Installing and Uninstalling Distributed R

Platform Requirements

To install and run Distributed R, your server must meet one of the following platform requirements:

- Red Hat Enterprise Linux 6.x (x86-64) or later
- 64-bit CentOS 6.x (x86-64) or later
- Ubuntu 12.04 LTS
- Debian GNU/Linux 7.0

Software Requirements

Distributed R requires the following Linux packages:

- Build tools: make, gcc, and gcc-c++
- protobuf
- libxml2-devel
- zeromq
- libaio
- R (version 3.0.1 and later)

Note: Distributed R was successfully built and tested on R version 3.0.1. For instructions about installing any of the Linux packages listed, please see the Installation section.

Installing Distributed R on RHEL or CentOS

Distributed R requires the following Linux packages:

- Build tools: make, gcc and gcc-c+
- protobuf
- libxml2-devel
- zeromq
- libaio
- R (version 3.0.1 and later)

When you install Distributed R, following the steps described in this section, you also install the following packages:

- **HPDGLM**: For distributed Regression
- **HPdcluster**: For distributed Clustering
- **HPdgraph**: For distributed algorithms for graph analytics
- **HPdclassifier**: For distributed algorithms for learning classifiers
- **HPdata**: Contains general functions to load distributed data structures supported by Distributed

R.

Note: Before installing this version of Distributed R, uninstall any earlier version (s) of Distributed R and any other R packages from each node in the cluster. The current version of the Distributed R package must be the only R package on each node. You must have root or sudo privileges to install/uninstall Distributed R. See Uninstalling Distributed R form more information.

To install Distributed R

1. Obtain the prerequisite packages specified in Software Requirements.
2. Install the build tools with yum using the following command:

   ```
   yum install make gcc gcc-c++
   ```
3. Some of the prerequisite packages are available from The Fedora Project's EPEL (Extra Packages for Enterprise Linux) RPM repository. You can download the file from: http://dl.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
4. Install the EPEL configuration file using the following command:

   ```
   sudo rpm -Uhv epel-release-6-8.noarch.rpm
   ```
5. Install the R, protobuf, libxml2-devel, and zero mq packages.
6. Go to HP Vertica Marketplace, and download the RPM for Distributed R.
7. Install the Distributed R RPM using the following command:

   ```
   sudo rpm -Uhv vertica-distributedR-x.x.x-xxx.el6.x86_64.rpm
   ```

Distributed R is now installed to /opt/hp/distributed R.

To install Distributed R on a cluster or in multiserver mode

- To run Distributed R on a cluster of machines, install Distributed R on each of the machines in the cluster following the same steps described in this procedure.
- To run Distributed R in multiserver mode, you must install Distributed R on each of the additional servers and create a cluster configuration file that identifies the additional nodes.

Installing Distributed R on Ubuntu or Debian OS Distribution

Distributed R requires the following Linux packages:

- Build Tools: make, libevent-dev, pkg-config, g++, libssl-dev
- libprotobuf-dev, protobuf-compiler
- libxml2-dev
- libxmq-dev
- libaio-dev
- R (version 3.0.1 and later)

When you install Distributed R, following the steps described in this section, you also install these packages:

- **HPDGLM**: For distributed Regression
- **HPdcluster**: For distributed Clustering
- **HPdgraph**: For distributed algorithms for graph analytics
- **HPdclassifier**: For distributed algorithms for learning classifiers
- **HPdata**: Contains general functions to load distributed data structures supported by Distributed R

Note: Before installing this version of Distributed R, uninstall any earlier version (s) of Distributed R and any other R packages from each node in the cluster. The current version of the Distributed R package must be the only R package on each node. You must have root or sudo privileges to install/uninstall Distributed R. See Uninstalling Distributed R form more information.

To install Distributed R on a single node or server

1. Obtain the prerequisite packages specified in Software Requirements.
2. Install the build tools using the following command:
   ```
   sudo apt-get install make libevent-dev pkg-config g++ libssl-dev
   ```
3. Install the R, protobuf, libxml2-devel, and zeromq packages, using the following commands:
   ```
   sudo apt-get install r-base
   sudo apt-get install libprotobuf-dev-protobuf-compiker
   sudo apt-get install libxml2-dev libzmq-deve libaio-dev
   ```
4. Go to HP Vertica Marketplace, and download the RPM for Distributed R.
5. Install the Distributed R Debian package using the following command:
   ```
   sudo dpkg –i vertica-distributedR-0. <version>.0-
   0.xxx.0.DEBIAN7.x86_64.deb
   ```

Distributed R is now installed to `/opt/hp/distributed`.

To install Distributed R on a cluster or in multiserver mode:

- **For a cluster:** To run Distributed R on a cluster of machines, install Distributed R on each of the machines in the cluster following the same steps as in the single-node procedure.
- **For multiple-server mode:** To run Distributed R in multiserver mode, install Distributed R on each of the additional servers, and create a cluster configuration file identifying the additional nodes.

4. Installation from Source

With the open source release of Distributed R, you can build and install Distributed R from its source rather than an rpm. You must have root or sudo privileges to install Distributed R.

The software requirements remain the same for each OS distribution as described in Section 2 *Installing Distributed R on RHEL or CentOS* and Section 3 *Installing Distributed R on Ubuntu or Debian OS distribution*

To install distributedR from source:

1. Download/Check-out distributedR source code from DistributedR Github repository to local machine.
2. Install required software packages

   - If your OS distribution is RHEL or CentOS

   ```
   sudo yum install R

   sudo yum install libtool zlib-devel automake pkgconfig gcc-c++ curl

   sudo yum install protobuf protobuf-devel protobuf-compiler zeromq-devel libaio-devel

   sudo yum install libxml2-devel
   ```

   - If your OS distribution is Ubuntu or Debian

   ```
   sudo apt-get install r-base

   sudo apt-get install libevent-dev pkg-config g++ libssl-dev curl

   sudo apt-get install libprotobuf7 libprotobuf-dev libzmq-dev protobuf-compiler

   sudo apt-get install libaio1 libaio-dev

   sudo apt-get install libxml2-dev
   ```

3. Install R packages distributedR requires.

   - Change directory (cd) to source code trunk. Folder third_party contains R packages required by distributedR build.

   - Change directory (cd) to third_party folder and install R packages using `R CMD INSTALL`

   ```
   cd third_party

   sudo R CMD INSTALL Rcpp_0.10.6.tar.gz

   sudo R CMD INSTALL RInside_0.2.10.tar.gz

   sudo R CMD INSTALL XML_3.98-1.1.tar.gz

   sudo R CMD INSTALL randomForest_4.6-7.tar.gz

   sudo R CMD INSTALL data.table_1.8.10.tar.gz
   ```

4. Build DistributedR

   - Change directory(cd) back to trunk and build distributedR

   ```
   cd ..
   ```

```
make
```

5. Finally, to complete installation, two environment variables need to be setup:

   ```
   export DISTRIBUTEDR_HOME=<Absolute path of DistributedR trunk>

   export R_LIBS_USER=$DISTRIBUTEDR_HOME/install
   ```

   It is advisable to add them to user profile (e.g. .bashrc file) to preserve their values between sessions.

6. To install Distributed R on a cluster of machines, execute same steps as above on each machine of the cluster.

Installation of Distributed R is complete and you can run it in single-server mode. To run in multiple-server mode you must first install Distributed R on each of the additional servers and create a cluster configuration identifying the additional nodes. See Section 6 - Running Distributed R – Multiple Machine mode for further details on the configuration file.

Running Distributed R – Single Machine Mode

This section describes how to run Distributed R on a single server. In single machine mode, the master and worker client are on the same machine. To run Distributed R, the server must have a passwordless and promptless login using `ssh 127.0.0.1`.

The following steps describe how to run Distributed R with a **passwordless** login:

1. Go to the following directory: `cd $HOME/.ssh`

2. Generate the ssh key on the server, using the following command:
```
ssh-keygen -t rsa
```

3. Add the generated ssh key to server's authorized keys file, using the following command:
```
cat .ssh/id_rsa.pub >> .ssh/authorized_keys
```

4. Set permission of the server’s authorized key files, using the follow command:
```
chmod 600 .ssh/authorized_keys
```

The following steps describe how to run Distributed R with a promptless login. When `ssh` is unknown to hosts, you must add it to the list of known hosts on the server.

You can add `ssh` in one of the following ways:

- Set `StrictHostKeyChecking` option in `$HOME/.ssh/config` file to **`'no'`** instead of **`'ask'.`**
- Issue a `ssh` command to the server so it will remember unknown hosts before running Distributed R. Users can provide an input to the `ssh` prompt as required. Hosts will now be added to the server’s known hosts.

Running Distributed R in single-machine mode:

1. Open an R session by typing:

```
$ R
```

2. Inside the R session, load the Distributed R package:

```
> library(distributedR)
Loading required package: Rcpp
```

```
Loading required package: RInside
```

3. Start Distributed R. The number of R instances started on the server is configured by the variable *`inst`*.

```
  > distributedR_start(inst=4)
```

```
Workers registered - 1/1.
All 1 workers are registered.
[1] TRUE
```

4. Check the status of worker running in Distributed R.

```
  > distributedR_status()
```

```
         Workers Inst SysMem MemUsed DarrayQuota DarrayUsed
1 127.0.0.1:9090    4   3833    3548        1724          0
```

5. Shut down Distributed R.

```
> distributedR_shutdown()
Shutdown complete
[1] TRUE
```

Note: You can use multiple cores on a single machine by distributing execution across multiple cores.

Running Distributed R – Multiple Machine Mode

To run Distributed R in multiple machine mode, verify that Distributed R is installed on all nodes as specified in the Installation section. You must also define the master and worker machines. These settings are described in the Cluster Configuration File section.

Cluster Configuration File

The Cluster Configuration file defines the master and worker nodes in the cluster.

Please note that starting DistributedR version 0.7.0 cluster configuration file format has changed. Hence, a cluster configuration file applicable to DistributedR version 0.6.0 or prior will not work with DistributedR version 0.7.0 onwards. A sample compatible Cluster Configuration file is available at */opt/hp/distributedR/conf/cluster_conf.xml* upon a particular distributedR version installation.

Hewlett-Packard recommends users create a new Cluster Configuration file, named cluster.xml at any location on the master node with the following format:

```
<MasterConfig>
   <ServerInfo>
     <Hostname>eng63</Hostname>
     <StartPortRange>50000</StartPortRange>
     <EndPortRange>50100</EndPortRange>
   </ServerInfo>
<Workers>
   <Worker>
     <Hostname>eng64</Hostname>
     <StartPortRange>50000</StartPortRange>
```

```
      <EndPortRange>50100</EndPortRange>
      <SharedMemory>0</SharedMemory>
      <Executors>10</Executors>
   </Worker>
   <Worker>
      <Hostname>eng10</Hostname>
      <StartPortRange>50000</StartPortRange>
      <EndPortRange>50100</EndPortRange>
      <SharedMemory>0</SharedMemory>
      <Executors>15</Executors>
   </Worker>
   <Worker>
      <Hostname>eng34</Hostname>
      <StartPortRange>50000</StartPortRange>
      <EndPortRange>50100</EndPortRange>
      <SharedMemory>0</SharedMemory>
      <Executors>15</Executors>
   </Worker>
</Workers>
</MoasterConfig>
```

Configuration Options

1. The `<ServerInfo>` tag specifies master node configuration and the `<Workers>` tag specifies the worker nodes configuration. Each `<Worker>` tag specifies the configuration for each worker node.
2. **Hostname** specifies the hostname of your machine. Enter the master node's hostname under the `<ServerInfo>` tag and the worker node's hostname under the `<Worker>` tag.
3. **Port Range** is defined in mandatory `<StartPortRange>` and `<EndPortRange>` tags. These tags define the range of port numbers allowed for use by a particular node (Master or Worker). At run time, a random port number within the port range defined will be opened. If the chosen port is not available, an exception is thrown. Master node port range should have a minimum of `two` available ports, while each Worker node port range should have a minimum of `2*(total number of worker nodes)+1` available ports at all times of distributedR task execution.
4. If `<Executors>` and `<SharedMemory>` options in the Worker nodes are 0, Distributed R automatically determines the settings using System Information.

Running Distributed R in multiple-machine mode.

To run Distributed R on the cluster, it is required that all machines in the cluster have passwordless and promptless login to one another. Also, each machine in the cluster should have passwordless and promptless login for the command `ssh 127.0.0.1.`

1. Open an R session by typing:.

```
$ R
```

2. Inside the R session, load the Distributed R package.

```
> library(distributedR)
  Loading required package:  Rcpp
  Loading required package:  RInside
```

3. Start Distributed R. Specify cluster and worker configuration paths.

```
> distributedR_start(cluster_conf=”<path to cluster.xml>”)
  Workers registered – 3/3.
  All 3 workers are registered.
  [1] TRUE
```

4. Check the status of workers running in Distributed R.

```
> distributedR_status()
       Workers Inst SysMem MemUsed DarrayQuota DarrayUsed
  1 eng10:9090   15  96682   88558       43506          0
  2 eng34:9090   15  96682   67212       43506          0
  3 eng64:9090   10  96682   73216       43506          0
```

5. Shut down Distributed R.

```
> distributedR_shutdown()
  Shutdown complete
  [1] TRUE
```

Uninstalling Distributed R and Supported Packages

Note: You must have root or sudo privileges to complete the Uninstallation steps described in this section.

RHEL and CentOS OS Distribution:

To uninstall Distributed R and its supported packages on a node in the cluster:

1. Verify that the Distributed R RPM is installed on the node:

```
> sudo rpm -qa | grep "distributedR"
  vertica-distributedR-0.3.0-xxx.el6.x86_64
```

2. Uninstall the Distributed R RPM that the `grep` command returned in Step 1:

```
> sudo rpm -e vertica-distributedR-0.3.0-xxx.el6.x86_64
```

After you complete this step, Distributed R and its supported packages are uninstalled. Follow these same steps to uninstall on each node in the cluster.

If you currently have HPDGLM installed from the 0.2.0 release of Distributed R, you must uninstall it before installing a new version of Distributed R. To uninstall HPDGLM, run the following command on each node in the distributedR cluster:

```
> sudo R CMD REMOVE HPDGLM
```

Ubuntu and Debian OS Distribution

To uninstall Distributed R and its supported packages on a node in the cluster:

1. Check that the Distributed R RPM is installed on the node.

```
> sudo dpkg -1 | grep “distributed”
  vertica-distributedR
```

2. Uninstall the Distributed R RPM that is returned by Step 1:

```
sudo dpkg –r vertica-distributedr
sudo dpkg –purge vertica-distributedr
```

After you complete this step, Distributed R and its supported packages are uninstalled. Follow these same steps to uninstall on each node in the cluster.

Installed from Source

To uninstall a Distributed R version installed from Source, as described in Section 4. *Installation from Source:*

1. Change directory (cd) to distributedR trunk
2. Clean the installation

```
make distclean
```

This command removes all Distributed R related components. The software packages and R packages remain.

Distributed R and its supported packages are uninstalled. Follow steps 1-2 to uninstall Distributed R and supported packages in each node in the cluster.

Using DistributedR with Rstudio

To use Distributed R with RStudio follow these steps:

1. Install RStudio on the master node of the Distributed R cluster.
2. Start RStudio on the master node of the Distributed R cluster.
3. Run Distributed R as described in the RStudio Installation guide.