An Oracle White Paper
July 2019

Scaling R to the Enterprise

Using R for Enterprise-level Performance, Scalability, Ease of Production Deployment, and Security

Disclaimer

The following is intended to outline our general product direction. It is intended for informational purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle products remains at the sole discretion of Oracle.

Introduction: Beyond the Laptop

Formulated in 1994 as an alternative to proprietary statistical environments, R is an integrated suite of software facilities for statistical analysis, data manipulation, machine learning, and graphical display. This open source scripting language has become an important part of the analytical arsenal for data scientists, business and data analysts, and statisticians. With millions of R users worldwide leveraging thousands of open source R, the R ecosystem enhances user productivity in a wide range of domains, including bioinformatics, spatial statistics, financial market analysis, and linear/non-linear modeling.

While data scientists often run R programs on their personal computers or workstations, increasingly they need to do advanced computations on large volumes of data quickly. To enable using R on data at scale, Oracle has created a wide range of options for conducting statistical and graphical analyses on data stored in Hadoop or Oracle Database, bringing enterprise-level capabilities to projects that require high levels of security, scalability and performance, as well as the ability to deploy their R scripts into production quickly and easily, either on premises or in the Oracle Public Cloud.

Oracle Machine Learning Platform combines the advantages of R with the power and scalability of Oracle Database and Hadoop. Oracle Machine Learning for R is supported by Oracle R Enterprise from the Oracle Advanced Analytics option to Oracle Database Enterprise Edition and is available with Oracle Cloud offerings. Oracle Machine Learning for Spark is supported by Oracle R Advanced Analytics for Hadoop, a component of the Oracle Big Data Connectors, and is available with Oracle Big Data Service. R programs and packages can be used in conjunction with these assets to process large volumes of data in a secure environment. Customers can build statistical models and execute them against local data stores as well as run R commands and scripts against data stored in a secure corporate database. Using Oracle Advanced Analytics, R scripts can be executed via SQL. The R script results, including structured results and images, can be immediately consumed by applications and dashboard tools.

These capabilities are especially important for many of today's big data projects. Data Scientists can obtain controlled access to data in Oracle Database, thereby accelerating productivity while enforcing IT security policies. Oracle's integrated approach simplifies data analysis, minimizes or eliminates data movement, and shortens the time it takes to transform raw data into actionable information. Using Oracle Big Data SQL extends the reach of Oracle Machine Learning for R to other data sources and allows R users to manipulate data in Oracle Database, a Hadoop environment, and NoSQL.

Oracle Machine Learning has two components that provide support for users of both SQL and R. Through Oracle Machine Learning for R (OML4R), supported by Oracle R Enterprise (ORE), R users transparently manipulate data in Oracle Database using standard R syntax, without data movement, leveraging Oracle Database as a high-performance compute engine. By translating R function invocations to SQL, users leverage in-database statistical techniques and data-parallelism for enhanced scalability and performance. Oracle Machine Learning provides a powerful set of in-database machine learning algorithms that execute within Oracle Database. It also provides the ability to execute user-defined R scripts on database server machine R engines, under the control of Oracle Database. These user-defined R scripts can leverage Comprehensive R Archive Network (CRAN) – the open-source R repository – packages, which can be invoked from either R or SQL. Oracle Machine Learning users have access to a range of Oracle-provided and third-party R GUI and IDE options targeting the user spectrum from business analysts to data scientists.

In addition, Oracle Machine Learning for Spark, supported by Oracle R Advanced Analytics for Hadoop (ORAAH), one of the Oracle Big Data Connectors, enables R users to manipulate data transparently in Apache Hive and Apache Impala using standard R syntax. It also provides a rich set of Spark-based parallel

distributed machine learning algorithms, including exposing many Spark MLlib algorithms through a well-integrated R interface. Users can also execute custom MapReduce jobs from R. The mapper and reducer functions are written in R and can leverage CRAN packages.

OML4R can be used in conjunction with Oracle's redistribution of open-source R, called Oracle R Distribution, as well as with the open-source R distribution. Oracle R Distribution can be readily enhanced with high-performance libraries such as Intel's MKL for enhanced linear algebra and matrix processing. Oracle R Distribution is supported by Oracle.

In this paper, we highlight how Oracle enhances open-source R by enabling developers to:

- Transparently analyze and manipulate data in Oracle Database or Hadoop
- Execute R scripts through the database with data and task parallelism
- Use in-database SQL-based algorithms seamlessly through R
- Score R models in the database
- Easily execute R scripts from SQL statements
- Integrate R into the IT software stack

By integrating R with both Oracle's core database and infrastructure offerings and Hadoop, your organization can realize the best of both worlds: obtain a familiar yet powerful statistical environment along with vastly improved scalability, performance, and security.

Run R Code at Oracle Database

Oracle developed a set of R packages that allows R computations to be executed within Oracle Database. This *transparency layer* makes Oracle tables and views accessible to the R environment as if they were native R objects through ore.frames, which are a subclass of data.frame. This allows users to execute a wide range of R functionality. Data Scientists can use their favorite R IDE for exploratory data analysis and to access a wide range of analytical capabilities in a natural statistical language – R – without the need to know SQL. R users can also leverage the OREdplyr package, which provides overloaded functionality from the popular open source R dplyr package. These features allow users to focus on data analytics opportunities rather than data access, scalability, and performance challenges.

The transparency layer allows R developers to use familiar environments, languages and tools. Under the covers, overloaded R functions execute within Oracle Database – taking advantage of database parallelism, query optimization, column indexing and partitioning – leveraging the rich in-database library of statistical functionality. R users can execute these complex computations within the database using their standard R development skills and tools.

Users can build, evaluate, share, and deploy predictive analytics methodologies, while also utilizing high-performance parallel and distributed machine learning algorithms from Oracle. These in-database algorithms also accept text columns from tables and views for integrated text mining automated term and theme extraction. The extracted data is then combined with other predictors in building models and scoring data.

Further enhancing data scientist productivity, users can automatically create ensembles of models – called *partition models* – where each component model is built on a user-specified partition of the data. Scoring is enabled and simplified using a single integrated model.

Data Scientists and application developers can easily scale their analytic projects as data volumes increase by bringing the algorithms to where the data reside. For example, they can use native parallel distributed in-database algorithms like decision trees, support vector machine, k-means, neural networks, stepwise regression, and random forest for scalable machine learning. They can analyze data and make predictions even faster when they run on Oracle Exadata – one of Oracle's powerful engineered systems - since this processing can take place at the storage tier. This allows organizations to gain further benefits from the extreme performance provided by Oracle Engineered Systems. The benefits of the Oracle approach are clear:

- Work solely from within R for data preparation, analysis, and visualization
- Use the database as a high-performance compute engine with query optimization, column indexing, and parallelism, and optional functionality for in-memory execution and partitioning
- No need to manage flat file data, or wrestle with the associated complexity of storage, backup, recovery, and security
- Minimize R memory constraints so you can handle big data requirements
- Execute R scripts from SQL for ease of deployment and integration with enterprise applications and dashboards

Enjoy In-Database Support for CRAN Packages

Oracle's *embedded R execution* capability allows Data Scientists to leverage thousands of specialized algorithms from the Comprehensive R Archive Network (CRAN) repository. They can write their own algorithms or download existing ones, and then install these packages in database server-side R engines. This architecture makes it easy to send and receive data to and from the database and feed it directly to their chosen algorithms.

By taking advantage of parallel feeds through indexing it is possible to run advanced and complex algorithms. For example, you might divide a customer table by zip code and run multiple R engines in parallel to process groups of customers from many different zip codes concurrently, all without leaving the R environment. R scripts that expose a wide variety of statistical techniques—some accessible through the transparency layer and some through CRAN packages—can be built and stored in Oracle's in-database R script repository.

- Crate your own packages in R and execute them at the database server machine under control of Oracle Database
- Leverage CRAN open-source packages
- Enable "lights-out" execution of R scripts via a SQL interface using Oracle Database scheduling
- Speed up large jobs with data-parallel and task-parallel R script execution under the control of Oracle Database
- Integrate results with applications and BI dashboards and reports

Enhance Machine Learning with Graph Analytics

For those interested in leveraging the powerful graph analytics present in the Oracle Spatial and Graph option (licensed separately), OML4R provides the package OAAgraph that eases working with both in-database machine learning algorithms and the Parallel Graph AnalytiX (PGX) engine. Prepare your data using R with Oracle Database, build models and score data to augment graph data and analysis, and

compute graph metrics to augment data provided to in-database machine learning algorithms – all with the goal of boosting model quality and graph analytics.

Deploy R Analytics in Production

Oracle Machine Learning enables R developers to use the database to execute R scripts within SQL queries. This makes it easy to operationalize R scripts within a standard business intelligence environment. Any SQL query to Oracle Database can contain a call to an R script that is registered in the database R script repository. Using the script name, users can initiate a query to call that script and receive the results in a new table, images, or as XML. For example, parameters controlling R scripts can be passed as run-time arguments to programmatically update BI dashboards and graphical reporting applications.

One telecommunications provider used OML4R to power complex survey research. Analysts at this firm maintain analytic functions in Oracle Database and then filter data and display results through a parameterized BI dashboard. Both the database and the BI infrastructure are standard components of the architecture, further enhanced by their connection to R scripts. These capabilities make R a more powerful language that can execute advanced statistical models directly on database data.

Run R with Hadoop and Spark

Oracle Machine Learning for Spark (OML4Spark), supported by Oracle R Advanced Analytics for Hadoop, is a component with an R package front-end that provides best-in-class Spark-based machine learning algorithms for data in Hadoop clusters, as well as transparent access to Hadoop and data stored in HDFS, Apache Hive, Apache Impala, and Spark DataFrames. OML4Spark enables users to run R models efficiently against large volumes of data, as well as to leverage Spark in-memory processing without leaving the R environment. They can use R to analyze data stored in HDFS with Oracle-supplied machine learning algorithms, as well as using CRAN R packages.

When it comes to Machine Learning, OML4Spark provides several parallel distributed algorithms whose execution benefits from a Hadoop Cluster with Spark. OML4Spark custom algorithms – including linear model, generalized linear model, and multi-layer perceptron neural network – scale better on Spark because they do not require that all data fit in memory. They also run faster than similar open-source Spark MLlib functions. OML4Spark provides enhanced interfaces to MLlib that take advantage of the full R-formula specification and surpass those provided by Spark MLlib.

The algorithms that run in Spark support both model build and apply (prediction scoring) with input datasets in the form of HDFS, Apache Hive, Apache Impala, Spark DataFrames, and JDBC data sources. The models themselves can be stored in binary format on HDFS and the local file system for execution on a different cluster.

OML4Spark enables R commands to run on data accessible from Apache Hive and Apache Impala tables by leveraging the transparency layer supported by OML4R. This transparency layer allows R developers to use the familiar R environment and commands, while under the covers functions are automatically converted to HQL (Hive Query Language) and are executed on the Hadoop Cluster in parallel.

R programs that take advantage of MapReduce can be deployed on a Hadoop cluster and benefit from the data-parallel nature of a Hadoop cluster for performance. Users do not need to know about Hadoop internals, MapReduce, command line interfaces, or the IT infrastructure to create, run, and store these R scripts.

Big Data IoT Use Case with Oracle Database

The Internet of Things (IoT) presents new opportunities for applying advanced analytics. Sensors are everywhere collecting data – on airplanes, trains, and cars, in semiconductor production machinery and the Large Hadron Collider, and even in our homes. One such sensor is the home energy smart meter, which can report household energy consumption every 15 minutes. This data enables energy companies to not only model each customer's energy consumption patterns, but also to forecast individual customer usage. Across all customers, energy companies can compute aggregate demand, which enables more efficient deployment of personnel, redirection or purchase of energy, and so on, often a few days or weeks out.

Building one predictive model per customer, when an energy company may have millions of customers, poses some interesting challenges. Consider an energy company with 1 million customers. Over the course of a single year, these smart meters will collect over 35 billion readings. Each customer, however, generates only about 35,000 readings. On most hardware, R can easily build a model on 35,000 readings. Note that if each model requires even only 10 seconds to build a forecast model, doing this serially will require roughly 116 days to build all models. Since the results are needed a few days or weeks out, a delay of months makes this project a non-starter. If powerful hardware, such as Oracle Exadata, can be leveraged to compute these models in parallel, say with degree of parallelism of 128, all models can be computed in less than one day.

While users can leverage parallelism enabled by various R packages, several factors need to be taken into account. For example, what happens if certain models fail? Will the models be stored as 1 million separate flat files – one per customer? For flat files, how will backup, recovery, and security be handled? How can these models be used for forecasting customer usage and where will the forecasts be stored? How can these R models be incorporated into a production environment where applications and dashboards normally work with SQL?

Using the *embedded R execution* capability of OML4R, Data Scientists can focus on the task of building a model for a single customer. This model is stored in the R Script Repository in Oracle Database. OML4R enables invoking this script from a single function, such as ore.groupApply, relying on Oracle Database to spawn multiple R engines, load one partition of data from the specified database table to the function produced by the Data Scientist, and then store the resulting model immediately in the R Datastore, again in Oracle Database. This greatly simplifies the process of computing and storing models. Moreover, standard database backup and recovery mechanisms already in place can be used to avoid having to devise separate specialized practices. Forecasting using these models is handled in an analogous way.

To put these R scripts into production, users can invoke the same R scripts produced by the Data Scientist from SQL, both for model building and forecasting. The forecasts can be immediately available as a database table that can be read by applications and dashboards, or used in other SQL queries. In addition, these SQL statements that invoke the R functions can be scheduled for periodic execution using the DBMS_SCHEDULER package of Oracle Database.

Leveraging the built-in functionality of Oracle Machine Learning, Data Scientists, application developers, and administrators do not have to reinvent complex code and testing strategies, which must often be done for each new project. Instead, they benefit from Oracle's integration of R with Oracle Database to easily design and implement R-based solutions for use with applications and dashboards, and scale to the enterprise.

Big Data Use Cases with Oracle Database and Hadoop

Oracle's big data technologies are designed to easily move data between Hadoop environments, R, and Oracle Database. Analysts can access data stored in Oracle or Hadoop and can code MapReduce processes, Apache Hive or Spark queries, and run machine learning algorithms in R without having to resort to Java. As described below, this flexible architecture enables organizations to analyze large tables and large data sets easily. In addition to SQL, R is now a good option for enterprise analytics to solve the pressing big data challenges of today.

Use Case 1: Analyzing Credit Risk

Banks continually offer new services to their customers, but the terms of these offers vary based on each customer's credit status. Do they pay the minimum amount due on credit balances, or more? Are their payments ever late? How much of their credit lines do they use and how many other credit lines do they have? What is the overall debt-to-income ratio?

All of these variables influence policies about how much credit to award to each customer, and what type of terms to offer them. A bureau like Equifax or Transunion examines an individual's overall credit history, but banks can examine a much more detailed set of records about their customers—down to the level of every discrete transaction. They need big data analytics to get down to this level of precision with this volume of data.

For example, one Oracle customer in Brazil is running multiple neural network algorithms against hundreds of millions of records to examine thousands of attributes about each of its customers. Previously the bank had trouble crunching this massive volume of data to generate meaningful statistics. They solved this problem by running a specialized algorithm using OML4Spark to analyze the data in parallel on the same cluster that is running the Hadoop file system, Apache Hive, and other tools. OML4Spark enables analysts to execute R analyses, statistics and models on tables stored in the bank's large Hadoop file systems. They can now run complex statistical algorithms against these files systems and Apache Hive tables.

The algorithms use standard R approaches, such as the R formula object. Behind the scenes, OML4Spark provides the interface for executing Spark-based implementations or MapReduce jobs in parallel on multiple processors throughout the bank's cluster. Analysts can create these MapReduce processes and Spark algorithms in R and store them in Hadoop as well as easily surface these models for review, plotting and analysis—and then push the results to Oracle Database—without having to utilize Java.

Use Case 2: Detecting Fraud

Another popular use case involves detecting fraud by analyzing financial transactions. Banks, retailers, credit card companies, telecommunications firms and many other large organizations wrestle with this issue. When scoring data to detect possible fraud, you typically study transactions as they occur within customer accounts (scoring refers to predicting outcomes using a machine learning model.)

Once you understand normal customer behavior, you can then recognize unusual patterns and suspicious transactions. For example, if you normally shop in Los Angeles and there is a sudden series of transactions in Rome this would indicate a high likelihood of fraud. Or would it? If you are somebody who travels a lot, is a surge of activity in Rome an anomaly or a regular pattern? By capturing all previous transactions and studying these patterns, you can develop a model that reflects normal behavior.

While R has algorithms and the environment for creating a predictive model that can analyze these transactions, the algorithms as found in CRAN packages are typically not multi-threaded. Hence, the

algorithm is limited by the memory and single CPU processing power of the machine on which it runs. R typically does not leverage the CPU capacity of a multi-processor laptop without special packages and programming.

Oracle Machine Learning can handle the massive computational requirements associated with analyzing customer-purchasing patterns using the R language to define scripts that are stored in the R Script Repository and run in the database. Organizations can leverage Oracle Exadata and Oracle Big Data Appliance engineered systems to scale the effort, and integrate with Oracle Business Intelligence Enterprise Edition and Oracle Data Visualization Desktop to display the results. R developers can use the results of a fraud model built in Hadoop using open source R and deploy that model in Oracle Database where it can rapidly predict behavior at the transactional level and be part of an enterprise application for real-time predictions.

Real-time analysis is important in fraud prevention. It is one thing to identify a fraudulent transaction that happened eight hours ago (the length of time it might take to stage data into a laptop and run a detailed analysis of yesterday's activities). It is clearly much more valuable to score that transaction against a model in real-time, with the potential to block the transaction or flag it for further scrutiny.

Use Case 3: Preventing Customer Churn

Customer churn is a major problem for many businesses, especially in highly competitive markets such as telecommunications. For example, as a mobile phone user, if you have problems with reception or experience too many dropped calls you might think about looking for another service provider. Your existing service provider is constantly analyzing your behavior to predict how likely you are to defect. They have a statistical model that says, "90% of our customers with similar issues and behavior have left us for another service provider." They can apply that model to your data to create a score that reveals your likelihood of defecting. Your score relates you to millions of other customers with similar behavior.

Using Oracle Machine Learning, you can run these R models while a customer is browsing a webpage or using a mobile app and then make on-the-spot recommendations based on current actions and real time analytics against an operational data store or data warehouse.

Scoring can also be done offline – in batches. For example, you might want to predict which of your 100 million customers will respond to each of a dozen offers so you can identify which customers should be targeted with a special offer or ad campaign. With enough processing power and the right predictive model, analysts can provide insight not only into what the churn rate is but also into the reasons behind the churn. One telecommunications company used OML4Spark to make richer, more informed decisions by examining payment records, calling plans and service histories to detect similarities and trends within its customer databases. This permitted them to run batch jobs in parallel on a large Hadoop cluster.

Oracle provides options for executing these computations from the R environment on data resident in HDFS, Apache Hive, Apache Impala, Spark Data Frames, Oracle Database, JDBC sources, and local files.

Conclusion: R for the Enterprise

Most organizations depend on databases to store information securely with rigorous, enterprise-level controls. Oracle has made R highly compatible with large-scale machine learning, analytics tools, and big data initiatives. Developers can use the familiar R environment in conjunction with Oracle Database, Hadoop, and analytics tools as they apply massive scalability and performance to big data problems. Analysts who are accustomed to working with file extracts can adopt a database-centric architecture,

pushing data from their desktop R implementations to the database and using desktop models to process data that reside in Oracle Database.

R-to-SQL transparency improves user efficiency by allowing analysts to use R directly against data in an Oracle database, Apache Hive or Apache Impala. R users can leverage in-database SQL analytic and data mining functions and open source R packages in combination with Oracle Database for task-parallel execution.

With Oracle Machine Learning, you can remain in the R environment. You can leverage CRAN packages and other R assets that you have created, and invoke R scripts from SQL to deploy R-based analytics into production dashboards and applications. For big data problems, you can leverage the scalability of OML4Spark with different Hadoop clusters or with the Oracle Big Data Appliance and its optimized Hadoop cluster.

Customers that purchase the Oracle Advanced Analytics option, Big Data Connectors, or Oracle Linux receive enterprise class support for Oracle R Distribution.

In summary, by extending R to work with Oracle Database and Hadoop you can bring your analyses to the data, rather than the other way around. By pushing R functionality to Oracle Database via Oracle Machine Learning and invoking Spark jobs from R on Hadoop cluster nodes via OML4Spark, analysts can minimize data movement and decrease latency time from raw data to actionable information. Integrating these three popular environments provides a powerful, cost-effective solution for big data analytics.

ORACLE

Bringing R to the Enterprise
July 2019

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com