Putting all pieces together - an R example

<table>
<thead>
<tr>
<th>Quick Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPC Home</td>
</tr>
<tr>
<td>Getting an account</td>
</tr>
<tr>
<td>Getting started on Prince</td>
</tr>
<tr>
<td>Prince How-to Articles</td>
</tr>
<tr>
<td>Logging in Windows</td>
</tr>
<tr>
<td>Mac / Linux</td>
</tr>
<tr>
<td>Clusters and Storage</td>
</tr>
<tr>
<td>Prince (HPC)</td>
</tr>
<tr>
<td>Dumbo (Hadoop)</td>
</tr>
<tr>
<td>Dalmatia (NYU Abu Dhabi)</td>
</tr>
<tr>
<td>Transferring data to/from the clusters</td>
</tr>
</tbody>
</table>
Transferri ng data to/from Prince cluster using Globus

Submitting jobs with sbatch

Available software

Licensed Software Available on the HPC Cluster

Building Software

Slurm Tutorial

Tutorials

FAQs

Scratch Area Cleanup

Programming for Biologists

Research Gallery

HPC People

HPC Policies

(Tip: click "<<" at bottom left to close Confluence sidebar)
<table>
<thead>
<tr>
<th>Running jobs on the Prince Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessing the Prince Cluster</td>
</tr>
<tr>
<td>From Windows workstation</td>
</tr>
<tr>
<td>From Mac workstation</td>
</tr>
<tr>
<td>Software and Environment Module</td>
</tr>
<tr>
<td>Job script and resource request</td>
</tr>
<tr>
<td>Introduction to job scheduling</td>
</tr>
<tr>
<td>Submitting jobs with sbatch</td>
</tr>
<tr>
<td>Requesting resources</td>
</tr>
<tr>
<td>Using computing nodes interactively</td>
</tr>
<tr>
<td>Monitoring batch jobs</td>
</tr>
<tr>
<td>Monitoring batch jobs - squeue</td>
</tr>
<tr>
<td>What is running and where?</td>
</tr>
<tr>
<td>slurmtop</td>
</tr>
<tr>
<td>Canceling your jobs</td>
</tr>
<tr>
<td>Compiling your own software</td>
</tr>
</tbody>
</table>
Putting all pieces together

An Amber example
A R example

Summary

Pulling it all together - Preparing, submitting and monitoring a job on Prince

In this section we will prepare, submit and monitor a small R job. Our test case comes from the NYU Data Services "Introduction to R" tutorial

Exercise
Start a terminal session on Mercer and replicate this example in it.

Choose your own example
After - or instead of - following this example through, prepare and submit a run of something genuinely relevant to your research. This way, if you are doing this tutorial in a classroom, the presenter will be available should you have questions or strike difficulties.

We're using R, so first we'll look for available modules. On mercer:

```
$ module avail r
----------------------------- /share/apps/modulefiles
---------------------------------
  gstreamer/intel/1.10.2    mothur/intel/1.35.1    r/intel/3.3.2
```

There's a few modules starting with r, and a couple of versions of R. We'll use the latest version, 3.1.2.

```
$ module purge
$ module list
No Modulefiles Currently Loaded.
$ module load r/intel/3.3.2
```

Take a look at what it did:

```
$ module list
Currently Loaded Modulefiles:
  1) jdk/1.8.0_111  2) intel/17.0.1  3) openmpi/intel/2.0.1  4) r/intel/3.3.2
```

... clearly, R uses a lot of other packages. The modulefile has looked after loading the correct ones.
$ module show r/intel/3.3.2
--------------------------------------------------------------------------------------
-----
/share/apps/modulefiles/r/intel/3.3.2.lua:
--------------------------------------------------------------------------------------
-----
whatis("R: a language and environment for statistical computing and graphics")
whatis("Name: r version: 3.3.2 compilers: intel")
load("intel/17.0.1")
load("jdk/1.8.0_111")
load("openmpi/intel/2.0.1")
prepend_path("MANPATH","/share/apps/r/3.3.2/intel/share/man")
prepend_path("PATH","/share/apps/r/3.3.2/intel/bin")
prepend_path("LD_LIBRARY_PATH","/share/apps/r/3.3.2/intel/lib64/R/lib")
prepend_path("PKG_CONFIG_PATH","/share/apps/r/3.3.2/intel/lib64/pkgconfig")
setenv("R_ROOT","/share/apps/r/3.3.2/intel")
setenv("R_INC","/share/apps/r/3.3.2/intel/lib64/R/include")
setenv("R_LIB","/share/apps/r/3.3.2/intel/lib64/R/lib")
family("R")

For our example, we'll get some code and data from /share/apps/examples:

Hint: there are usage examples for a few common packages here

$ mkdir /beegfs/$USER/R-example
$ cd !$
$ cp /share/apps/examples/r/basic/* .

Take a look at the job script:

$ cat my_R_job.s
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --time=5:00
#SBATCH --mem=1GB
#SBATCH --job-name=myRtest
#SBATCH --mail-type=END
#SBATCH --mail-user=bob.smith@nyu.edu

module purge
module load r/intel/3.3.2

RUNDIR=$SCRATCH/R-example
mkdir -p $RUNDIR

cp /share/apps/examples/r/basic/example.R $RUNDIR

cd $RUNDIR
srun R CMD BATCH example.R example.out

There are a few steps we can try here:

1. Start an interactive batch session, and run the example.R script interactively
2. Close the interactive session, and submit the batch script as a job:
$ sbatch my_R_job.s

You'll get a job id returned.

Is it running yet?

$ squeue -u $USER

You could watch the output in the run directory:

$ ls -l ${SCRATCH}/R-example

Finally, when the job finishes, you should see a .out file in the directory you submitted from.

**Exercise**

Experiment with sbatch options for the job name, output and error file merging and location, resource limits.