Tutorial on Parallel Programming in R

Contact information

Presentation webpage: http://goo.gl/oHvKFk

Henry Scharf: henry.scharf@colostate.edu

Miranda Fix: miranda.fix@colostate.edu

Josh Hewitt: joshua.hewitt@colostate.edu

Conference themes addressed

- **Theme 3: Big Data Prediction and Analytics**
  The techniques we outline provide a basis for performing demanding compuations on large datasets.
- **Theme 4: Software, Programming, and Graphics**
  The techniques we outline supplement basic to fluent programming skills in R.

Section 1: The `foreach` package in R

This section will last approximately 40 minutes. Participants are encouraged to follow along on their laptops.

Learning goals

1. How to identify parallelizable tasks
2. How to convert from a `for` loop to a `foreach` loop
3. Familiarity with iterators as a way to manage memory use
4. Familiarity with special considerations required for random numbers in parallelized tasks

Extra materials

1. Presentation slides: http://goo.gl/Xssgv2
2. R script: http://goo.gl/FzPd4R
3. Verbose presentation handout: http://goo.gl/NTT5Sl
4. Data: http://goo.gl/A8LSVi

Key ideas

1. With minimal additional programming knowledge, several demanding computational tasks may be spread over multiple cores.
2. A `foreach` loop attempts to mimic the syntax of a `for` loop, but is technically a function.
3. `foreach` loops allow for the user to manage memory useage (through iterators) and random number generation (through the `doRNG` package).

Section 2: The `parallel` package in R

This section will last approximately 30 minutes. Participants are encouraged to follow along on their laptops.

Learning goals

1. When to use the `parallel` package in R
2. How to parallelize `apply` functions
3. How to parallelize a process that generates random numbers
4. How to parallelize bootstrapping (example)

Extra materials (helpful, but not required)

1. Presentation slides: http://goo.gl/5wQh0f
2. R script: http://goo.gl/d2XDcF

Key ideas

1. The base R package `parallel` is a merger of `multicore` and `snow`.
2. For embarassingly parallel tasks, a small amount of effort can produce a large gain in efficiency.
3. Some care must be taken when parallelizing processes that involve random number generation.

Section 3: Computing with R and Hadoop

This section will last approximately 40 minutes. The first 20-25 minutes introduces Hadoop and how people can use it with R, and the remaining 15-20 minutes walks participants through a demonstration lab.

Learning goals

1. What Hadoop is
2. Current R/Hadoop integrations
3. *When* to use R with Hadoop (guidelines)
4. *How* to use R with Hadoop (lab)

Extra materials (helpful, but not required)

1. Presentation slides: http://goo.gl/Ew6jOP
2. Virtual machine used to run R/Hadoop for the lab (requires ~4GB RAM): http://goo.gl/R5Okcr
3. Lab instructions: http://goo.gl/6fkQr9
4. Lab R script: http://goo.gl/URPJdD
5. PuTTY (SSH client for Windows): http://goo.gl/vMv6ra

Key definitions

Hadoop: Open source software for enabling distributed storage and computing capabilities on networked servers.

MapReduce: Hadoop's model for parallel programming.

R: Open source statistical computing software designed with strong built-in support for statistical needs like fitting models, making predictions, and drawing inferences.

R/Hadoop integration: Extra R packages that let practitioners run R code in Hadoop MapReduce jobs.

Key ideas

1. R and Hadoop integrate best when using the strengths of both technologies.
2. R and Hadoop do not integrate well for all projects. Simple data processing and iterative algorithms may be best implemented with other languages or technologies.
3. Hadoop facilitates distributed computing with the MapReduce programming model.