Distributed R Frequently Asked Questions

HP Vertica Development Team

1 Overview

What is Distributed R?

Distributed R is a High-Performance Scalable Platform for the R language. It enables R to leverage multiple cores and multiple servers to perform advanced Big Data analytics. It consists of new R language constructs to easily parallelize algorithms across multiple R processes.

When should I use Distributed R?

Distributed R allows you to overcome the scalability and performance limitations of R to analyze very large data sets. Distributed R provides the ability to run analysis on the complete data set and not just the sample.

Which algorithms are implemented using Distributed R?

Generalized Linear Models (glm) for performing many forms of regressions - includes logistic, linear, and poisson regression. For more details refer to HPDGLM-Manual.pdf and HPDGLM-UserGuide.pdf.

How can I convert existing R programs to scale using Distributed R?

Distributed R provides simple and powerful tools for distributed computing in R. Data-structures such as distributed array, *darray*, enables R algorithms to handle big data by distributing data across machines in a cluster. Language primitives such as *foreach* in Distributed R enables R developers to easily parallelize algorithms. To convert an R program to scale using Distributed R, the programmer should use data-parallelism techniques to break the program into smaller sequential functions, apply them on data partitions, and then aggregate partial results. Please refer to Distributed-R-UserGuide.pdf to understand the programming model

Can I use other R packages with Distributed R?

Yes. Distributed R packages can be loaded with other R packages in the Comprehensive R Archive Network (CRAN). Distributed R can execute parallel programs that call existing R packages. For more details refer to the Parallel execution using existing packages section in Distributed R-UserGuide.pdf.

Can I benefit from running Distributed R in a single multi-core machine?

Yes. Distributed R can leverage multiple cores by distributing processing across multiple cores similar to multi-node cluster.

How do I use algorithms developed on Distributed R?

Usage of Distributed R algorithm functions are similar to any other R based algorithm function. However, Distributed R algorithms use distributed datastructures instead of simple arrays. For example, hpdglm implements a distributed alternative for R glm. The signature of hpdglm is the same as R glm, except that it uses distributed arrays as input instead of simple arrays. For more details refer to HPDGLM-UserGuide.pdf

What are the hardware requirements to run Distributed R?

You can run Distributed R on commodity hardware. You need enough total memory to hold all of the data you want to analyze, along with a buffer for R's bookkeeping. If you have hardware with more memory, you will need fewer to-

tal nodes to do the analysis. We recommend a stand-alone cluster of HP DL380s.

Can I run Distributed R in the cloud or on an appliance?

Distributed R is not currently tested in the cloud or on HP Vertica appliances.

Is Distributed R part of HP Vertica? Can I use any database with Distributed R?

Distributed R is a collaboration project between HP Labs and HP Vertica. It was created at HP Labs. Today, we are working together to make an offering of it. You can store your data in HP Vertica and use our optimized connector vRODBC to load data into Distributed R. You can load data from other sources as well.

Does Distributed R replace HP Vertica R UDX?

No. Distributed R does not replace HP Vertica R UDx. R UDx works best when your data can fit into memory on a single node and you are satisfied with the processing time. Distributed R is a way for us to improve our offering in this space.

Does Distributed R work with RStudio?

Distributed R works with common R development environments. However, we have primarily tested it with the default R console.

2 Installation

What are the steps to install Distributed R?

Prior to installing Distributed R ensure you have a supported Linux operating system and software pre-requisites installed. The installation document, Distributed-R-Installation-Guide.pdf, describes detailed installation steps.

Can I install Distributed R on a single server?

Yes. If you have a single server with multiple cores, Distributed R enables you to leverage multiple-cores to improve performance. The installation document section Running Distributed R Single Machine mode provides more details of how to install and use Distributed R in a single machine.

How do I verify the successful installation of Distributed R in a cluster environment?

Use `distributedR_status()` command to verify that all nodes up and running in a cluster.

3 Performance and scalability

What is the upper limit of the size of data that Distributed R can analyze?

We have tested regression on more than 1 terabyte of data. We tested few permutations of this data set size, varying the number of rows and columns. We would like to understand your scalability and data size needs. Please either post your requirements to our beta email distribution list, *VerticaDistributedRBeta@external.groups.hp.com*, or email us at *sunil.venkayala@hp.com* and *geeta.aggarwal@hp.com*.

How do you characterize the scalability of Distributed R?

We have noticed near linear scaling with regression. This behavior is with a fixed data set and increasing number of nodes. We have also observed weak scaling (execution time remains nearly constant when number of nodes and data set is increased proportionally). We are working on characterizing scalability based on the type of algorithm you are using.

I have a small dataset. Should I use Distributed R?

It is known that distributed processing systems incur additional overheads due to communication and bookkeeping. Therefore, Distributed R may not be beneficial to you if your dataset size is small and the execution time of your program is in seconds or a minute. Use vanilla R in such cases.

How do I load data from HP Vertica to Distributed R?

Use connectors such as RODBC to load data

from HP Vertica. We provide a fast ODBC connector called vRODBC. Additionally, the HPDGLM package has a default `dataLoader` function that uses ODBC connections to load data in parallel from HP Vertica.

What happens if a node goes down?

Distributed R detects that the node is down and prompts you to restart the session.

What happens if I load Distributed R with too much data?

Distributed R prompts you that there is not enough memory and asks you to restart.

How much hardware do I need for a particular data set size?

The rule of thumb is that you need enough servers so that the whole dataset fits into the total memory of the machines, along with a buffer for bookkeeping by R. If you want the computations to run faster, you need to use more CPU cores.

4 Limitations

What are the current limitations of Distributed R?

Distributed R is in beta stage and we want to use this opportunity to obtain feedback. We are actively improving the system. Some limitations are listed below:

- Distributed R currently supports only one session on a cluster. Multiple users may not be able to run concurrently on the same machines.
- Debugging distributed programs is, in general, challenging. We plan to add support for debugging tools for Distributed R. If you make a programming mistake or have a bug in the code, you may need to restart the Distributed R session.