This guide is the result of our implementation of a separate Storage Element server at Florida Tech.

Listed are some important acronyms or terms related to the OSG Middleware.

- **VDT - Virtual Data Toolkit**: The package that contains and installs all OSG software.
- **DN - Distinguished Name**: A personal identity linked to a grid certificate.
- **VO - Virtual Organization**: An organization that is responsible for a set of DN.
- **GUMS - Grid User Management System**: A piece of software that maps grid users (DN) to local cluster accounts.
- **CE - Compute Element**: System that runs most of the critical software (e.g. globus, apache, RSV).
- **SE - Storage Element**: System that runs a storage manager (such as BeStMan or dCache), sometimes is the same as the CE.
- **BeStMan - Berkeley Storage Manager**: System that manages globus file transfers in and out of the cluster.
- **PhEDEx - Physics Experiment Data Exports**: System that allows CMS grid users to request data transfers to/from particular sites and admins to control the data flow for their site.
- **Squid - Squid Web Proxy Cache**: System that stores important CMS job data on the local cluster where all running jobs can access it.

1. **System Setup**

For our SE I used the Development node commissioned for jobs that require large amounts of memory. As a result, the machine has 8 Xeon cores and 64GB RAM, as well as a 1TB RAID5 array. I decided to reconfigure this machine to be our storage element, given the high memory overhead of the storage managers.

First I set up the node and kickstart graph profiles for the Rocks appliance “dev” which we created. The instructions for creating a Rocks appliance can be found at:

http://www.rocksclusters.org/roll-documentation/base/5.0/customization-adding-appliance.html

The node and graph xml files can be found in the Scripts area on the NAS. Basically they automatically install the JDK1.6 (required for BeStMan to work), and create links from the new appliance to the NAS appliance (effectively making the machine a NAS) but also adding X11 and devel software - which could come in handy for certain software.

As a simple installation, the machine will not have external web access. I had to perform several critical steps to get this to work.
First, I followed the instructions on this page.
http://www.rocksclusters.org/roll-documentation/base/5.0/customization-extra-nic.html

Specifically I used the following commands for our machine:
# rocks list host interface dev-0-0
# rocks set host interface subnet dev-0-0 eth1 public
# rocks set host interface name dev-0-0 eth1 uscms1-se.fltech-grid3.fit.edu
# rocks set host interface gateway dev-0-0 eth1 163.118.206.254

And also had to make sure that the SE was not set to be a compute node (otherwise it
would be included in cluster-fork operations.
# /opt/rocks/bin/mysql -u root -p
# show databases;
# use cluster;
# SELECT * FROM memberships;
# UPDATE memberships SET Compute = 'no' where Name = 'Dev Appliance';

In addition to setting up the external interface, I had to make some modifications to the
/etc/hosts file on the SE. Below is the new file. Notice that the external interface is first.
127.0.0.1 localhost.localdomain localhost
163.118.206.157 uscms1.fltech-grid3.fit.edu
163.118.206.162 uscms1-se.fltech-grid3.fit.edu uscms1-se
10.255.255.230 dev-0-0.local dev-0-0

I also had to change the hostname of the SE from dev-0-0.local to uscms1-se.fltech-
grid3.fit.edu in order for Gridftp to work properly. This was done by editing the
HOSTNAME field in /etc/sysconfig/network.

After these tweaks, the network configuration of the SE should be enough to run the
storage software. However I performed a few tweaks of the TCP buffers as well:
Edit /etc/sysctl.conf
With:
# increase TCP max buffer size
net.core.rmem_max = 4194304
net.core.wmem_max = 4194304
# increase Linux autotuning TCP buffer limits
net.ipv4.tcp_rmem = 4096 87380 4194304
net.ipv4.tcp_wmem = 4096 65536 4194304

After setting up the network, I sought to find a better way to enable XFS support on the
SE (for the 1TB array). I used the following method, which should supersede the method
described in the CE installation:
Enable the Extras repository in /etc/yum.repos.d/CentOS-Base.repo
Then
# yum list available kmod-xfs*
# yum install kmod-xfs.x86_64 xfsdump xfsprogs

This allowed mounting the big disk to /sandbox as XFS. I also added the entry to /etc/
fstab just like on the NAS (for automounting at boot).
I also edited /etc/exports on the NAS (nas-0-0) to allow access to the dev node (if it
wasn’t done already), and then added the automount in /etc/fstab (on the SE) so that the /
nas0 big disk would be automatically accessible by the SE. An example of these two /etc/
fstab entries is:
nas-0-0.local:/nas0 /mnt/nas0 nfs defaults 0 0
/dev/sdb1 /sandbox xfs defaults 1 2

2. Squid Setup

The frontier-squid (CMS deployed squid server) is very easy to set up.
Download the latest version of the frontier-squid source.
# su dbfrontier
$ mkdir /sandbox/squid
$ cd /sandbox/squid
$ wget http://frontier.cern.ch/dist/frontier_squid-4.0rc9.tar.gz
$ tar xvzf frontier_squid-4.0rc9.tar.gz
$ cd frontier_squid-4.0rc9
$ ./configure

Make the network/netmask:
163.118.206.157/255.255.255.255,10.255.255.255/255.255.255.0
And choose memory and cache appropriate to your system.

Then I just ran:
$ make
$ make install
# /sbin/chkconfig --add frontier-squid.sh

Now would be a good time to update iptables with the Squid and SE related ports.
These ports are:
#### Squid
tcp 3128
udp 3401
#### 3ware
tcp 888
#### OSG
tcp 8443, 8080, 2811, 40000:45000 IN
Finally (as dbfrontier):
$ /etc/init.d/frontier-squid.sh start

3. BeStMan Setup

I installed a host and http certificate on the SE following these instructions:
https://twiki.grid.iu.edu/bin/view/ReleaseDocumentation/GetGridCertificates

Following the directions to install the VDT BeStMan is acceptable:
https://twiki.grid.iu.edu/bin/view/ReleaseDocumentation/BestmanGateway

Be sure to follow all OSG1.2 instructions.
Install pacman like with the CE, and source the setup script. I did the following:

# mkdir /sandbox/osg-se
# cd /sandbox/osg-se
# export VDT_GUMS_HOST=uscms1.fltech-grid3.fit.edu
# pacman -get http://software.grid.iu.edu/osg-1.2:Bestman
# source setup.sh
# $VDTLOCATION/vdt/bin/vdt-ca-manage setupca --location local --url osg
# vdt-post-install

# $VDTLOCATION/vdt/setup/configure_bestman --server y \
 --user daemon \
 --cert /etc/grid-security/containercert.pem \
 --key /etc/grid-security/containerkey.pem \
 --http-port 8080 \ 
 --https-port 8443 \ 
 --gums-host uscms1.fltech-grid3.fit.edu \ 
 --gums-port 8443 \ 
 --enable-gateway \ 
 --with-allowed-paths=/mnt/nas0/OSG/BeStMan \ 
 --with-transfer-servers gsiftp://uscms1-se.fltech-grid3.fit.edu

Be sure to follow the instructions for adding the SRM_CMD lines to /etc/sudoers

# cp $VDTLOCATION/post-install/prima-authz.conf /etc/grid-security
# cp $VDTLOCATION/post-install/gsi-authz.conf /etc/grid-security

Add to $VDTLOCATION/vdt/etc/vdt-local-setup.sh:
GLOBUS_TCP_SOURCE_RANGE=40000,45000
GLOBUS_TCP_PORT_RANGE=40000,45000
export GLOBUS_TCP_SOURCE_RANGE
export GLOBUS_TCP_PORT_RANGE

# source $VDT_LOCATION/setup.sh
# vdt-control --enable gsiftp
# $VDT_LOCATION/vdt/setup/configure_gratia --probe gridftp-transfer --report-to
# http://gratia-osg.fnal.gov:8880/gratia-reporting/ --probe-cron --site-name FLTECH
# vdt-control -on

Note that I had to manually install xinetd service from an RPM to start BeStMan.

4. PhEDEx Setup

The following instructions are useful.
https://twiki.cern.ch/twiki/bin/view/CMS/PhedexAdminDocsInstallation

Note that I had to use a hacked Bootstrap script to install the slc4 files on CentOS5, this script is located in the Scripts directory on the NAS.

$ su phedex
$ mkdir /sandbox/phedex
$ cd /sandbox/phedex
$ mkdir -p state logs sw gridcert
$ chmod 700 gridcert
$ export sw=$PWD/sw
$ export version=3_2_0
$ export myarch=slc4_amd64_gcc345
$ sh -x bootstrap-slc4_amd64_gcc345.CentOS.5.sh setup -path $sw
$ source /sandbox/phedex/sw/slc4_amd64_gcc345/external/apt/0.5.15lorg3.2-CMS3/etc/profile.d/init.sh
$ apt-get update
$ apt-cache search PHEDEX | grep PHEDEX+PHEDEX
$ apt-get install cms+PHEDEX+PHEDEX_$version
$ rm -f PHEDEX; ln -s $sw/$myarch/cms/PHEDEX/PHEDEX_$version PHEDEX

Be sure to check for the latest version of PhEDEx before installing and change the
$version variable accordingly.
Now install the srmclient and the Java JRE (correct JRE RPM is in the RPMs directory on the NAS).

$ mkdir $sw/$myarch/external/srm
$ wget http://www.dcache.org/downloads/1.8.0/dcache-srmclient-1.8.0-4.noarch.rpm
$ rpm -i --nodeps -v --prefix $sw/$myarch/external/srm ./dcache-srmclient-1.8.0-4.noarch.rpm
$ echo . $sw/$myarch/external/srm/srm/conf/init.sh >> PHEDEX/etc/profile.d/dependencies-setup.sh
$ echo "export SRM_PATH=$sw/$myarch/external/srm/srm
$ export PATH=$sw/$myarch/external/srm/srm/bin:`PATH
$ export SRM_CONFIG=$sw/$myarch/external/srm/srm/conf/config.xml
(multiline command)
$ if [ ![ -f $SRM_CONFIG ]; then
  srmcp -save_conf=$SRM_CONFIG srm://qq/ww srm://ww//qq
cat $SRM_CONFIG | sed 's/<retry_num> 20/<retry_num> 0/'> 
$SRM_CONFIG.anew
mv $SRM_CONFIG.anew $SRM_CONFIG
fi" > $sw/$myarch/external/srm/srm/conf/init.sh
$ source $sw/$myarch/external/srm/srm/conf/init.sh

$ chmod +x jre-6u4-linux-x64.bin
$ mkdir $sw/$myarch/external/java
$ cd $sw/$myarch/external/java
$ ~/jre-6u4-linux-i586.bin
$ mkdir $sw/$myarch/external/java/etc/init.d/
$ echo "export JAVA_INSTALL_PATH=$sw/$myarch/external/java/jre1.6.0_04
export JAVA_HOME=$sw/$myarch/external/java/jre1.6.0_04
export PATH=$sw/$myarch/external/java/jre1.6.0_04/bin:`PATH" > 
$sw/$myarch/external/java/etc/init.d/init.sh
$ echo . $sw/$myarch/external/java/etc/init.d/init.sh >> PHEDEX/etc/profile.d/dependencies-setup.sh
$ touch $sw/$myarch/external/java/etc/init.d/init.sh

Now get the configuration templates:

$ cd /sandbox/phedex
$ export CVSROOT=:pserver:anonymous@cmscvs.cern.ch:/cvs_server/repositories/CMSSW
$ cvs login # password is "98passwd"
$ cvs co SITECONF/T3_US_FIT

The configuration files for our PhEDEx instance are in the Scripts directory of the NAS. They need to be placed in SITECONF/T3_US_FIT/PhEDEx/

Set up a long term proxy (simplest option) with the grid user certificate.

$ grid-proxy-init -valid 720:00
$ cp /tmp/x509up_u`id -u` /home/phedex/gridcert/proxy.cert
After registering our node with CMS and registering my personal grid user cert with CMS. Phedex can be started. Note that I had to configure a Config.Debug (almost identical to Config.Prod) and start the Debug instance to do loadtests before Production is authorized.

```
$ cd /sandbox/phedex
$ ./PHEDEX/Utilities/Master -config SITECONF/T3_US_FIT/PhEDEx/Config.Prod start
```

The log file Prod_T3_US_FIT/logs/download-srm should be watched for any errors as jobs start. I had many of these errors at the start, but our finalized configuration should not give any operational errors - only perhaps authentication ones. If job errors occur, you can navigate to the individual job logs as their locations are shown in the above log file. PhEDEx needs to be manually started after a reboot.