

NCCS Brown Bag Series



NCCS User Getting-Started Package

Chongxun (Doris) Pan

doris.pan@nasa.gov

January 15, 2013



Welcome!



Congratulations on your NCCS user account!

This presentation will guide you through all the basics of using NCCS's computing, storage, data analysis systems, and services. Links are included for some topics where in-depth discussions are available. **New users are recommended to read through the entire package prior to login.**

We are here for YOU. With an emphasis on enabling science and providing user-oriented systems and services, we encourage you to ask a lot of questions of NCCS Support!

Email to support@nccs.nasa.gov

or

Call 301-286-9120 Monday through Friday 8am-6pm Eastern



NCCS Getting Started Package - Roadmap

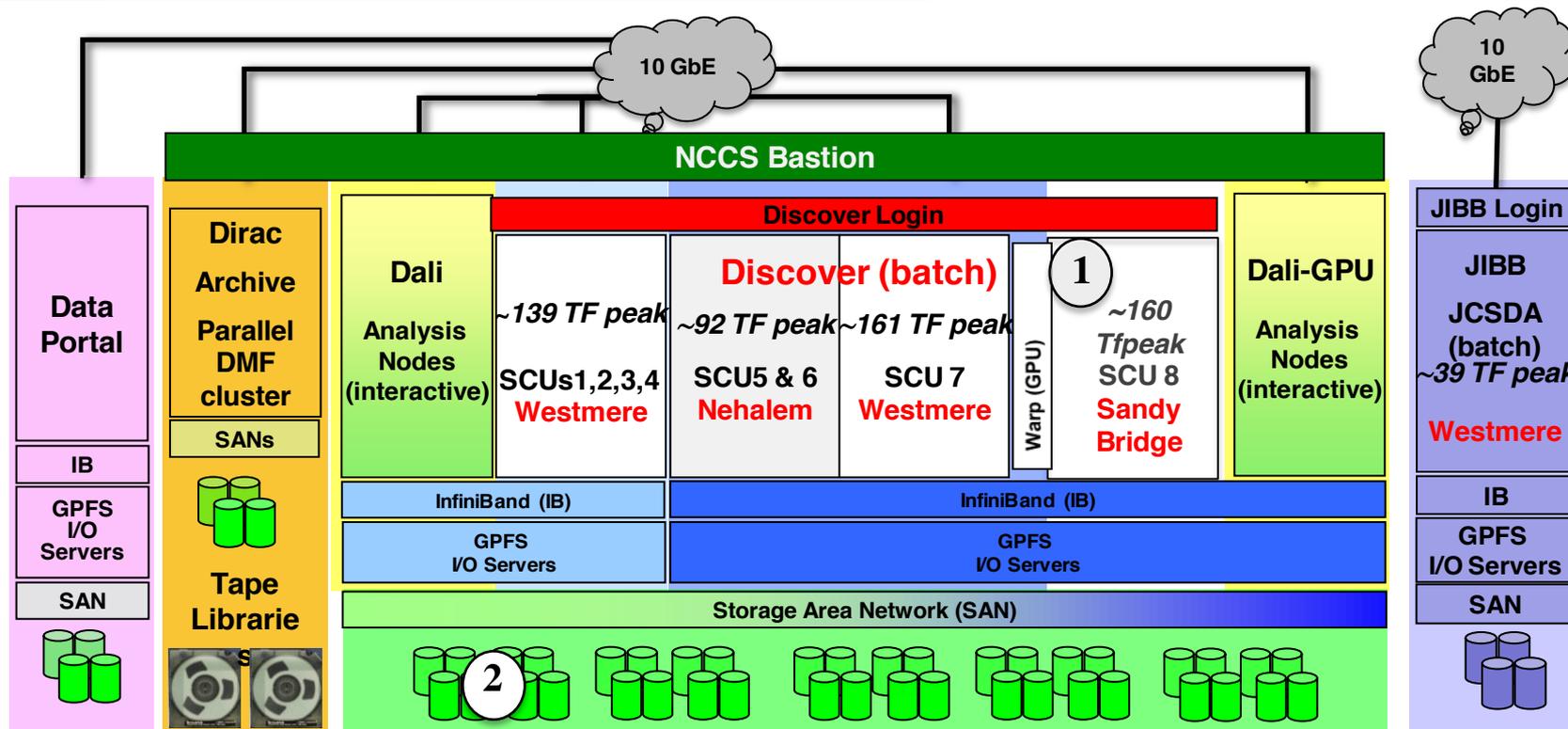


- ◆ **NCCS Systems Overview**
- ◆ **Systems and Components**
- ◆ **System Login**
- ◆ **Files and Data**
 - ◆ Discover & Dali File Systems
 - ◆ GPFS
 - ◆ Node-local scratch space
 - ◆ Archive file systems
 - ◆ Managing Your Files
 - ◆ Data Sharing
 - ◆ Quota Limits
 - ◆ File Transfer
 - ◆ To and from NCCS Systems
 - ◆ Between Dirac Archive & Discover/Dali
 - ◆ Archive Best Practice

- ◆ **Discover Linux Cluster and Dali**
 - ◆ Default Shell
 - ◆ Cron Jobs
 - ◆ Module
 - ◆ Compilers
 - ◆ MPI Libraries
 - ◆ Intel Math Kernel Libraries
 - ◆ Standard Billing Units
 - ◆ Specs of Discover and Dali
 - ◆ Running Jobs via PBS
 - ◆ Common PBS commands
 - ◆ PBS Resource Request Examples
 - ◆ Interactive-batch PBS Jobs
 - ◆ Running PBS Jobs Best Practice
 - ◆ Licensed Application Software
 - ◆ Open Source Software Packages



NCCS Systems Overview



① New SCU8 includes 480 Sandy Bridge nodes and Xeon Phi co-processors.

② New 4PB disk augmentation.



Systems and Components



➔ Computing

- ◆ **Discover**: A Linux cluster containing multiple generations of Intel processors
- ◆ **Dali or Dali-gpu** nodes with large memory are designed for interactive, large-scale data analysis. They share the same file system with Discover

➔ Mass Storage/Archive

- ◆ **Dirac**: A number of SGI servers that uses DMF to manage long term archive data, currently 30+ PB on tapes. Users can also access Dirac via NFS mounted file systems on Discover/Dali

➔ Data Portal

- ◆ Providing public access to some NCCS data and services



System Login



➔ From outside, two ways to access the NCCS systems:

1. Standard login mode:

```
ssh -XY user_id@login.nccs.nasa.gov
```

PASSCODE: PIN followed by the six-digit TOKENCODE

host: discover (or dali, dali-gpu, dirac)

password: your-LDAP-password

2. Proxy Mode:

Recommended for command-line users. **On your local workstation, create a `$HOME/.ssh/config` file as follows**

```
$ cat $HOME/.ssh/config  
host discover dirac dali dali-gpu discover.nccs.nasa.gov  
dirac.nccs.nasa.gov  
  LogLevel Quiet  
  ProxyCommand ssh -l user_id login.nccs.nasa.gov direct %h  
  Protocol 2  
  ConnectTimeout 30  
$ ssh -XY user_id@discover (or)  
$ ssh -XY user_id@dali-gpu
```

Click [here](#) for details



System Login



- From inside the NCCS, you can ssh to other NCCS systems without entering either a PASSCODE or password, e.g., “ssh dirac” from Discover.

- **But first make sure** you set up password-less ssh/scp within NCCS systems.

- If your RSA token is locked, call Enterprise Service Desk 301-286-3100 for token reset

- Change your NCCS LDAP password at this link

https://www.nccs.nasa.gov/LDAP_pwchange_Gateway.php

```
On Discover/Dali:  
$ chmod -R 0700 $HOME/.ssh  
$ cd $HOME/.ssh  
$ ssh-keygen -t dsa (hit enter two times)  
$ cat id_dsa.pub >> authorized_keys  
$ scp $HOME/.ssh/id_dsa.pub  
user_id@dirac:~/.ssh/id_dsa.pub.discover
```

```
On Dirac:  
$ cat $HOME/.ssh/id_dsa.pub.discover >>  
$HOME/.ssh/authorized_keys
```



So many different passwords!



➔ **IDMAX password**

NASA Identity and Access Management password, used for IT access, training, or updating personal information in NASA public information directory

➔ **NDC password**

Password used to access your nasa.gov email and NASA-issued laptop/desktop

➔ **RSA token passcode**

Your given PIN followed by a six-digit TOKENCODE shown in your SecurID key to access the NCCS firewall

➔ **NCCS LDAP password**

Password to access the NCCS systems.

The NCCS Support is only able to help with LDAP password issues.



Files and Data



Discover and Dali File Systems



- GPFSS (Global Parallel File System). Accessible from all Discover login and compute nodes and Dali nodes.
 - ◆ `$HOME` or `/home/user_id`
 - ◆ Quota controlled. **User disk space limit is 1GB.**
 - ◆ Fully backed up. Ideal for storing source codes and scripts.
 - ◆ `$NOBACKUP` or `/discover/nobackup/user_id`
 - ◆ Quota controlled for both the disk space and number of files (inodes)
 - ◆ **User limit is 250GB and 100K inodes**
 - ◆ NOT backed up. Long-term storage data should be moved to the archive system.
- ◆ Use the command “`showquota`” to check usage on `$HOME` and `$NOBACKUP`. See [here](#) for details
- ◆ `/usr/local/other` stores most of open source packages, tools, and libraries.



File Systems (*Cont'd*)



- Node-specific local scratch file system
 - ◆ Access via `$LOCAL_TMPDIR`. Fast performing file system, but NOT global
 - ◆ Consider using it if your applications create/read/write a large number of small-size files
 - ◆ Files generated in `$LOCAL_TMPDIR` should be copied to `$NOBACKUP` once the job completed. Files under `$LOCAL_TMPDIR` are scrubbed periodically
- Archive file system on Dirac
 - ◆ `$ARCHIVE`, or `/archive/u/user_id`, mounted on Discover/Dali via NFS with 1Gigabit network connectivity
 - ◆ You can also access `$ARCHIVE` data by ssh/scp to dirac via 10 Gigabit network connectivity
 - ◆ **No specified user quota for mass storage space. Only inode limit,250K, is enforced**
 - ◆ Recommend looking at the presentation, [Your data on Tape](#), for useful details on how to use the archive system efficiently



Managing Your Files



- \$NOBACKUP is NOT backed up. It is your responsibility to copy valuable data to either \$HOME, \$ARCHIVE, or to remote systems
- \$NOBACKUP or /discover/nobackup/*user_id* is a symlink that points to the actual disk your nobackup directories reside, e.g., /gpfs/dnbxx/*user_id*. **ALWAYS use the symlink in your scripts to specify paths**, because the actual path may be changed due to disk augmentations or system events.
- The quota system on Discover/Dali (using “**showquota**” command) differs from that on Dirac (using “**quota**” command)



Data Sharing



- ➔ A common way to share files/directories with group members and others is to change permissions using *chmod* command

```
$ ls -l
drwx----- 2 cpan2 k3001 8192 2013-01-07 16:17 tmp/
$ chmod -R go+rx tmp | ls -l
drwxr-xr-x 2 cpan2 k3001 8192 2013-01-07 16:17 tmp/
$ chmod -R o-rx tmp | ls -l
drwxr-x--- 2 cpan2 k3001 8192 2013-01-07 16:17 tmp/
$ groups cpan2
cpan2 : k3001 k3002
$ chgrp -R k3002 tmp | ls -l
drwxr-x--- 2 cpan2 k3002 8192 2013-01-07 16:17 tmp
```

- ➔ File permissions are retained when you copy data from Discover to Dirac, regardless what your umask setting is on Dirac.
- ➔ **Do NOT make files/directories world-writable.** Think twice before making files/directories world-readable. If you have a specific need to share data with group members or others, send a ticket to NCCS Support and we will help you.



Quota limits



- Two kinds of quotas are enforced:
 - ◆ **Limits on the total disk space occupied** by a user or a group's files on either \$HOME and \$NOBACKUP
 - ◆ **Limits on how many files (inodes)** a user can store on \$NOBACKUP and \$ARCHIVE, irrespective of size. For quota purposes, directories count as files

- Two types of quota limits are in place:
 - ◆ **Hard limits** -- can never be exceeded. Any attempt to use more than your hard limit will be refused with an error
 - ◆ **Soft limits** -- can be exceeded temporarily. When you exceed your soft limit, you can continue to use your account normally -- but the grace period would begin and you would have a limited amount of time to bring usage back below the soft limit value. Failure to do so within the grace period will cause the soft limit to become a hard limit.



File Transfer to and from NCCS Systems



- To copy data from a remote system to NCCS, a user must use the **Bastion Service Proxy Mode**
- For command line users:
 - ◆ **Initiating commands from a remote system:**
 1. Make sure to first set up for the Proxy Mode, i.e., the `$HOME/.ssh/config` file is created on the remote system (described in Slide 6)
 2. Use either **scp**, **sftp**, or **rsync** from the remote system:
 - `scp -r user_id@dirac.nccs.nasa.gov:~/mydir .`
 - `sftp user_id@discover.nccs.nasa.gov`
 - `rsync -auPv ~/mydir user_id@dali:/discover/nobackup/user_id/mydir`
 - ◆ **Initiating commands from Discover/Dali to pull/push data from a remote system is also possible**
- For WinScp users:
 - ◆ Check [here](#) for configuration details
 - ◆ Or, look at [this brownbag presentation](#)



File Transfer between Dirac and Discover/Dali



- ◆ It is recommended to use the cluster gateway node (the datamove queue) to transfer large files instead of using the Discover/Dali front end nodes.
- ◆ The gateway node has 10 Gigabit interface and larger memory to handle multiple and large file transfers

On Discover/Dali:

```
$ qsub -l -q datamove -l walltime=01:00:00
```

```
qsub: waiting for job 922008.borgpbs1 to start .....ready
```

```
borggw06 $ cp /archive/u/myplace/bigfile /discover/nobackup/myplace/
```

- ◆ Alternative ways to move files include:
 - ◆ bbcp or scp from Discover/Dali login nodes to Dirac
 - ◆ bbcp or scp from the gateway node to Dirac
 - ◆ cp to \$ARCHIVE from Discover/Dali login nodes

Click [here](#) to find more on the above mentioned methods and their performance comparisons



Archive File System Best Practice



- ➔ **It is better to store a few large files than many small ones in the archive.** An example to pipe the tar file directly to the archive through a shell command:

```
tar zvcf - ./work | ssh dirac "cat > /archive/u/myuserid/work.tar.gz"
```

- ➔ **Do not untar anything in the archive area.** If you do, you will create many small files in the archive area, which will then be written to many tapes.
- ➔ If you are transferring data from Dirac, make sure your file is online (i.e., on disk) first, not just on tape. Use “dmis -l” to check the state of the file; if it is offline, use “dmget” to recall it. Click [here](#) for details.
- ➔ If you use NFS to move a file to the archive, **we strongly recommend using cp instead of mv.** If there is an error in the created archive version you may lose data with mv. Remove the file after verification. Click [here](#) for details.



Discover Linux Cluster and Dali



Default Shell



- ➔ “echo \$SHELL” to check your default shell
- ➔ To change the default shell, contact NCCS Support

Shell	Startup files to edit
sh or ksh	\$HOME/.profile
bash	\$HOME/.bashrc if it exists; or \$HOME/.bash_profile if it exists; or \$HOME/.profile if it exists (in that order)
csh	\$HOME/.cshrc
tcsh	\$HOME/.tcshrc if it exists; or \$HOME/.cshrc if it exists (in that order)

- ➔ Click [here](#) for example startup files for bash and csh/tcsh



Cron Jobs



- ➔ Manage your cron jobs at [discover-cron](#). [discover-cron](#) is an alias for a Discover login-style node that runs cron. “ssh discover-cron” from any of the Discover and Dali nodes.
- ➔ If you want to run a cron job on Dali nodes to take advantage of their large memory, then you should setup a cron command like this:

```
1 00 * * * ssh dali <command> 1>> FULLPATH/output 2>&1
```

or

```
1 00 * * * ssh dali <script_name> 1>> FULLPATH/output 2>&1
```

- ➔ See [here](#) for detailed examples of crontabs.



Module



- ◆ The “module” command allows you choose compilers, libraries, and packages to create/change your own personal environment.
- ◆ When you initially log into the NCCS system, **no modules** are loaded by default
- ◆ Common Module commands

module avail (av)	Display a complete list of available modules
module list	Display loaded modules
module load <i>module_name1</i> ...	Load new modules
module purge	Unload all loaded modules
module swap <i>old_name new_name</i>	Switch between different versions of software
module show <i>module_name</i>	Display the environmental variables set by the module

- ◆ Use the module commands in either your shell startup file, your job script, or at the command line, depending on which sessions you want the change to take effect in



Compilers



- ➔ To accommodate the needs of a broad range of user groups, multiple versions of compilers from different vendors are provided:

GNU	Default 4.3.4 (usr/bin/gcc), OR module load other/comp/gcc-* (check "module av" for versions)
Intel	module load comp/intel-* (check "module av" for versions)
PGI	module load comp/pgi-* (check "module av" for versions)
NAG	module load comp/nag-5.3-886
CUDA	module load other/gpgpu/cuda/* (check "module av" for versions)

- ➔ Click [here](#) for a few commonly used compiler options for the GNU, Intel, and PGI compilers



MPI Libraries



- Various MPI implementations available on Discover. Prior to loading an MPI module, you will have to load an appropriate module for a supported compiler suite.

Vendor	Modules	Supported Compilers
Intel MPI	mpi/impi-*	Intel Compiler only
OpenMPI	mpi/openmpi/*	GNU, Intel, and PGI compilers
MVAPICH2	other/mpi/mvapich2-*	GNU, Intel, and PGI compilers

- For new users, we recommend starting with Intel compiler and Intel MPI, for example,

`module load comp/intel-12.1.0.233 mpi/impi-4.0.1.007-beta`

- **Do NOT need to add “-Impi” to link your program with the MPI library.** Always invoke “mpif90” or “mpicc” to compile and link MPI programs, e.g., `mpif90 -o foo foo_mpi.f90`



Intel Math Kernel Library (MKL)



- Intel MKL is the primary numerical libraries with comprehensive math functionality, including BLAS, LAPACK, FFTs, Vector math, Statistics, and data fitting.
- MKL libraries are already included in LD_LIBRARY_PATH if you use Intel Compiler version ≥ 11
- If you use Intel Compiler version 10, PGI, or GNU compiler, and want to use MKL, you will need to load an MKL module, lib/mkl-*, e.g.,

`module load lib/mkl-10.1.2.024`



Running Jobs via PBS



- PBS is a distributed workload management system that handles the computational workload on Discover.
- To access the compute nodes on Discover, you must submit jobs to the batch queues, managed by PBS.

Queues	Wall Time Limit	CPUs allowed per job	Max jobs per user
debug	1 hr	1-32	4
general_small	12 hr	1-24	50
general	12 hr	17-1024	20
general_long	24 hr	1- 516	2

- In order to run multi-node PBS jobs, you have to set up a **\$HOME/.ssh/authorized_keys** file. See Slide 8.



Common PBS command



- Either a Shell script or a Python script are allowed by PBS. To submit a job:

`qsub <job_script>`

- To list the available queues as well as their status and some basic information:

`qstat -q`

- To list detailed info for a particular queue, e.g.,

`qstat -Qf general`

- To list the status of all jobs from a particular user:

`qstat -u <user_id>`

- To list the status of a particular job, e.g.,

`qstat 1672381.pbsa1` (Or simply just, `qstat 1672381`)

- To delete a job, e.g.,

`qdel 1672381.pbsa1` (Or simply just, `qdel 1672381`)



Standard Billing Units (SBUs)



- ➔ Computer resource allocations are quantified with SBUs. You can no longer run batch jobs if your allocated SBUs are used up.
- ➔ Command to check SBU balance and CPU hours used is:
`/usr/local/bin/allocation_check`
- ➔ Usage of Discover login nodes and Dali nodes is NOT charged against SBUs
- ➔ SBU rates differ on various types of nodes, including Nehalem, Westmere, and Sandy Bridge, **so be mindful of the type of the nodes your jobs are running on for efficient usage of your allocations.**



Specs of Discover compute and Dali nodes



Node Type	Cores per Node	Memory per node	Memory per core	Swap Space per node	Environment
Nehalem	8	24 GB	3 GB	8 GB	PBS11
Westmere	12	24 GB	2 GB	8 GB	PBS11
Sandy Bridge	16	32 GB	2 GB	8 GB	PBS11
Warp Westmere	12	48 GB	4 GB	8 GB	PBS11
Dali (01-08)	16	256 GB	16 GB	--	No PBS
Dali-gpu (09-20)	12	192 GB	16 GB	--	No PBS

Jobs will be killed when using $\geq 60\%$ of the swap space on one or more nodes.



Resources Request Examples



➔ Click [here](#) for example PBS job scripts to run serial, OpenMP, and MPI applications

➔ Resource request for a 128-CPU job:

```
#PBS -l select=16:ncpus=8:mpiprocs=8
```

“**ncpus**” decides the type of node you request (at least 8 cores per node in the above example). “**mpiprocs**” decides how many MPI ranks per node you want to run your application with.

➔ Adding “**proc=**” may be necessary if you intend to run on a certain type of nodes.

```
#PBS -l select=16:ncpus=8:mpiprocs=8:proc=neha  
#PBS -l select=11:ncpus=12:mpiprocs=12:proc=west  
#PBS -l select=16:ncpus=12:mpiprocs=8:proc=west  
#PBS -l select=8:ncpus=16:mpiprocs=16  
...  
mpirun -np 128 ./foo.exe
```

proc=sand is unnecessary in this case because currently the SandyBridge nodes are the only choice satisfying ncpus=16



Interactive-batch PBS Jobs



- An interactive-batch job is a regular batch job, but allows you to get on to the compute nodes. Very useful for debugging and computational steering.

```
$ xsub -I -l select=4:ncpus=12:proc=west,walltime=2:00:00 -q general
```

```
Establishing X forwarding and submitting batch job...
```

```
qsub: waiting for job 845848.pbsa1 to start
```

```
qsub: job 845848.pbsa1 ready
```

```
borge107:$ xterm &
```

(Now you are on the headnode. And you can open another terminal!)

```
borge107:$ cat $PBS_NODEFILE
```

```
borge107.prv.cube
```

```
borge108.prv.cube
```

```
borge118.prv.cube
```

```
borge119.prv.cube
```

```
borge107:$ mpirun -perhost 6 -np 24 ./GEOSgcm.x
```

```
borge107:$ mpirun -perhost 1 -np 4 killall -v GEOSgcm.x
```

(in case you want to kill the previous executions and restart the mpirun)



Running PBS Jobs Best Practice



- Pay attention to monitor job memory usage. Jobs that require more memory than physically available on the nodes requested would crash the nodes or even cause system issues. To prevent that from happening, users are highly recommended to look at [this presentation](#)
- You can run multiple serial jobs concurrently, using the PoDS utility, within a single batch job to reduce wall time. Click [here](#) to see details.
- When your PBS job is running, its standard output and error files are kept in `/discover/pbs_spool/<job_id>.OU` or `.ER` files. Monitor the files to check the progress of the job.



Licensed Application Software



- A few licensed applications from different vendors are installed on the NCCS systems:

- ◆ **Matlab : Must be used on Dali nodes ONLY**

```
$ ssh dali-gpu
```

```
$ module load tool/matlab-R2011b
```

To find out how many licenses are currently in use, issue:

```
$ /discover/vis/matlab/matlab_r2008a/etc/lmstat -a
```

- ◆ **IDL: Must be used on Dali nodes ONLY**

```
$ module load tool/idl-8.1
```

```
$ /discover/vis/itt/idl/idl81/bin/lmstat -a
```

- ◆ **TOTALVIEW**

```
$ module load tool/tview-8.9.2.2
```



Open Source Software Packages



- A variety of open source software packages are installed under (most recent first):
 - `/usr/local/other/SLES11.1`
 - `/usr/local/other/SLES11`
 - `/usr/local/other/`
- After each system OS upgrade, some software are recompiled. Users should always try to use the more recent build of a software.
- With few exceptions, e.g. Python or gcc, you can use most of third party software directly WITHOUT loading modules
- A user may request installing a new package through NCCS Support



Open Source Software Packages (*Cont'd*)



- An inventory of the packages (frequently updated) is maintained [here](#)
- Here are a few commonly used software:
 - ◆ **Python** : Python distributions for scientific computing based on various versions of Python are available via various modules, e.g.,
[module load other/SIVO-PyD/spd_1.7.0_gcc-4.5-sp1](#)
Find detail [at this presentation](#)
 - ◆ **HDF4** and **HDF5** : /usr/local/other/SLES11.1/hdf4 (hdf5)
 - ◆ **Netcdf3** and **Netcdf4** : /usr/local/other/SLES11.1/netcdf (netcdf4)
 - ◆ **R** : /usr/local/other/SLES11.1/R/gcc-4.6.3/bin/R
 - ◆ **GrADS** : Version 2.0.1.oga.1 -- /discover/nobackup/projects/gmao/share/dasilva/opengrads/Contents/opengrads
 - ◆ **NCL** : /usr/local/other/SLES11/ncarg/bin/ncl. **And make sure to add the two libs into LD_LIBRARY_PATH,**
/usr/local/other/SLES11/jpeg/6b/intel-11.1.069/lib:/usr/local/other/SLES11/udunits2/2.1.23/intel-11.1.069/lib
 - ◆ **NCO** : /usr/local/other/SLES11/nco/4.0.8/intel-11.1.069/bin



Questions?

We are here for YOU

Email to support@nccs.nasa.gov

or

Call 301-286-9120 Monday through Friday 8am-6pm Eastern

Check NCCS Primer for up-to-date User Guide

<http://www.nccs.nasa.gov/primer/>