Making R faster

2015-09-25

Today:

- Local parallelization with foreach
  - Using the SCF cluster
- C++ integration with Rcpp
- Lab3 introduction
  - You'll want parallelization and Rcpp

Local parallelization

Parallelization has a few different flavors:

- Multicore processors
- GPUs
- Computer clusters

This is as far as we'll go in this class

Resources

Chris Paciorek (of STAT 243) is a local expert. The material today is mostly his.

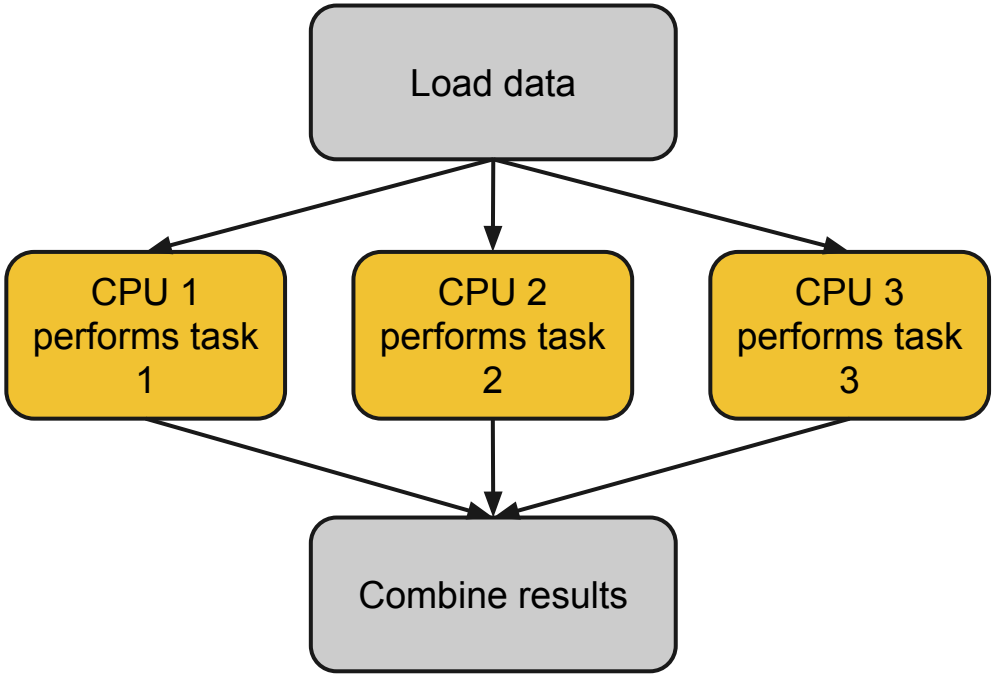
Check out the SCF page on the stat department website for in depth tutorials

Parallel Programming Workshops:

Session 1: Monday Oct. 12, 4 pm, Evans 1011

Session 2: Monday Oct. 26, 4 pm, Evans 1011

Local Parallelization



The parallel tasks cannot talk to one another.

You can parallelize to either speed up computation or split up a large data set.

How would you parallelize:

- A bootstrap?
- K-means?

R Example

`foreach_example.R`

Exercise: use foreach to parallelize kmeans

Using the SCF cluster

Read section 2 of Chris's document. Long story short:

- Set up a shell script that runs your job (e.g. shell_example.sh).
- Choose a computer and ssh to it.
- Copy your files to that computer
  - Clone a git repository there or
  - Use scp
- Submit your job using something like:
  - `qsub -pe smp 4 shell_example.sh`

This is how many cores you need

This will run your analysis and produce output

Anatomy of shell script

```
#!/bin/bash
R --no-save < shell_example.R
```

This basically just runs commands as if you had typed them. Make sure it's executable:

```
chmod 755 shell_example.sh
```

Command line arguments

Try optparse:

```
library(optparse)
option_list <- list(
  make_option(c("--cores"), type="integer", default=1,
              help="Number of cores to use [default %default]",
              metavar="cores")
)

# Get command line options.
opt <- parse_args(OptionParser(option_list=option_list))
registerDoParallel(cores = opt$cores)
------------------------
```

R CMD BATCH --args --cores=5 my_sript.R

Exercise

- As a group, run one member’s parallelized kmeans on the SCF

Rcpp

R is inefficient with memory and for loops.

Rcpp allows you to easily integrate C++ code into R.

Demo:
`Rcpp_demo.R`

Exercises:

- Write a function using `Rcpp` to calculate the average distance between a set of points in R^2

Lab 3

Clustering stability of k-means.

Algorithm 1 Calculation of clustering similarities in k-means

for $k = 2$ **to** k_{max} **do**
  for $i = 1$ **to** n **do**
    $\text{sub}_1 = \text{subsample}\,(X, m)$, a subsample of fraction m of dataset X
    $\text{sub}_2 = \text{subsample}\,(X, m)$, a subsample of fraction m of dataset X
    $L_1 = \text{cluster}\,(\text{sub}_1)$
    $L_2 = \text{cluster}\,(\text{sub}_2)$
    $\text{intersect} = \text{sub}_1 \cap \text{sub}_2$
    $S\,(i, k) = \text{similarity}\,(L_1\,(\text{intersect})\,, L_2\,(\text{intersect}))$
  end for
end for