REVOLUTION ANALYTICS

RHadoop: Installation and Configuration

RevolutionAnalytics

December 3, 2013

Outline

1 RHadoop Background

2 RHadoop Installation and Configuration

3 rhdfs

4 rmr2

Introduction

Hadoop Streaming enables the creation of mappers, reducers, combiners, etc. in languages other than Java. Increasingly viewed as a lingua franca of statistics and analytics, R is a natural match for Big Data-driven analytics.[1]

There are a number of options to work with R and Hadoop:

- RScript (â□□nakedâ□□ streaming)
- JDBC/ODBC connections to Hive
- R packages: HadoopStreaming, hive, RHIPE, segue, RHadoop (rhdfs, rhbase, rmr2)

[1] Along with Python

Hadoop-related packages

Package	Latest Release *(as of 2012-07-09)*	Comments
hive	v0.1-15: 2012-06-22	misleading name: stands for "Hadoop interactIVE" & has nothing to do with Hadoop hive. On CRAN.
HadoopStreaming	v0.2: 2012-10-29	focused on utility functions: I/O parsing, data conversions, etc. Available on CRAN.
RHIPE	v0.71: 2012-11-18	comprehensive: code & submit jobs, access HDFS, etc. Unfortunately, most links to it are broken. Look on github instead: https://github.com/saptarshiguha/RHIPE/
segue	v0.05: 2012-07-09	Very clever way to use Amazon EMR with small or no data. http://code.google.com/p/segue/
RHadoop (rmr2, rhdfs, rhbase)	rmr2 2.0.2: 2012-12-04 rhdfs 1.0.5: 2012-08-02 rhbase 1.1: 2012-10-18	Divided into separate packages by purpose: •rmr2 - all MapReduce-related functions •rhdfs - management of Hadoop's HDFS file system •rhbase - access to HBase database Sponsored by Revolution Analytics & on github: https://github.com/RevolutionAnalytics/RHadoop

Figure : Various Hadoop-related packages

RHadoop Package Overview

- rmr2 - all MapReduce-related functions
- rhdfs - interaction with Hadoopâ□□s HDFS file system
- rhbase - access to HBase database

RHadoop Advantages

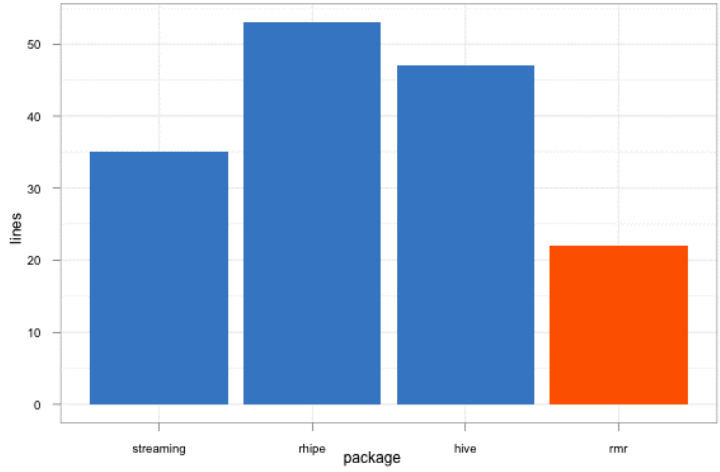
- Modular
- Packages group similar functions
- Only load (and learn) what you need
- Minimizes prerequisites and dependencies
- Open Source
- Cost: Low (no) barrier to start using
- Transparency: Development, issue tracker, Wiki, etc. hosted on github
- Supported and Sponsored by Revolution Analytics
- Training & professional services available

RHadoop Advantages: rmr2

- Well designed API: Your code only needs to deal with R objects: strings, lists, vectors & data.frames
- Very flexible I/O subsystem: Handles common formats like CSV and allows you to control the input parsing without having to interact with stdin/stdout directly (or loop)
- Integrates seem The result of the primary mapreduce() function is simply the HDFS path of the jobâ□□s output Since one jobâ□□s output can be the next jobâ□□s input, mapreduce calls can be daisy-chained to build complex workflows …and **shorter code**

RHadoop Advantages: LoC

Jonathan Seidman (now at Cloudera) compared â□□nakedâ□□ streaming (i.e., RScript) with the RHIPE, hive, and RHadoopâ□□s rmr packages (available on github)



Outline

1 RHadoop Background

2 RHadoop Installation and Configuration

3 rhdfs

4 rmr2

Hadoop in a (virtual) box

Clouderaâ□□s Hadoop Demo VM provides everything you need to run small jobs in a virtual environment: Hadoop 0.20 + Flume, HBase, Hive, Hue, Mahout, Oozie, Pig, Sqoop, Whirr, Zookeeper. Based on CentOS 5.7 & available for VMware, KVM and VirtualBox and available here. Older versions came with training exercises, but fortunately theyâ□□re still available on github.

Provides a common base which can be used to launch clusters on cloud services like EC2, etc.

RHadoop Prerequisites

Short answer: CDH3 and higher, Apache 1.0.2 and higher Detailed answer: Try `R CMD check <path-to-rmr-pkg-dir>`

Environment variables (adjust as needed):

```
HADOOP_HOME=/usr/lib/hadoop
HADOOP_CONF=/etc/hadoop/conf
HADOOP_CMD=/usr/bin/hadoop
HADOOP_STREAMING=/usr/lib/hadoop/contrib/streaming/hadoop-streaming-<ve
```

- rhdfs: rJava
- rmr2: RJSONIO (0.95-0 or later), itertools, digest, Rcpp, functional, plyr
- rhbase: Thrift server (and its prerequisites) more details on the wiki

Downloading RHadoop

Stable and development branches are available on github with releases available as packaged downloads.

Outline

1 RHadoop Background

2 RHadoop Installation and Configuration

3 rhdfs

4 rmr2

Install rhdfs package

The rhdfs package contains functions to administer and interact with Hadoopâ□□s HDFS distributed file system. Note: `rhdfs` *does not need to be installed on all cluster nodes*â□“only where scripts will be using it.

First install prerequisite packages (run R as root to install system-wide)

```
> install.packages( c('rJava'), repos='http://cran.revolutionanalytics.
```

Download and install `rhdfs`

```
wget --no-check-certificate https://github.com/downloads/RevolutionAnaly
R CMD INSTALL rhdfs_1.0.5.tar.gz
R -e "library(rhdfs)" # Test that it loads
```

rhdfs Function Overview

File & directory manipulation

- hdfs.ls(), hdfslist.files()
- hdfs.delete(), hdfs.del(), hdfs.rm()
- hdfs.dircreate(), hdfs.mkdir()
- hdfs.chmod(), hdfs.chown(), hdfs.file.info()
- hdfs.exists()

rhdfs Function Overview

Copying, moving & renaming files to/from/within HDFS

- hdfs.copy(), hdfs.move(), hdfs.rename()
- hdfs.put(), hdfs.get()

rhdfs Function Overview

Reading files directly from HDFS

- hdfs.file(), hdfs.read(), hdfs.write(), hdfs.flush()
- hdfs.seek(), hdfs.tell(con), hdfs.close()
- hdfs.line.reader(), hdfs.read.text.file()

rhdfs Function Overview

Misc.

- hdfs.init() # required initialization
- hdfs.defaults() # rhdfs options

rhdfs Example: populate HDFS

```
# Create /rhadoop-training directory on HDFS if it doesnâ^^80^^99t alre
hdfs.root = '/rhadoop-training'
if ( !hdfs.exists( hdfs.root ) ) {
    hdfs.dircreate( hdfs.root )
}
# For each project, copy local files to HDFS
data.root = 'data'
for (project in c('wordcount', 'airline', 'marketing'))
{
    data.path = file.path(data.root, project)    # local
    target.path = file.path(hdfs.root, project) #HDFS

    if ( !hdfs.exists( target.path ) )
        hdfs.dircreate( target.path )

    target.path = file.path(target.path, 'data')
    if ( !hdfs.exists( target.path ) )
```

rhdfs Example: check results

On local file system:

```
$ ls -l data/*
data/airline:
total 916
-rw-rw-r-- 1 1120 games 1875213 Feb 27 10:52 20040325.csv

data/marketing:
total 108
-rw-r--r-- 1 1120 games 219341 May 22 15:24 marketing-1000.csv

data/wordcount:
total 2609
-rw-r--r-- 1 1120 games 5342761 May 22  2009 all-shakespeare
```

rhdfs Example: check results

on HDFS:

```
> hdfs.ls('/rhadoop-training', recurse=T)
  permission    owner      group    size          modtime
1 -rw-r--r-- cloudera supergroup 1875213 2012-05-22 16:09
2 -rw-r--r-- cloudera supergroup  219341 2012-05-22 16:09
3 -rw-r--r-- cloudera supergroup 5342761 2012-05-22 16:09
                                                    file
1         /rhadoop-training/airline/data/20040325.csv
2 /rhadoop-training/marketing/data/marketing-1000.csv
3    /rhadoop-training/wordcount/data/all-shakespeare
```

Outline

1 RHadoop Background

2 RHadoop Installation and Configuration

3 rhdfs

4 rmr2

Install rmr2 package

The rmr2 package contains all the mapreduce-related functions, including generating Hadoop streaming jobs and basic data exchange with HDFS. Note: `rmr2` *needs to be installed on all cluster nodes.* First install prerequisite packages (run R as root to install system-wide)

```
> install.packages( c('RJSONIO', 'itertools', 'digest', 'Rcpp', 'functi
repos='http://cran.revolutionanalytics.com')
```

Download and install the latest stable release (2.0.2) from github

```
wget --no-check-certificate https://github.com/downloads/RevolutionAnaly
sudo R CMD INSTALL rmr2_2.0.2.tar.gz
R -e "library(rmr2)" # Test that it loads
```