RStudio, rslurm, haven and WRDS

Steven Finch, sfinch@mit.edu

Research Computing Specialist, Sloan Technology Services

It is possible to submit parallel batch jobs to the Engaging HPC cluster from within RStudio, using an R package called rslurm. It is further possible to read *.sas7bdat files into RStudio, using an R package called haven. These packages together enable us to efficiently perform analysis on several files from the Wharton Research Data Services site. In our demonstrations, the R program resides on Engaging and accesses a copy of WRDS elsewhere on Engaging.

To open an interactive RStudio session on Engaging, you will need to use an X forwarding client such as MobaXterm (for Windows) or XQuartz (for Mac). With this in place, the SLURM commands

```
module load engaging/rstudio-desktop/0.98.1103
module load sloan/R/CRAN
salloc --cpus-per-task=2 --mem=8G -p sched_mit_sloan
rstudio
```

open a new GUI window (resembling the familiar RStudio console) into which you can type commands and perform analysis. Here we've requested 2 cores on the cluster using a total of 8 GB of RAM (hence 4 GB per core).

Here is the RStudio program for the demo:

rslurm_haven-eo.txt

I recommend that you copy and paste at most several lines at a time, e.g., select just the "library(rslurm)" command and, only when finished executing in R, select the "library(haven)" command.

The demo contains three tasks labeled f, g and h. I recommend performing f_read, g_read and h_read initially: these are all parallel batch jobs, governed by slurm_apply

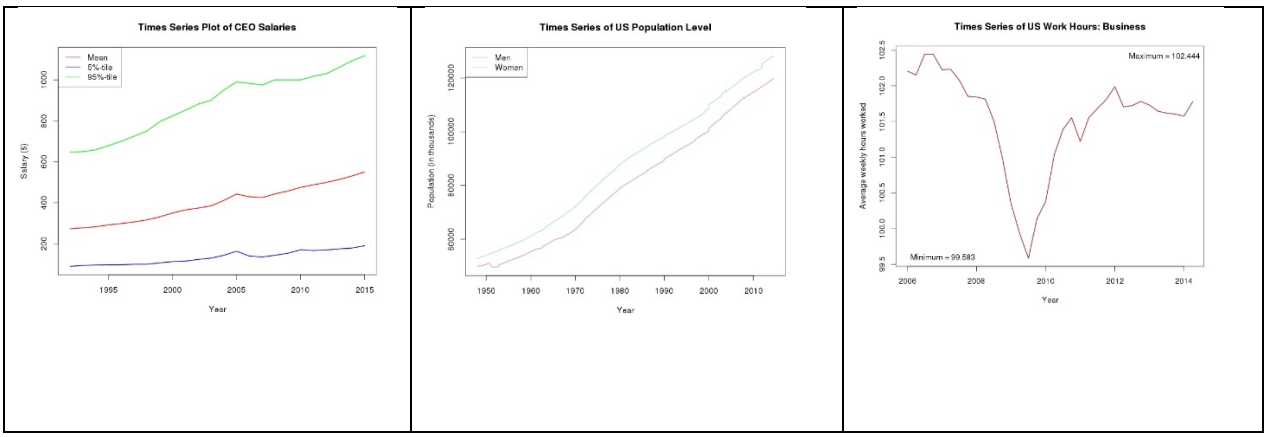
within rslurm. The data assimilation is quickly done by read_sas within haven. The dataframes pars_f, pars_g and pars_h can be used to pass parameters to the parallel processing (e.g., different values for different scenarios), but in this case, they serve only to assist in distributing the computation among various Engaging nodes. For f_read and h_read, exactly one dataset is assimilated; for g_read, exactly ten are assimilated via a "for" loop structure. The dataset for h_read is especially large; the h_read jobs will continue running in the background even after f_read and g_read are completed.

The print_job_status() function, applied to sjob_f, will indicate when f_read is finished. We could also check the progress via sacct in SLURM; the whole point is that shifting from RStudio to SLURM and back again repeatedly is no longer necessary. Everything can be done conveniently within RStudio. Do not apply the get_slurm_out() command to sjob_f until f_read has completed. It pieces together the various results_*.RData files gathered in the new _rslurm_f_read folder within your home directory; the wildcard character * represents integers 0, 1, 2, 3,…, each corresponding to a separate parallel processing job. Once done, we define a dataframe AC (for "Annual Compensation"), compute CEO salary statistics for 2015 and create a time series plot for salaries over a twenty year span.

Do not apply the get_slurm_out() command to sjob_g until g_read has completed. Once done, we define three dataframes bls (for "Bureau of Labor Statistics"), ln (for "Labor Force Statistics: NAICS") and pr (for "Major Sector Productivity"), leaving the remaining seven dataframes for your enjoyment. For the first part (bls), we print a table and compute a correlation coefficient. For the second part (ln), we create a time series plot for the US population over a sixty year span, stratified by gender. For the third part (pr), we create a time series plot for US work hours in the business sector from 2006 to 2014. The data underlying our three plots have differing units of time – yearly, monthly and quarterly – and incorporating such frequency scales in our code is an important detail. We use the ts() function to convert a numeric dataframe into an R time series object especially appropriate for graphing.

Do not apply the get_slurm_out() command to sjob_h until h_read has completed. Several minutes would be required just to assemble the full SC dataframe (which is why we examine only a subset): of many possible avenues for analysis of the scorecard of US colleges, we focus on men's-only and women's-only colleges and list those schools with a competitive admission rate.

The PDF output file fgh.pdf contains three plots. All text output (tables and statistics) are written to the console. If you wish to explore other statistical aspects of the WRDS datasets, simply edit my RStudio code.



Please feel free to write to me (sfinch@mit.edu) with questions and suggestions on how to improve this tutorial.