Using R on the HPC

Alex Townsend

Introduction to R

- R is a statistical analysis tool and a programming language.
- R is widely used for data analysis in a variety of fields.
- R is free and open source!

Using R on the HPC

- R is available on both the Spear and HPC Systems.
  - There are two versions of R available.
    - The default version can be loaded by typing the following into the terminal
      - `R`
    - The other version, which is the most up-to-date and is recommended for use can be loaded with the following commands:
      - `module load R`
      - `R`

Using R on the HPC

- You can run R on the Login Nodes
  - This limits you to one node and up to 24 cores.
- You can also submit your R job to the Compute Nodes.
  - This allows you to use multiple nodes, each of which have a large number of cores, usually between 24 and 48.
  - This also allows you to use GPUs.

The Advantages of R on HPC

- R can be run normally on any laptop. Why use HPC?
  - Parallel Computing!
    - The parallel package ~ Multicore
    - The rmpi package ~ Distributed
  - GPU Computing!
    - gpuR and tensorflow Packages

Today's Example

- For the example today, we will be tuning a simple R script using a built-in dataset of SAT and ACT scores. This will go in 2 steps.
  - Step 1: Write a simple script for Serial Computation
  - Step 2: Modify the script for Multicore Computation

Step 1

```r
# load the Package with the Data
library(psych)

# Get the SAT and ACT Score Dataset
dataset <- sat.act

# Clean the Dataset (Remove Missing Data)
cleandata <- na.omit(dataset)

# Now get the Column Means for the data
means <- colMeans(cleandata, na.rm=TRUE)

# Now compute the Column-Wise Variances for the data
vars <- apply(cleandata, 2, var)
```

Step 1

```
#!/bin/bash

#SBATCH --job-name="R Template"


#SBATCH --mail-type=ALL

#SBATCH -n 24

#SBATCH -p genacc_q

#SBATCH -t 14-00:00:00









module load R


R CMD BATCH parallel_script.R
```

Step 2

```r
# load the Package with the Data
library(psych)

# Load the PARALLEL library for Multicore Computing!
library(parallel)

# Get the SAT and ACT Score Dataset
dataset <- sat.act

# Clean the Dataset (Remove Missing Data)
cleandata <- na.omit(dataset)

# Determine how many cores you have available
cores <- detectCores() - 1
print(cores)

# Build a virtual "cluster" out of these cores
cluster <- makeCluster(cores)

# Now get the Column Means for the data
means <- parLapply(cluster, cleandata, mean)
print(means)

# Now compute the Column-Wise Variances for the data
vars <- parLapply(cluster, cleandata, var)
print(vars)

# Now stop the cluster and release the resources
stopCluster(cluster)
```

Step 2

```
#!/bin/bash

#SBATCH --job-name="R Template"


#SBATCH --mail-type=ALL

#SBATCH -n 24

#SBATCH -p genacc_q

#SBATCH -t 14-00:00:00









module load R


R CMD BATCH parallel_script.R
```

Advanced Topics

- R also has packages available to make that same script useful for Distributed Computation
  - For this, we would use the Rmpi package.

Advanced Topics

- R also has packages available to make that same script useful for GPU Computation
  - gpuR and tensorflow