



To test switching from Windows to Linux, you need to fully configure Moab for switching. See section 3.0.

## 3.2.1 Switching Operating Systems with xCAT

The optional `os.switch.pl.xcat` script included in MSMHPC tools can be used in place of the PXE boot process, allowing you control MSMHPC using xCAT commands. The script only works with a full xCAT setup, including TFTP and DHCP servers. xCAT 2.6x is recommended.

To configure xCAT to work correctly with the `os.switch.pl.xcat` script, do the following:

1. Install xCAT and define all your nodes in it.
2. Deploy Windows/Linux on the nodes.
3. Copy `chain.32` from to `/tftpboot`. To do so:
  1. Locate the `chain.c32` file included in the MSMHPC tools.
4. Add a `nodes.netboot` entry to make the node boot via PXE instead of XNBA. A sample looks like this:

```
"vm4" , , "pxe" , "000.00.00.000" . . .
```

5. Add the OS entries to the `boottargets` table in `nodes.netboot`, adapting it to your disk or partition layout.
6. Ensure that the `os.switch.pl.xcat` script is copied into the `$MOAB_HOME/tools` directory and is configured in `moab.cfg` as the `NODEMODIFYURL` for the provisioning resource manager. See the [Provisioning and Load Balancing](#) documentation for more information.

## 4.0 Moab Configuration



Copy the `moab-private.cfg` file you generated in [1.1 Microsoft Windows HPC Server 2008](#) to `/opt/moab/etc` so Moab can load MSMHPC's private key.

The following is a sample Moab configuration file (`moab.cfg`) that is configured for a hybrid environment:

```
# Scheduler configuration
SCHEDCFG[moab]      SERVER=moab:42559
SCHEDCFG[moab]      MODE=NORMAL
SCHEDCFG[moab]      FLAGS=ALLOWMULTICOMPUTE
NOLOCALUSERENV      TRUE
DISPLAYFLAGS        SHOWSYSTEMJOBS

# Logging
LOGFILE              moab.log
LOGLEVEL              1
LOGFILEMAXSIZE        10000000
LOGFILEROLLDEPTH      7

RMPOLLINTERVAL 15
DEFERTIME 60

# Primary admin must be first in admin1 user list
# <http://adaptivecomputing.com/resources/docs/mwm/a.esecurity.php>
ADMINCFG[1]          USERS=root

#Resource Manager configuration
RMCFG[torque]        TYPE=PBS
RMCFG[torque]        PARTITION=local
RMCFG[torque]        NODESTATEPOLICY=OPTIMISTIC
RMCFG[torque]        DEFOS=linux
RMCFG[torque]        FLAGS=USERSPACEISSEPARATE

RMCFG[prov]          TYPE=NATIVE RESOURCETYPE=PROV
RMCFG[prov]
```



To change the `PROXY` variable from its default port 5343, use the MSMHPC manager on the Windows head node in the **Configuration Page** → **Server Port** → **Save Settings**.



The `DOMAIN` variable does not need to be the full domain name, but should match the login domain, such as `DOMAIN\user`.

- [4.1 Scheduler Configuration](#)
- [4.2 On-Demand Provisioning](#)
- [4.3 Provisioning & Load Balancing](#)
  - [4.3.1 Switching from Dual to Single OS Provisioning](#)
  - [4.3.2 Configuring Multiple Operating Systems in Windows](#)
- [4.4 Resource Manager Configuration](#)
  - [4.4.1 Linux Resource Manager Configuration](#)
    - [4.4.1.1 TORQUE Setup](#)
    - [4.4.1.2 Sun Grid Engine Configuration](#)
  - [4.4.2 Windows Resource Manager Configuration](#)

## 4.1 Scheduler Configuration

```
SCHEDCFG [moab]      SERVER=moab:42559
SCHEDCFG [moab]      MODE=NORMAL
SCHEDCFG [moab]      FLAGS=ALLOWMULTICOMPUTE
NOLOCALUSERENV      TRUE
DISPLAYFLAGS        SHOWSYSTEMJOBS
```

Some parameters are needed to make OS tracking work correctly. `FLAGS=ALLOWMULTICOMPUTE` must be set on the scheduler. For example:

```
SCHEDCFG [moab]      FLAGS=ALLOWMULTICOMPUTE
```

Moab does not automatically validate all user accounts or directories on the head node. If `NOLOCALUSERENV` is set to `TRUE`, Moab relies on the resource manager to validate user accounts and directories.

If the administrator would like to have separate user spaces between operating systems, for example having Linux users that do not exist in active directory, be sure to assign the `USERSPACEISSEPARATE` flag to each of the resource managers.

Also, please note that it is required to place all the compute nodes in the same partition. In the current example, the partition `local` was used, but any name can be assigned to this partition. This change affects each of the resource managers as well as every one of the compute nodes.

By default in Moab 5.4 and later, system jobs (such as provisioning jobs) do not display when you run the `showq` command. Setting the `DISPLAYFLAGS` parameter to `SHOWSYSTEMJOBS` enables you to see such jobs when you run `showq`.



Additional information on scheduler configuration is available in the section titled [Installing Moab](#), and in the canonical [Moab Documentation](#).

## 4.2 On-Demand Provisioning

The current model for provisioning control involves the just in time provisioning of nodes according to priorities in Moab's job queue.

To properly configure this provisioning model, please include the following:

```
QOSCFG[ondemand]      QFLAGS=PROVISION
USERCFG[DEFAULT]      QLIST=ondemand
```

## 4.3 Provisioning & Load Balancing

- [4.3.1 Switching from Dual to Single OS Provisioning](#)
- [4.3.2 Configuring Multiple Windows Operating Systems](#)

You must define a provisioning resource manager for Moab to be able to change operating systems on nodes.

Create a provisioning resource manager by adding an RMCFG line in the moab.cfg file. The only attribute it needs is a NODEMODIFYURL that Moab can call to change the operating system on a given node. You can define and adjust PROVDURATION to specify how long the provisioning process takes to finish so that it can schedule around it.

```
RMCFG[prov]           TYPE=NATIVE RESOURCETYPE=PROV
RMCFG[prov]           PROVDURATION=5:00
RMCFG[prov]           NODEMODIFYURL=exec://$TOOLS_DIR/os.switch.pl
```

The script following NODEMODIFYURL should contain the logic necessary to swap the OS of the compute node based on the parameters received by its arguments. Moab calls this script, passing it the following arguments:

```
$NODEMODIFYURL <node id> --set OS=<os>
```

If you use external cluster management software (such as xCAT) rather than a local unmanaged DHCP/TFTP server, you must ensure that the NODEMODIFYURL script obeys the following algorithm:

- Control the PXE boot sequence or change the boot loader to point to the correct partition so that the node boots from the selected OS.
- Connect to the node and reboot it. For Linux systems, set up SSH keys on all the compute nodes so that the os.switch.pl.pxe script can SSH into the compute nodes and reboot them. For Windows systems, the os.switch.pl.pxe script uses the MSMHPC Perl Library (`msmhpc_linux/moab_tools/Moab/MSMHPC.pm`). The library connects to the MSMHPC Service on the Windows head node. The head node then uses the MSMHPC Service to connect to the compute nodes and issue a reboot command.

If you are integrating with xCAT, add all compute nodes to the xCAT tables, use xCAT's nodeset boot command to point the node to the expected OS, and reboot the compute node using the algorithm detailed previously.

Specify node configuration information on the NODECFG lines in the moab.cfg file. The NODECFG line includes the host name of a given compute node and OSLIST, a comma-separated list of the operating systems supported by that particular node. For example:

```
NODECFG[compute000] OSLIST=windows PARTITION=local FEATURES=compute000
NODECFG[compute001] OSLIST=linux PARTITION=local FEATURES=compute001
NODECFG[compute002] OSLIST=linux, windows PARTITION=local
FEATURES=compute002
NODECFG[compute003] OSLIST=linux, windows PARTITION=local
FEATURES=compute003
NODECFG[compute004] OSLIST=linux, windows PARTITION=local
FEATURES=compute004
NODECFG[compute005] OSLIST=linux, windows PARTITION=local
FEATURES=compute005
NODECFG[compute006] OSLIST=linux, windows PARTITION=local
FEATURES=compute006
NODECFG[compute007] OSLIST=linux, windows PARTITION=local
FEATURES=compute007
```

### 4.3.1 Switching from Dual to Single OS Provisioning

If you no longer want Moab to provision multiple operating systems to a compute node, it is not enough to

just change the OSLIST parameter in the Moab configuration file. You must prevent the operating system resource manager from directing Moab to provision multiple operating systems.

#### 4.3.1.1 Removing Linux OS Provisioning

If you are running TORQUE, use the following steps to remove Linux operating system provisioning:

1. Open the moab.cfg file and edit the appropriate NODECFG line. For example, if compute004 is the node you want to run Windows only, remove "linux" from the line so that it reads as follows:

```
NODECFG[compute004] OSLIST=windows PARTITION=local FEATURES=compute004
```

2. Use the **qterm** command to terminate pbs\_server (TORQUE).

```
> qterm
```

3. Remove the node from the nodes file, which is commonly located in /var/spool/torque/server\_priv/nodes.

4. Restart pbs\_server (TORQUE).

```
> pbs_server
```

5. Restart Moab.

#### 4.3.1.2 Removing Windows OS Provisioning

If you are running HPC 2008 R2, use the following steps to remove Windows operating system provisioning:

1. Open the moab.cfg file and edit the appropriate NODECFG line. For example, if compute003 is the node you want to run Linux only, remove "windows" from the line so that it reads as follows:

```
NODECFG[compute003] OSLIST=linux PARTITION=local FEATURES=compute003
```

2. Open HPC Cluster Manager and click **Node Management**.
3. Right-click the specified node in the node list and choose **Take Offline**.
4. After taking the node offline, right-click the node again and choose **Delete**.
5. Launch Moab Services for Microsoft Windows HPC 2008 R2 and click **Configure**.
6. Click **Flush DBs** to make sure the changes made to the HPC cluster manager are immediately recognized by the integration service.

### 4.3.2 Configuring Multiple Operating Systems in Windows

Multiple Windows operating systems can be supported by allowing the environmental variable OSSTRING to set the cluster.query.hpc.pl and os.switch.pl scripts. To do so:

1. Ensure that Moab can identify the different operating systems that each resource manager reports. By default, the MSMHPC service reports OS=windows for all the nodes it manages.
2. Customize your cluster.query.hpc.pl. For instance, remap the OS= wiki parameter supported by Moab with the value from the OSSTRING environmental variable.
3. Modify your moab.cfg file:

```
RMCFG[HPC] TYPE=NATIVE:MSMHPC
RMCFG[HPC] PARTITION=local
RMCFG[HPC] NODESTATEPOLICY=OPTIMISTIC
RMCFG[HPC] DEFOS=windowsA
RMCFG[HPC] FLAGS=USERSPACEISSEPARATE
RMCFG[HPC] ADMINEXEC=jobsubmit
RMCFG[HPC]
ENV=OSSTRING=windowsA;RMNAME=MSMHPC;PUBKEY=myspubkey;DOMAIN=yourdomain

RMCFG[HPC] CLUSTERQUERYURL=exec://$TOOLS_DIR/cluster.query.hpc.pl
RMCFG[HPC] WORKLOADQUERYURL=exec://$TOOLS_DIR/workload.query.hpc.pl
RMCFG[HPC] JOBSUBMITURL=exec://$TOOLS_DIR/job.submit.hpc.pl
RMCFG[HPC] JOBSTARTURL=exec://$TOOLS_DIR/job.start.hpc.pl
RMCFG[HPC] JOBCANCELURL=exec://$TOOLS_DIR/job.cancel.hpc.pl
RMCFG[HPC] JOBQUEUEURL=exec://$TOOLS_DIR/job.requeue.hpc.pl
```

```
RMCFG[HPC2] TYPE=NATIVE:MSMHPC
RMCFG[HPC2] PARTITION=local
RMCFG[HPC2] NODESTATEPOLICY=OPTIMISTIC
RMCFG[HPC2] DEFOS=windowsB
RMCFG[HPC2] FLAGS=USERSPACEISSEPARATE
RMCFG[HPC2] ADMINEXEC=jobsubmit
RMCFG[HPC2]
ENV=OSSTRING=windowsB;RMNAME=MSMHPC;PUBKEY=myspubkey;DOMAIN=yourdomain

RMCFG[HPC2] CLUSTERQUERYURL=exec://$TOOLS DIR/cluster.query.hpc.pl
RMCFG[HPC2] WORKLOADQUERYURL=exec://$TOOLS DIR/workload.query.hpc.pl
RMCFG[HPC2] JOBSUBMITURL=exec://$TOOLS DIR/job.submit.hpc.pl
RMCFG[HPC2] JOBSTARTURL=exec://$TOOLS DIR/job.start.hpc.pl
```

Doing so passes all the configuration values through the environmental variables, so you can use the same set of scripts.

4. Customize your `os.switch.pl` script to reflect your environment's setup. You need a script that can change an OS to a `DESTINATION_OS`. To find the source of the operating system, run `mdiag -n` in Moab, since Moab processes the command from the cache.

## 4.4 Resource Manager Configuration

- 4.4.1 Linux Resource Manager Configuration
  - 4.4.1.1 TORQUE Configuration
  - 4.4.1.2 Sun Grid Engine Configuration
- 4.4.2 Windows Resource Manager Configuration
  - 4.4.2.1 Microsoft HPC Resource Manager Configuration

### 4.4.1 Linux Resource Manager Configuration

It is required that all resource managers support the same classes so Moab knows which classes are supported on which nodes. By default, MSMHPC reports the following queues:

- HIGHEST
- ABOVENORMAL
- NORMAL
- BELOWNORMAL
- LOWEST

Verify that all resource managers have these queues configured.



Queue names are case sensitive.

#### 4.4.1.1 TORQUE Configuration

For instructions on how to install TORQUE, point your browser to the following URL:

<http://www.adaptivecomputing.com/resources/docs/torque/a.torquequickstart.php>

Ensure that the resource managers on both operating systems are set to start on bootup. For example, make sure the pbs\_mom init script is installed and that it has been added to the default run level. It is also helpful to set the polling interval on polling resource managers fairly low. The more responsive the resource managers are, the more responsive Moab can be.

Moab must control walltime instead of TORQUE. For Moab to control the walltime, add a configuration directive to `/var/spool/torque/mom_priv/config` on all the compute nodes with the following:

```
> ignwalltime 1
```

The following additional queues must be configured for TORQUE to integrate with Moab Adaptive HPC Suite:

```
create queue HIGHEST
set queue HIGHEST queue_type = Execution
set queue HIGHEST resources_default.walltime = 01:00:00
set queue HIGHEST enabled = True
set queue HIGHEST started = True

create queue ABOVENORMAL
set queue ABOVENORMAL queue_type = Execution
set queue ABOVENORMAL resources_default.walltime = 01:00:00
set queue ABOVENORMAL enabled = True
set queue ABOVENORMAL started = True

create queue NORMAL
set queue NORMAL queue_type = Execution
set queue NORMAL resources_default.walltime = 01:00:00
set queue NORMAL enabled = True
set queue NORMAL started = True
```

```

create queue BELOWNORMAL
set queue BELOWNORMAL queue_type = Execution
set queue BELOWNORMAL resources_default.walltime = 01:00:00
set queue BELOWNORMAL enabled = True
set queue BELOWNORMAL started = True

create queue LOWEST
set queue LOWEST queue_type = Execution
set queue LOWEST resources_default.walltime = 01:00:00
set queue LOWEST enabled = True
set queue LOWEST started = True

```

To submit jobs to TORQUE that will translate nodes to cores, ensure that TORQUE is aware it has the necessary resources by running the following:

```
qmgr -c 'set server resources_available.nodect = X'
```

Set X to a number greater than or equal to the total number of cores in your system. Failing to do so will cause jobs to fail during submission and produce the following output:

qsub: Job exceeds queue resource limits MSG=cannot locate feasible nodes.

#### 4.4.1.2 Sun Grid Engine Configuration

Refer to the [SGE integration instructions](#) for details on integrating SGE with Moab. The following are additional instructions specific to integrating with Moab Adaptive HPC Suite.

Normal Moab/SGE installs require [adding a complex variable](#) to SGE. The `qconf -mc` command calls the assigned editor; add the following lines:

```

nodelist          nodelist  RESTRING == YES          NO          NONE 0
opsys             os        RESTRING == YES          NO          NONE 0

```

The second step is similar to [example 5](#) in the SGE integration documentation, but needs to reflect the additional complex variable:

```

for i in `qconf -sel | sed 's/\..*//'\`
do
  echo $i
  qconf -rattr exechost complex_values nodelist=$i,opsys=linux $i
done

```

Queues must be configured in SGE. To do so, use the following commands:

```

qconf -aq HIGHEST.q
qconf -aq ABOVENORMAL.q
qconf -aq NORMAL.q
qconf -aq BELOWNORMAL.q
qconf -aq LOWEST.q

```

#### 4.4.2 Windows Resource Manager Configuration

In addition to the default priorities and queues mentioned, optional queues may be configured using job templates. Job templates are configured using the HPC Cluster Manager. Additionally, if you create queues in other resource managers, such as TORQUE or SGE, you must also configure them as job templates in Windows.

To do so, right click the HPC Cluster Manager Configuration Job Templates screen. The Job Template Wizard opens, and you may create the queue there. It is possible to limit the user options when creating the new template, but because Moab schedules the resources, any specific policies should be set in Moab so that it is safe to leave the default values.

To associate a job with a specific queue:

- If you are submitting the job from Windows, select the desired job template during job submission.
- If you are submitting the job from Linux, specify the queue name during job submission.

```
echo ping -n 100 localhost | msub  
-los=windows,walltime=100 -q Department 1
```



Job templates in Windows must not contain spaces.



The nodes must be reached after a job template is created in order for MSMHPC to pick up the new template.



You may still use the five static queues from previous versions (HIGHEST, ABOVENORMAL, NORMAL, BELOWNORMAL and LOWEST) if the default job template is selected.

The following lines of code define the interface to the HPC resource manager and call the specified Perl scripts to perform any action on the HPC cluster. You must edit the moab.cfg file by adding the following lines, adjusting the paths to reflect your directory structure:

```
RMCFG[HPC]          TYPE=NATIVE:MSMHPC  
RMCFG[HPC]          PARTITION=local  
RMCFG[HPC]          NODESTATEPOLICY=OPTIMISTIC  
RMCFG[HPC]          DEFOS=windows  
RMCFG[HPC]          FLAGS=USERSPACEISSEPARATE  
RMCFG[HPC]          ADMINEXEC=jobsubmit  
RMCFG[HPC]          ENV=OSSTRING=windows;RMNAME=MSMHPC;PUBKEY=myspubkey;DOMAIN=yourdomain;PR  
  
RMCFG[HPC]          CLUSTERQUERYURL=exec://$TOOLS DIR/cluster.query.hpc.pl  
RMCFG[HPC]          WORKLOADQUERYURL=exec://$TOOLS DIR/workload.query.hpc.pl  
RMCFG[HPC]          JOBSUBMITURL=exec://$TOOLS DIR/job.submit.hpc.pl  
RMCFG[HPC]          JOBSTARTURL=exec://$TOOLS DIR/job.start.hpc.pl  
RMCFG[HPC]          JOBCANCELURL=exec://$TOOLS DIR/job.cancel.hpc.pl  
RMCFG[HPC]          JOBREQUEUEURL=exec://$TOOLS DIR/job.requeue.hpc.pl
```



Setting the OSSTRING variable allows MSMHPC tools to report a custom operating system. This enables you to run multiple HPC resource managers. It is recommended to set each resource manager's DEFOS parameter to the same string set in the OSSTRING variable.

## 5.0 Submitting Jobs through Moab

- [5.1 Cache the User Credentials](#)
- [5.2 Submitting Jobs with msub](#)
- [5.3 Verify that the Job has Migrated to HPC 2008 R2](#)
- [5.4 Verify that the Job Runs Correctly](#)

## 5.1 Cache User Credentials

Before you can submit jobs, your credentials must be cached on the HPC 2008 R2 head node, which allows the MSMHPC service to authenticate when it submits the job to HPC 2008 R2. Credential caching only needs to be done once for each user. Once cached, you can submit jobs normally with the **msub** command. To cache user credentials, use either MSMHPC Manager, the MSMHPCCacheCredentialsCLI / MSMHPCCacheCredentialsGUI application, or the `user.cache.pl` script that was provided with the MSMHPC Perl scripts.

If the credentials are not cached on submission or start, the job will be deferred and have a message attached to it in Moab indicating that the credentials are not cached and which user attempted to start the job.

The `./user.cache.pl` script is the recommended method. It may have one of the following three syntaxes:

- `./user.cache.pl` - prompts for username and password twice.
- `./user.cache.pl<username>` - prompts for password twice.
- `./user.cache.pl<username><password>` - no prompt given.

### Items for Special Consideration

- Configure the `env.hpc` file (shipped with the MSMHPC tools) to reflect your configuration. Then source, or load, the file when you wish to cache user credentials:

```
> . env.hpc
> ./user.cache.pl
```

- These credentials are sent in *plain text* over the network.
- The domain configured in `moab.cfg` is automatically appended to the username.
- If a job will not start and gets stuck in the "Blocked" queue, run the **checkjob** command to diagnose the issue:

```
> checkjob -v <jobid>
```

Note whether the following message appears as a "block reason": "The supplied username (<username>) was not cached." If you see this message, recache your credentials (because they were not cached, have expired, or the user password has changed).

- Moab cannot process jobs submitted directly from Windows if user credentials are not cached through MSMHPC / Moab. If Moab detects this, Windows jobs are canceled with the following reason:

```
Moab won't be able to process this job until you cache the credentials for user
<user>. Please refer to the documentation for more information.
```

- Administrators can check the Activation Filter's log files in the MSMHPC Manager for the reason jobs are canceled.

To use the MSMHPCCacheCredentialsCLI and MSMHPCCacheCredentialsGUI applications instead of your script to cache your credentials, copy both the application you want and the MSMHPCConnections.dll file to the

```
msmhpc_cache_credentials_gui
```

The public key refers to the key set up in [Installing the MSMHPC Services](#). Host refers to the host name or IP address of the machine running the MSMHPC service.



Both of the clients (CLI and GUI) must reach the Windows head node's port 5343.

## 5.2 Submit the Job via msub

You can use the **msub** command to submit jobs. You should submit a job now to test the system. Note that all executables called must exist on the Windows compute nodes or the script will fail. For example:

```
> echo ping -n 300 localhost | msub -l walltime=300,os=windows
```

## 5.3 Verify that the Job has Migrated to HPC 2008 R2

To verify that the job successfully migrated to HPC 2008 R2, on the HPC 2008 R2 head node, open HPC Cluster Manager (**Start**→**All Programs**→**Microsoft HPC Pack R2**→**HPC Job Manager**). You should see the newly submitted job.

## 5.4 Verify that the Job Runs Correctly

In order to verify that Moab starts the job correctly, do the following:

1. Using the `showq` command, verify the job is in the Moab queue and that the job's state changes to "Running."
2. Use `checkjob` on the Linux head node to determine which Windows node is currently running the job. RDP (remote desktop) into that node and open Task Manager. Verify the job is running, and that it is running as the correct user.
3. When the job finishes, the stdout and stderr files should be staged back to the user's home directory (if Samba is configured) or to the shared directory on the HPC 2008 R2 head node. Verify these files are present and have the correct content.

## 6.0 Configuring Moab for Dual-Boot Systems

Moab requires configuration to ensure proper functionality in a dual-boot system. The Moab OS tracking feature must be configured to recognize the dual system.

- [6.1 Node Setup](#)

## 6.1 Node Setup

For OS tracking to work correctly, both operating systems of the dual-boot nodes must have identical host names. Moab host names are case sensitive (even though DNS is not). You don't need to have all the host names capitalized on each machine. You only need to capitalize them in the TORQUE nodes file (`/var/spool/torque/server_priv/nodes` by default) for them to be reported in caps. You can change the case of node names in the MSMHPC Manager. Also, if SSH is enabled on both operating systems, the SSH keys should be identical to avoid SSH errors.

To make the SSH keys identical, boot all of the nodes into one operating system. Copy all SSH keys from the nodes onto the head node. (You can usually find keys in `/etc/ssh` and usually named `ssh_host_*`.) Then reboot the nodes and copy the keys to the other operating system. You may need to edit `/etc/ssh/sshd_config` to point to the new key files. Also, make sure the hostnames and IP addresses are identical.

## 7.0 Creating SSH Shared Keys for Cluster Users

So that you do not have to manually copy the files created by the job, set up SSH shared keys for all your regular users on all of the compute nodes. To create a public and private key, do the following:

1. Run the following on the Linux head node:

```
> ssh-keygen -t rsa
```

2. Accept default settings and do not submit a passphrase (press **Enter** three times).
3. From the head node, SSH onto each and every node in the cluster by hostname (**including** the head node) to append the host's public key to the `.ssh/known_hosts` file.

```
> for i in node01 node02 node03; do ssh -o StrictHostKeyChecking=no  
${i} hostname; done
```

4. Append the contents of `id_rsa.pub` to the `authorized_keys` file.

```
> cat .ssh/id_rsa.pub >> .ssh/authorized_keys
```

5. Create the `.ssh` directory on all compute nodes in case it does not exist.

```
> for i in node04 node05 node06; do ssh ${i} mkdir .ssh; done
```

6. Copy the `.ssh` folder to the nodes.

```
> for i in node01 node02 node03; do scp -r .ssh/id_rsa.pub  
${i}:~/.ssh; done
```

7. (Optional step) To allow users to access the compute nodes from the head node, copy the `authorized_keys` to the nodes.

```
> for i in node01 node02 node03; do scp -r .ssh/authorized_keys  
${i}:~/.ssh; done
```



For information about automating SSH key creation, see the [Automating the Home Directory and SSH Key Creation](#) documentation.

# 8.0 Centralizing Authentication with Active Directory (AD)

- [8.1 Installing MSMHPC Services](#)
- [8.2 Synchronizing UIDs on the Master Node](#)
- [8.3 Synchronizing UIDs across All Linux Machines](#)

Three packages are required to configure authentication against AD.

- Kerberos
- Winbind
- Samba

Install the packages and their dependencies by using the following command:

```
> apt-get install krb5-user samba winbind
```



This example assumes a Debian-based system.

## 8.1 Configuring the Linux System

If you are using CentOS or RHLES, **authconfig** configures everything.

```
authconfig --update --kickstart --enablewinbind --enablewinbindauth --  
-smbsecurity=ads --smbworkgroup=MYDOMAIN --smbrealm=MYDOMAIN --  
smbservers=MYSERVER.MYDOMAIN --winbindjoin=Administrator --  
winbindtemplatehomedir=/home/%U --winbindtemplateshell=/bin/bash --  
enablewinbindusedefaultdomain --enablelocauthorize
```

Set MYDOMAIN and MYSERVER.MYDOMAIN to reflect your environment. If you are using some other Linux distribution, follow the steps below.

The variables and names used in the examples below have the following meanings:

Variable	Description
<b>cridomain</b>	Public network domain name
<b>sge.local</b>	Private network domain name
<b>SGE.LOCAL</b>	Kerberos realm name
<b>headnode</b>	Head node Linux host name
<b>winhead</b>	Windows Active Directory server host name
<b>winadmin</b>	Windows user name for Windows domain administrator
<b>winuser</b>	Windows user name for normal user

1. Configure the Windows AD server domain name and name server.

Add the Windows domain name and name server IP addresses to `/etc/resolv.conf`.

```
root@x36-lhn:~# vi /etc/resolv.conf  
domain sge.local  
nameserver 10.0.0.100  
nameserver 192.168.0.1
```

2. Configure the Windows AD server as a recognized host.

Add a line to `/etc/hosts` that contains the:

- o Windows AD server IP address
- o fully qualified host name
- o host name with the Kerberos realm domain name
- o simple host name

```
root@headnode:~# vi /etc/hosts
...
10.0.0.100 winhead.cridomain winhead.sge.local winhead
```

3. Configure the Linux system to look up users using Winbind.

Add the Windows AD server and domain name to `/etc/nsswitch.conf`:

```
root@headnode:~# vi /etc/nsswitch.conf
...
passwd:          compat winbind
group:           compat winbind
shadow:         compat winbind
```

4. Configure Kerberos.

Set up Kerberos by adding the following to `/etc/krb5.conf`:

```
root@headnode:~# vi /etc/krb5.conf
...
[logging]
default=FILE:/var/log/krb5libs.log
kdc=FILE:/var/log/krb5kdc.log
admin_server=FILE:/var/log/kadmind.log

[libdefaults]
default_realm = SGE.LOCAL
dns_lookup_realm = true
dns_lookup_kdc = true
ticket_lifetime = 24h
forwardable = yes

[kdc]
profile = /var/kerberos/krb5kdc/kdc.conf

[appdefaults]
pam = {
debug = false
ticket_lifetime = 36000
renew_lifetime = 36000
forwardable = true
krb4_convert = false
}

[realms]
SGE.LOCAL = {
kdc = winhead.sge.local
```

5. Synchronize the Linux system clock with Windows AD server and make sure the domain name is uppercase.

```
root@headnode:~# ntpdate winhead
 2 Dec 09:37:58 ntpdate[6495]: adjust time server 10.0.0.100 offset -
0.120004 sec
```

## 6. Test Kerberos authentication.

```
root@headnode:~# kinit winuser@SGE.LOCAL
Password for winuser@SGE.LOCAL:
```

If no error messages are returned, Kerberos authentication was successful.

You can now check existing authentication tickets.

```
root@headnode:~# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: winuser@SGE.LOCAL

Valid starting      Expires            Service principal
12/02/09 09:38:38  12/02/09 19:38:41  krbtgt/SGE.LOCAL@SGE.LOCAL
    renew until 12/03/09 09:38:38
```

## 7. Configure Samba.

Set up Samba by adding the following to `/etc/samba/smb.conf`:

```
workgroup = sge
max log size = 50
security = ads
password server = winhead.sge.local
realm = SGE.LOCAL
idmap uid = 16777216-33554431
idmap gid = 16777216-33554431
template shell = /bin/bash
template homedir = /home/%U
winbind use default domain = true
winbind enum users = yes
winbind enum groups = yes
winbind separator = +
```

## 8. Configure the pluggable authentication modules (PAM) to authenticate Windows AD users.

Set up PAM by adding the following to the specified `pam.d` files:

```
root@headnode:~# vi /etc/pam.d/common-account

account sufficient pam_winbind.so
account required pam_unix.so
```

```
root@headnode:~# vi /etc/pam.d/common-auth

auth sufficient pam_winbind.so
auth required pam_unix.so nullok_secure use_first_pass
```

```
root@headnode:~# vi /etc/pam.d/common-session
...
session required pam_mkhomedir.so umask=0077 skel=/etc/skel
```



The `/etc/pam.d/common-session` file makes PAM create the user's home directory on successful authentication.

## 9. Join the Linux system to the Windows domain.

```
root@headnode:~# net ads join -U winadmin
Enter winadmin's password:
Using short domain name -- SGE
```

```
Joined 'HEADNODE' to realm 'sge.local'
```

You can ignore any DNS update errors.

#### 10. Restart Samba and Winbind.

Restart Samba and Winbind in the following order:

```
root@x36-lhn:~# service samba stop
* Stopping Samba daemons [ OK ]
root@x36-lhn:~# service winbind stop
* Stopping the Winbind daemon winbind [ OK ]
root@x36-lhn:~# service samba start
* Starting Samba daemons [ OK ]
root@x36-lhn:~# service winbind start
* Starting the Winbind daemon winbind [ OK ]
```

#### 11. Restart Moab.

If Moab is running, restart it so that it can recognize the Windows AD users.

```
root@headnode:~# mschedctl -k

moab will be shutdown immediately

root@headnode:~# moab
```

#### 12. Test Linux authentication of Windows AD users.

Verify Windows AD users can log in to the Linux system.

```
root@headnode:~# finger -m winuser
Login: winuser Name:
Directory: /home/winuser Shell: /bin/bash
Last login Tue Dec 1 18:07 (MST) on pts/4 from winhead.cridomain
No mail.
No Plan.
```

## 8.2 Synchronizing UIDs on the Master Node

Winbind generates local random user IDs in the order that users are first queried. This makes accessing NFS shares difficult; however, it can be resolved by syncing all the user IDs on the master node (the Linux head node or machine that exports the NFS share) and syncing Winbind's DBs on all the compute nodes for every reboot.



HPCUsers is the default user group. Users are automatically placed in it when they are created through MSMHPC or MSMHPC tool scripts (./create.ad.account.hpc.pl).

#### 1. Populate the winbind\_idmap file on the Linux head node.

```
[root@x36-lhn samba]# for i in `wbinfo -u`; do id ${i} ; done
uid=16777216(administrator) gid=16777216(HPCUsers)
groups=16777216(HPCUsers),16777217(group policy creator
owners),16777218(domain admins),16777219
(enterprise admins),16777220(schema admins),16777221(denied rodc
password replication group)
uid=16777217(guest) gid=16777222(domain guests) groups=16777222(domain
guests)
uid=16777223(krbtgt) gid=16777216(HPCUsers)
groups=16777216(HPCUsers),16777221(denied rodc password replication
group)
uid=16777220(lmsilva) gid=16777216(HPCUsers) groups=16777216(HPCUsers)
uid=16777221(test) gid=16777216(HPCUsers) groups=16777216(HPCUsers)
```

#### 2. Dump the winbind\_idmap mapping DB on the Linux head node and restore it across all the compute

nodes.

```
[root@x36-lhn samba]# for i in node01 node02 node03 node04; do net
idmap dump /var/cache/samba/winbindd_idmap.tdb | ssh ${i} net idmap
restore ;done
```

3. Query a single user ID on the entire cluster to verify the synchronization.

```
[root@x36-lhn samba]# for i in node01 node02 node03 node04; do ssh
${i} id test1; done
uid=16777224(test1) gid=16777216(HPCUsers) groups=16777216(HPCUsers)
uid=16777224(test1) gid=16777216(HPCUsers) groups=16777216(HPCUsers)
uid=16777224(test1) gid=16777216(HPCUsers) groups=16777216(HPCUsers)
uid=16777224(test1) gid=16777216(HPCUsers) groups=16777216(HPCUsers)
```

## 8.3 Synchronizing UIDs across All Linux Machines

To synchronize UIDs across all Linux machines, do the following every time the active directory adds or removes any user account.

1. Populate the winbind\_idmap file on the Linux head node.

```
[root@x36-lhn samba]# for i in `wbinfo -u`; do id ${i} ; done
uid=16777216(administrator) gid=16777216(HPCUsers)
groups=16777216(HPCUsers),16777217(group policy creator
owners),16777218(domain admins),16777219
(enterprise admins),16777220(schema admins),16777221(denied rodc
password replication group)
uid=16777217(guest) gid=16777222(domain guests) groups=16777222(domain
guests)
uid=16777223(krbtgt) gid=16777216(HPCUsers)
groups=16777216(HPCUsers),16777221(denied rodc password replication
group)
uid=16777220(lmsilva) gid=16777216(HPCUsers) groups=16777216(HPCUsers)
uid=16777221(fchism) gid=16777216(HPCUsers) groups=16777216(HPCUsers)
uid=16777222(test) gid=16777216(HPCUsers) groups=16777216(HPCUsers)
```

2. Dump the winbind\_idmap mapping DB on the Linux head node and restore it across all the compute nodes and synchronize ID maps.

```
[root@x36-lhn samba]# for i in node01 node02 node03 node04; do net
idmap dump /var/cache/samba/winbindd_idmap.tdb | ssh ${i} net idmap
restore ;done
```

3. Query a single user ID on the entire cluster to verify the synchronization.

```
[root@x36-lhn samba]# for i in node01 node02 node03 node04; do ssh
${i} id test1; done
uid=16777222(test1) gid=16777216(HPCUsers) groups=16777216(HPCUsers)
uid=16777224(test1) gid=16777216(HPCUsers) groups=16777216(HPCUsers)
uid=16777223(test1) gid=16777216(HPCUsers) groups=16777216(HPCUsers)
uid=16777223(test1) gid=16777216(HPCUsers) groups=16777216(HPCUsers)

# for i in `wbinfo -u`; do id ${i} ; done
uid=16777216(administrator) gid=16777216(HPCUsers)
groups=16777216(HPCUsers),16777217(group policy creator
owners),16777218(domain admins),16777219(enterprise
admins),16777220(schema admins),16777221(denied rodc password
replication group)
uid=16777217(guest) gid=16777222(domain guests)
groups=16777222(domain guests)
uid=16777223(krbtgt) gid=16777216(HPCUsers)
groups=16777216(HPCUsers),16777221(denied rodc password replication
group)
uid=16777220(lmsilva) gid=16777216(HPCUsers)
groups=16777216(HPCUsers)
uid=16777221(fchism) gid=16777216(HPCUsers)
```

```
groups=16777216(HPCUsers)
uid=16777222(test) gid=16777216(HPCUsers) groups=16777216(HPCUsers)
uid=16777224(test1) gid=16777216(HPCUsers) groups=16777216(HPCUsers)
uid=16777225(test2) gid=16777216(HPCUsers) groups=16777216(HPCUsers)
uid=16777226(test3) gid=16777216(HPCUsers) groups=16777216(HPCUsers)

# for i in node01 node02 node03 node04; do net idmap dump
/var/cache/samba/winbindd idmap.tdb | ssh ${i} net idmap restore
```

## 9.0 Reservation Tracking

Reservation tracking allows you to configure a dynamic job to be a certain size at a certain time. Reservations allow you to specify how many (or which specific) nodes a job should use at certain times. These reservations are standing reservations that you configure in the `moab.cfg` file.

For example, consider an application with three defined reservations that is heavily used during the afternoon:

**Reservation 1:** (12:00 a.m. - 12:00 p.m.)—node001, node002, node003, node007

**Reservation 2:** (12:00 p.m. - 5:00 p.m.)—node001, node002, node003, node007, node008, node009

**Reservation 3:** (5:00 p.m. - 12:00 a.m.)—node001, node002

The application is then configured to track the reservation.

At 12:00 a.m., it uses 4 nodes—node001, node002, node003, node007.

At 12:00 p.m., it expands to use 6 nodes—node001, node002, node003, node007, node008, node009.

At 5:00 p.m., it contracts to use only 2 nodes—node001, node002.

A dynamic partition can also be configured to use reservation tracking. This allows the system to change the operating system pools according to a calendar.

The following parameters are required. The example used is our hybrid cluster.

Configure the job to use partition tracking. This is the dynamic job associated with the partition for a dynamic partition. `ADVRES` specifies the name of the reservation group the job will track.

```
JOBCFG[win] FLAGS=ADVRES:RG1
```

In addition to normal parameters, each reservation must define the following:

- `Partition=ALL` allows the reservation to span partitions. Required for dynamic partitions.
- `RSVGROUP=<GROUP>` specifies the reservation group (defined above). In addition all reservations must have a `HOSTLIST` or `TASKCOUNT`, `STARTTIME`, and `DURATION` or `ENDTIME`.
- `FLAGS=BYNAME` is also useful for non-partition dynamic jobs.

```
SRCFG[RG1S1]      PARTITION=ALL
SRCFG[RG1S1]      RSVGROUP=RG1

SRCFG[RG1S1]      COMMENT="Resource Group 1 Step 1"
SRCFG[RG1S1]      RSVGROUP=RG1
SRCFG[RG1S1]      STARTTIME=1:00:00
SRCFG[RG1S1]      ENDTIME=12:00:00
SRCFG[RG1S1]      DAYS=MON,TUE,WED,THU,FRI
SRCFG[RG1S1]      HOSTLIST=CCS1,CCS2
SRCFG[RG1S1]      USERLIST=root,user1
SRCFG[RG1S1]      PARTITION=ALL
SRCFG[RG1S1]      FLAGS=BYNAME

SRCFG[RG1S2]      COMMENT="Resource Group 1 Step 2"
SRCFG[RG1S2]      RSVGROUP=RG1
SRCFG[RG1S2]      STARTTIME=12:00:00
SRCFG[RG1S2]      ENDTIME=17:00:00
SRCFG[RG1S2]      DAYS=MON,TUE,WED,THU,FRI
SRCFG[RG1S2]      HOSTLIST=CCS1,CCS2,LAB1
SRCFG[RG1S2]      TASKCOUNT=6
SRCFG[RG1S2]      USERLIST=root,user1
SRCFG[RG1S2]      PARTITION=ALL
SRCFG[RG1S2]      FLAGS=BYNAME

SRCFG[RG1S3]      COMMENT="Resource Group 1 Step 3"
SRCFG[RG1S3]      RSVGROUP=RG1
```

```
SRCFG[RG1S3] STARTTIME=17:00:00
SRCFG[RG1S3] ENDTIME=12:00:00
SRCFG[RG1S3] DAYS=MON,TUE,WED,THU,FRI
```

# Appendix A: GRUB Dual-Boot OS Switching

Installing Windows and then Linux operating systems on each node ensures that the GRUB is installed as the bootloader. Before continuing, verify that GRUB is able to manually boot both operating systems. Leave the Linux partition as the default. The partition table should appear as follows when finished:

1. NTFS (Windows) (hd0,0)
2. EXT3 (Linux) (hd0,1)
3. Swap (Linux)

If your partition scheme is different, you can look up the partitions using a partition editor, or by reading GRUB's menu.lst file (usually located at /boot/grub/menu.lst).



For sites planning to redeploy frequently, consider installing GRUB to the Linux partition instead of the master boot record (MBR) because Windows overwrites the MBR (every subsequent deployment), which could cause you to lose access to an MBR-based GRUB configuration.

PXE booting is the recommended dual-boot method, but the following offers instructions for automating dual-boot OS switching using GRUB.

Instructions for automating dual-boot operating system switching assume that you are installing a Linux system that uses the GRand Unified Bootloader (GRUB). Note that you should set up dual-booting only on compute nodes. To automate dual-boot operating system switching, during the Linux installation, manually partition the drive. To do so:

- A. 1 (ext3) primary partition, right after the Windows partition, mounted at "/>.
- B. 1 FAT32 partition (primary or extended) of at least 256 MB mounted at /boot/otheros.
- C. 1 extended partition for swap.

Note which partition number the FAT32 partition is. If it is the third primary partition, it will usually be sda3, or if it is an extended partition, it will be greater than 4. You can look this up using the gparted partitioning utility.



After following these steps, there should be two grub.conf files, one at /boot/grub and one at /boot/otheros. The grub.conf file at /boot/otheros should not be a symlink to the grub.conf file at /boot/grub.

## Post Linux Installation Steps for Dual-Boot Setup

1. Copy all the files in /boot/grub to /boot/otheros:

```
> cp /boot/grub/* /boot/otheros
```

2. Create a symbolic link from /boot/grub/menu.lst to /boot/grub/grub.conf. To create the link, issue the following commands:

```
> cd /boot/grub
> rm menu.lst
> ln -s grub.conf menu.lst
```

3. Copy and paste the following text into the /boot/grub/grub.conf file, immediately preceding the first "title" line:

```
title BOOT REDIRECT: PLEASE WAIT
root (hd0,2)
configfile /menu.lst
boot
```

Please note that the preceding text should be the first "title" entry in the list of boot options in `/boot/grub/grub.conf`. Also, make this title the default boot option (i.e. default 0); this will allow you to boot your system to a known partition if anything goes wrong during the boot redirect setup.

Also note that in the preceding sample code, the numeral 2 in the line `root (hd0,2)` represents the FAT32 partition; you should replace the 2 in that line with your FAT32 partition number.

4. Save and close the file.

5. Copy the following sample code and save it in a file named `switchos.pl` and place it in `/boot/otheros/`.

```
#Contents of the switchos.pl file (used by Moab to change the OS)
#!/usr/bin/perl

use strict;
use warnings;

my $default;
my $file;

my ($os) = @ARGV;
$os = "" unless defined $os;
$file = 'menu.lst';

if ( $os eq 'linux' )
{
    $default = 0;
}
elsif ( $os eq 'windows' )
{
    $default = 2;
}
else
{
    die "Usage: switch.pl <windows|linux>";
}

my $menu = <<__END__;
# grub.conf generated by anaconda
#
```

6. Edit `/boot/otheros/switchos.pl` to reflect your system configuration. That is, verify the specified disk and FAT32 partition values are correct. If you want, replace the in-line `grub.conf` (between the `__END__` lines) in the file with your own `grub.conf`. Note the injection of the current boot OS via the Perl variable: `default=$default`.

- o If you want to speed up the boot process, change the timeout variable, which has a default of 30 seconds, to a lower value.

7. Create a file named `bootlin.bat` that includes the following:

```
w:
cd \
perl switchos.pl linux
```

8. Boot into the Windows partition and ensure that you have installed the latest version of Perl.

9. Assign a drive letter to the FAT32 partition. (Note that `w` is used in the previous example.)

10. To switch between the Windows and Linux environments, do the following:

- A. Boot into Windows using the following command (run from `/boot/otheros`):



```
> ./switchos.pl windows  
> reboot
```

B. Change the drive letter of the FAT32 partition to *w*.

C. Switch back to Linux by running the following command (from *w:\*):

```
> perl switchos.pl linux  
> shutdown /r /t 00 /f
```

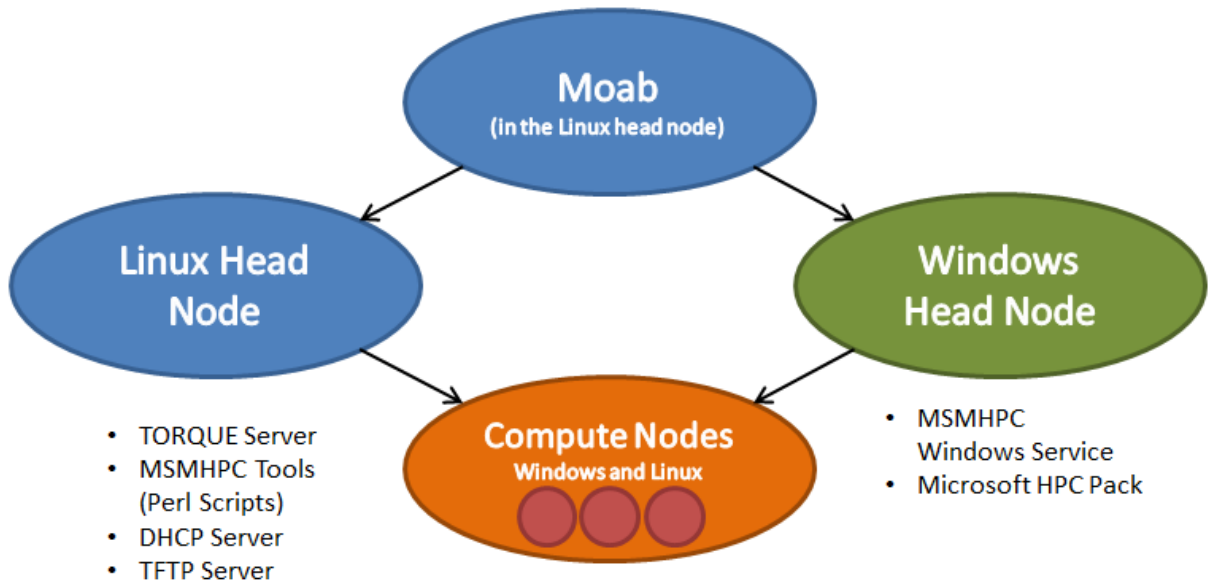
# Appendix B: Troubleshooting Common Problems

The following are common issues that are encountered when installing and configuring Adaptive HPC Suite:

- Verify that the following ports are open and reachable to prevent firewall issues:
  - The TFTP server port on the Linux head node.
  - The HTTP server port on the Linux head node.
  - The TORQUE server (`pbs_server`) port on the Linux head node.
  - The MSMHPC web service port on the Windows head node. By default an exception is added to the firewall on install and when the port is changed.
- Verify that GCC is installed on the Linux head node. This is especially important when installing the required Perl modules or compiling packages.
- Verify that the required Perl modules are installed for MSMHPC tools by running the `module_test.pl` script in the `extra_toolkit` directory.



See the [Moab Adaptive HPC Knowledgebase](#) for more troubleshooting information.



### Path of Communication

