RHadoop and MapR

Accessing Enterprise-Grade Hadoop from R

Table of Contents

Introduction 3

Environment 3

R 3

Installation Prerequisites 4

Install R 4

Install RHadoop 5

- Install rhdfs 5
- Install rmr2 6
- Install rhbase 8

Conclusion 10

Resources 11

Introduction

RHadoop is an open source collection of three R packages created by Revolution Analytics that allow users to manage and analyze data with Hadoop from an R environment. It allows data scientists familiar with R to quickly utilize the enterprise-grade capabilities of the MapR Hadoop distribution directly with the analytic capabilities of R.
This paper provides step by step instructions to install and use RHadoop with MapR and Revolution R on RedHat Enterprise Linux.

RHadoop consists of the following packages:

- rmr2 - functions providing Hadoop MapReduce functionality in R
- rhdfs - functions providing file management of the HDFS from within R
- rhbase - functions providing database management for the HBase distributed database from within R

Each of the RHadoop packages can be installed and used independently or in conjunction with each other.

Environment

The integration testing described in this paper was performed in December 2012 on a 3-node Amazon EC2 cluster. The product versions tested are listed in the table below. Note that Revolution Analytics currently provides linux support only on RedHat.

Product	Version
EC2 AMI	RHEL-6.3-Starter-x86_64-1-Hourly2 (ami-5453e055)
Type	m1.medium
Root/Boot	7GB EBS standard
MapR storage	(3) 4GB EBS standard
RedHat Enterprise Linux 6.3 64-bit	2.6.32-276.el6.x86_64
Java	java-1.6.0-openjdk.x86_64
	java-1.6.0-openjdk-devel.x86_64
MapR M5	2.0.1 (2.0.1.15869.GA)
HBase	0.92.2
Revolution R Community	6.0
RHadoop	
rhdfs	1.0.5
rmr2	2.0.2
rhbase	1.1
Apache Thrift	0.9.0

R

R is both a language and environment for statistical computation and is freely available as a GNU project. Revolution Analytics provides two versions of R: the free Revolution R Community and the premium Revolution R Enterprise for workstations and servers. Revolution R Community is an enhanced distribution of the open source R for users looking for faster performance and greater stability. Revolution R Enterprise adds commercial

enhancements and professional support for real-world use. It brings higher performance, greater scalability, and stronger reliability to R—at a fraction of the cost of legacy products. R is easily extended with libraries that are distributed in packages. Packages are collections of R functions, data, and compiled code in a well-defined format. The directory where packages are stored is called the library. R comes with a standard set of packages. Others are available for download and installation. Once installed, they have to be loaded into the session to be used.

Installation Prerequisites

These instructions are specific to the product versions specified in the Environment section. Some modifications may be required for your environment. A package repository must be available to install dependent packages. The MapR cluster must be installed and running and root privilege must be available for all nodes in the cluster. MapR installation instructions are available on the MapR Documentation web site http://mapr.com/doc/display/MapR/Installation+Guide.

All commands entered by the user are in **`bold courier`** font. Commands entered from within the R environment are preceded by the default R prompt ">". Linux shell commands are not preceded by a prompt. Output from many commands is lengthy and not shown. Any output that is shown is in `normal courier` font. Unless otherwise indicated, all commands should be run as the root user.

Install R

Testing for this paper was done with Revolution R Community and instructions specify the tar file for the Community edition. If you have Revolution R Enterprise, you can use that version of R instead - follow Revolution Analytics installation instructions for the Enterprise edition. A version of R must always be installed on the system accessing the cluster using the RHadoop libraries. Additionally, to execute MapReduce jobs with the rmr2 library, R must be installed on all tasktracker nodes in the cluster. As root, follow the installation steps below to install it in the standard location under /usr/lib64/Revo-6.0.

1) Download Revolution R Community.
   Request download from http://www.revolutionanalytics.com/downloads.
   You will receive an email with download instructions for multiple platforms. Download the Red Hat Enterprise Linux 6 Installer Revo-Co-6.0-RHEL6.tar.gz

2) Install Revolution R Community all nodes in the MapR cluster. The install script will install R under /usr/lib64/Revo-6.0. Symbolic links to the R run script are create in /usr/bin/R and /usr/bin/Revo64. The options for the install script will cause it to perform a non-interactive install with default answers. The install.py script can be run with --help for details on options.

   Note: Revolution R Community depends on certain packages. The R installation script will invoke yum to install package dependencies. A yum repository must be available for this installation.

   ```
   tar xzvf Revo-Co-6.0-RHEL6.tar.gz
   cd RevolutionR_6.0
   ./install.py -n -d -l -s
   ```

3) Confirm installation was successful by running R as a non-root user. At the command line, type **`R`**. At the > prompt, type **`q()`** to quit and **`y`** to save the current R workspace. You can also invoke **`R`** with the **`--save`** option to quit immediately without being prompted to save the workspace.

   ```
   $ R
   ```

```
R version 2.14.2 (2012-02-29)
Copyright (C) 2012 The R Foundation for Statistical Computing
...
Type 'revo()' to visit www.revolutionanalytics.com for the latest
Revolution R news, 'forum()' for the community forum, or 'readme()'
for release notes.

> q()
Save workspace image? [y/n/c]: y
```

Install RHadoop

The installation instructions that follow are complete for each RHadoop package (rhdfs, rmr2, rhbase). System administrators can skip to installation instructions of just the package(s) they want to install. Of course, R must be installed before installing any of the RHadoop packages.

Install rhdfs

The rhdfs package uses the hadoop command to access MapR file services. To use rhdfs, R and the rhdfs package only need to be installed on the client system that is accessing the cluster. This can be a node in the cluster or it can be any client system that can access the cluster with the hadoop command.

As root, perform the following steps:

1) Confirm that you can access the MapR file services by listing the contents of the root directory. You should be able to do this as both root and a non-root user.

```
hadoop fs -ls /
```

Important: Linux UID and GID must be consistent across all nodes in the cluster and client systems accessing the cluster. This is true whether accessing a Hadoop system with the hadoop command or via the RHadoop libraries.

2) Install the rJava R package that is required by rhdfs. At the command line, type **`R --save`**. From the > prompt, install the rJava package. This will download, compile, and install the package. Exit R with the `q()` function.

```
> install.packages('rJava')
> q()
```

Note: All the components of each R library are installed in /usr/lib64/Revo-6.0/R-2.14.2/lib64/R/library. After installing rJava, you can see the directory rJava in this location. You can list all available libraries from within R using the library() command with no parameters.

3) Download the rhdfs gzipped tar file.

```
wget
https://github.com/downloads/RevolutionAnalytics/RHadoop/rhdfs_1.0.5.ta
r.gz
```

4) Set the HADOOP_CMD environment variable to the hadoop command script and install the rhdfs package. Whereas the rJava package in the previous step was downloaded and installed from a CRAN repository, rhdfs is installed from the previously downloaded gzipped tar file.

```
export HADOOP_CMD=/opt/mapr/hadoop/hadoop-0.20.2/bin/hadoop
R CMD INSTALL rhdfs_1.0.5.tar.gz
```

Note: After installing rhdfs, you can see the rhdfs directory in /usr/lib64/Revo-6.0/R-2.14.2/lib64/R/library or from within the R environment list all libraries with the library() command.

5) Set required environment variables. In addition to the HADOOP_CMD environment variable set in the previous step, LD_LIBRARY_PATH must include the location of the MapR client library libMapRClient.so and HADOOP_CONF must specify the Hadoop configuration directory. Any user wanting to use the rhdfs library must set these environment variables.

```
export HADOOP_CMD=/opt/mapr/hadoop/hadoop-0.20.2/bin/hadoop
export LD_LIBRARY_PATH=/opt/mapr/lib:$LD_LIBRARY_PATH
export HADOOP_CONF=/opt/mapr/hadoop/hadoop-0.20.2/conf
```

6) From R, load the rhdfs library and confirm that you can access the MapR cluster file system by listing the root directory. You should confirm that this can be done as both a root and non-root user.

```
> library(rhdfs)
> hdfs.init()
> hdfs.ls('/')
> q()
```

Note: When loading an R library using the library() command, dependent libraries will also be loaded. For rhdfs, the rJava library will be loaded if it has not already been loaded.

7) Run rhdfs package check. This will run sanity tests on the rhdfs package and the examples that are included in the package.

Note: When running the hdfs package check, you can safely ignore the warning referring to LaTeX errors:

```
* checking PDF version of manual ... WARNING
LaTeX errors when creating PDF version.
```

Note: When running the rhdfs package check, the last check will fail if the pdflatex command is not available on your system.:

```
* checking PDF version of manual without hyperrefs or index ... ERROR
```

This failure can safely be ignored. However, you can optionally install pdfjam and its dependencies from the Fedora Project's EPEL repository. If pdflatex is available, there only be the one warning about LaTeX errors when checking PDF version of manual. The instructions specify the -y option to yum which will answer “yes” to all install questions. Run yum without the -y option if interactive installation of packages is preferred.

```
yum install -y pdfjam  # optional
R CMD check rhdfs_1.0.5.tar.gz
```

If no errors are reported, you have successfully installed rhdfs and can use it to access the MapR file services from R. As with any access to Hadoop files, users must have required permissions to read and write files. Also, users must set the environment variables specified in the instructions.

Use the `help()` function within R to see the complete list of functions available in the rhdfs library and for usage details about any particular function.

```
> help('rhdfs')
> help('hdfs.ls')
```

Install rmr2

The rmr2 package uses Hadoop Streaming to invoke R map and reduce functions. To use rmr2, R and the rmr2 package must be installed on the server that is accessing the cluster, the client, as well as every tasktracker node in the cluster.

Note: rmr2 is the second version of the rmr package. It is not intended for use with MapReduce2, also known as YARN.

Install rmr2 on every tasktracker node AND on every client system:

1) Install required R packages: 'Rcpp', 'RJSONIO', 'itertools', 'digest', 'functional' from Revolution Analytics' repository. At the command line, type **`R --save`**. From the > prompt, install the packages. This will download, compile, and install the packages. Exit R with the `q()` function.

```
> install.packages(c('Rcpp', 'RJSONIO', 'itertools', 'digest',
'functional', 'stringr', 'plyr'))
```

   Note: Warnings may be safely ignored while the packages are being built.

2) Download the quickcheck and rmr2 gzipped tar files.

   Note: RHadoop includes the quickcheck package to support writing randomized unit tests performed by the rmr2 package check.

```
wget
https://github.com/downloads/RevolutionAnalytics/RHadoop/quickcheck_1.0
.tar.gz
wget
https://github.com/downloads/RevolutionAnalytics/RHadoop/rmr2_2.0.2.tar
.gz
```

3) Install the quickcheck and rmr2 packages.

```
R CMD INSTALL quickcheck_1.0.tar.gz
R CMD INSTALL rmr2_2.0.2.tar.gz
```

   Note: Warnings may be safely ignored while the packages are being built.

Validate rmr2 on a client system as a non-root user with the following steps:

1) [Optional] Confirm that your MapR cluster is properly set up and configured to run a simple wordcount MapReduce job outside of the RHadoop and R environment. Users must have write permission to a /tmp directory on the cluster. You should be able to do this as both root and a non-root user. You only need to perform this step on the node from which you intend to run your R MapReduce programs. It is not necessary to perform this step on all task tracker nodes.

```
hadoop fs -mkdir /tmp/rmr2setup/wc-in
hadoop fs -put /opt/mapr/NOTICE.txt /tmp/rmr2setup/wc-in
hadoop jar /opt/mapr/hadoop/hadoop-0.20.2/hadoop-0.20.2-dev-
examples.jar wordcount /tmp/rmr2setup/wc-in /tmp/rmr2setup/wc-out
hadoop fs -rmr /tmp/rmr2setup
```

   Important: Linux UID and GID must be consistent across all nodes in the cluster and client systems accessing the cluster. This is true whether accessing a Hadoop system with the hadoop command or via the RHadoop libraries.

2) Set required environment variables. Since rmr2 uses Hadoop Streaming, it needs access to both the hadoop command and the streaming jar. These are read from the following environment variables. Any user wanting to use the rhdfs library must set these environment variables.

```
export HADOOP_CMD=/opt/mapr/hadoop/hadoop-0.20.2/bin/hadoop
export HADOOP_STREAMING=/opt/mapr/hadoop/hadoop-
0.20.2/contrib/streaming/hadoop-0.20.2-dev-streaming.jar
```

3) Extract the wordcount test from the rmr2 gzipped tar file
```
tar zxf rmr2_2.0.2.tar.gz rmr2/tests/wordcount.R
```

4) Run the wordcount program from the R environment. This will load the required R packages and run a streaming MapReduce job.
```
> source ('rmr2/tests/wordcount.R')
```

5) Run the full rmr2 check. The examples run by the rmr2 check below will sequentially generate 81 streaming MapReduce jobs on the cluster. 80 of the jobs have just 2 mappers so a large cluster will not speed this up. On a 3 node medium EC2 cluster with two tasktrackers, the examples take just over 1 hour.
```
R CMD check rmr2_2.0.2.tar.gz
```

Install rhbase

The rhbase package accesses HBase via the HBase Thrift server which is included in the MapR HBase distribution. The rhbase package is a Thrift client that sends requests and receives responses from the Thrift server. The Thrift server listens for Thrift requests and in turn uses the HBase HTable java class to access HBase.

Since rhbase is a client-side technology, it only needs to be installed on the client system that will access the MapR HBase cluster. Any MapR HBase cluster node can also be a client.

For the client system to access a local Thrift server, the client system must have the mapr-hbase-internal rpm installed which includes the MapR HBase Thrift server. If your client system is one of the MapR HBase Masters or RegionServers, it will already have this rpm installed. These rhbase installation instructions assume that mapr-hbase-internal is already installed on the client system.

In addition to the HBase Thrift server, the rhbase package requires the Thrift include files to compile and the C++ thrift library at runtime in order to be a Thrift client. Since these Thrift components are not included in the MapR distribution, Thrift must be installed before rhbase.

By default, rhbase connects to a Thrift server on the local host. A remote server can be specified in the rhbase hb.init() call, but the rhbase package check expects the Thrift server to be local. These installation instructions assume the Thrift server is running locally.

As root, perform the following installation steps.

1) Install prerequisite packages. The instructions specify the -y option to yum which will answer "yes" to all install questions. Run yum without the -y option if interactive installation of packages is preferred.
```
yum -y install automake libtool flex bison pkgconfig gcc-c++ boost-
devel openssl-devel
```

2) Download, build, and install Thrift.
```
wget https://dist.apache.org/repos/dist/release/thrift/0.9.0/thrift-
0.9.0.tar.gz
tar xzvf thrift-0.9.0.tar.gz
cd thrift-0.9.0
./configure
make
make install
```

3) When rhbase is installed and compiled, it uses pkg-config to determine the compiler flags for includes and libraries. Set the PKG_CONFIG_PATH environment variable to find the thrift.pc package configuration file.
```
export PKG_CONFIG_PATH=$(pwd)/lib/cpp
cd ..
```

4) Download the rhbase gzipped tar file.

```
wget
https://github.com/downloads/RevolutionAnalytics/RHadoop/rhbase_1.1.tar
.gz
```

5) Install the rhbase package. The LD_LIBRARY_PATH must be set to find the Thrift library (libthrift.so) which was installed as part of the Thrift installation. Also, PKG_CONFIG_PATH must still be set to point to the location of the thrift.pc package configuration file.

```
export LD_LIBRARY_PATH=/usr/local/lib
R CMD INSTALL rhbase_1.1.tar.gz
```

6) Modify the file /opt/mapr/hbase/hbase-0.92.2/conf/hbase-site.xml to configure hbase zookeeper servers. For HBase Master Servers and HBase Region Servers, zookeeper servers should already be properly configured in hbase-site.xml. For client only systems, edit the `hbase.zookeeper.quorum` and `hbase.zookeeper.property.clientPort` properties to correspond to your zookeeper servers.

```
<property>
<name>hbase.zookeeper.quorum</name>
<value>zkhost1,zkhost2,zkhost3</value>
</property>

<property>
<name>hbase.zookeeper.property.clientPort</name>
<value>5181</value>
</property>
```

Note: The zookeeper servers are the hostnames that were passed in to the MapR configure.sh script with the -Z option when MapR was configured. You can use the following command to retrieve the hostnames and port:

```
maprcli node listzookeepers
```

7) Start the MapR HBase Thrift server as a background daemon.

```
/opt/mapr/hbase/hbase-0.92.2/bin/hbase-daemon.sh start thrift
```

Note: Pass the parameters **`stop thrift`** to the hbase-daemon.sh script to stop the daemon.

8) Run the rhbase package checks. LD_LIBRARY_PATH and PKG_CONFIG_PATH must still be set.

```
R CMD check rhbase_1.1.tar.gz
```

Note: When running the rhbase package check, warnings can be safely ignored.

The rhbase package is now installed and ready for use by any user on the system. Validate that a non-root user can access HBase via the HBase shell (optional) and from rhbase.

1) [Optional] Start the HBase shell, create a table, display its description, and drop it.

```
hbase shell
hbase(main):001:0> create 'testtable', 'colfam1'
hbase(main):002:0> describe 'testtable'
hbase(main):003:0> disable 'testtable'
hbase(main):004:0> drop 'testtable'
hbase(main):005:0> quit
```

2) Now perform the same test with rhbase. The LD_LIBRARY_PATH environment variable must be set to include the path to the libthrift.so library.

```
export LD_LIBRARY_PATH=/usr/local/lib
```

```
R --save
> library(rhbase)
> hb.init()
> hb.new.table('testtable', 'colfam1')
> hb.describe.table('testtable')
> hb.delete.table('testtable')
> q()
```

Use the `help()` function within R to see the complete list of functions available in the rhbase library and for usage details about any particular function.

```
> library(rhbase)
> help('rhbase')
> help('hb.init')
```

Conclusion

With Revolution Analytics' RHadoop packages and MapR's enterprise grade Hadoop distribution, data scientists can utilize the full potential of Hadoop from the familiar R environment.

Resources

More information can be found on RHadoop, and other technologies referenced in this paper at the links below.

RHadoop wiki: https://github.com/RevolutionAnalytics/RHadoop/wiki
RHadoop source: https://github.com/RevolutionAnalytics/RHadoop
Hadoop Streaming: http://hadoop.apache.org/docs/mapreduce/current/streaming.html
Apache Thrift: http://thrift.apache.org
Revolution Analytics: http://www.revolutionanalytics.com
MapR Technologies: http://www.mapr.com

About MapR Technologies
MapR's advanced distribution for Apache Hadoop delivers on the promise of Hadoop, making the management and analysis of Big Data a practical reality for more organizations. MapR's advanced capabilities, such as streaming analytics, and mission-critical data protection, expands the breadth and depth of use cases across industries.

About Revolution Analytics
Revolution Analytics (formerly Revolution Computing) was founded in 2007 to foster the R Community, as well as support the growing needs of commercial users. Our name derives from combining the letter "R" with the word "evolution." It speaks to the ongoing development of the R language from an open-source academic research tool into commercial applications for industrial use.

12/17/2012