# Distributed R FAQ

## GENERAL

**1. 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 Big Data Advanced Analytics. It consists of new R language constructs to easily parallelize algorithms across multiple R processes.

**2. When should I use Distributed R?**
Distributed R should be used when you have performance and scalability issues with R. Distribute 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 for when you want to use all the data, and not just the sample.

**3. Which algorithms are implemented using Distributed R?**
Distributed R 0.4 version includes following parallel algorithms

- Generalized Linear Models
  - Logistic Regression
  - Linear Regression
  - Poisson Regression
- Classification
  - Random Forest
- Clustering
  - k-Means
- Graph Analysis
  - PageRank (In-development)

**4. How can I convert existing R programs to scale using Distributed R capabilities?**
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. 'foreach' and other new language constructs in Distributed R enables R developer to easily parallelize algorithms.
To convert an R programs to scale using Distributed R, the programmer should use data-parallelism techniques to break the program into smaller sequential functions that are applied on data partitions then aggregate partial results.
Please refer to Distributed-R-UserGuide.pdf to understand the programming model

**5. Can I be able to use other R packages with Distributed R?**
Yes. Distributed R packages can be loaded with other R packages just like any other package 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.

**6. Can I benefit running Distributed R in a single machine with multiple cores?**
Yes, Distributed R can leverage multiple cores by distributing execution across multiple cores similar to multi-node cluster.

**7. How do I use algorithms developed on Distributed R?**
Usage of Distributed R algorithm functions are similar to any other R based algorithm functions. However, Distributed R algorithm functions use distributed arrays 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'

**8. Can Distributed R work with R-Studio?**
Yes, Distributed R works with common R development environments. However, we have primarily tested it with the default R console.

**9. What hardware is required to run 'Distributed R'?**
You can run Distributed R on commodity hardware. In total, you need enough total memory to hold all of the data you want to analyze, along with a buffer for R bookkeeping. If you have hardware with more memory, you will need fewer total nodes to do the analysis. We recommend a stand-alone cluster of HP DL380's.

**10. Can you run "Distributed R" in the cloud or on an appliance?**
Distributed R is not currently tested in the cloud or on HP Vertica appliances.

**11. 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've tested a 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.

**12. Is 'Distributed R' a part of HP Vertica? Can I use any database with 'Distributed R'?**
The Distributed R effort is a collaboration 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 we have optimized data loading through vRODBC to Distributed R. You can load data from other sources as well.

**13. Does Distributed R replace the HP Vertica R UDX?**
Distributed R does not replace the 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.

## INSTALLATION

**1. How is Distributed R installed?**
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.

**2. Can I install Distributed R in a single server?**
Yes. When 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.

**3. How do I verify the successful installation of Distributed R in multiple node cluster environment?**
Use distributedR_status() command and verify the status log to ensure all nodes up and running in a cluster.

## MISCELLANEOUS

**1. How do I load data from HP Vertica to Distributed R?**
Data can be loaded from HP Vertica using an RODBC connector. We provide a fast ODBC connector called vRODBC. HPDGLM package has dataLoader function that makes concurrent ODBC connections to load data in parallel from HP Vertica.

**2. What happens if a node goes down?**
Distributed R detects that the node is down and prompts you to restart the session.

**3. 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.

**4. How do I know how much hardware I need for a particular size data set?**
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 need your answer faster, you need to use more CPU cores.

**5. How do you characterize the scalability of Distributed R?**
We have noticed near linear strong scaling with logistic regression. This is with a fixed data set and increasing the number of nodes. We have also noticed weak scaling (increase number of nodes, increase data set linearly, expect same time for results) with logistic regression as well. We are working on characterizing scalability based on the type of algorithm you are using.