Scalable Machine Learning using Big Data SQL, Hadoop and Spark

Advanced Analytics at Scale

BIWA Summit 2016

Marcos Arancibia
Product Manager, Oracle Data Science
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.
Use Cases

1. Root Cause Analysis of semiconductor manufacturing that uses custom-built R algorithms running against data in the Database and against Hadoop to verify the potential of Big Data SQL.

2. Predicting Airline flight cancellations using Logistic Regression directly against a Big Data Cluster
Data Analytics Challenge

Separate data access interfaces...

{APIs}  {MapReduce}  SQL
Data Analytics Challenge

...which require separate Predictive Analytics interfaces.
Oracle Big Data SQL

Expanding the reach of OAA with Predictive Analytics via SQL

R

ORAAS

OAA

NoSQL

ORACLE
1. Root Cause Analysis - Requirements

Semiconductor Industry

- BISTel is an Oracle Partner in the USA and in Korea, and it is the leading provider of equipment engineering systems and services for the fabrication of semiconductor chips and flat panel displays.

- BISTel offers solutions and services that enable the customers to achieve yield improvement and increase productivity.

- The requirement was to be capable of using their original 8,000+ lines of R code that express some of their custom algorithms on large scale data using both Oracle Database data as well as data stored in Hadoop.
Traditional Manufacturing Yield Mgt Process

Identify yield loss patterns manually or through pre-defined pattern libraries

Extract items/lots with similar patterns

Hours/Days to Find Root Cause

Use multiple 3rd party tools to do different analysis to find cause down to suspected tool/chamber level

Ask equipment engineers to look at their tools for any possible cause

Engineer searches through various systems to see if there is any correlation

Report on possible causes

Report Logs
Tool
FDC
EPT
SPC
RMS
EES
FDC
EPT
SPC
RMS
EES
Report

3rd party BI tools
YMS
JMP
SAS
SPSS
3rd party YMS
In-House
New Analytical Approach for Manufacturing

New Paradigm: Changing the way you analyze

Run in manual or auto mode

Problem with yield

Run pattern detection and classification, root cause analysis together for enhanced accuracy and TTRCI

Engineer view result and make decision/report

ROI from Customer:

<table>
<thead>
<tr>
<th>Customer</th>
<th>Time to Root Cause Identification (TTRCI) for Traditional Method</th>
<th>TTRCI using BISTel MA + IM</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>7 days</td>
<td>Within 4 hours</td>
</tr>
<tr>
<td>B</td>
<td>3 weeks</td>
<td>Within 4 hours</td>
</tr>
<tr>
<td>B</td>
<td>Could not find within 4 weeks</td>
<td>Found within 3 hours</td>
</tr>
</tbody>
</table>
E.g.: Auto Pattern Recognition and Classification

Data
- Coordinate (i.e. x, y)
- Quality Measurement (i.e. thickness, defect)
- Yield data (i.e. good/bad, grade)

Manipulate data and convert to map (i.e. circle, rectangle, etc...)

Root cause analysis of pattern

Dynamic pattern recognition and classification

Extrapolate and interpolate data to full map
E.g.: Root Cause Analysis with Machine Learning

Data
- Effect data
- Cause candidate data
- Continuous/categorical

Select Effect

Select Cause Candidates

Ranked Result

Engineer Decision
Solution Architecture Overview

- **Oracle R Distribution**
- **EXADATA**
- **BIG DATA APPLIANCE**
- **Oracle R Enterprise**
- **BISTel’s algorithms**
- **R Console**
- **R Studio**
- **eDataLyzer**
- **SQL Developer or PL/SQL**
- **JDBC**

- MA
- Intellimine
- TA

---

Copyright © 2016 Oracle and/or its affiliates. All rights reserved.
Interfaces of the usage in Production: Database

Various Database for OLTP

ETL

Auto Request

User Request

.Net Client

JAVA Analysis Server

Notify Analysis Result

EXADATA

ExtProc

Intellimine

...
Features of Oracle Advanced Analytics
From either an R Client or a SQL Client, OAA in-Database algorithms, R Engines and Open-Source R Packages can be accessed.

Oracle Database Server
with Advanced Analytics Option

- SQL Basic Statistics and Joins
- Data Mining Predictive Analytics
  15 PL/SQL In-Database algorithms
- ORE Parallel algorithms: MLP Neural, Stepwise, LM, GLM, PCA
- Access to open-source R packages

R Client
R Analytics
Oracle R Enterprise

SQL Client
SQL Developer
Other SQL Apps
Features of Oracle Advanced Analytics

Oracle R Distribution x64 running on the Intel Platform can make use of Intel’s MKL for additional performance, even with open-source R packages.

<table>
<thead>
<tr>
<th></th>
<th>ORD with internal BLAS/LAPACK 1 thread</th>
<th>ORD + MKL 1 thread</th>
<th>ORD + MKL 2 threads</th>
<th>ORD + MKL 4 threads</th>
<th>ORD + MKL 8 threads</th>
<th>Performance gain ORD + MKL 4 threads</th>
<th>Performance gain ORD + MKL 8 threads</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Matrix Calculations</strong></td>
<td>11.2</td>
<td>1.9</td>
<td>1.3</td>
<td>1.1</td>
<td>0.9</td>
<td>9.2x</td>
<td>11.4x</td>
</tr>
<tr>
<td><strong>Matrix Functions</strong></td>
<td>7.2</td>
<td>1.1</td>
<td>0.6</td>
<td>0.4</td>
<td>0.4</td>
<td>17.0x</td>
<td>17.0x</td>
</tr>
<tr>
<td><strong>Matrix Multiply</strong></td>
<td>517.6</td>
<td>21.2</td>
<td>10.9</td>
<td>5.8</td>
<td>3.1</td>
<td>88.2x</td>
<td>166.0x</td>
</tr>
<tr>
<td><strong>Cholesky Factorization</strong></td>
<td>25</td>
<td>3.9</td>
<td>2.1</td>
<td>1.3</td>
<td>0.8</td>
<td>18.2x</td>
<td>29.4x</td>
</tr>
<tr>
<td><strong>Singular Value Decomposition</strong></td>
<td>103.5</td>
<td>15.1</td>
<td>7.8</td>
<td>4.9</td>
<td>3.4</td>
<td>20.1x</td>
<td>40.9x</td>
</tr>
<tr>
<td><strong>Principal Component Analysis</strong></td>
<td>490.1</td>
<td>42.7</td>
<td>24.9</td>
<td>15.9</td>
<td>11.7</td>
<td>29.8x</td>
<td>40.9x</td>
</tr>
<tr>
<td><strong>Linear Discriminant Analysis</strong></td>
<td>419.8</td>
<td>120.9</td>
<td>110.8</td>
<td>94.1</td>
<td>88.0</td>
<td>3.5x</td>
<td>3.8x</td>
</tr>
</tbody>
</table>

This benchmark was executed on a 3-node cluster, with 24 cores at 3.07GHz per CPU and 47 GB RAM, using Linux 5.5. More details at https://blogs.oracle.com/R/entry/oracle_r_distribution_3_0
Features of Oracle Advanced Analytics

R: Transparency through function overloading. E.g. in-database aggregation function

```
Oracle Distribution of R version 3.2.0 (-- "Full of Ingredients"

> aggdata <- aggregate(ONTIME_S$DEST,
+                  by = list(ONTIME_S$DEST),
+                  FUN = length)

> class(aggdata)
[1] "ore.frame"
attr("package")
[1] "OREbase"
> head(aggdata)

               Group.1 x
1 ABE          237
2 ABI          34
3 ABQ         1357
4 ABY           10
5 ACK           3
6 ACT           33
```

Oracle SQL
```
select DEST, count(*)
from ONTIME_S
group by DEST
```
Features of Oracle Advanced Analytics

R: Scalable Machine Learning Models. E.g. custom parallel distributed lm model

Oracle Distribution of R version 3.2.0 (-- "Full of Ingredients"

```r
options(ore.parallel=4)
> lm_mod <- ore.lm(ARRDELAY ~ DISTANCE + DEPDELAY,
data=ONTIME_S)

> summary(lm_mod)
Call:
ore.lm(formula = ARRDELAY ~ DISTANCE + DEPDELAY, data = ONTIME_S)
Residuals:
   Min     1Q  Median     3Q    Max
-1462.45 -6.97  -1.36     5.07  925.08
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.254e-01  5.197e-02   4.336  1.45e-05 ***
DISTANCE   -1.218e-03  5.803e-05  -20.979 < 2e-16 ***
DEPDELAY    9.625e-01  1.151e-03   836.289 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 14.73 on 215144 degrees of freedom
(4785 observations deleted due to missingness)
Multiple R-squared:  0.7647, Adjusted R-squared:  0.7647
F-statistic: 3.497e+05 on 2 and 215144 DF,  p-value: < 2.2e-16
```
Features of Oracle Advanced Analytics
Server execution of open-source R package: ore.tableApply()

Oracle Distribution of R version 3.2.0 (-- "Full of Ingredients"

```r
mod_biglm <- ore.tableApply(dat = ONTIME_S, # Database table
function(dat) {
  library(biglm) # Load open-source package
  biglm(ARRDELAY ~ DISTANCE + DEPDELAY, dat)
});

library(biglm) # Load open-source package locally to interpret results
summary(mod_biglm) # Summary of the resulting Model
```

Large data regression model: biglm(ARRDELAY ~ DISTANCE + DEPDELAY, dat)
Sample size =  392805

<table>
<thead>
<tr>
<th></th>
<th>Coef</th>
<th>(95% CI)</th>
<th>SE</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>0.0638</td>
<td>-0.7418</td>
<td>0.8693</td>
<td>0.4028</td>
</tr>
<tr>
<td>DISTANCE</td>
<td>-0.0014</td>
<td>-0.0021</td>
<td>-0.0006</td>
<td>0.0004</td>
</tr>
<tr>
<td>DEPDELAY</td>
<td>1.0552</td>
<td>1.0373</td>
<td>1.0731</td>
<td>0.0090</td>
</tr>
</tbody>
</table>

Oracle R Distribution
Open-source R Packages

Oracle Distribution of R version 3.2.0 (-- "Full of Ingredients" --)

---

Copyright © 2016 Oracle and/or its affiliates. All rights reserved.
Features of Oracle Advanced Analytics

Server execution of open-source R package: ore.groupApply() for data parallelism

Oracle Distribution of R version 3.2.0 (-- "Full of Ingredients"
> options(ore.parallel=4)
> modList <- ore.groupApply(dat = ONTIME_S, # Database table
INDEX = ONTIME_S$DEST,# groupby col
function(dat) {
library(biglm) # Load open-source package
biglm(ARRDELAY ~ DISTANCE + DEPDELAY, dat)
});
> library(biglm) # Load open-source package locally to interpret results
> summary(modList) # Checks how many models we have in the model list
Length Class   Mode
 325 ore.list S4
> summary(modList$BOS) # Request the resulting Model for Boston Logan Airport
Large data regression model: biglm(ARRDELAY ~ DISTANCE + DEPDELAY, dat)
Sample size = 3928

<table>
<thead>
<tr>
<th>Coef</th>
<th>(95% CI)</th>
<th>SE</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>0.0638</td>
<td>-0.7418</td>
<td>0.8693</td>
</tr>
<tr>
<td>DISTANCE</td>
<td>-0.0014</td>
<td>-0.0021</td>
<td>-0.0006</td>
</tr>
<tr>
<td>DEPDELAY</td>
<td>1.0552</td>
<td>1.0373</td>
<td>1.0731</td>
</tr>
</tbody>
</table>
Features of Oracle Advanced Analytics

SQL interface rqEval for R scripts – can also generate XML string for graphic output

Oracle PL/SQL

```
begin
  sys.rqScriptCreate('Example6',
    'function(){
      res <- 1:10
      plot(1:100, rnorm(100), pch = 21,
           bg = "red", cex = 2)
      res
    }');
end;
/
```

Oracle SQL

```
select value
from   table(rqEval(NULL,'XML','Example6'));
```

- R script output is often dynamic – not conforming to pre-defined structure
- R apps generate stats, new data, graphics
- Example
  - Plot 100 random numbers
  - Return a vector with values 1 to 10
  - Return the results as XML
Performance on Auto Pattern Classification: EXADATA only

Tested on 2-node EXADATA X3
OAA with **Big Data SQL**: EXADATA + BDA

Using the in-Database algorithms, plus R Engine and Open-Source R Packages if desired
Performance on Micro Level Data Analytics: EXADATA + BDA
With data either as Database tables or as HDFS files on the BDA, the performance was the same, highlighting the throughput of Big Data SQL, and OAA’s agility on running open-source R code against group-by problems.

EXADATA+Big Data SQL+OAA on a full rack EXADATA X5-2, via Infiniband to a 9-node BDA X5-2. Degree of Parallelism set to 288.

- Algorithm Execution Time
- DoP is 288
## Conclusion: ROI using BISTel Analytics with OAA

<table>
<thead>
<tr>
<th>ROI for Solution Provider</th>
<th>ROI for Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Shorten Time to Market</strong></td>
<td><strong>Shorten TTRCI (Time to Root Cause Identification)</strong></td>
</tr>
<tr>
<td>Faster POC (Proof of Concept) at customer site with new ideas/apps – Reduced by more than 30 days</td>
<td>Within 1 day from 2 weeks to 2 month</td>
</tr>
<tr>
<td>Dramatic reduction of development and test time – At least by 50%</td>
<td>Reduce Investment Cost</td>
</tr>
<tr>
<td></td>
<td>Use of proven EXADATA and ORE with minimal data migration from existing Oracle DB</td>
</tr>
</tbody>
</table>
2. Flight Cancellation prediction in USA airports

Airline Industry

• On-time arrival data for non-stop domestic flights by major air carriers can be found at the Bureau of Transportation Statistics website, and is free to download: http://www.transtats.bts.gov/Fields.asp?Table_ID=236

• Several benchmarks have been executed against this dataset, known as ONTIME. The original combined dataset had 123mi records and contained data from October 1987 to April 2008. It is available at many websites, including the American Statistical Association: http://stat-computing.org/dataexpo/2009/

• We augmented the original data with information until September 2014 to get 159mi unique records. For scalability testing, we appended the file onto itself to get 1Bi total records
## Oracle R Advanced Analytics for Hadoop

**Advanced Analytics algorithms in a Hadoop Cluster: Map-Reduce and Spark based**

<table>
<thead>
<tr>
<th>Classification</th>
<th>Clustering</th>
<th>Statistical Functions</th>
<th>Feature Extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generalized Linear Model</td>
<td>Hierarchical k-Means</td>
<td>Correlation</td>
<td>Nonnegative Matrix Fact (NMF)</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td></td>
<td>Covariance</td>
<td>Collaborative Filtering (LMF)</td>
</tr>
<tr>
<td>Regression</td>
<td>Attribute Importance</td>
<td>Cross Tabulation</td>
<td></td>
</tr>
<tr>
<td>Linear Regression</td>
<td>Principal Components Analysis</td>
<td>Summary statistics</td>
<td></td>
</tr>
<tr>
<td>Multi-Layer Neural Networks</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Oracle R Advanced Analytics for Hadoop – vs. Rhadoop (RMR)

Best platform available to run Map-Reduce R jobs vs. Revolution Analytics’ RHadoop

Performance on a 6-node BDA X3-2, 16 cores and 47 GB of Total RAM assigned

Covariance computation on 100 GB HDFS/200 columns input dataset

More info at https://blogs.oracle.com/R/entry/oraah_enabling_high_performance_r
Oracle R Advanced Analytics for Hadoop: Integration

Using ORAAH’s Hadoop and HIVE Integration, plus R Engine and Open-Source R Packages

Hadoop Cluster
with Oracle R Advanced Analytics for Hadoop

- HQL Basic Statistics, Data Prep, Joins and View creation
- ORAAH distributed algorithms: MLP Neural Nets*, Logistic Reg*, GLM, LM, PCA, k-Means, NMF, LMF. Open-source R packages via Map-Reduce
  * Spark-Caching enabled

Oracle Database Server
with Advanced Analytics option

R Client
- R Analytics
  - Oracle R Advanced Analytics for Hadoop

SQL Client
- SQL Developer
- Other SQL Apps

Other SQL Apps
Join the two tables by one common variable

```r
> joined <- merge(tab_input, tab_input2, by="value")
```

The new table is a temporary HIVE table not seen in `ls()`

```
[1] "tab_input" "tab_input2"
```

But, it's part of the local R objects

```
> ls()
[1] "joined"
```

```r
> names(joined)
[1] "value" "v1.x" "v2.x" "v3.x" "v4.x" "v5.x" "v6.x" "v7.x" "v8.x" "v9.x"
[11] "v10.x" "v11.x" "v12.x" "v13.x" "v14.x" "v15.x" "v16.x" "v17.x" "v18.x" "v19.x"
[21] "v20.x" "v21.x" "v1.y" "v2.y" "v3.y" "v4.y" "v5.y" "v6.y" "v7.y" "v8.y"
[31] "v9.y" "v10.y" "v11.y" "v12.y" "v13.y" "v14.y" "v15.y" "v16.y" "v17.y" "v18.y"
[41] "v19.y" "v20.y" "v21.y"
```
ORAAH: Transparency functions against HIVE
Invoke ORAAH transparent functions for HIVE: Create new Variables and Summary Statistics

We can create a new variable that is a combination of other columns in the joined table, using just R’s plain syntax.

```r
> joined$NEW_VARIABLE <- (join$v1.x + join$v1.y)/2
```

The new variable can be used in any type of computation.

Checking the content of the new Variable:

```r
> summary(joined$NEW_VARIABLE)
Min. 1st Qu. Median Mean 3rd Qu. Max.
-14890 -12040 -10190 -10230 -8621 -2609
```

```r
> head(joined$NEW_VARIABLE)
```
**ORAAH: Machine Learning models against HDFS data**

Invoke open-source Im model and collect graphical results in Map-Reduce using just R

---

**ORAAH Client Packages**

**Map/Reduce Call**

```r
dfs <- hdfs.put(iris, key='Species')
res <- NULL
dfs.res <- hadoop.run(
  dfs,
  mapper = function(key, vals) {
    keyval(key, vals)
  },
  reducer = function(key, vals) {
    dat <- do.call(rbind.data.frame, vals)
    orch.dlogv(colnames(dat))
    mod = lm(Petal.Length ~ Sepal.Length+Petal.Width, data=dat)
    png(fname)
    par(mfrow=c(2, 2), cex=0.6, mar=c(6, 6, 6, 4), mex=0.8)
    plot(mod, id.n=1, cex.caption=0.8, which=1:4)
    dev.off()
    hdfs.fdir <- "/user/oracle/iris"
    hdfs.fname <- paste(hdfs.fdir,"/", fname, sep="")
    system(paste("hadoop fs -copyFromLocal",fname, hdfs.fdir))
    pred <- predict(mod, dat)
    keyval(NULL, orch.pack(pred, hdfs.fname))
  }
)
```

**YARN: Hadoop Map Reduce Job**

1. Map/Reduce Call
2. R Result Object Stored in HDFS
3. /user/oracle/iris
4. Mapper(s)
5. Reducer(s)

---

Invoke open-source lm model and collect graphical results in Map-Reduce using just R

---

**Oracle Distribution of R version 3.2.0 (-- "Full of Ingredients"

---

**ORAAH:** Machine Learning models against HDFS data

---

**Invoke open-source lm model and collect graphical results in Map-Reduce using just R**

---

**YARN: Hadoop Map Reduce Job**

1. Map/Reduce Call
2. R Result Object Stored in HDFS
3. /user/oracle/iris
4. Mapper(s)
5. Reducer(s)
Oracle Distribution of R version 3.1.1 (-- "Sock it to Me"

```r
ontime <- hdfs.attach("/user/oracle/ontime_s")
lm_mod <- orch.lm(ARRDELAY ~ DISTANCE + DEPDELAY,
                  dfs.dat=ontime, nMappers = 4,
                  nReducers = 2)
summary(lm_mod)
```

Call:
```r
ore.lm(formula = ARRDLY ~ DISTANCE + DEPDELAY, data = ONTIME\_S)
```

Residuals:
```
            Min 1Q Median 3Q    Max
-1462.45 -6.97  -1.36  5.07  925.08
```

Coefficients:
```
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)     2.254e-01  5.197e-02  4.336 1.45e-05 ***
DISTANCE     -1.218e-03  5.803e-05 -20.979 < 2e-16 ***
DEPDELAY      9.625e-01  1.151e-03  836.289 < 2e-16 ***
```

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.73 on 215144 degrees of freedom
(4785 observations deleted due to missingness)
Multiple R-squared: 0.7647, Adjusted R-squared: 0.7647
F-statistic: 3.497e+05 on 2 and 215144 DF, p-value: < 2.2e-16
Invoke ORAAH custom parallel distributed GLM Model using Spark Caching

ORAAH: Machine Learning in Spark against HDFS data

# Connects to Spark and reserves a dedicated Context
spark.connect("yarn-client",memory="24g")

# Creates a pointer to the HDFS file for use within R
ont1bi <- hdfs.attach("/user/oracle/ontime_1bi")

# Checks the size of dataset (rows and columns)
formatC(hdfs.dim(ont1bi),digits=4,format="fg",big.mark = "","1,000,000,000" " 30"

# Formula definition: Cancelled flights (0 or 1) based on other attributes
form_orah_glm2 <- CANCELLED ~ DISTANCE + ORIGIN + DEST + F(YEAR) + F(MONTH) + + F(DAYOFMONTH) + F(DAYOFWEEK)

system.time(m_spark_glm <- orch.glm2(formula=form_orah_glm2, ont1bi))

ORCH GLM: processed 6 factor variables, 25.806 sec
ORCH GLM: created model matrix, 100128 partitions, 32.871 sec
ORCH GLM: iter 1, deviance 1.38433414089348300E+09, elapsed time 9.582 sec
ORCH GLM: iter 2, deviance 3.39315388583931150E+08, elapsed time 9.213 sec
ORCH GLM: iter 3, deviance 2.06855738812683250E+08, elapsed time 9.218 sec
ORCH GLM: iter 4, deviance 1.75868100359263200E+08, elapsed time 9.104 sec
ORCH GLM: iter 5, deviance 1.700231875961580E+08, elapsed time 9.132 sec
ORCH GLM: iter 6, deviance 1.69476890425481350E+08, elapsed time 9.124 sec
ORCH GLM: iter 7, deviance 1.69467586045954760E+08, elapsed time 9.077 sec
ORCH GLM: iter 8, deviance 1.69467574351380850E+08, elapsed time 9.164 sec

user system elapsed
84.107 5.606 143.591

YARN:
1. Spark Context Creation
2. Spark Job Execution
3. Spark Job Execution

1. Reserve Memory in a dedicated Context
2. Loads data from HDFS and runs ORAAH's in-Memory Custom Machine Learning algorithm

/user/oracle/ontime_1bi
ORAAH’s Spark-based GLM against HDFS data
Performance against ORAAH’s Map-Reduce GLM

GLM - Logistic Regression Model
ONTIME Dataset - 845 weights

Performance on a 6-node BDA X3-2 with CDH 5.3.0, 24 cores and 96 GB of RAM per Node, Spark 1.2.0 configuration with 24 cores and 24 GB of RAM per Node
Oracle R Advanced Analytics for Hadoop – **Spark vs. Map-Reduce**

Neural Networks Performance – Linear Model, 845 weights, linear model

**Neural Network - Linear Model**

ONTIME Dataset, 7 Predictors - 845 weights

Performance on a 6-node BDA X3-2 with CDH 5.3.0, 24 cores and 96 GB of RAM per Node, Spark 1.2.0 configuration with 24 cores and 24 GB of RAM per Node

More details at https://blogs.oracle.com/R/entry/oracle_r_advanced_analytics_for
ORAAH on Spark: 1 Bi records and Large number of coefficients
Scalable GLM-Logistic Regression and Complex Nonlinear Deep Neural Networks

Performance on a 6-node BDA X3-2 with CDH 5.3.0, 24 cores and 96 GB of RAM per Node
Spark 1.2.0 configuration with 24 cores and 24 GB of RAM per Node

### GLM-Logistic Regression and Non-Linear MLP Neural Networks

<table>
<thead>
<tr>
<th>Model Description</th>
<th>Seconds (Log Scale)</th>
<th>Total # of Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLM - Logistic</td>
<td>2m24s</td>
<td>845 coef.</td>
</tr>
<tr>
<td>Neural - Linear</td>
<td>3m04s</td>
<td>845 coef.</td>
</tr>
<tr>
<td>Neural 3 Layers (50,25,10)</td>
<td>35m27s</td>
<td>43,846 coef.</td>
</tr>
<tr>
<td>Neural 4 Layers (100,50,25,10)</td>
<td>1h50m07s</td>
<td>91,196 coef.</td>
</tr>
<tr>
<td>Neural 5 Layers (200,100,50,25,10)</td>
<td>8h30m13s</td>
<td>195,896 coef.</td>
</tr>
</tbody>
</table>

More details at https://blogs.oracle.com/R/entry/oracle_r_advanced_analytics_for
ORAAH’s Spark-based GLM vs. Spark MLLib GLM
Performance measured on same Hardware and same HDFS input Dataset

**ORAAH Spark GLM performance vs. Spark MLLib GLM**
ONTIME Dataset with 123mi records. Modeling Probability of Flight Cancellation
Initial formula: Cancelled ~ Year + Month + DayOfMonth + DayOfWeek

<table>
<thead>
<tr>
<th>Formula</th>
<th>ORAAH (s)</th>
<th>Spark MLLib (s)</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Formula (=IF)</td>
<td>21</td>
<td>304</td>
<td>14.5x</td>
</tr>
<tr>
<td>IF + FlightNum &amp; Distance (=F2)</td>
<td>63</td>
<td>873</td>
<td>13.9x</td>
</tr>
<tr>
<td>F2 + Origin &amp; Destination</td>
<td>315</td>
<td>1586</td>
<td>5x</td>
</tr>
</tbody>
</table>

Hardware: 24-core Intel(R) Xeon(R) CPU E5-2690 @2.9GHz, Spark 1.5.1 with a 64GB of RAM Context
Oracle R Advanced Analytics for Hadoop

Efficient use of Spark Caching memory, even at minimum levels

Performance on 159mi records on an X4-2 Server, 40 threads, 128GB of RAM, CDH 5.3.0 Spark 1.2.0 configuration with 24 cores and 24 GB of RAM per Node

GLM – Logistic Regression model with 845 Coefficients

Neural Networks - Model using 1 Layer of Neurons, linear activation function, 838 Coefficients

ORAAH’s Spark interface performs well even under small memory settings
Roadmap Architecture

GUI  SQL  R

Algorithms
Common core, parallel, distributed

Hadoop  Spark  Relational  Cloud
Oracle Advanced Analytics—On Premise or Cloud

100% Compatibility Enables Easy Coexistence and Migration

CoExistence and Migration

Same Architecture

Same Analytics

Same Standards

transparently move workloads and analytical methodologies between On-premise and public cloud
Data Science already available on the Oracle Cloud

Cloud-Based Advanced Analytics

Oracle Advanced Analytics option including the Oracle Data Mining and Oracle R Enterprise on:

• Oracle EXADATA Cloud Service

• Oracle Database as a Service: Included in High Performance and Extreme Performance services

Oracle R Advanced Analytics for Hadoop

• Included in the Big Data Cloud Service
Hardware and Software
Engineered to Work Together