HPC-R Workshop Prerequisites

XSEDE
Extreme Science and Engineering
Discovery Environment

A UT/ORNL PARTNERSHIP
NATIONAL INSTITUTE FOR COMPUTATIONAL SCIENCES
NICS

NIMBioS
National Institute for Mathematical
and Biological Synthesis

Attendee Background

The attendee should ideally have some familiarity with working with R. Service providers looking for information to better support R users need not have direct experience with R, provided they have a strong programming background and some familiarity with a high-level language such as Python or Matlab.

Tutorial Outline

The tutorial will be four hours in length (including breaks), scheduled for 1–5 p.m. Eastern Standard Time on Friday, February 27, 2015. The tutorial will be held physically at the National Institute for Mathematical and Biological Synthesis (NIMBioS), 1122 Volunteer Blvd., Claxton Complex. Room 206, on the campus of the University of Tennessee, Knoxville. The event will also be streamed online.

The tutorial is divided into 4 separate 1 hour modules, consisting of 45 minutes of instruction and 15 minutes of hands-on exercises. The general structure of the tutorial will be:

1. Basics
    - Instruction (45 minutes)
        1. Introduction
        2. Debugging
        3. Profiling
        4. Benchmarking
    - Exercises (15 minutes)
2. Improving R Performance
    - Instruction (45 minutes)
        1. Free Improvements
        2. Writing Better R Code
    - Exercises + break (30 minutes)
3. Interfacing to Compiled Code
    - Instruction (45 minutes)
        1. Introduction to Rcpp
        2. Using Rcpp
        3. Examples
    - Exercises (15 minutes)
4. Parallelism
    - Instruction (45 minutes)
        1. Shared memory parallelism
        2. Distributed parallelism

Note that the final section, parallelism, will not have dedicated time for exercises (though “take-home” exercises will be provided).

Software Requirements

Due to the volume of registrants for this tutorial, we will not be making any XSEDE resources available for the hands-on portions of the training. If you are able to utilize an XSEDE resource, you of course can choose to use one of your personal accounts. But we can not provide instructions or support for you if you do so. We will assume that you have the necessary software installed on your workstation.

In order to complete the exercises, you will need to have R as well as some R packages installed. You can download R here (http://cran.r-project.org). The packages you will need are:

- Rcpp
- rbenchmark
- microbenchmark
- foreach
- doParallel

To install a package named `pkg`, start up an R session and enter

```
install.packages("pkg")
```

I would strongly encourage you to install the RStudio Desktop editor (http://www.rstudio.com/products/rstudio/download/). RStudio is the most popular editor used in the R community, so even if you prefer another editor/ide, it is still useful to get a sense for the tools that R users find most appealing.

Finally, in order to follow along with the examples and exercises of the Rcpp section, you will need to be able to build packages from source. This requires some extra software to be installed depending on the platform you use.

Windows

Install Rtools (http://cran.r-project.org/bin/windows/Rtools/).

Mac

First, you will need to install XCode (http://itunes.apple.com/us/app/xcode/id497799835?mt=12) from the Mac App Store. Then, open XCode and choose “Preferences”, “Downloads”, then “Install Command Line Tools”.

Linux, FreeBSD, etc.

You’re probably good.

Testing

To make sure that the installation went smoothly (and assuming you already installed Rcpp), you can test your installation with the following script:

```
library(Rcpp)

code <- '
#include <Rcpp.h>

// [[Rcpp::export]]
int test()
{
  Rprintf("It worked!");
  return 0;
}
'

sourceCpp(code=code)
invisible(test())
```

If all goes well, you you should see:

```
## [1] "It worked!"
```