

Weekly Journal

Leila Uy

May 24, 2021

1 Work Update

For this past week, I focused mainly on reading the assigned material and practicing parallel programming in R. I also had a brief meeting with Diljot to help her with GitHub.

1.1 Parallel programming

1.1.1 My Computer Specs and AWS

The first function that any parallel R tutorial tells you to call is `detectCores(logical = True)` and `detectCores(logical = False)`. This function returns 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 started discussions on possibilities for small-scale testing to avoid unnecessary AWS charges. We are hoping to test future code on his computer with smaller data sets before moving to AWS. This would include experimenting with parsec (access via cloud), live coding, or constant communication of GitHub updates (pull requests).

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 R 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 and Amdahl's Law

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.

Figure 1: Example of a flowchart of a simple parallel process [1]



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 the parallelization then time the process to see if the prediction was close.

1.2 Future Tasks

This is what I am hoping to do next.

- Comment through the code base
- Experiment with parallel GRASS + R
- Set up a code sharing process for small scale testing
- Create a flow chart for a parallel version of the code base
- Time the serial version of the code base with small data sets and calculate Amdahl's Law

2 Literature Review

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. The assigned reading this week focused on greenhouse gas (GHG) scenerios ("alternative images of how the future might unfold" [4]). They determined that there are 4 mains storylines (A1, A2, B1, B2) with 40 resulting scenerios. These scenerios were derived from several factors including the 3 main driving forces: population prediction, economic development, and structural and technological change.

After reading the article, I had an idea that requires further discussion. It would be interesting to create interpolated GHG emission maps along with our ecoregion maps to find patterns or spatial autocorrelation between the anthropo-scene and emissions.

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.