Slicing and dicing big data with RHadoop/rmr.

Antonio Piccolboni[1,*]

1. Revolution Analytics
*Contact author: antonio@piccolboni.info

Keywords: Hadoop, RHadoop, **rmr**, mapreduce, big data

Hadoop has become the de-facto standard for storing and processing extremely large data sets. Some predict it will host half of all the world's data by 2016. Part of Hadoop is a parallel distributed computational model implementation known as Hadoop MapReduce whose abstract definition has roots in functional programming languages. As such, MapReduce is a natural fit for *R* where the `lapply-tapply` pair represents the closest analog. The package **rmr**, part of the open source RHadoop[1] project, provides an abstraction over Hadoop MapReduce that is tightly integrated with the R language but is capable of processing terabytes of data, taking advantage of clusters of up to thousands of machines. A simple library that works with most other R packages and in any IDE, **rmr** provides straightforward bridges between the in-memory an on-disk data, promoting a pragmatic and incremental approach to big data. Moreover, **rmr** supports the use of any *R* objects in connection with MapReduce, upholds for the most part usual *R* variable scoping rules, and doesn't force you to use any esoteric *R* constructs. This makes **rmr**-based programs look and feel and work like regular *R* programs providing what we believe is the easiest and most productive path into Hadoop and big data for *R* users and developers.

In this talk, after introducing Hadoop and the RHadoop project, we will illustrate, by way of simple examples, how to sift through and summarize large data sets with a few lines of code and how to go from simple one-liners to reusable functions.

References

[1] Revolution Analytics (2011). The RHadoop project, http://github.com/RevolutionAnalytics/RHadoop.