Installing and Uninstalling Distributed R

Platform Requirements

To install and run Distributed R, your servers run one of the following platforms:

- 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++
- libxml2-devel
- R (version 3.1.0 and later)

Note: Distributed R was successfully built and tested on R version 3.1.2. For instructions about installing any of the Linux packages listed, please see the Installation section.

Installing Distributed R

A binary version of Distributed R is available as a single tar.gz file. It contains easy-to-run scripts to install requisite packages and libraries, platform and all parallel algorithm R packages offered by Distributed R. Please follow link https://my.vertica.com/distributedr/ to download and install it. Complete product documentation including installation instructions is available at http://www.vertica.com/hp-vertica-documentation/hp-vertica-distributed-r-1-0-x-product-documentation.

The open source version of Distributed R is available at https://github.com/vertica/DistributedR. Following section describes instructions to install Distributed R from source. Installing Distributed R through the steps described in this section also installs following packages supported by Distributed R:

- 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 Distributed R

Note: You must have root or sudo privileges to install Distributed R.

To install Distributed R from source:

1. Download/Check-out distributedR source code from Distributed R Github repository https://github.com/vertica/DistributedR to local machine.
2. Install required software packages

   - If your OS distribution is RHEL or CentOS

   ```
   sudo yum install R
   sudo yum install -y make gcc gcc-c++ libxml2-devel
   ```

   - If your OS distribution is Ubuntu or Debian

   ```
   sudo apt-get install r-base
   sudo apt-get install -y make gcc g++ libxml2-dev rsync
   ```

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_0.11.2.tar.gz
   sudo R CMD INSTALL RInside_0.2.11.tar.gz
   sudo R CMD INSTALL XML_3.98-1.1.tar.gz
   sudo R CMD INSTALL randomForest_4.6-10.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. Install Distributed R and parallel algorithm packages. Upon installation, Distributed R and algorithm packages will be available at `/opt/hp/distributedR`.

   ```
   make install
   ```

   Please note that this step will replace the binary version of Distributed R, if installed on your machine, with the open sourced version of Distributed R.

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 *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 set of server's authorized keys 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:

1. Set `StrictHostKeyChecking` option in `$HOME/.ssh/config` file to **`'no'`** instead of **`'ask'`**.
2. 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

Loading required package: XML
```

3. Start Distributed R. The number of R instances to be started in the Server is configured by the variable *inst*.

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

```
Workers registered - 1/1.
All 1 workers are registered.
Master address:port - 127.0.0.1:50000
```

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

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

5. Shutdown Distributed R.

```
> distributedR_shutdown()
Shutdown complete
```

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 under *conf/cluster_conf*.*xml* in Distributed R source. The defined cluster configuration file should be specified when starting distributedR using distributedR_start() API using “cluster_conf=” argument.

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>mach01</Hostname>
     <StartPortRange>50000</StartPortRange>
     <EndPortRange>50100</EndPortRange>
   </ServerInfo>
<Workers>
   <Worker>
     <Hostname>mach01</Hostname>
     <StartPortRange>50000</StartPortRange>
     <EndPortRange>50100</EndPortRange>
      <SharedMemory>0</SharedMemory>
     <Executors>10</Executors>
   </Worker>
   <Worker>
```

```
    <Hostname>mach02</Hostname>
    <StartPortRange>50000</StartPortRange>
    <EndPortRange>50100</EndPortRange>
    <SharedMemory>0</SharedMemory>
    <Executors>15</Executors>
  </Worker>
  <Worker>
    <Hostname>mach03</Hostname>
    <StartPortRange>50000</StartPortRange>
    <EndPortRange>50100</EndPortRange>
    <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 the master machine. Enter the Master node's Hostname under `<ServerInfo>` tag and Worker node's Hostname under `<Worker>` tag.
3. **Port Range**. It is defined in mandatory `<StartPortRange>` and `<EndPortRange>` tags. These tags define the range of port numbers that is allowed be used by the 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 is 0 or not defined, 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 R session, load Distributed R package.

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

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

   ```
   > distributedR_start(cluster_conf=”<path to cluster.xml>”)
   Workers registered - 3/3.
   All 3 workers are registered.
   Master address:port - mach01:50000
   ```

4. Check status of workers running in Distributed R.

   ```
   > distributedR_status()
        Workers Inst SysMem MemUsed DarrayQuota DarrayUsed
   1 mach01:9090   15  96682   88558       43506          0
   2 mach02:9090   15  96682   67212       43506          0
   3 mach03:9090   10  96682   73216       43506          0
   ```

5. Shutdown Distributed R

   ```
   > distributedR_shutdown()
   Shutdown complete
   ```

Uninstalling Distributed R and parallel algorithm packages

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

To uninstall the binary version of Distributed R, please follow the instructions in the Distributed R Product documentation available at http://www.vertica.com/hp-vertica-documentation/hp-vertica-distributed-r-1-0-x-product-documentation.

To uninstall the open source version of Distributed R and supported parallel algorithm packages:

1. Change directory (cd) to Distributed R trunk
2. Uninstall Distributed R and parallel algorithm packages:

   ```
   make uninstall
   ```

3. Clean Distributed R build, including third party compilation, from local directory:

   ```
   make distclean
   ```

Distributed R and its supported packages are uninstalled. However, requisite software packages and standard R packages such as Rcpp, Rinside, data.table, randomForest and XML are not uninstalled. Follow steps 1-3 to uninstall Distributed R and supported packages in each node in the cluster.

Using Distributed R with R Studio

To use Distributed R with R Studio.

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 descibed in Section 5 - *Running Distributed R – Single Machine mode* and Section 6 - *Running Distributed R – Multiple Machine mode.*