Getting Started on Maverick

Botanical Information and Ecology Network
University of Arizona,
Nathan Casler
ncasler@email.arizona.edu

April 29, 2014

Abstract

Reference Material for *BIEN Species Range and Future Forest Distribution Modeling* at TACC

1 Accessing TACC

Once you receive access to a TACC server, you can use an SSH client such as PuTTY or OpenSSH to log onto your allocation.

```
ssh yourname@maverick.tacc.utexas.edu
```

Once logged in, you will see documentation about Maverick along with numbers regarding disk space/usage in your home and work directories. The work directory should be equivalent to the /scratch directory from the last project. To load the project data you can clone the code repository with:

```
git clone git@github.com:npcasler/bien-range
```

The repository will provides version control project codebase. You can use `git pull` in your clone of the repository to load any code updates.

2 Loading Modules

TACC functionality is module-based, allowing users to load modules depending on their needs. The list of available modules can be found using `module avail`. `module load Rstats` will load the R Statistical Library which will allow you to start an interactive session with `R`.

3 Processing with SLURM

TACC uses the Simple Linux Utility for Resource Management (SLURM) for job scheduling and cluster management. SLURM works to schedule(squeue), call(srun) and terminate (scancel) and monitor(sinfo) processes running on computing clusters. From our discussions with John Fonner and the Brian McGill, it looks like we will be interacting more with the TACC Parametric launcher than SLURM directly. For more detailed information, SLURM documentation can be found at slurm.schedmd.com.

3.1 Parametric Launcher

The Bien Species Range Modeling project relied heavily on the TACC parametric launcher to queue and manage jobs last time around. The parametric launcher has been upgraded since the last run, but it seems most of the syntax has remained unchanged. The parametric launcher consists of several files:

launcher.slurm
A SLURM batch submission script that requests resources

paramlist
A plain-text file containing the tasks to execute(1 per line)

paramrun
Central caller that calls assigns init_launcher an individual node

init_launcher
Caller for various launch scripts on an individual node

launcher
Calls a script using parameters from a paramlist entry

The paramlist structure is as follows:
`program [args...]  [<infile] [> outfile]`
The parametric launcher also relies on a series of user-defined parameters to manage the amount of resources dedicated to each job. These environmental variables are defined in the task layout. Examples of the environmental variables are:

TACC_LAUNCHER_NHOSTS The number of hosts for the job

TACC_LAUNCHER_PPN The number of tasks per node for this job

TACC_LAUNCHER_JID The id of the current job(line in paramlist)

TACC_LAUNCHER_TSK_ID The id of the current task

These parameters can be set in the launcher.slurm script with the following format:

```
#----------------Scheduler Options--------------
#SBATCH -J Parametric # Job Name
```

```
#SBATCH -N 1 # Total number of nodes(16 cores/node)
#SBATCH -n 16 # Total number of tasks
#SBATCH -p normal-mic # Queue Name
#SBATCH -o Parametric.o%j # Name of stdout output file
#SBATCH -t 01:00:00 # Run time (hh:mm:ss)
# <----------Account String--------------->
```

More detailed information regarding the use of the parametric launcher can be found at: `https://github.com/TACC/launcher`

4 Installing External Packages

Although TACC has a large variety of pre-built modules at our disposal, we will still need to build and install some packages locally. For example, the **GDAL Library** is a requirement for several of the R libraries used in our scripts including *rgdal*, and *raster*.

You can download the source from:
`http://download.osgeo.org/gdal/1.11.0/gdal-1.11.0.tar.gz`
and push to your *Home* directory using scp or pscp if you have putty installed.
Once the binary is download and unpacked, you can move into the gdal directory and run the following commands:

```
cd gdal
./configure --prefix=/home/your-account/gdal/
make
make install
```

We are still in the process of making local R libraries recognize the local package installations, which will be added to this documentation.