

TERADATA ASTER

white paper

Aster Data *n*Cluster In - database Analytics with R

Contents

Introduction to Aster Data <i>nCluster</i> and SQL-MapReduce	3
R in Aster Data <i>nCluster</i>	3
Proprietary Scoring using R <i>without</i> In-database Analytics	4
Proprietary Scoring using R <i>with</i> In-database Analytics	4
Summary.....	6
About Aster Data	6

Introduction to Aster Data *n*Cluster and SQL-MapReduce

Advanced In-Database Analytics is a unique capability of Aster Data *n*Cluster™ that moves beyond the constraints of running analytics in a traditional RDBMS database or on external application servers. By pushing complete analytic applications, not just analytic logic, into the data tier, Aster Data marries the power of standard SQL development with existing programming languages and new frameworks for analytic processing. This includes languages like R, Java, C/C#/C++, .NET, and Python, as well as new big data analytic processing techniques like MapReduce.

All Advanced in-database analytics execute within Aster Data *n*Cluster, a massively parallel processing (MPP) database with an integrated analytics engine that runs on commodity hardware. Uniquely architected to support advanced in-database analytics, *n*Cluster maintains isolation between data management and analytic application processes in the database while at the same time ensuring that both data and application processes are treated as first-class citizens. This means that standard database services like fault-tolerance, workload management, and monitoring apply equally to data management and analytic application processes. In *n*Cluster any existing analytic applications, including pre-packaged applications like R and SAS, run in-database, fully parallelized, and achieve significantly faster response times.

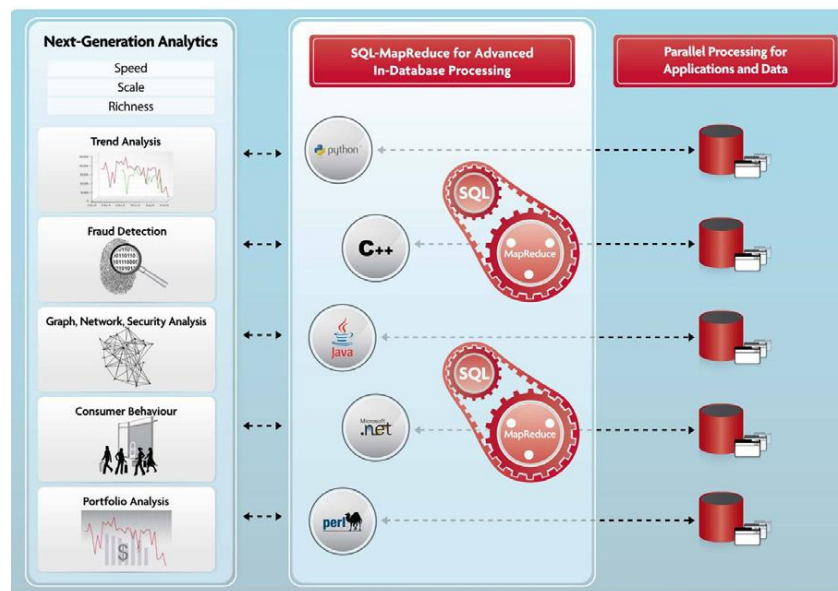


Figure 1: Architectural overview of advanced in-database analytics

R in Aster Data *n*Cluster

The *n*Cluster Stream API allows you to run R scripts and functions to operate upon distributed tables in the database. Once the R code is installed in *n*Cluster, using a Stream function is simple. In your SQL query, you use a SELECT statement in the form,

```
SELECT ... FROM STREAM (ON input_table SCRIPT)
```

The R function then runs on each worker machine in nCluster and can work on pieces of the table in parallel. nCluster manages the execution of the R code across the cluster and the output of the R function can be redirected to an output table or piped into another MapReduce function as desired.

Proprietary Scoring using R *without* In-database Analytics

In financial services, scoring the stocks of interest based on repeated models that are built by analysts is very common. With traditional R (no parallelization), the model building, training and scoring would involve the following high level steps:

Model Building and Training

- Sample data from data store (database or flat files) on to the analyst's machine.
- Use R libraries to build and train the model on the sample data set. The size of the sample data set is generally limited by the available memory (RAM) on the analyst's machine.

Scoring

- Pull out all data to be scored (no sampling) on to the analyst's machine.
- Use the model built in R above to score each row in memory.
- Push the results of the scoring back to the database

This approach is very time consuming and does not scale because:

1. The data can easily go up to 100s of GBs and is typically in the several **10s of TBs** for users wanting to score on a large amount of tick history. Pulling data out of the database and pushing the scoring results back to the database are expensive data transfer operations which increase the cycle time for the analysis.
2. R is **not parallelized** in this mode of operation since it runs on the analyst's machine and has to cycle through all the data in a serial fashion.
3. R execution is restricted to the **RAM of a single machine**.

Proprietary Scoring using R *with* In-database Analytics

In the section, we demonstrate how to use R to run the proprietary scoring function on a tick data set inside nCluster.

This approach provides the following benefits:

1. Eliminates the need to move big data in and out of nCluster.
2. Runs R in parallel across all worker machines of nCluster.
3. Allows the use of RAM across all the worker machines in the nCluster.

Consider a sample table tickdata with the following schema:

Field Name	Data Type
stock_id	VARCHAR

open_price	INTEGER
low_price	INTEGER
high_price	INTEGER
...	...

Figure 2: Sample table tickdata

The tickdata table contains a history of prices and other variables for each stock_id. A score for the stock id takes is calculated using the model which has been built by the analyst.

The R scoring function written by the analyst can be used in-database by modifying it to read input from and write results to the database.

```
#
# ProprietaryScoring.R
#
DELIMITER='\t'
# This is main score function
score_function <- function(input)
{
  # Proprietary Scoring Function
}
stdin = file(description="stdin",open="r")
while (1)
{
  # Read a tuple from stdin into a vector
  input = scan(stdin, sep=DELIMITER, nlines = 1, quiet=TRUE)
  if (length(input) == 0)
    break
  # Compute a score for the input vector
  score = score_function(input)
  # Output original tuple with attached score
  result = c(input, score)
  write(result, stdout(), sep=DELIMITER, ncolumns = length(result))
}
```

In the code snippet above, the scoring function reads rows from the database through the stdin, and after computing the scores writes out to stdout the results.

This function would be executed in parallel on all vworkers (typically 6-8 per worker machine) in the nCluster. These invocations would be done in parallel and the distributed table “tickdata” would be fed in slices to each of these invoked functions.

nCluster provides the SQL-MapReduce interface for invoking these functions in the database. To invoke the script “ProprietaryScoring.R” the analyst would just issue the following SQL-MapReduce command:

```
SELECT * FROM STREAM
  (ON tickdata
   SCRIPT ('ProprietaryScoring.R')
  );
```

The above SQL instructs the database to run 'ProprietaryScoring.R' in parallel on the 'tickdata' table and sends the output of the scoring to the analyst. Using familiar SQL constructs we can combine the analysis done via R to prune out the results and see the top 20 stock_ids and their scores.

```
SELECT stock_id, score FROM STREAM
  (ON tickdata
    SCRIPT ('ProprietaryScoring.R')
  )
ORDER BY score
LIMIT 20;
```

Similarly, extending the flexibility afforded by SQL-MapReduce, the analyst can now:

- Store the results of the analysis in a table for further use.
- Feed the results of scoring to another MapReduce function (written in R or other languages) and perform a streaming analysis through multiple functions.

Summary

In summary, executing R inside Aster Data nCluster provides the following benefits:

1. Eliminates the need to move big data in and out of nCluster
2. Runs R in parallel on all worker machines and each vworker of nCluster
3. Allows the use of RAM across all the worker machines in the nCluster

This approach allows use of SQL-MapReduce to **combine analyses** across multiple MapReduce and SQL functions in a single pass.

About Aster Data

Aster Data is a market leader in data management and advanced analytics for diverse and big data, enabling the powerful combination of cost-effective storage and ultra-fast analysis of relational and non-relational data.

Aster Data nCluster is an analytic platform that incorporates a massively parallel processing (MPP) hybrid row and column database with an integrated analytics engine, allowing application logic to execute with data to deliver breakthrough performance and scalability. Aster Data's solution utilizes Aster Data's patent-pending SQL-MapReduce to parallelize processing of data and applications and deliver rich analytic insights at scale. Companies including Barnes & Noble, Intuit, LinkedIn, Akamai, Full Tilt Poker, and MySpace use Aster Data to deliver applications such as deep clickstream analysis, recommendation and personalization analytics, real-time fraud detection, and churn analysis.

For more information please visit us at <http://www.asterdata.com>, write to info@asterdata.com, or call 1-888-ASTER-DATA.