

Installing sims (lsst stack) on carver at NERSC

Michael Wood-Vasey update

1. Installed "v10_0"ish stack to "lsst_apps"
2. Installed most of "lsst_sims -t sims" as of 2015 March 5

I started with Simon's very usefully recorded prior experience below and then set about installing the v10_0 version of the stack.

My default shell is bash. Almost all of the same issues that Simon originally raised still apply.

- default python version (syntax errors, need 2.7)
- default gcc version (can't compile boost)
- default git version (no http)
- no m5 on nodes
- no gettext
- compiling afw exceeds login node memory limit

Related notes.

- numpy needs to be >1.9.0 so I had updated Anaconda to 2.1.0 and couldn't just use the v1.9.0 installed from last summer.
- flex is now available on the nodes. m5 is not.

1. Log in to NERSC:

```
ssh carver.nersc.gov
```

2. Make a working directory in the project space:

```
mkdir -p /project/projectdirs/m1727/lsst_S14/lsstStack
```

3. setup the needed modules:

```
module swap pgi gcc
module load git
```

5. Install Anaconda

```
cd /project/projectdirs/m1727/lsst_S14/
curl -s -L -o installer.sh http://repo.continuum.io/archive/Anaconda-2.1.0-Linux-x86\_64.sh
bash installer.sh -b -p anaconda
```

5. Install Gettext

```
mkdir src
cd src
curl -O http://ftp.gnu.org/pub/gnu/gettext/gettext-0.19.tar.gz
tar zxvf gettext-0.19.tar.gz
cd gettext-0.19
./configure --prefix $PWD/../../
make install
```

6. Setup environment

```
export PATH=/project/projectdirs/m1727/lsst_S14/bin:/project/projectdirs/m1727/lsst_S14/anaconda/bin:${PATH}
export PYTHON=/project/projectdirs/m1727/lsst_S14/anaconda/bin/python
source "/project/projectdirs/m1727/lsst_S14/lsstStack/loadLSST.bash"
```

7. Install the basic stack

```
cd /project/projectdirs/m1727/lsst_S14/lsstStack
curl -O https://sw.lsstcorp.org/eupspkg/newinstall.sh
bash newinstall.sh
#Answer no to install git? and install anaconda?
source "/project/projectdirs/m1727/lsst_S14/lsstStack/loadLSST.bash"
```

8. Start compiling

Either

```
eups distrib install lsst_apps
```

or

```
eups distrib install lsst_sims -t sims
```

This will go along fine until afw, at which point it will die because it will exceed the virtual memory available on the login node.

So we'll go to a compute node to finish afw

```

qsub -I -q interactive -l walltime=00:30:00 #need this long since afw takes a while to compile
cd /project/projectdirs/m1727/lsst_S14
module swap pgi gcc
module load m4

export PATH=/project/projectdirs/m1727/lsst_S14/bin:/project/projectdirs/m1727/lsst_S14/anaconda/bin:${PATH}
export PYTHON=/project/projectdirs/m1727/lsst_S14/anaconda/bin/python

source lsstStack/loadLSST.bash

#So here this gets a little tricky because as of (2015-03-05) "lsst_sims -t sims" requires an afw version that's different than "lsst_apps" current
#If you're installing lsst_apps [current], then just
eups distrib install afw
#If you're installing lsst_sims -t sims as of 2015-03-05 then
eups distrib install afw master-gcb7650adda

Finish up the install after exiting the interactive allocation
Either

eups distrib install lsst_apps

or

eups distrib install lsst_sims -t sims

...wait...a...very...long...time

setup lsst_apps

or

setup lsst_sims -t sims

This last succeeded except for "sims_catUtils", "sims_maf", and "lsst_sims".

To use this lsst_apps stack under bash:

export PATH=/project/projectdirs/m1727/lsst_S14/anaconda/bin:$PATH
source /project/projectdirs/m1727/lsst_S14/lsstStack/loadLSST.bash
setup lsst_apps

=====

In summary
ssh carver.nersc.gov

mkdir -p /project/projectdirs/m1727/lsst_S14/lsstStack

module swap pgi gcc
module load git

cd /project/projectdirs/m1727/lsst_S14/
curl -s -L -o installer.sh http://repo.continuum.io/archive/Anaconda-2.1.0-Linux-x86\_64.sh
bash installer.sh -b -p anaconda

mkdir src
cd src
curl -O http://ftp.gnu.org/pub/gnu/gettext/gettext-0.19.tar.gz
tar zxvf gettext-0.19.tar.gz
cd gettext-0.19
./configure --prefix $PWD/../../
make install

export PATH=/project/projectdirs/m1727/lsst_S14/bin:/project/projectdirs/m1727/lsst_S14/anaconda/bin:${PATH}
export PYTHON=/project/projectdirs/m1727/lsst_S14/anaconda/bin/python

cd /project/projectdirs/m1727/lsst_S14/lsstStack
curl -O https://sw.lsstcorp.org/eupspkg/newinstall.sh
bash newinstall.sh
#Answer no to install git? and install anaconda?
source "/project/projectdirs/m1727/lsst_S14/lsstStack/loadLSST.bash"

eups distrib install lsst_apps

```

```
qsub -I -q interactive -l walltime=00:30:00 #need this long since afw takes a while to compile
# The following are executed on the interactive node
cd /project/projectdirs/m1727/lsst_S14
module swap pgi gcc
module load m4
export PATH=/project/projectdirs/m1727/lsst_S14/bin:/project/projectdirs/m1727/lsst_S14/anaconda/bin:${PATH}
export PYTHON=/project/projectdirs/m1727/lsst_S14/anaconda/bin/python
source lsstStack/loadLSST.bash
```

```
eups distrib install afw
```

```
exit # log out of the interactive node session
```

```
#Finish up the install after exiting the interactive allocation
```

```
eups distrib install lsst_apps
```

```
=====
```

To use this stack to:

```
setenv PATH /project/projectdirs/m1727/lsst/anaconda/bin:$PATH
```

```
source /project/projectdirs/m1727/lsst/lsstStack/loadLSST.csh
```

```
setup lsst_sims -t sims
```

```
=====
```

[Simon Krughoff](#) originally put this together 2014 June.

This was not as straight forward as I had hoped. Issues were:

- default python version (syntax errors, need 2.7)
- default gcc version (can't compile boost)
- default git version (no http)
- no flex on nodes
- no m5 on nodes
- no gettext
- compiling afw exceeds login node memory limit

How I did it. I tried to use bash, but my default shell is tcsh and I had some shell issues (path to module command wasn't defined):

1. Log in to NERSC: `$> ssh krughoff@carver.nersc.gov`
2. make a working directory in the project space: `$> mkdir -p /project/projectdirs/m1727/lsst/lsstStack`
3. setup the needed modules: `$> module swap pgi gcc; module load git`
4. get anaconda:


```
$> cd /project/projectdirs/m1727/lsst;
$> curl -s -L -o installer.sh http://repo.continuum.io/archive/Anaconda-1.9.1-Linux-x86_64.sh
$> bash installer.sh -b -p anaconda
```
5. get gettext:


```
$> mkdir src
$> cd src
$> curl -O http://ftp.gnu.org/pub/gnu/gettext/gettext-0.19.tar.gz
$> tar zxvf gettext-0.19.tar.gz
$> cd gettext-0.19
$> ./configure --prefix $PWD/../../
$> make install
```
6. Setup environment:


```
$> cd /project/projectdirs/m1727/lsst
$> setenv PATH ${PWD}/bin:${PWD}/anaconda/bin:$PATH
$> setenv PYTHON /project/projectdirs/m1727/lsst/anaconda/bin/python
```
7. Install basic stack:


```
$> module load git # Default git is too old
$> cd lsstStack
$> curl -O sw.lsstcorp.org/eupspkg/newinstall.sh
$> bash newinstall.sh #Answer no to install git? and install anaconda?
$> source "/project/projectdirs/m1727/lsst/lsstStack/loadLSST.csh"
```
8. Start by installing afw. Can't do this on the nodes because flex is not installed and is not available on carver as a module. This will die when trying to install afw, but we can log on to a node to finish the compile of afw.


```
$> eups distrib install afw
...wait...for...a...long...time
After it dies, spin up an interactive node.
$> qsub -I -q interactive -l walltime=00:30:00 #need this long since afw takes a while to compile
```
9. Compile afw on the node:


```
$> cd /project/projectdirs/m1727/lsst
```

```
$> module swap pgi gcc
$> module load m4
$> setenv PATH ${PWD}/bin:${PWD}/anaconda/bin:$PATH
$> setenv PYTHON `which python`
$> source lsstStack/loadLSST.csh
$> eups distrib install afw
10. Finish up the install after exiting the interactive allocation
$> eups distrib install lsst_sims -t sims
...wait...a...very...long...time
$> setup lsst_sims -t sims
11. Done.
```

To use this stack to:

```
$> setenv PATH /project/projectdirs/m1727/lsst/anaconda/bin:$PATH
$> source /project/projectdirs/m1727/lsst/lsstStack/loadLSST.csh
$> setup lsst_sims -t sims
```