Taking R to New Heights for Scalability and Performance

Mark Hornick
Director, Advanced Analytics and Machine Learning
mark.hornick@oracle.com
@MarkHornick
blogs.oracle.com/R

January 31, 2017
Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information 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’s products remains at the sole discretion of Oracle.
Why statisticians | data analysts | data scientists use R

R is a statistics language similar to Base SAS or SPSS statistics

- Powerful
- Extensible
- Graphical
- Extensive statistics
- Ease of installation and use
- Rich ecosystem
  - ~10K open source packages
  - Millions of users worldwide
- Free
  - http://cran.r-project.org/
Traditional R and Data Source Interaction

- Access latency
- Paradigm shift: R → *Data Access Language* → R
- Memory limitation – data size, call-by-value
- Single threaded
- Ad hoc production deployment
- Issues for backup, recovery, security

Deployment
R script
cron job

Read/Write using built-in R capabilities

Data Source

[Diagram showing data flow between R and data source through Flat Files and RODBC / RJDBC / ROracle]
How to take R to new heights for scalability and performance?

i.e., to work on Big Data
big data

noun COMPUTING

extremely large data sets that may be analyzed computationally to reveal patterns, trends, and associations, especially relating to human behavior and interactions.
"much IT investment is going towards managing and maintaining big data"

https://www.google.com/search?q=big+data&ie=utf-8&oe=utf-8

Big data is a term for data sets that are so large or complex that traditional data processing applications are inadequate to deal with them. Challenges include analysis, capture, data curation, search, sharing, storage, transfer, visualization, querying, updating and information privacy. The term "big data" often refers simply to the use of predictive analytics, user behavior analytics, or certain other advanced data analytics methods that extract value from data, and seldom to a particular size of data set. [2] "There is little doubt that the quantities of data now available are indeed large, but that’s not the most relevant characteristic of this new data ecosystem." [3]

https://en.wikipedia.org/wiki/Big_data
Capabilities that take R to new heights...

- Transparent Data Access, Analysis, and Exploration
- Production Deployment
- Scalable Machine Learning
- Data & Task Parallelism
Transparent data access, analysis, and exploration

- Offload processing to more powerful machines and environments using data.frame proxies
- Avoid data movement

- Translate R invocations

- Data Source

- Light local Processing

- Main Processing
  - SQL
  - Spark
    - Java, Python, Scala, R
  - HiveQL

- All Processing
Transparent data access and manipulation

• Maintain language features and interface
• Transparently translate R to language of powerful data processing engines
• Reference data to eliminate data movement
• Analyze all of your data
Proxy objects for Big Data

`data.frame`

Inherits from

Proxy `data.frame`
library(ORE)
ore.connect("rquser", "orcl",
  "localhost", "rquser", all=TRUE)
ore.ls()

df <- with(ONTIME_S,
  ONTIME_S[DEST=="SFO"|DEST=="BOS",1:21])

df$LRGDELAY <-
  ifelse(df$ARRDELAY > 20,1,0)

head(df)
summary(df)

hist(MY_TABLE$ARRDELAY,breaks=100)
merge (TEST_DF1, TEST_DF2,
  by.x="x1", by.y="x2")

# with OREdplyr in ORE 1.5.1...
select(FLIGHTS, year, month, dep_delay)
rename(FLIGHTS, tail_num = tailnum)
filter(FLIGHTS, month == 1, day == 1)
arrange(FLIGHTS, year, month, day)
mutableate(FLIGHTS, speed=air_time/distance)

ore.frame Proxy Object
Scalable Machine Learning

Offload model building to parallel software and powerful machines and environments
Enable new technologies as they arise

Build models using powerful hardware and parallel algorithms...

RDBMS
Spark
Flink
Hadoop

Main Processing
RDAH, MLlib
FlinkML
ORAAH, Mahout

Main Processing
Main Processing
Main Processing
Scalable Machine Learning

• Maintain R machine learning interface
  – Easy to specify formula – minimal lines of code
  – Include transformations, interaction terms, etc.

\[
\text{Target} \quad \log(\text{ARRDELAY}) \sim \text{DISTANCE} + \text{ORIGIN} + \text{DEST} + \\
\quad \quad \quad \quad \quad \text{as.factor(MONTH)} + \text{as.factor(YEAR)} + \text{as.factor(DAYOFMONTH)} + \\
\quad \quad \quad \quad \quad \text{as.factor(DAYOFWEEK)} + \text{as.factor(FLIGHTNUM)}
\]
Scalable Machine Learning

• Maintain R machine learning interface
  – Easy to specify formula – minimal lines of code
  – Include transformations, interaction terms, etc.

• Bring the algorithm to the data
  – Eliminate or minimize data movement
  – Leverage proxy objects to reference data
Scalable Machine Learning

• Maintain R machine learning interface
  – Easy to specify formula – minimal lines of code
  – Include transformations, interaction terms, etc.

• Bring the algorithm to the data
  – Eliminate or minimize data movement
  – Leverage proxy objects to reference data

• Parallel, distributed algorithm implementations
  – Oracle-proprietary parallel, distributed algorithms
  – Leverage other open source packages and toolkits
e.g., Apache Spark Mllib, Apache FlinkML
Linear Model Performance Comparison

- Predict “Total Revenue” of a customer based on 31 numeric variables as predictors, on 184 million records using SPARC T5-8, 4TB of RAM
- Data in an Oracle Database table

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Threads Used*</th>
<th>Memory required**</th>
<th>Time for Data Loading***</th>
<th>Time for Computation</th>
<th>Total</th>
<th>Relative Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open-Source R Linear Model (lm)</td>
<td>1</td>
<td>220Gb</td>
<td>1h3min</td>
<td>43min</td>
<td>1h46min</td>
<td>1x</td>
</tr>
<tr>
<td>Oracle R Enterprise lm (ore.lm)</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>42.8min</td>
<td>42.8min</td>
<td>2.47X</td>
</tr>
<tr>
<td>Oracle R Enterprise lm (ore.lm)</td>
<td>32</td>
<td>-</td>
<td>-</td>
<td>1min34s</td>
<td>1min34s</td>
<td>67.7X</td>
</tr>
<tr>
<td>Oracle R Enterprise lm (ore.lm)</td>
<td>64</td>
<td>-</td>
<td>-</td>
<td>57.97s</td>
<td>57.97s</td>
<td>110X</td>
</tr>
<tr>
<td>Oracle R Enterprise lm (ore.lm)</td>
<td>128</td>
<td>-</td>
<td>-</td>
<td>41.69s</td>
<td>41.69s</td>
<td>153X</td>
</tr>
</tbody>
</table>

*Open-source R lm() is single threaded
**Data moved into the R Session's memory, since open-source lm() requires all data to be in-memory
***How long it takes to load 40Gb of raw data into the open-source R Session's memory
Not all parallel implementations are the same
Comparing performance with varying Spark memory footprints

Benchmark on single X5-2 Node with 74 threads and 256 GB of Total RAM, Spark 1.6.0 on CDH 5.8.0

Input Data is 15GB "Ontime" airline dataset with 123mi records, predicting 8,926 total coefficients

**LM formula used**

\[
\text{ARRDELAY} \sim \text{DISTANCE} + \text{ORIGIN} + \text{DEST} + \text{as.factor(MONTH)} + \text{as.factor(YEAR)} + \text{as.factor(DAYOFMONTH)} + \text{as.factor(DAYOFWEEK)} + \text{as.factor(FLIGHTNUM)}
\]

**GLM formula used**

\[
\text{CANCELLED} \sim \text{DISTANCE} + \text{ORIGIN} + \text{DEST} + \text{as.factor(MONTH)} + \text{as.factor(YEAR)} + \text{as.factor(DAYOFMONTH)} + \text{as.factor(DAYOFWEEK)} + \text{as.factor(FLIGHTNUM)}
\]
Data and Task Parallel Execution

• Easily specify parallelism and data partitioning
  – Simplified API – all-in-one
  – Build and score with millions of models
• Automated management of parallel R engines
  – Insulation from hardware details
  – Limit resources as appropriate
  – Startup and shutdown automatically
• Automated loading of data into parallel R engines
• Leverage CRAN packages
Data and Task Parallelism

- **Hand-code logic** to spawn R engines and partition and feed data to R engines as they become available.
- **All Processing**
  - Rserve
  - Rmpi
  - snow
- **Parallel UDF Invoke**
  - Execute user-defined R function on back-end servers in data-parallel or task-parallel manner.
  - Auto-partition and feed data while also leveraging CRAN packages.
- **Partition data by value**
- **Partition data by count**
- **Invoke function with index**
- **Spawn & control R engines**
  - Provide function and data
- **R Script Repository**
- **R Object Repository**

---

Copyright © 2017 Oracle and/or its affiliates. All rights reserved.
Example API

- Supply data
- Specify function
- Use CRAN packages
- Store and load R objects
- Pass Arguments
- Specify parallelism
- Get/use results
  - R objects
  - structured data
  - Images
  - etc.

```r
library(e1071)
mod <- ore.tableApply(IRIS_TABLE,
  function(dat, datastore) {
    library(e1071)
    dat$Species <- as.factor(dat$Species)
    mod <- naiveBayes(Species ~ ., dat)
    ore.save(mod, name = datastore)
  },
  datastore = "NB_Model-1")

scoreNBmodel <- function(dat, datastore) {
  library(e1071)
  ore.load(datastore)
  dat$PRED <- predict(mod, newdata = dat)
  dat
}

IRIS_PRED <- IRIS_TABLE[1,]
IRIS_PRED$PRED <- "A"
res <- ore.rowApply(IRIS_TABLE, scoreNBmodel, datastore = "NB_Model-1",
  parallel = 4, FUN.VALUE = IRIS_PRED, rows = 10)

DAT <- ONTIME_S[ONTIME_S$DEST %in% c("BOS", "SFO", "LAX", "ORD", "ATL", "PHX", "DEN"),]
modList <- ore.groupApply(
  X = DAT, INDEX = DAT$DEST, parallel = 3,
  function(dat) lm(ARRDELAY ~ DISTANCE + DEPDELAY, dat))
```
Deployment

- Avoid costly recoding or translating R code
- Invoke R easily from non-R environments
- Map data structures and types naturally
- Seamlessly return data.frames, images, XML, JSON in local environment data structures
Deployment

Invoke R from SQL
results as table: structured, image, XML, JSON
automate parallel and concurrent execution

Allow applications and dashboard tools to use familiar, existing SQL protocols for invoking R

Application
Business Logic
C, C++, Java, etc.

Application / Dashboards
Business Logic
C, C++, Java
SQL

RDBMS
HDFS
NoSQL
Hive
File system

Data
Stores

R

rJava API
RServe – client/server TCP/IP
rpy2
system()
cron – independent
...

data.frame
images
R objects

R Scripts

R

SQL - R Engine
R Scripts
Data

R

R

R

R

Copyright © 2017 Oracle and/or its affiliates. All rights reserved.
Deploy R using SQL

- Store named R function in Script Repository from R or SQL
- Return values
  - Images as PNG BLOB column
  - data.frame content as database table
  - XML with data.frame and image
- Benefits
  - Fewer moving parts
  - IPC data transfer speeds at backend
  - Invoke same function from R or SQL
  - Security
  - Integrated backup and recovery

```sql
begin
  sys.rqScriptDrop('RandomRedDots');
  sys.rqScriptCreate('RandomRedDots',
  'function()
    id <- 1:10
    plot( 1:100, rnorm(100), pch = 21,
         bg = "red", cex = 2, main="Random Red Dots"
    )
    data.frame(id=id, val=id / 100)
  }');
end;
```

```sql
select    ID, IMAGE
from      table(rqEval( NULL,'PNG','RandomRedDots'));
```

```sql
select    id, val
from      table(rqEval( NULL,'select 1 id, 1 val from dual',
                        'RandomRedDots'));
```

```sql
-- Return structured and image content within XML string
select    *
from      table(rqEval(NULL, 'XML', 'RandomRedDots'));
```

```sql
-- In R, invoke same function by name
ore.doEval(FUN.NAME='RandomRedDots')
```
Architectural Elements: Enabling R for Big Data

- Leverage powerful back-ends for the heavy lifting...transparently
- Leverage new, more powerful back-ends more easily as they appear
- Enable parallelism quickly and easily for big data processing
- Immediately leverage data scientist R scripts and results in production environments
Oracle R Enterprise

Part of Oracle Advanced Analytics option to Oracle Database

- Use Oracle Database as HPC environment
- Use in-database parallel and distributed machine learning algorithms
- Manage R scripts and R objects in Oracle Database
- Integrate R results into applications and dashboards via SQL
Oracle R Enterprise

Part of Oracle Advanced Analytics option to Oracle Database

• Transparency layer
  – Leverage proxy objects (ore.frames) - data remains in the database
  – Overload R functions that translate functionality to SQL
  – Use standard R syntax to manipulate database data

• Parallel, distributed algorithms
  – Scalability and performance
  – Exposes in-database algorithms from ODM
  – Additional R-based algorithms executing and database server

• Embedded R execution
  – Manage and invoke R scripts in Oracle Database
  – Data-parallel, task-parallel, and non-parallel execution
  – Use open source CRAN packages
# OAA / Oracle R Enterprise 1.5.1

## Predictive Analytics algorithms in-Database

### Classification
- Decision Tree
- Logistic Regression
- Naïve Bayes
- Support Vector Machine
- Random Forest

### Regression
- Linear Model
- Generalized Linear Model
- Multi-Layer Neural Networks
- Stepwise Linear Regression
- Support Vector Machine

### Clustering
- Hierarchical k-Means
- Orthogonal Partitioning
- Expectation Maximization

### Attribute Importance
- Minimum Description Length
- Expectation Maximization

### Anomaly Detection
- 1 Class Support Vector Machine

### Market Basket Analysis
- Apriori – Association Rules

### Feature Extraction
- Nonnegative Matrix Factorization
- Principal Component Analysis
- Singular Value Decomposition
- Explicit Semantic Analysis

### Time Series
- Single Exponential Smoothing
- Double Exponential Smoothing

---

*New in ORE 1.5.1*

*ODB 12.2 only*

---

...plus open source R packages for algorithms in combination with embedded R data- and task-parallel execution
Oracle R Advanced Analytics for Hadoop
Using Hadoop/Hive/Spark Integration, plus R Engine and Open-Source R Packages

Oracle R Advanced Analytics for Hadoop (ORAAH) on Hadoop Cluster

R interface to HQL Basic Statistics, Data Prep, Joins and View creation

Parallel, distributed algorithms:
- ORAAH Spark algorithms: Deep Neural, GLM, LM
- Spark MLlib algorithms: LM, GLM, LASSO, Ridge Regression, Decision Trees, Random Forests, SVM, k-Means, PCA

Use of Open-source R packages via custom R Mappers / Reducers

R Client
Oracle R Advanced Analytics for Hadoop
R Analytics
SQL Client
SQL Developer
Other SQL Apps
SQL Client
Oracle Database with Advanced Analytics option

Copyright © 2017 Oracle and/or its affiliates. All rights reserved.
## Oracle R Advanced Analytics for Hadoop 2.7.0

### Predictive Analytics algorithms

#### Classification
- GLM ORAAH
- Logistic Regression ORAAH
- Logistic Regression
- Random Forests
- Decision Trees
- Support Vector Machines

#### Regression
- MLP Neural Networks ORAAH
- LASSO
- Ridge Regression
- Support Vector Machines
- Random Forest
- Linear Regression

#### Feature Extraction
- Non-negative Matrix Factorization
- Collaborative Filtering (LMF)
- Singular Value Decomposition

#### Clustering
- Hierarchical k-Means
- Hierarchical k-Means
- Gaussian Mixture Models

#### Basic Statistics
- Correlation/Covariance

---

**Copyright © 2017 Oracle and/or its affiliates. All rights reserved.**
Cloud-Based Machine Learning

• **Oracle Advanced Analytics option including Oracle Data Mining and Oracle R Enterprise on:**
  – Oracle Exadata Cloud Service
  – Oracle Database Cloud Service: Included in High Performance and Extreme Performance services

• **Oracle R Advanced Analytics for Hadoop**
  – Included in the Oracle Big Data Cloud Service
Demonstration of ORE
Join us for the Oracle R Enterprise Hands-on Lab
Wednesday @ 9:00

Using R for Big Data Advanced Analytics and Machine Learning
→ data exploration / attribute importance
→ clustering
→ regression
→ OREdplyr, and more
Join us for Big Data with ORAAH
Wednesday @ 1:00

Using Machine Learning to unlock the Business Value in Big Data
Join us for new technology intro Thursday @ 9:50

Combining Graph and Machine Learning Technologies using R
Join us for new technology intro Thursday @ 10:55

Introducing Oracle Machine Learning
→ new notebook technology from Oracle
Learn More about Oracle’s Advanced Analytics R Technologies...

http://oracle.com/goto/R

R Technologies from Oracle

Bringing the Power of R to the Enterprise