Homework 12 Solution - Yidan Xu

Statistical computing on greatlakes

From homework 11, we understand the relevance of being able to implement embarrassingly parallel calculations. In the context of statistical computing in R, we are motivated to learn to use `foreach` on the greatlakes cluster accessed via the terminal. In this homework we carry out a simple example, since many of us are new to Linux clusters in general, and working at a greatlakes terminal in particular. Once you can run a simple multicore `foreach` on greatlakes you are prepared for a large fraction of all statistical parallel computing tasks. Work through the questions below and write brief answers to the following questions, by editing the tex file available at `https://github.com/ionides/810f21`, and submit the resulting pdf file via Canvas.

1. R has a reputation of being convenient for data analysis but slow in some situations, both features shared with Python. Loops in R can be slow, though this may be avoidable by careful coding (`https://adv-r.hadley.nz/control-flow.html#common-pitfalls`). If R run time starts limiting your research—i.e., whenever there is a numerical result you would like to see but you don't have the patience to wait for it to be computed—you have various options, including:

   (a) Use smaller examples.

   (b) Write the critical part of your code in C called from R. This can make use of the C functions underpinning the R language (`https://cran.r-project.org/doc/manuals/R-exts.html#The-R-API`).

   (c) Work on rewriting your R code to make it more efficient.

   (d) Run your code on a more powerful machine, maybe moving from your laptop to greatlakes.

   Comment on situations when each approach might be suitable. Do you have advice, or relevant experiences?

   (a) It's good to use smaller example to test codes in the early stage of doing simulations etc.

   (b) When e.g. matrix multiplication is used extensively then acceleration can be achieved using C.

   (c) Exploit sparsity of the data - use sparse matrix whenever appropriate; use mathematically equivalent numerically stable expression ...

(d) When much more computation power is required when running large-scale/parallisable tasks.

2. Follow the instructions at `https://ionides.github.io/810f21/gl/slides.pdf` to run a job testing foreach on greatlakes. Report back on your results, and any problems that arose or advice to share.

   Comparing the `elapsed` time to run test.R in Table (1) with Great Lakes cluster to that in Table (2) with Macbook Air, it is clear that parallelisation does take advantage of multiple cores with the cluster as time elapsed is reduced; however running one sequence on cluster is slower comparing to running it on personal computer in this case.

   I was having trouble sending the result back to my laptop using scp, I did specify the correct path on my computer, however it is saying that it is not recognised.

Table 1: Run time of test.R on Great Lakes cluster.

	user.self	sys.self	elapsed	user.child	sys.child
time0	6.36	0.315	6.691	0	0
time1	0.288	0.883	2.184	0	0
time2	0.223	0.811	1.439	1.107	0.273
time3	0.344	0.751	1.529	6.851	2.143
time4	1.103	1.436	2.786	7.356	3.596

Table 2: Run time of test.R on Mac.

	user.self	sys.self	elapsed	user.child	sys.child
time0	4.754	0.154	4.912	0	0
time1	0.196	0.402	2.335	3.776	0.567
time2	0.164	0.420	2.863	1.350	0.290
time3	0.228	0.469	2.879	5.446	1.218
time4	0.905	0.513	24.543	5.539	1.264