Installation Guide : distributedR

Author Shreya Prasad <sprasad@vertica.com>

Description This document provides instructions on how to install distributedR and supported packages on single or multiple servers.

Table of Contents

1. Platform Requirements

To install and run distributedR, Servers 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

2. Software Requirement

distributedR requires the following Linux packages:

- Build tools: make, gcc, and gcc-c++
- protobuf
- libxml2-devel
- zeromq
- libaio
- R (version 3.0.1 and above) – distributedR is successfully built and tested on R version 3.0.1.

Instructions on how to get and install these prerequisite packages is in the *Installation* section.

3. Installation

Installing distributedR through the steps described in this section will also install following packages supported by distributedR:

- HPDGLM, package for distributed Regression
- HPdcluster, package for distributed Clustering.
- HPdgraph, package for distributed algorithms for graph analytics.
- HPdclassifier, package for distributed algorithms for learning classifiers.
- HPdata, package containing general functions to load distributed data structures supported by distributedR

Note: Before installing, uninstall any earlier version(s) of distributedR and any other R packages from each node in the cluster. The current version of the distributedR package must be the only R package on each node. Section 6 - *Uninstalling distributedR and supported packages* provides instructions for uninstalling distributedR and its supported packages from a node. You must have root or sudo privileges to install/uninstall distributedR.

To install distributedR:

1. Obtain the prerequisite packages specified in *Software Requirements* section:

   A. Install build tools with yum:

   ```
   yum install make gcc gcc-c++
   ```

   B. Some of the prerequisite packages are available from The Fedora Project's EPEL (Extra Packages for Enterprise Linux) RPM repository.

   1. Download the EPEL yum repository configuration file from http://epel.mirror.freedomvoice.com/6/i386/epel-release-6-8.noarch.rpm
   2. Install the EPEL configuration file:

      ```
      sudo rpm -Uhv epel-release-6-8.noarch.rpm
      ```

   3. Install R, protobuf, libxml2-devel and zeromq packages:

      ```
      sudo yum install R protobuf libxml2-devel zeromq libaio
      ```

2. Download the RPM for distributed from the FTP site indicated on your cover page.
3. Install distributedR RPM:

   ```
   sudo rpm -Uhv vertica-distributedR-0.4.0-xxx.el6.x86_64.rpm
   ```

4. distributedR is now installed to `/opt/hp/distributedR`.

5. To run distributedR on a cluster of machines, install distributedR on each of the machines in the cluster by following the same steps above.

Installation of distributedR is complete and can be run in single-server mode. To run in multiple-server mode you must first install distributedR on each of the additional servers then create cluster configuration file to add information about the additional nodes. See Section 5 - *Running distributedR – Multiple Server mode* for further details on the configuration file.

4. Running distributedR – Single Machine mode

This section shows how to run distributedR on a single server. In Single Machine mode, Master and Worker client are on the same machine.

To run distributedR, the Server must have password-less and prompt-less login with "*ssh 127.0.0.1*".

Password-less login. To establish password-less login to a server

1. `cd $HOME/.ssh`
2. Generate ssh key on the server

   ```
   ssh-keygen -t rsa
   ```

3. Add the generated ssh key to server's authorized keys file.

   ```
   cat .ssh/id_rsa.pub >> .ssh/authorized_keys
   ```

4. For multiple-machine mode (*Section 5*), add ssh key of each server to file `$HOME/.ssh/authorized_keys` of other servers in the cluster.
5. Set permission set of server's authorized keys file

```
chmod 600 .ssh/authorized_keys
```

Prompt-less login. When `ssh` is issued to an unknown hosts, it must automatically be added to the list of known hosts (`$HOME/.ssh/known_hosts`) of the Server. It must not prompt for any user input like `Are you sure you want to continue connecting (yes/no)?`

This can be achieved by one of the following ways:

1. Set `StrictHostKeyChecking` option in `$HOME/.ssh/config` file to `'no'` instead of `'ask'.`
2. Make Server remember an unknown hosts by issuing a `ssh` to it before running distributedR. User can provide input to the ssh prompt as required. Hosts will now be added to Server's known hosts (`$HOME/.ssh/known_hosts`) and `ssh` will not make any prompt while running distributedR.

Running distributedR in single-machine mode:

1. Run R by typing 'R' in console. A R-session is opened.

```
$ R
```

2. Inside R session, load distributedR package.

```
> library(distributedR)
```

```
Loading required package: Rcpp
Loading required package: RInside
```

3. Start distributedR. The number of R instances to be started in the Server is configured by parameter "*inst*".

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

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

4. Check status of Worker running in distributedR.

```
> distributedR_status()
```

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

distributedR is ready to run code/algorithms which are written to run in distributed manner on Single Server.

5. Shutdown distributedR.

```
> distributedR_shutdown()
```

```
Shutdown complete
[1] TRUE
```

5. Running distributedR – Multiple Machine mode

To run distributedR in multiple Machine mode, verify that distributedR is installed on all nodes as specified in *Installation section*.

Few additional configurations are required for running distributedR in Multiple machine mode for defining Master and Worker machines and configuring them. These setting are done in Cluster Configuration file.

Cluster Configuration file.

The Cluster Configuration file defines the Master and Worker nodes in the cluster. A sample Cluster Configuration file is in */opt/hp/distributedR/conf/cluster_conf.xml*.

Users are recommended to create a new Cluster Configuration file, namely cluster.xml, at any location on the Master Node with format as:

```
<MasterConfig>
   <ServerInfo>
     <Hostname>eng63</Hostname>
     <Port>8989</Port>
   </ServerInfo>
<Workers>
   <Worker>
     <Hostname>eng64</Hostname>
     <Port>9090</Port>
     <SharedMemory>0</SharedMemory>
     <Executors>10</Executors>
   </Worker>
   <Worker>
     <Hostname>eng10</Hostname>
     <Port>9090</Port>
     <SharedMemory>0</SharedMemory>
     <Executors>15</Executors>
   </Worker>
   <Worker>
     <Hostname>eng34</Hostname>
     <Port>9090</Port>
     <SharedMemory>0</SharedMemory>
     <Executors>15</Executors>
   </Worker>
</Workers>
</MasterConfig>
```

Configuration Options.

1. <ServerInfo> tag specifies Master node configuration and <Workers> tag specifies Worker nodes configuration details. Each <Worker> tag specifies configuation for each Worker node.

2. **Hostname**. Specifies Hostname of your machine. Enter the Master node's Hostname under <ServerInfo> tag and Worker node's Hostname under <Worker> tag.
3. **Port**. This can be any available port on the servers. Please note that Master's port number (<Port> under <ServerInfo>) should be different from Worker node's port number (<Port> under <Worker>) as Master and Worker communicate using these Port Numbers. However, the same Port number can be used for the Worker nodes.
4. If <Executors> and <SharedMemory> options in the Worker nodes is 0, distributedR will automatically determine the settings using System information.

Running distributedR in multiple-machine mode.

To run distributedR on the cluster, it is required that all machines in the cluster should have password-less and prompt-less login to one-another. Also, each machine in the cluster should have password-less and prompt-less login for the command *"ssh 127.0.0.1"*. See instructions in Section 4 - *Running distributedR – Single Machine mode* on how to enable password-less and promptless-less login.

1. Run R by typing 'R' in console. An R-session is opened.

```
  $ R
```

2. Inside R session, load distributedR package.

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

3. Start distributedR. Specify cluster and worker configuration file paths.

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

4. Check status of worker running in distributedR.

```
 > 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
```

distributedR is ready to run code/algorithms on multiple-machine cluster.

5. Shutdown distributedR

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

6. Uninstalling distributedR and supported packages

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

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

1. Check the distributedR RPM installed on the node

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

2. Uninstall distributedR RPM returned by Step 1

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

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

If you currently have HPDGLM installed from distributedR release 0.2.0 on the distributedR cluster, you must uninstall it manually before installing the new version of distributedR.

To uninstall HPDGLM, run following command on each node in the distributedR cluster:

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