Installation Walkthrough

CentOS 6.5
Warewulf 3.5
OpenMPI 1.4.5
Ganglia 3.1.7
SLURM 2.3.4
HPL 2.0

Compiled By:

Jason Somers
Red Barn Technology Group, Inc.
jsomers@thinkredbarn.com
http://www.thinkredbarn.com

Last Updated: March 21, 2014

1. Operating System Installation

a. Download and burn and boot from the "Minimal Install" CD
b. Set up your partitions however you wish
c. Configure your network interfaces:

- *Please note that in this walkthrough, the server's hostname will be 'wwmaster'. eth0 will be assigned via DHCP, and will provide the master with internet access. eth1 will be wwmaster's interface to the nodes and will be assigned a static IP of 10.10.10.1/255.255.255.0)*

- *I also strongly recommend disabling – or even removing – NetworkManager on the master.*

d. Disable SELINUX by editing /etc/selinux/config:
(NOTE: The new selinux setting will not take effect until the next reboot)

```
# -- /etc/selinux/config --

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

e. Install required OS packages and reboot

```
# yum -y groupinstall "Development Tools"
# yum -y install httpd dhcp tftp-server tftp mod_perl mysql mysql-server wget tcpdump
nfs-utils nfs-utils-lib
# yum -y update
# reboot
```

f. Make sure Apache and MySQL start automatically – and iptables doesn't.

(Yes – I know this is a bad security practice. Once you get everything thing working, feel free to re-enable this after adding all of your rules)

```
# service httpd start
# service mysqld start
# service iptables stop
# chkconfig httpd on
# chkconfig mysqld on
# chkconfig iptables off
```

g. Make sure the tftp-server is enabled in /etc/xinetd.d/tftp

```
# -- /etc/xinetd.d/tftp --

# default: off
# description: The tftp server serves files using the trivial file transfer \
#       protocol.  The tftp protocol is often used to boot diskless \
#       workstations, download configuration files to network-aware printers, \
#       and to start the installation process for some operating systems.
service tftp
{
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -s /var/lib/tftpboot
        disable                 = no
        per_source              = 11
        cps                     = 100 2
        flags                   = IPv4
}
```

h. And restart xinetd.

```
# service xinetd restart
```

2. Installing Warewulf

a. For simplicity, we configure yum Warewulf's own RHEL repository, along with Red Hat's EPEL repo:

```
# wget http://warewulf.lbl.gov/downloads/repo/warewulf-rhel6.repo -O
/etc/yum.repos.d/warewulf-rhel6.repo
# rpm -ivh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
```

b. Using yum, we install Warewulf and its required dependences.
(Perl-Term-ReadLine allows for command completion in the Warewulf shell)

```
# yum -y install warewulf-common warewulf-provision warewulf-vnfs warewulf-provision-
server warewulf-ipmi Warewulf-mic
# yum -y install perl-Term-ReadLine-Gnu
```

c. Configure the cluster interface (the interface wwserver will use to communicate with the nodes) in /etc/warewulf/provision.conf

```
# -- /etc/warewulf/provision.conf --

# What is the default network device that the nodes will be used to
# communicate to the nodes?
network device = eth1  ###change to eth0 if needed
```

3. Configuring a Warewulf node image

a. The following command will build a simple vanilla distribution of Centos into the /var/chroots/centos6 folder using yum.

```
# wwmkchroot centos-6 /var/chroots/centos6
```

b. Next, we convert the new node OS into a VNFS (Virtual Node File Systsem) usable by Warewulf – and build the bootstrap from our current running environment.

```
# wwvnfs --chroot /var/chroots/centos6
# wwbootstrap `uname -r`
```

c. In order to register each node, we will run wwnodescan and then boot up each node on the cluster network – and each DHCP request will be recorded and the MAC addresses stored on the Warewulf master.

NOTE: It is imperative that you understand how the IPMI works on your node motherboards! Many server boards with IPMI will give you the option between using a dedicated IPMI port –or- sharing the LAN1 port. If no link is detected in the IPMI port when the power cord is plugged in, it will assume you wish to share LAN1. Then - if the IPMI is set to use DHCP, this request will come first – and confuse the Warewulf server. Warewulf will register the IPMI MAC address as the node's LAN1 and not the actual one. So – in order to address this, either make sure there is separate link to the IPMI port before boot (so LAN1 is not shared) or makes sure you have a static address set for the IPMI interface.

```
# wwsh dhcp update
# wwsh dhcp restart
# wwnodescan --netdev=eth0 --ipaddr=10.10.10.10 --netmask=255.255.255.0 --vnfs=centos6 --
bootstrap=`uname -r` n0000

*Note: A range of IPs can be indicated as n00[02-19]
*Note: netdev is the cluster interface of the nodes
*Note: ipaddr is the first IP to be assigned to the nodes. It will increment for each new
node
```

d. Once the nodes have been registered, we need to update the master's DHCP conf to include all of the node's MACs and IPs.

```
# wwsh dhcp update
# wwsh dhcp restart
```

e. And a reboot for good measure…

```
# reboot
```

Your nodes should be booting successfully now. Power them on in the order you want them numbered.

4. Customizing Warewulf

a. **Configuring the Parallel Distributed Shell (pdsh)**

- We need to install pdsh and its dependency to the master. Then we share root's ssh key to the node image. Then, we rebuild the node image and manually reboot each node.

```
# yum -y install http://apt.sw.be/redhat/el6/en/x86_64/rpmforge/RPMS/libgenders-1.14-
2.el6.rf.x86_64.rpm
# yum -y install http://pkgs.repoforge.org/pdsh/pdsh-2.27-1.el6.rf.x86_64.rpm
# cd /root
# chmod 700 /var/chroots/centos6/root/.ssh
# ssh-keygen     (use empty passphrases)
# cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
# chmod 400 ~/.ssh/authorized_keys
# cp ~/.ssh/authorized_keys /var/chroots/centos6/root/.ssh/
# wwvnfs --chroot /var/chroots/centos6

       -- Manually reboot each node -
```

- Once each node has rebooted, we must establish an ssh connection with each one so that the keys get properly associated with each node's MAC address. *(I know this is a pain – if anyone has a more automated way of doing this, please let me know)*

```
# ssh n0000
       -- answer 'yes' and exit
# ssh n0001
       -- answer 'yes' and exit
# ssh n0002
       -- answer 'yes' and exit
…
```

- Once finished, the following command should work:

```
# pdsh -w n00[00-05]

--Output--
n0000: n0000
n0001: n0001
n0002: n0002
n0003: n0003
n0004: n0004
n0005: n0005
```

b. **Hosts file**

- Add wwmaster's cluster interface IP to the hosts template that will get passed to the nodes

```
#-- /etc/warewulf/hosts-template

127.0.0.1               localhost   localhost.localdomain
10.10.10.1              wwmaster  wwmaster.mydomain.local
```

- Propagate Warewulf-generated /etc/hosts file to the nodes

```
# wwsh provision set --fileadd dynamic_hosts n0000 n0001 n0002

*Note: A range of nodes can be indicated as n00[02-19]
```

- Propagate Warewulf-generated /etc/hosts file to wwmaster

```
# wwsh file show dynamic_hosts > /etc/hosts

*Note: This should be run every time new nodes are added to the cluster. A cron job is
recommended…
```

c. **Sharing wwmaster's /home with the nodes.**

Each node will mount this as their own /home and gives us a good launching point for jobs across the cluster.

- On the master, edit /etc/exports (changes in red):

```
#-- /etc/exports --

/home/            10.10.10.0/255.255.255.0(rw,no_root_squash,async)
```

- Next is to edit the nodes' /etc/fstab. In order to do this, we directly modify the chrooted OS we made earlier (/var/chroots/centos6):

```
#-- /var/chroots/centos6/etc/fstab --

#GENERATED_ENTRIES#
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0

10.10.10.1:/home/     /home/      nfs rw,soft,bg    0  0
```

- Tell wwmaster to run the nfs server at startup

```
# chkconfig nfs on

# service rpcbind start

# service nfs start
```

- And rebuild the node image from the modified chroot.

```
# wwvnfs --chroot /var/chroots/centos6
```

Once complete, reboot the nodes to load the modified VNFS.

d. **Synchronizing user/group accounts to the nodes.**

- Create new user

```
# adduser rbc
# passwd  rbc
```

- Provision the master's /etc/passwd file to the nodes

```
# wwsh file import /etc/passwd
# wwsh provision set n[0000-0099] --fileadd passwd
```

- Provision the master's /etc/group file to the nodes

```
# wwsh file import /etc/group
# wwsh provision set n[0000-0099] --fileadd group
```

- In order to provide password-less access for users to the nodes, each time you create a user, make sure you generate a key pair and add it to authorized_keys. Since the home folder is shared between all nodes, you only need to do this once for each user at inception. Our user will be 'rbc'

```
# su - rbc
$ cd ~
$ mkdir .ssh
$ chmod 700 .ssh
$ ssh-keygen -t dsa -P ""
$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
$ chmod 400 ~/.ssh/authorized_keys
$ exit
```

Once complete, reboot to allow password-less logins.

e. **Installing the EPEL repo on the nodes:**

- So as to avoid any versioning differences between the master and the nodes, let's make sure the nodes' filesystem is using the same EPEL repo as the master.

```
# rpm --root /var/chroots/centos6 -ivh
http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
```

- Rebuild the VNFS

```
# wwvnfs --chroot /var/chroots/centos6
```

Reboot the nodes to deliver them the latest VNFS

f. **Configuring a node's hard drive as swap and scratch space (if available)**

- Create the following file and save it as */home/rbc/node-partitions*

```
#-- /home/rbc/node-partitions

# This will create 2 partitions on device. First is a swap of about 16Gb,
# and the second is the remainder of the filesystem

,2034,82
,,83
```

- Partitioning the drive (this must be done on each node! – it can be done much easier with the help of pdsh)

```
# chmod 777 /home/rbc/node-partitions
# mkdir /var/chroots/centos6/scratch
# chmod 777 /var/chroots/centos6/scratch
# pdsh -w n00[00-99] "cat /home/rbc/node-partitions | sfdisk /dev/sda"
```

- Formatting the drive:

```
# pdsh -w n00[00-99] "mkswap /dev/sda1"
# pdsh -w n00[00-99] "mkfs.ext4 /dev/sda2"
```

- Mounting the drive (do this on the master, NOT the nodes)
  Edit /var/chroots/centos6/etc/fstab

```
#-- /var/chroots/centos6/etc/fstab



#GENERATED_ENTRIES#
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0

10.10.10.1:/home/            /home/            nfs rw,soft,bg 0 0

# To use a local disk on the nodes (make sure the mountpoints exist in
# the VNFS!!!)

/dev/sda2      /scratch               ext4    defaults        0 0
/dev/sda1      none                   swap    defaults        0 0
```

- Rebuild the VNFS and reboot the nodes

```
# wwvnfs --chroot /var/chroots/centos6
# pdsh -w n00[00-99] reboot
```

3. Installing Ganglia

*** This procedure assumes that you have installed the EPEL repo on both the master and nodes, as described in Section 3: Installing Warewulf***

a. Using yum, install the ganglia binaries on the master node:

```
# yum install -y ganglia*
```

b. Ganglia has two conf files that need editing. The first is for the server daemon, gmetad. This resides only on the server. We must edit it to contain the name of the cluster and the name of the master node (which in this case, is simply localhost).

```
#-- /etc/ganglia/gmetad.conf - (Line 39)

Data_source "Warewulf" localhost
```

c. Next, edit /etc/ganglia/gmond.conf. This information will reside on both the master and the nodes – indicating where the gmetad daemon is and how to communicate with it. (changes in red)

```
#-- /etc/ganglia/gmond.conf - (lines 19-47)

cluster {
  name = "Warewulf"
  owner = "unspecified"
  latlong = "unspecified"
  url = "unspecified"
}

/* The host section describes attributes of the host, like the location */
host {
  location = "unspecified"
}

/* Feel free to specify as many udp_send_channels as you like.  Gmond
   used to only support having a single channel */
  udp_send_channel {
  host = wwmaster
  port = 8649
}

/* You can specify as many udp_recv_channels as you like as well. */
udp_recv_channel {
  port = 8649
}

/* You can specify as many tcp_accept_channels as you like to share
   an xml description of the state of the cluster */
tcp_accept_channel {
  port = 8649
}
```

d. Start gmetad and gmond on the master and configure them to start automatically every boot.

```
# /etc/init.d/gmetad start
# /etc/init.d/gmond start
# chkconfig gmetad on
# chkconfig gmond on
```

e. Now, install only gmond on the nodes, copy the config file from the server, then make sure it starts at boot time on the nodes.

```
# yum --installroot=/var/chroots/centos6 install ganglia-gmond
# cp /etc/ganglia/gmond.conf /var/chroots/centos6/etc/ganglia/gmond.conf
# chroot /var/chroots/centos6
# chkconfig gmond on
# exit
```

f. One of the ganglia RPMs we installed (ganglia-web) actually builds the root of the ganglia web app, but we need to make sure apache is running and configured to start at boot time first. *If you installed Warewulf – this should already be done.*

```
# /etc/init.d/httpd start
# chkconfig httpd on
```

g. By default, ganglia-web only allows connections from localhost. If you would like the ganglia utility viewable by all on your network, adjust the /etc/httpd/conf.d/ganglia.conf file to be:

```
#-- /etc/httpd/conf.d/ganglia.conf

  #
  # Ganglia monitoring system php web frontend
  #

  Alias /ganglia /usr/share/ganglia

  <Location /ganglia>
    Order allow,deny
    Allow from all
    # Allow from .example.com
  </Location>
```

h. Rebuild the image and restart the nodes. Browse to http://<wwmaster_external_IP>/ganglia from your LAN to verify installation.

```
# wwvnfs --chroot /var/chroots/centos6
# pdsh -w n00[00-99] reboot
```

4. Installing OpenMPI

*** This procedure assumes that you have installed the EPEL repo on both the master and nodes, as described in Section 2: Installing Warewulf. We also assume that you are using a Gigabit network, and not Infiniband. This can be done, but is not detailed here.***

a. Get the needed files from openmpi.org

```
# cd ~
# wget http://www.open-mpi.org/software/ompi/v1.4/downloads/openmpi-1.4.5.tar.gz
# wget http://svn.open-mpi.org/svn/ompi/branches/v1.4/contrib/dist/linux/buildrpm.sh
# wget http://svn.open-mpi.org/svn/ompi/branches/v1.4/contrib/dist/linux/openmpi.spec
```

b. Make sure the buildrpm.sh script is executable:

```
# chmod +x buildrpm.sh
```

c. For some strange reason Centos 6.x does not give you a default RPM-building environment. You must create one yourself:

```
# cd ~
# mkdir -p ~/rpmbuild/{BUILD,RPMS,SOURCES,SPECS,SRPMS}
# echo '%_topdir /root/rpmbuild' > ~/.rpmmacros
```

d. Edit the buildrpm.sh file so that it will build a single rpm file (changes in red)

```
#-- ~/buildrpm.sh - (Lines 33-41)

# Note that this script can build one or all of the following RPMs:
# SRPM, all-in-one, multiple.

# If you want to build the SRPM, put "yes" here
build_srpm=${build_srpm:-"yes"}
# If you want to build the "all in one RPM", put "yes" here
build_single=${build_single:-"yes"}
# If you want to build the "multiple" RPMs, put "yes" here
build_multiple=${build_multiple:-"no"}
```

- *Note: Optimally, I would prefer to build with the multiple RPMs option and just install the openmpi-runtime package on the nodes – to reduce VNFS size, but that has given me some compilation errors I have not been able to fix. Please feel free to send me any advice on this topic.*

e. Build the RPM (this will take a while)

```
# ./buildrpm.sh openmpi-1.4.5.tar.gz     // this will take a while!
```

f. Install the new RPM on the master

```
# rpm -ivh /root/rpmbuild/RPMS/x86_64/openmpi-1.4.5-2.x86_64.rpm
```

g. Install the new RPM on the nodes

```
# yum --installroot /var/chroots/centos6 install /root/rpmbuild/RPMS/x86_64/openmpi-
1.4.5-2.x86_64.rpm
```

h. Rebuild the VNFS and reboot the nodes

```
# wwvnfs --chroot /var/chroots/centos6
# pdsh -w n00[00-99] reboot
```

5. Installing SLURM

*** This procedure assumes that you have installed the EPEL repos on both the master and nodes, as described in Section 2: Installing Warewulf.***

a. We will be using Munge for authentication, so it must be configured and installed before SLURM.

i. Add the 'munge' and 'slurm' users

```
# useradd munge
# useradd slurm
```

ii. Find the new users' entries in /etc/passwd, and change the homedir and shell: (changes in red)

***Note, your uid and guid may be different than mine. This is ok!*

```
#-- /etc/passwd --
…
munge:x:502:502::/home/munge:/sbin/nologin
slurm:x:503:503::/home/slurm:/sbin/nologin
```

iii. Sync the users to the nodes

```
# wwsh file import /etc/passwd
# wwsh file import /etc/group
```

iii. Install Munge

```
# yum install munge munge-devel
```

iv. Create the authentication key for use with Munge

```
# dd if=/dev/urandom bs=1 count=1024 >/etc/munge/munge.key
# echo -n "foo" | sha1sum | cut -d' ' -f1 >/etc/munge/munge.key  //(where foo is your
password)
# chown munge.munge /etc/munge/munge.key
# chmod 400 /etc/munge/munge.key
# /etc/init.d/munge start
# chkconfig munge on
```

v. Copy the authentication key to the nodes, then reboot the nodes

```
# yum --installroot=/var/chroots/centos6 install munge
# chroot /var/chroots/centos6
# chkconfig munge on
# exit
# cp -p /etc/munge/munge.key /var/chroots/centos6/etc/munge/munge.key
# chown -R munge.munge /var/chroots/centos6/etc/munge
# chown -R munge.munge /var/chroots/centos6/var/log/munge
# chown -R munge.munge /var/chroots/centos6/var/lib/munge
# mv /var/chroots/centos6/etc/rc3.d/S40munge /var/chroots/centos6/etc/rc3.d/S96munge
```

vi. Rebuild your VNFS and reboot your nodes

```
# wwvnfs --chroot /var/chroots/centos6
# pdsh -w n00[00-99] reboot
```

vii. You can test munge by running the following commands:

```
# munge -n | unmunge                // test locally
# munge -n | ssh n0000 unmunge      // test remotely

NOTE - make sure your clocks are synced (or at least very close) or munge will report an
error
```

b. Install slurm on the master

```
# wget http://www.schedmd.com/download/archive/slurm-2.3.4.tar.bz2
# yum install readline-devel openssl-devel pam-devel
# rpmbuild -ta slurm*.tar.bz2
# cd ~/rpmbuild/RPMS/x86_64
# yum install slurm-2.3.4-1.el6.x86_64.rpm slurm-devel-2.3.4-1.el6.x86_64.rpm slurm-
plugins-2.3.4-1.el6.x86_64.rpm slurm-munge-2.3.4-1.el6.x86_64.rpm
```

c. Generate your config file. You can do this by going to
https://computing.llnl.gov/linux/slurm/configurator.html
and saving the result to /etc/slurm/slurm.conf
However, here is my conf file for your convenience:

*** Please note you may need to adjust the Sockets and CoresPerSocket parameters based on your hardware configuration. Also, please note that the last two lines in the file begin with "NodeName" and "PartitionName" respectively. I don't want the word wrap to make you think there should be line breaks where there shouldn't be)***

```
#-- /etc/slurm/slurm.conf

#
# Example slurm.conf file. Please run configurator.html
# (in doc/html) to build a configuration file customized
# for your environment.
#
#
# slurm.conf file generated by configurator.html.
#
# See the slurm.conf man page for more information.
#
ClusterName=Warewulf
ControlMachine=wwmaster
ControlAddr=10.10.10.1
#BackupController=
#BackupAddr=
#
SlurmUser=slurm
#SlurmdUser=root
SlurmctldPort=6817
SlurmdPort=6818
AuthType=auth/munge
#JobCredentialPrivateKey=
#JobCredentialPublicCertificate=
StateSaveLocation=/tmp
SlurmdSpoolDir=/tmp/slurmd
SwitchType=switch/none
MpiDefault=none
SlurmctldPidFile=/var/run/slurmctld.pid
SlurmdPidFile=/var/run/slurmd.pid
ProctrackType=proctrack/pgid
#PluginDir=
CacheGroups=0
#FirstJobId=
ReturnToService=0
#MaxJobCount=
#PlugStackConfig=
#PropagatePrioProcess=
#PropagateResourceLimits=
#PropagateResourceLimitsExcept=
#Prolog=
#Epilog=
#SrunProlog=
#SrunEpilog=
#TaskProlog=
#TaskEpilog=
#TaskPlugin=
#TrackWCKey=no
#TreeWidth=50
#TmpFs=
#UsePAM=
#
# TIMERS
SlurmctldTimeout=300
SlurmdTimeout=300
InactiveLimit=0
MinJobAge=300
KillWait=30
Waittime=0
#
# SCHEDULING
SchedulerType=sched/backfill
#SchedulerAuth=
#SchedulerPort=
#SchedulerRootFilter=
SelectType=select/linear
FastSchedule=1
#PriorityType=priority/multifactor
#PriorityDecayHalfLife=14-0
```

```
#PriorityUsageResetPeriod=14-0
#PriorityWeightFairshare=100000
#PriorityWeightAge=1000
#PriorityWeightPartition=10000
#PriorityWeightJobSize=1000
#PriorityMaxAge=1-0
#
# LOGGING
SlurmctldDebug=3
#SlurmctldLogFile=
SlurmdDebug=3
#SlurmdLogFile=
JobCompType=jobcomp/none
#JobCompLoc=
#
# ACCOUNTING
#JobAcctGatherType=jobacct_gather/linux
#JobAcctGatherFrequency=30
#
#AccountingStorageType=accounting_storage/slurmdbd
#AccountingStorageHost=
#AccountingStorageLoc=
#AccountingStoragePass=
#AccountingStorageUser=
#
# COMPUTE NODES
NodeName=n00[00-16] Sockets=2 CoresPerSocket=6 ThreadsPerCore=2 State=UNKNOWN
PartitionName=allnodes Nodes=n00[00-16] Default=YES MaxTime=INFINITE State=UP

#ALL OOF THE NODES ABOVE MUST BE RESOLVABLE OR ELSE SLURM WILL NOT START!
```

d. Start Slurm and configure to start automatically each boot

```
# /etc/init.d/slurm start
# chkconfig slurm on
```

e. Install Slurm and configure it to start automatically on the nodes

```
# cd ~/rpmbuild/RPMS/x86_64
# yum --installroot=/var/chroots/centos6 install slurm-2.3.4-1.el6.x86_64.rpm slurm-
plugins-2.3.4-1.el6.x86_64.rpm slurm-munge-2.3.4-1.el6.x86_64.rpm

# cp -p /etc/slurm/slurm.conf /var/chroots/centos6/etc/slurm/slurm.conf
# chroot /var/chroots/centos6
# chkconfig slurm on
# mv /etc/rc3.d/S90slurm /etc/rc3.d/S97slurm
# exit
```

f. Resave the image, reboot the nodes,

```
# wwvnfs --chroot /var/chroots/centos6
# pdsh -w n00[00-99] reboot
```

g. Check that SLURM sees the nodes

```
# scontrol show nodes

          -- If nodes' state shows as DOWN, run the following: --

# scontrol update  NodeName=ClusterNode0 State=Resume

# scontrol show nodes
```

6. Installing the HPL Benchmark

*** This procedure assumes that you have installed the EPEL repo on both the master and nodes, as described in Section 2: Installing Warewulf.*

a. Install the following Linear Algebra libraries and HPL prereqs on the master:

```
# yum install atlas atlas-devel blas blas-devel lapack lapack-devel compat-gcc*
# yum --installroot=/var/chroots/centos6 install compat-gcc*
```

b. Get the source for the benchmark and move it into the proper folder (to make our lives easier later)

```
# cd ~
# wget http://www.netlib.org/benchmark/hpl/hpl-2.0.tar.gz
# tar xfvz hpl-2.0.tar.gz
# mv hpl-2.0 hpl
# cd hpl
# cp setup/Make.Linux_PII_CBLAS .
```

c. Edit the Makefile (~/hpl/Make.Linux_PII_CBLAS) , giving it the location of OpenMPI and the ATLAS/BLAS. (changes in red)

```
# ----------------------------------------------------------------------
# - Message Passing library (MPI) --------------------------------------
# ----------------------------------------------------------------------
# MPinc tells the  C  compiler where to find the Message Passing library
# header files,  MPlib  is defined  to be the name of  the library to be
# used. The variable MPdir is only used for defining MPinc and MPlib.
#
MPdir        = /usr/lib64/openmpi
MPinc        = -I$(MPdir)
MPlib        = /usr/lib64/libmpi.so
#
# ----------------------------------------------------------------------
# - Linear Algebra library (BLAS or VSIPL) -----------------------------
# ----------------------------------------------------------------------
# LAinc tells the  C  compiler where to find the Linear Algebra  library
# header files,  LAlib  is defined  to be the name of  the library to be
# used. The variable LAdir is only used for defining LAinc and LAlib.
#
LAdir        = /usr/lib64/atlas
LAinc        =
LAlib        = $(LAdir)/libcblas.a $(LAdir)/libatlas.a
#
```

d. Then, compile the benchmark and copy the new binary to your user's home folder

```
# make arch=Linux_PII_CBLAS
# cp bin/Linux_PII_CBLAS/xhpl /home/rbc
# chown rbc.rbc /home/rbc/xhpl
```

e. Next you need to configure your HPL.dat file. Just plug your node specifications into the following online tool, and it will generate the file for you:

http://www.advancedclustering.com/faq/how-do-i-tune-my-hpldat-file.html

f. Once xhpl and HPL.dat are both in your user's home directory, you may run the benchmark in one of two ways:

i. Using OpenMPI directly:

i. Create a file named 'machines' in the user's home folder, and give it the following contents, adjusting for your own nodes

```
#-- /home/rbc/machines --

n0000
n0001
n0002
n0003
n0004
n0005
n0006
n0007
```

ii. Next, invoke the benchmark using mpirun, and make sure the –np attribute equals the number of your nodes, multiplied by the number of cores in each node. (eg, 8 nodes x 2 cpus x 4 cores = 64)

```
# su - rbc
$ mpirun --machinefile machines -np 64 ./xhpl
```

b. Using SLURM:

i. Allocate the resources for the benchmark and make sure the –n attribute equals the number of your nodes, multiplied by the number of cores in each node. (eg, 8 nodes x 2 cpus x 4 cores = 64)

```
$ salloc -n64 sh
```

ii. Inside the provided shell, call the benchmark using mpirun. Note: No parameters are needed – as they are already inferred from the allocation we did previously.

```
$ mpirun ./xhpl
```

iii. Here is a sample HPL.out file generated by the benchmark. The score in this example for our sample cluster was 328 Gflops.

```
================================================================================
HPLinpack 2.0  --  High-Performance Linpack benchmark  --   September 10, 2008
Written by A. Petitet and R. Clint Whaley,  Innovative Computing Laboratory, UTK
Modified by Piotr Luszczek, Innovative Computing Laboratory, UTK
Modified by Julien Langou, University of Colorado Denver
================================================================================

An explanation of the input/output parameters follows:
T/V    : Wall time / encoded variant.
N      : The order of the coefficient matrix A.
NB     : The partitioning blocking factor.
P      : The number of process rows.
Q      : The number of process columns.
Time   : Time in seconds to solve the linear system.
Gflops : Rate of execution for solving the linear system.

The following parameter values will be used:

N      :  150000
NB     :     128
PMAP   : Row-major process mapping
P      :       8
Q      :       8
PFACT  :   Right
NBMIN  :       4
NDIV   :       2
RFACT  :   Crout
BCAST  :  1ringM
DEPTH  :       1
SWAP   : Mix (threshold = 64)
L1     : transposed form
U      : transposed form
EQUIL  : yes
ALIGN  : 8 double precision words

--------------------------------------------------------------------------------

- The matrix A is randomly generated for each test.
- The following scaled residual check will be computed:
      ||Ax-b||_oo / ( eps * ( || x ||_oo * || A ||_oo + || b ||_oo ) * N )
- The relative machine precision (eps) is taken to be               1.110223e-16
- Computational tests pass if scaled residuals are less than                16.0

================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR11C2R4      150000   128     8     8            6859.11              3.280e+02
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0019772 ...... PASSED
================================================================================

Finished      1 tests with the following results:
              1 tests completed and passed residual checks,
              0 tests completed and failed residual checks,
              0 tests skipped because of illegal input values.
--------------------------------------------------------------------------------

End of Tests.
================================================================================
(END)
```