Weekly Journal

Leila Uy

May 24, 2021

1 Work Update

This past week, I focused mainly on reading the assigned material and practicing parallel programming in R. The week was busy with personal matters like my friend's wedding and holidays so I was not able to read up on GRASS or thoroughly read the code base as much I hoped to.

1.1 Parallel programming

1.1.1 My Computer Specs and AWS

One of the first thing that any parallel R tells you to do is call `detectCores(logical = True)` and `detectCores(logical = False)`. This function should tell the number of available hardware threads and cores. For my computer, I have 4 threads and 2 processors. Since I have only 2 cores, I have been very careful on trying small parallel programming tasks on my laptop, but Jishnu and I have been discussing and for now we are hoping to test future, bigger code on his computer with smaller sets up data before moving to AWS.

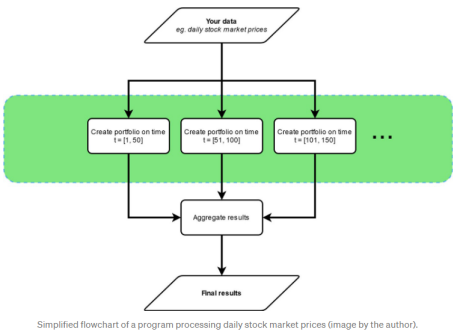
1.1.2 R

I went through **many** resources for parallel R. Some resources were good [1, 2, 3, 5] and some were bad. One of the main things that I discovered is that parallel R will work differently depending on the OS of your computer. For example, I was encountering problems with some functions in the parallel package in Windows. The function `mclapply` will not work in parallel because of how workers are managed. In addition, a socket cluster and fork cluster are different and forking will only work in a Unix-like system [2].

1.1.3 Code Base

I looked at the code base to get an idea of what it is trying to achieve and the steps it takes to achieve it, but one of the things I want to do with the code base in the future is go line-by-line in-depth. From there, I want to create a simple flowchart of the code and find the best place to preform parallelism similar to the figure below.

Figure 1: Flowchart of a simple parallel process [1]



Simplified flowchart of a program processing daily stock market prices (image by the author).

Also, I read about Amdahl's Law which is often used in parallel computing to predict theoretical speedup with multiple processors. [2] I really want to try predicting the speedup of parallelization and timing the proces afterwards to see if the prediction was close.

1.2 Checklist

So you know my current thought process, this is what I am hoping to look into doing next.

- Comment through the code base
- Experiment with parallel GRASS with parallel R
- Create a flow chart for a parallel version of the code base
- Time the serial version of the code base and calculate Amdahl's Law

2 Literature Review

Aside from the assigned readings, I did not look for other academic journals or publications but mainly blogs/videos/tutorials. The IPCC has many useful reports and resources that will be useful in the future like the "Climate Change and Land" special report for August 2019. Regarding the emissions scenerio report [4], it gave me the idea that it would be interesting to create predictive maps of land change using several factors. The emissions scenerio paper mentioned 3 main driving forces for the emissions scenerios: population, economic development, and structural and technological change. For example, the paper mentions the expansion of biomass energy production in places like Asia, Africa, and Latin America.

In addition, it would interesting to map emissions from the past along with our land change maps. It would be interesting to create a visual representation of the land-use emissions the article talks about like CO2 for deforestation and CH4 for farming. There is also the increase of GHG emissions from land use change and the decrease of emissions from developed countires.

References

[1] Imre Gera. Parallelization caveats in r #1: performance issues. https://towardsdatascience.com/parallelization-caveats-in-r-1-the-basics-multiprocessing-and-multithreading-performance-eb584b7gi=15eed98ffb65, 2020.

[2] Michael Hallquist. Parallel computing in r. https://psu-psychology.github.io/r-bootcamp-2018/talks/parallel_r.html#conceptual_overview_of_parallel_computing, 2018.

[3] Soren Martius. How to use multithreading and multiprocessing — a beginner's guide to parallel and concurrent programming. https://medium.com/mineiros/how-to-use-multithreading-and-multiprocessing-a-beginners-guide-to-parallel-and-concurrent-a69b9 2020.

[4] Nebojsa Nakicenovic and Rob Swart. Emissions scenarios. 2000.

[5] Tinakarimi. Parallel processing with r on windows. https://waterprogramming.wordpress.com/2020/03/16/parallel-processing-with-r-on-windows/, 2020.