EXECUTIVE WHITE PAPER

Delivering value from big data with Microsoft R Server and Hadoop

Microsoft Advanced Analytics Team

April 2016

Microsoft

ABSTRACT

Businesses are continuing to invest in Hadoop to manage analytic data stores due to its flexibility, scalability, and relatively low cost. However, Hadoop's native tooling for advanced analytics is immature; this makes it difficult for analysts to use without significant additional training and limits the ability of businesses to deliver value from growing data assets.

Microsoft R Server leverages open-source R, the standard platform for modern analytics. R has a thriving community of more than two million users who are trained and ready to deliver results.

Microsoft R Server runs in Hadoop. It enables users to perform statistical and predictive analysis on big data while working exclusively in the familiar R environment. The software uses Hadoop's ability to apportion large computations for transparently distributing work across the nodes of a Hadoop cluster. Microsoft R Server works inside Hadoop clusters without the complex programming typically associated with parallelizing analytical computation.

By leveraging Microsoft R Server in Hadoop, organizations tap a large and growing community of analysts—and all of the capabilities in R—with true cross-platform and open standards architecture.

HADOOP FOR ADVANCED ANALYTICS

By 2020, analysts are predicting that the Hadoop market will reach $50.2 billion. Hadoop is an open-source software framework for distributed data management; it supports resource management (YARN), a programming model (YARN/MapReduce), and a file system (HDFS) that work together to provide a data management and analysis platform that is flexible, highly scalable, and relatively inexpensive to implement.

For some types of data, Hadoop has emerged as the clear platform of choice:

- Clickstream data, including site-side clicks and web media tags
- Sentiment data, including news feeds, blog feeds, social media comments, and Twitter streams
- Telematics, such as vehicle-tracking data
- Sensor and machine-generated data
- Geotracking and location data
- Server and network logs
- Document and text repositories
- Digitized images, voice, video, and other media

Enterprises struggle, however, to analyze data held in Hadoop[1] due to the complex programming model required to embrace MapReduce. While Hadoop is an excellent platform for managing large and complex data sets, its tooling for advanced analytics is much less mature than the analytics tooling available to enterprises today on other platforms.

Hadoop's "native" advanced analytics project, Apache Mahout, is an eclectic mix of algorithms and hasn't captured wide adoption. New algorithm libraries available via the Apache Spark platform can also be combined with Hadoop, but are not yet fully integrated with the R language and can be insufficiently robust for enterprise usage.

Conventional server-based analytic software packages from vendors such as SAS or SPSS typically offer the ability to connect with Hadoop and extract data over a network, but they do not run inside Hadoop. Using server-based software with Hadoop is suitable for users working with small data sets or whose needs can be met using small samples of the data. However, this architecture is impractical when source data approaches terabyte scale, and sampling is not acceptable for the prescribed analytic applications.

Hosting the analytic software outside of Hadoop also creates a deployment problem, since custom programming may be required to implement a production scoring process. Moreover, data replication and movement can create data security issues. For "curated" data (e.g., clinical trials), the organization must recertify the data after replication or movement. This is time-consuming and expensive.

Some legacy vendors have added Hadoop-based alternatives, but most suffer from high pricing, require installation on dedicated clusters, and require users to convert legacy analytics applications in order to run them in the new systems. As a result, many users of legacy analytics packages are eschewing the path laid out by established analytics vendors in favor of open-source approaches such as Microsoft R Server.

Some vendors have introduced software that runs on freestanding "Analysis" servers within the Hadoop cluster or runs in memory on the Hadoop data nodes. While deployment within the Hadoop cluster reduces latency from data movement, this approach is still only practical with smaller data sets. Analytics that load the entire data set into memory typically cannot run on the commodity hardware commonly used as Hadoop data nodes.

As a result of the limited prebuilt analytics available in Hadoop, much of the analytic work performed in Hadoop is "homegrown" or built with custom programming in MapReduce, Java, or other languages. This is a serious constraint for organizations seeking to leverage Hadoop for analytics because data scientists, the individuals with the necessary skills, are scarce and in high demand. Moreover, custom "homegrown" applications are expensive to build, maintain, and support.

There is a pressing need for an analytics platform that runs inside Hadoop, analyzes data that scales into the terabytes, offers comprehensive support for advanced analytics, and is accessible to a broad user base without extensive retraining.

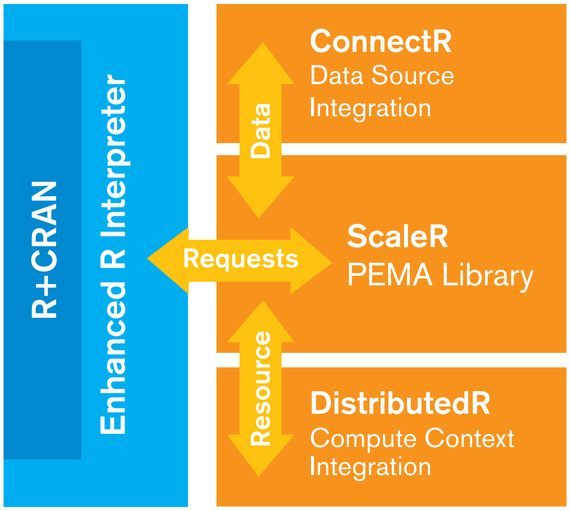
Microsoft R Server for big-data analytics

Open-source R is widely used by millions of statisticians, data miners, actuaries, researchers, and other professionals who need an agile and capable analytics platform. This user base is expanding rapidly. Recent surveys[2,3,4] show that working analytics professionals prefer R to all other analytic software options.

There are many reasons behind R's growing popularity, including its flexibility, rich graphics, broad ecosystem, and comprehensive library of more than seven thousand packages. However, since open-source R runs most functions and algorithms in memory and lacks a native distributed computing framework, R alone is ill-suited to working with large data sets.

Microsoft R Server is a big-data analytics platform built with open-source R at its core. Microsoft R Server extends an enhanced open-source R with performance improvements, support, and a distributed computing framework that includes scalable algorithms, data connectivity tools, an integrated development environment, and enterprise deployment tools.

Microsoft R Server includes the ScaleR package of high-performance analytic algorithms called Parallel External Memory Algorithms (PEMAs) that make open-source R scalable. PEMAs are rewritten into compiled languages, can execute work remotely, and are rearchitected to conduct computation on multiple machines in parallel. With these improvements, ScaleR PEMAs support operations at speed on data far larger than available memory through a combination of streaming, parallel processing on multiple cores, and distributed processing across multiple nodes.



Microsoft R Server's Write Once Deploy Anywhere (WODA) architecture enables users to develop scripts and algorithms on one platform and deploy on any other platform supported by Microsoft R Server or SQL Server R Services. These include workstations, servers, data warehouse appliances, and Hadoop clusters, running on-premises or in the cloud.

Microsoft R Server for Hadoop makes R analytics more deployable into production environments. The Microsoft R Server's DeployR component provides enterprise integration, allowing users to publish analytical results including visualizations and script results to a broad array of tools and platforms. DeployR enables two-way web services–based integration with popular tools like Excel, QlikView, Tableau, and Power BI.

In brief, Microsoft R Server for Hadoop offers a scalable, high-performance platform for the rich capabilities of R. It offers true cross-platform integration together with a wide choice of user interfaces and deployment options.

Microsoft R Server for Hadoop

In 2015, Microsoft purchased Revolution Analytics, a pioneer in R analytics for Hadoop users. In 2011, Revolution Analytics introduced the RHadoop open-source project, a popular project on GitHub.

Microsoft R Server supports interfaces to MapReduce, HDFS, and HBase, powering production analytics for enterprise customers:

- A life-sciences company delivers recommended treatments to its customers based on user-specified details of planned usage.
- A leading US bank has developed an innovative prediction engine.
- A marketing-sciences company analyzes billions of cookies and session records in a matter of hours instead of days.

Microsoft R Server does not simply connect to Hadoop—it runs inside Hadoop in YARN MapReduce, providing users with direct access to data stored in Hadoop from the familiar R interface. To do this, Microsoft R Server uses YARN MapReduce to distribute computational workload across the nodes of a Hadoop cluster. This section includes a detailed explanation of how this works. When Spark is installed within the cluster, a forthcoming version of Microsoft R Server will also enable distribution of workloads into the fast memory-based computational framework that is part of Apache Spark.

Deployment architecture

Figure 1 shows Microsoft R Server deployed for operations inside Hadoop. The diagram shows Microsoft R Server deployed on each node in the Hadoop cluster, with one node designated as the Microsoft R Server master node. The diagram also shows Microsoft R Server deployed on one or more desktops or servers outside of the Hadoop cluster. This is optional, but useful as a development environment or as a platform for analytics with small data sets.

Microsoft R Server's DeployR interface supports the web services connection shown in Figure 1. The Corporate Applications shown in the diagram can include interactive interfaces to popular BI platforms (such as Jaspersoft, QlikView, or Power BI), end-user tools like Microsoft Excel, custom web-based reports, application servers, and rules engines.

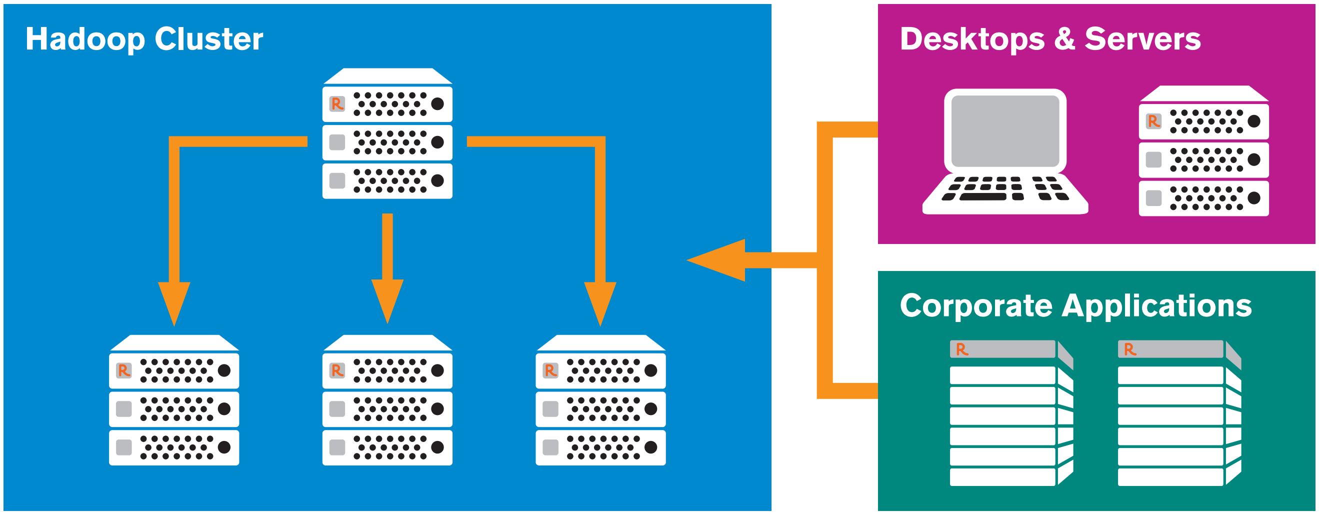


Figure 1: Leveraging inherent parallelism in Hadoop to accelerate big-data analytics

Execution in Hadoop: high-level flow

Users invoke Microsoft R Server in Hadoop in one of two ways: local or remote.

> **Local:** Microsoft R Server users submit individual commands or scripts through the R command line interface.
>
> **Remote:** Microsoft R Server users working on a separate platform run a script that includes a command to transfer execution to a remote Hadoop cluster.

In both cases, Microsoft R Server running on an edge node acts as the master process. It executes the user's Microsoft R Server script locally, using threads, cores, and sockets within the edge node server to achieve scale.

When the master process encounters calls to one or more PEMA algorithms in the user's script, it calls the PEMA's interface, allowing the PEMA to distribute subsets of the work to data nodes in the Hadoop cluster via the job request.

Once the data nodes finish their tasks, a single Reducer consolidates intermediate results and returns them to the master node. If the master node determines that the algorithm has completed its work, it returns the consolidated results to the script or command.

For iterative algorithms (such as logistic regression or k-means clustering), Microsoft R Server performs a convergence test and checks the number of iterations. If the tests succeed, Microsoft R Server returns consolidated results to the script or command; otherwise, it repeats the process.

Execution in Hadoop: detailed flow

This section details execution steps for Microsoft R Server running inside Hadoop.

1. An R script, running on a Microsoft R Server deployed to an edge node in the Hadoop cluster, encounters a call to a ScaleR algorithm.
2. The master process serializes instructions, stores them in an object, and saves the object to a GUID file in HDFS (where the instructions are accessible to Mappers and Reducers).
3. The master process prepares a job request for YARN MapReduce's JobTracker, passing the location of the input data file and the processing instructions for Mappers and Reducers.
4. Once invoked, JobTracker spawns a set of TaskTrackers on each worker node to ensure that all nodes of the distributed input file are analyzed. When TaskTrackers are started, they are passed the URL of the instructions file and the HDFS location of a single split of data.
5. Each TaskTracker invokes a Mapper, passing input data split location and the instruction file location. The Mapper consists of a Java wrapper housing a complete instance of Microsoft R Server. Data splits roughly equate to a single HDFS file segment. Once started, the TaskTracker awaits completion of the Mapper or Reducer.
6. TaskTracker provides each Mapper it starts with data split (distributed file segment) locale and instructions from a GUID instructions file. The Mapper task ingests the designated data split and performs the requested operations. Mappers operate independently of one another.
7. When each Mapper finishes its work, it produces an Intermediate Results Object, or IRO, in a key value format. The Mapper serializes the IRO and stores it in a GUID file at a location passed to it by TaskTracker.
8. As Mappers begin to complete, Hadoop's shuffle-sort sequentially moves all IROs to the Reducer's host node.
9. As IROs begin to arrive, TaskTracker invokes the Reducer and passes it the GUID of the instructions file directing the job.
10. The Reducer reads the instructions file and consolidates the inbound IROs into a single consolidated IRO.
11. When there are no remaining IROs, the Reducer saves the consolidated IRO to HDFS in a location directed by the master process.

12. The Reducer exits, its TaskTracker exits, and the master process assumes control.
13. Upon return of control, the master process retrieves and evaluates the consolidated IRO to see if the process has converged.
14. If the convergence test succeeds, the master process returns a final results object to the Microsoft R Server command or program.
15. If the convergence test does not succeed, the master process repeats steps 2–13 with new instructions.
16. Processing ends when the convergence test succeeds or the number of iterations reaches a user-defined limit.

All IROs produced by a particular job use the same key; this causes subsequent shuffle-sort operations in Hadoop to consolidate all IROs on a single node using a single Reducer instance. It is worth noting that IROs are tiny by comparison to input data, so this step is a performance improvement over the use of combiners or other multiple-Reducer schemes to consolidate results.

Microsoft R Server and Hadoop: the benefits

Microsoft R Server brings tools to Hadoop that enable organizations to deliver new value from big data:

- Understand site-side customer behavior.
- Optimize search term selection.
- Measure the impact of web media campaigns.
- Track brand perceptions among current and prospective customers.
- Track driving behavior of policyholders.
- Identify anomalies in fleet vehicle usage.
- Predict machine failures before they happen.
- Drive location-specific marketing offers.
- Detect unusual behavior in secure systems.
- Predict network outages.
- Identify plagiarized documents.

With Microsoft R Server, data remains in place in Hadoop; users perform analysis without data movement. This dramatically reduces the total cycle time to build and deploy predictive models, and eliminates potential security issues from data movement or replication.

Microsoft R Server makes Hadoop easier to use. Users can focus on what they do best—drawing insight from data—and do not need to learn parallel programming, distributed data management techniques, Java, MapReduce, or HDFS. By making Hadoop available to the R community of users, Microsoft R Server enables organizations to tap into a large and rapidly growing talent pool.

Microsoft R Server's Write Once Deploy Anywhere architecture offers flexibility and portability while helping organizations avoid vendor platform "lock-in." Users can build and test projects on a workstation, then deploy them on a production system essentially without change. Organizations can easily rehost projects, for any reason—for better performance, better economics, to move computing to the cloud, or simply for convenience.

Microsoft R Server brings rich analytical and statistical computation at scale to Hadoop and enables data analytics teams to achieve shorter, less costly, and lower-risk projects while attaining far higher performance, data scale, and portability than can be accomplished by other methods.

FAQs

Q. *What input sources does Microsoft support for use with Microsoft R Server?*

A. Microsoft R Server can read data from many sources using various connectors. Microsoft R Server ingests input data in parallel from HDFS when running inside Hadoop.

Q. *What data input formats can we use?*

A. For Hadoop users, Microsoft R Server supports text files in CSV or fixed-width file formats. For data residing in other systems, Microsoft R Server supports many other formats—including both SAS and SPSS files—plus a fast proprietary format called XDF. XDF is an efficient binary serialization format used by Microsoft R products to accelerate data access and manipulation.

Q. *Where can data be output?*

A. There are two options available to Hadoop users.

First, Microsoft R Server returns data in R data objects to the calling program or application. These objects are available for further processing and export, or they are published through the DeployR interface in a variety of formats.

Second, users can run prediction functions to score data and write it back to HDFS.

Q. *What formats can we use to return results?*

A. Users can write model scores directly to CSV or XDF, publish models as PMML documents, or pass R objects to DeployR server for broader integration.

Q. *Are there intermediate formats used?*

A. Between Mappers and Reducers, Microsoft R Server passes data in standard Hadoop key/value pairs. These are stored in RAM and swapped to local storage if necessary.

Q. *How many MapReduce jobs run?*

A. Typically, one MapReduce job runs per algorithm called. That job may be iterative in the case of multipass algorithms like logistic regression or clustering.

Q. *How many actual Mapper and Reducer instances run?*

A. Microsoft R Server schedules a variable number of Mappers depending on the size of the data set. For each split of input data, Microsoft R Server schedules one Mapper. Splits roughly approximate the block size setting of the HDFS file system instance, typically either 64 or 128 MB. The HDFS Administrator configures the HDFS block size.

Q. *Why run only a single Reducer or no Reducer at all?*

A. Unlike many MapReduce-based applications, ScaleR and Microsoft R Server typically use a single Reducer and, in some cases, such as scoring data, no Reducer at all. Reducers only run if needed to return data to the master process on the edge node; otherwise, Microsoft R Server skips this step.

Q. *What happens if a Microsoft R Server job fails while running in Hadoop?*

A. When running ScaleR algorithms, Microsoft R Server may invoke a large number of Mappers to accomplish required calculations. The results of each map instance are persisted to memory and files, allowing Hadoop to reschedule only failed Mapper instances in the event of incomplete execution.

Q. *How do we run model scoring?*

A. Microsoft R Server offers users two options for model scoring.

Most commonly, users run scoring natively using the rxPredict function. This function applies model objects produced by any of the ScaleR predictive modeling functions to score each data record in the input file. rxPredict computes model scores and writes the scores to a file in HDFS.

Less frequently, users can choose to export model objects for execution elsewhere using Predictive Model Markup Language (PMML). Organizations can then deploy PMML model documents into any PMML-compliant database.

Endnotes:

1. May 10, 2013, Hadoop Adoption Accelerates, But Not For Data Analytics, http://readwrite.com/2013/05/10/hadoop-adoption-accelerates-but-not-for-what-you-might-think#awesm=~onhD75n7uuzq23

2. September 02, 2013, Poll: R top language for data science three years running, http://blog.revolutionanalytics.com/2013/09/top-languages-for-data-science.html

3. October 15, 2013, R usage skyrocketing: Rexer poll, http://blog.revolutionanalytics.com/2013/10/r-usage-skyrocketing-rexer-poll.html

4. May 2013, The Popularity of Data Analysis Software, http://r4stats.com/articles/popularity/

This data sheet is for informational purposes only. Microsoft makes no warranties, express or implied, with respect to the information presented here.