Seamless C++ Integration with Rcpp Attributes

J.J. Allaire

1. RStudio

Contact author: jj@rstudio.com

Keywords: C++, Rcpp

The Rcpp [Eddelbuettel and François, 2013] package allows users to effortlessly pass rich objects between R and C++ code. The initial use cases for Rcpp were using C++ within R packages as well as embedding R in C++ applications. However, users are also highly interested in working with C++ interactively, a possibility afforded by the inline package [Sklyar et al., 2012]. Attributes are annotations added to C++ source files to provide additional information to the compiler [Maurer and Wong, 2008]. Rcpp Attributes [Allaire et al., 2013] are a new feature that provides a high-level syntax for declaring C++ functions as callable from R and automatically generating the code required to invoke them. The motivation for attributes is several-fold:

1. Reduce the learning curve associated with using C++ and R together
2. Eliminate boilerplate conversion and marshaling code wherever possible
3. Seamless use of C++ within interactive R sessions
4. Unified syntax for interactive work and package development

Rcpp supports attributes to indicate that C++ functions should be made available as R functions, as well as to optionally specify additional build dependencies. The C++ file can then be interactively sourced into R using the sourceCpp function, which makes all of the exported C++ functions immediately available to the interactive R session. As a result of eliminating both configuration and syntactic friction, the workflow for C++ development in an interactive session now approximates that of R code: simply write a function and call it.

Attributes can also be used for package development via the compileAttributes function. This enables ad-hoc interactive work done with sourceCpp to be easily migrated into a package for broader distribution.

This talk will cover the motivation for and implementation of attributes as well as review many practical examples of their use.

References


