SQL Server 2016 and Microsoft R Server

Tomaž Kaštrun

March 16, 2017

SQL User Group

Speaker Info

- BI Developer and data analyst
- SQL Server, SAS, R, Python, C#, SAP, SPSS
- 15years experience MSSQL, DEV, BI, DM
- Spar ICS Austria, Spar Slovenija
- Frequent community speaker
- Avid coffee drinker & Bicycle junkie

http://tomaztsql.wordpress.com

tomaz.kastrun@gmail.com

@tomaz_tsql

/in/tomaztsql

http://github.com/tomaztk

https://mvp.microsoft.com/PublicProfile/5002196

Microsoft

Agenda

1 R language and available RevoscaleR package for multi-threaded and parallelization computation

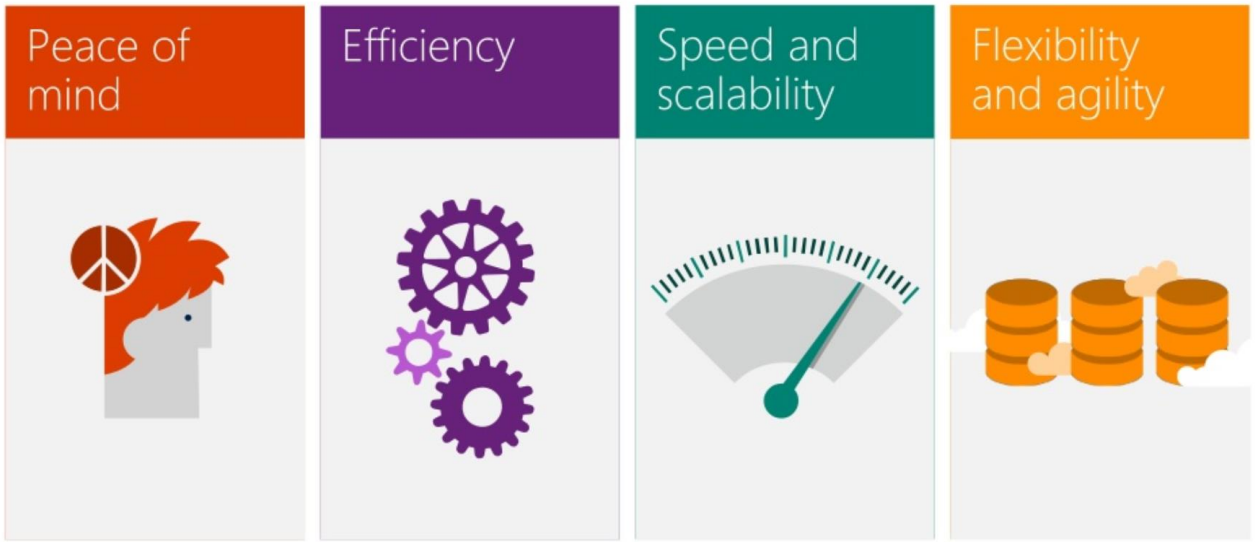
2 Using R language in T-SQL for data analysis and predictions

3 Visualizations (PowerBI)

Analytical Barriers

Common Challenges

Addressing Challanges with R from Microsoft



Microsoft

What is R?

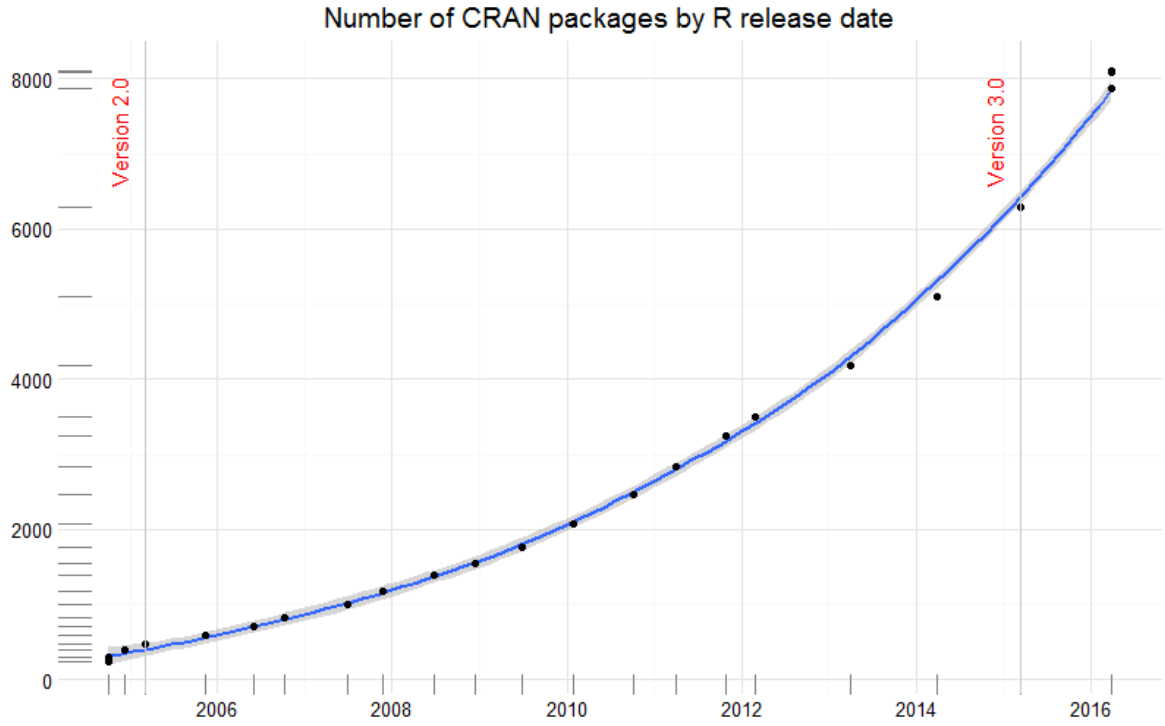
- A Language Platform
  - A Procedural Language optimized for Statistics and data science (and much more)
  - A Data Visualization framework
  - Provided as Free Software

- A Community and a system
  - Taught on universitieis and many active user groups across the world
  - Estimated 3Mio Users
  - Repositories (CRAN, BioConductor, Github,…)

In fact, R is a movement!

Power of R: R Language + Packages

- R is an open source (GNU) version of the S language developed by John Chambers *et al. at Bell Labs in 80's* *History of R*
- R was initially written in early 1990's by *Robert Gentleman and Ross Ihaka then with the* Statistics Department of the University of Auckland
- R is administered and controlled by the R Foundation
- Microsoft is founding member and Platinum Sponsor of R Consortium



3000 packages added in last 2 years

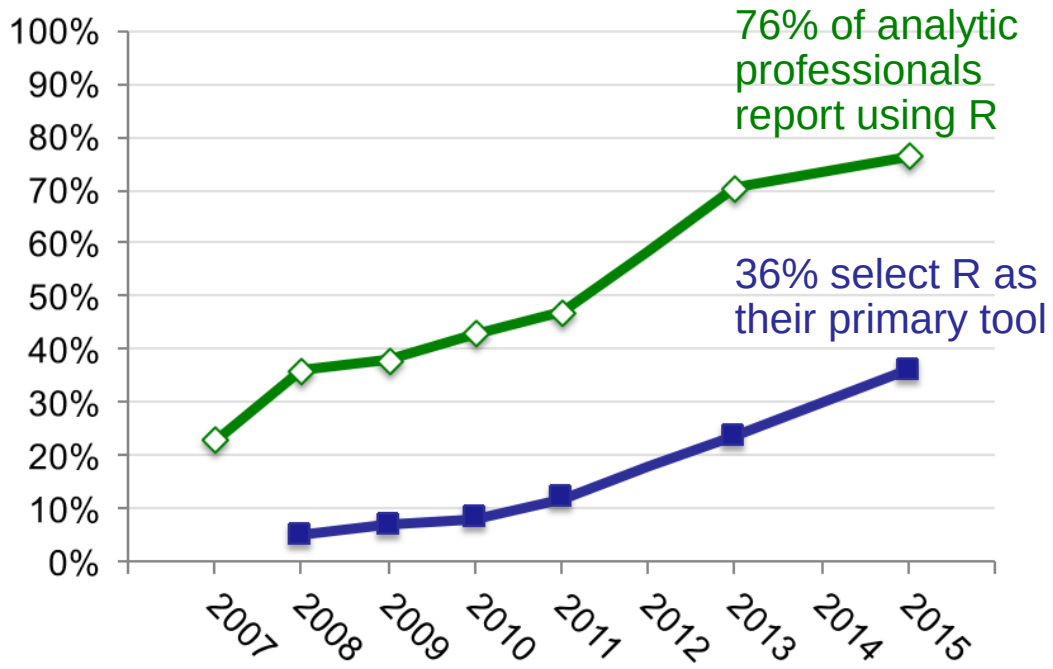
R Reference Card from CRAN

Microsoft

R: The #1 software for Advanced Analytics

R Usage Growth

Rexer Data Miner Survey, 2007-2015



Language Popularity

IEEE Spectrum Top Programming Languages, 2016

Language Rank	Types	Spectrum Ranking
1. C		100.0
2. Java		98.1
3. Python		98.0
4. C++		95.9
5. R		87.9
6. C#		86.7
7. PHP		82.8
8. JavaScript		82.2
9. Ruby		74.5
10. Go		71.9

Microsoft

R is Popular

Most widely used software for Data Mining and Analytics

Used by 2M+ data scientists, statisticians and analysts

Open Source (GPL) language and environment

Easy to bring and explore data, uncover insights and generate predictions

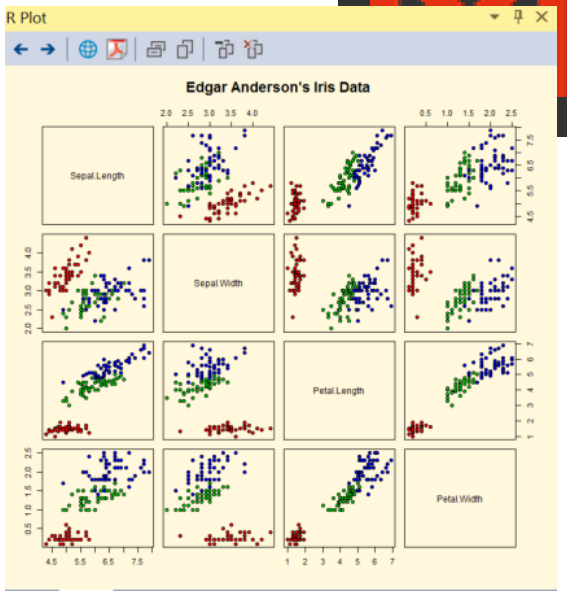
From the most trivial statistical function to the most complex ML technique

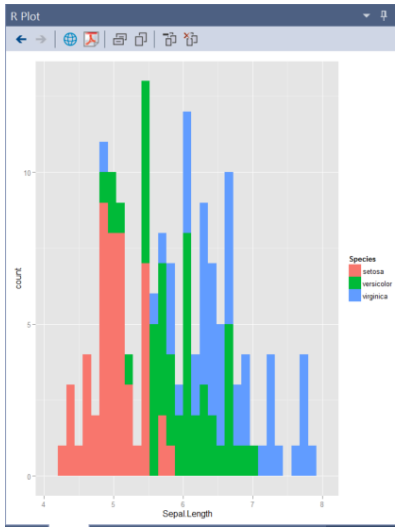
Easily create beautiful and unique data visualizations

As seen in New York Times, The Economist and FlowingData

Thriving open-source community

Over 8000 packages in CRAN and growing; Active forums and groups

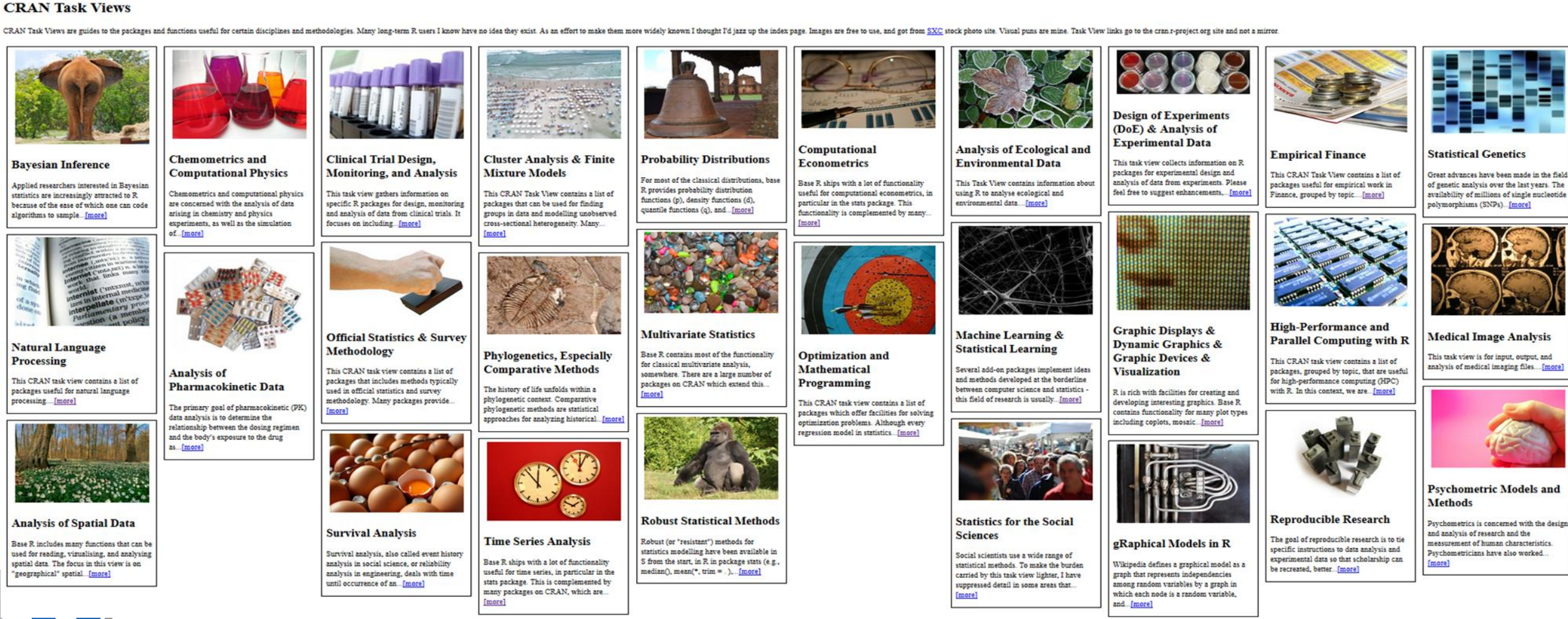




Microsoft

Power of R: R Language + Packages

CRAN: 9000+ Add-on packages for R



CRAN Task View by Barry Rowlingson: http://www.maths.lancs.ac.uk/~rowlings/R/TaskViews/

More packages on Github and BioConductor project

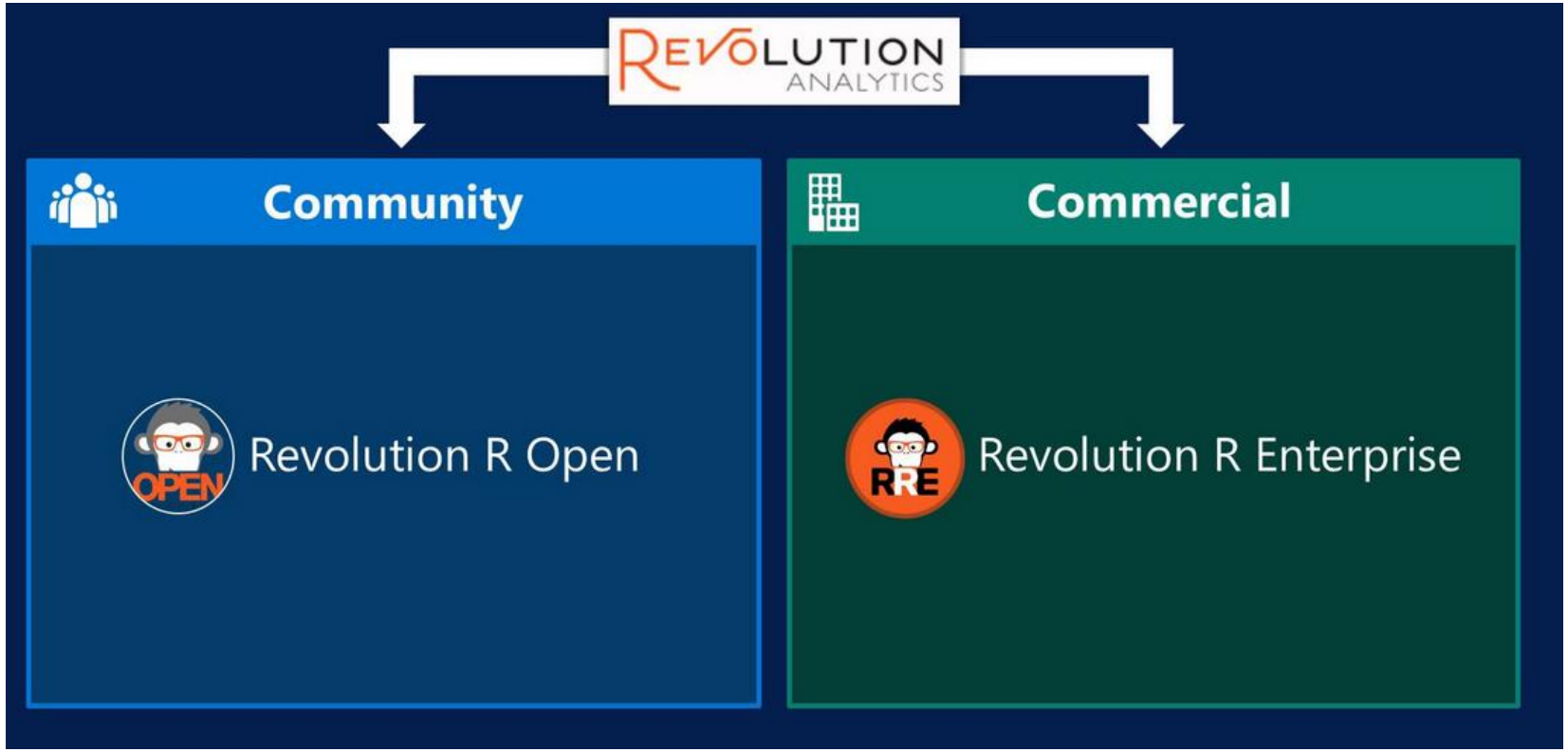
Microsoft

Limitations of R as a free software

- Memory Based Data access model
- Interpreted vs. Compiled Performance
- Lack of parallel computation
- Data movement & Duiplication Costs
- Governance and providence oversight
- Community support vs. Enterprise utilization

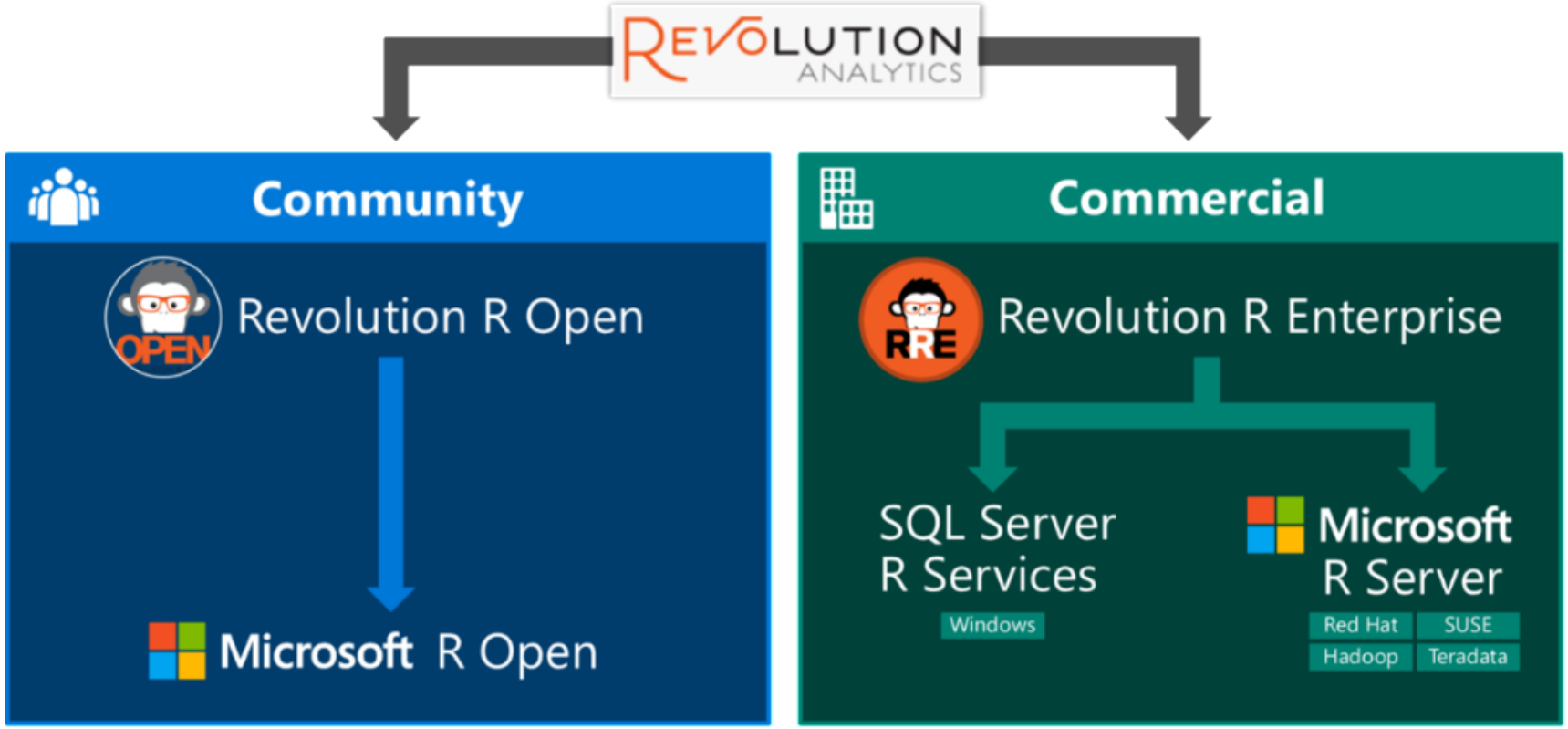
Microsoft

Revolution Analytics Product Integration



Microsoft

Microsoft R SQL Server platform



-> Free and open Source R distribution
-> Enhanced and distributed by Revolution analytics

-> Built in Advanced Analytics and Standalone Server Capability
-> Leverages the benefits of SQL Server 2016EE

Microsoft

Microsoft R Platform

Microsoft R Open

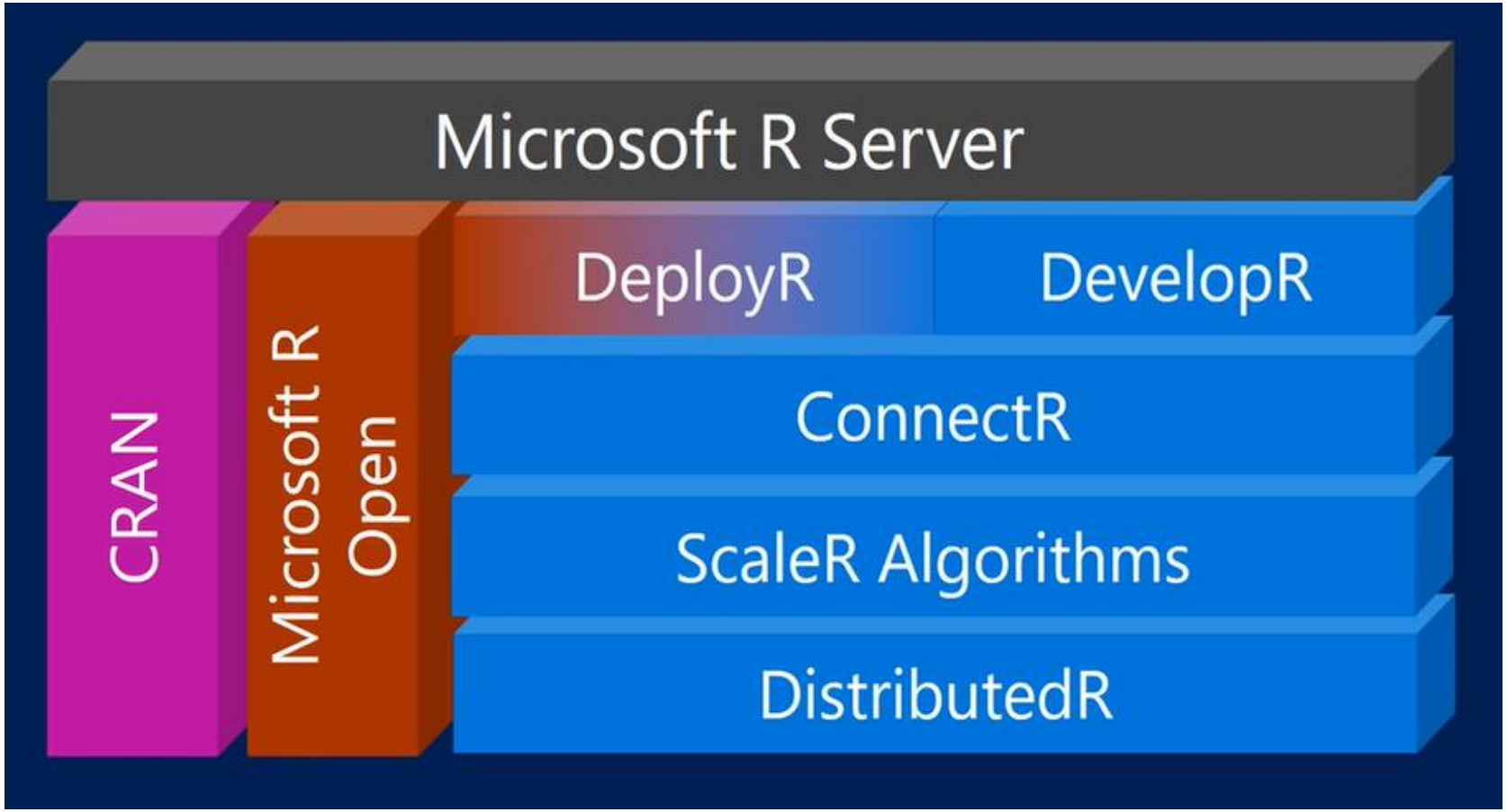
Microsoft R Client

Microsoft SQL R Services

Microsoft R Server

Different flavors:

- Microsoft R server for Linux,
- Microsoft R Server for Teradata,
- Microsoft R Server for Hadoop,
- Microsoft R HDInsight



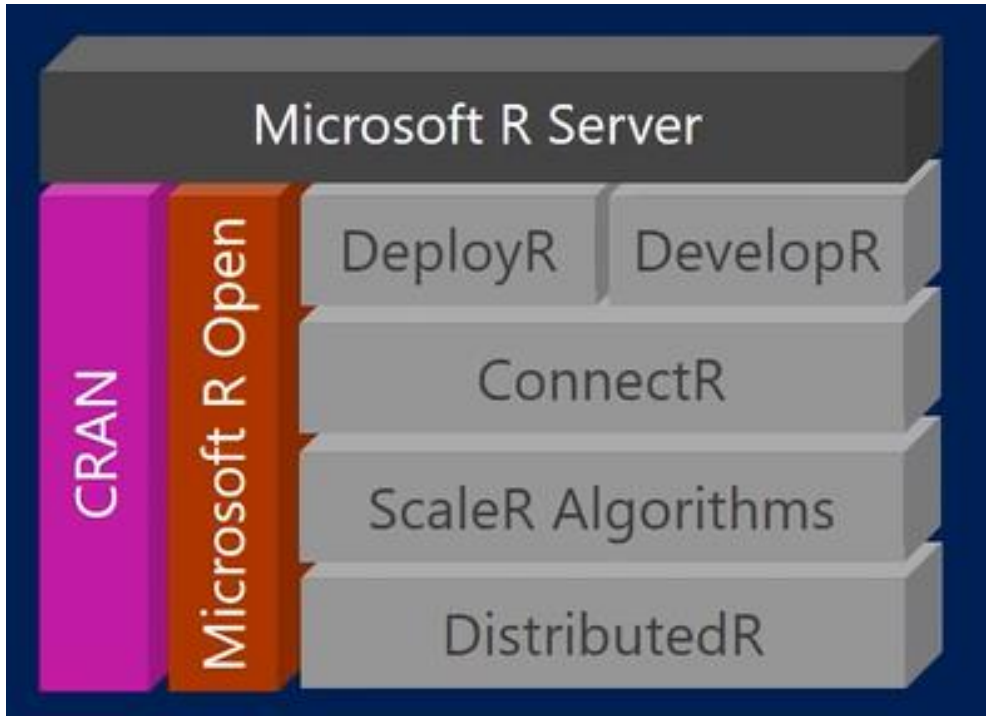
Microsoft

Microsoft R Server

- Evolved from Revolution R Enterprise
- Based on open Source R
- Adapted for Enterprise Scale
- For multiple platforms
  - Hadoop
  - Teradata
  - LinuX
  - Azure
  - Windows
- Interoperable
- On-premises + Cloud + Hybrid
- Operationzalize analytics for Big scale datasets and big data

Based on Open Source R

- Open source based
- Runs your normal R Script
- MetaCran / CRAN / Github / Bioconductor

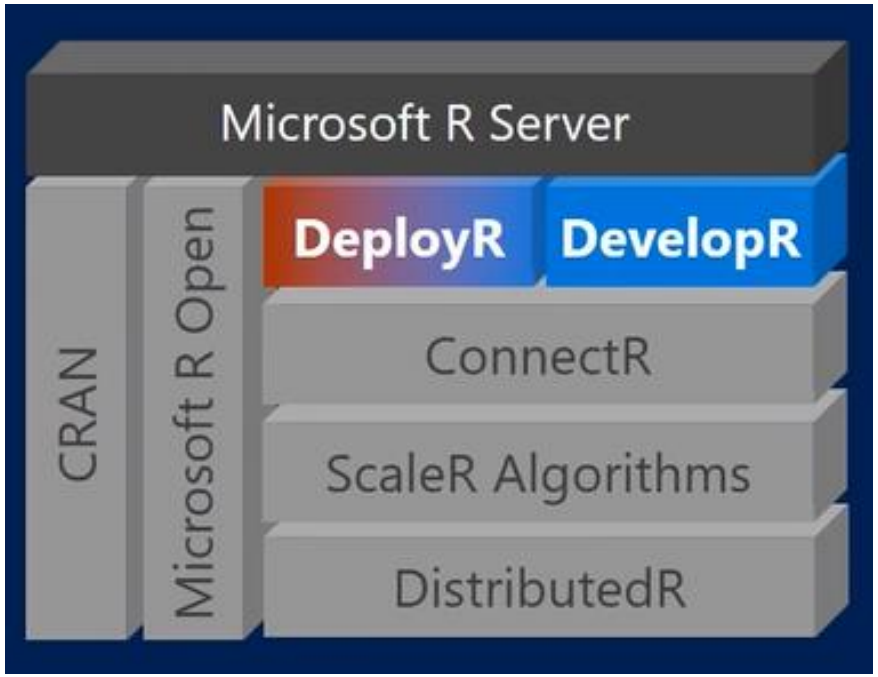


Microsoft

Microsoft R Server

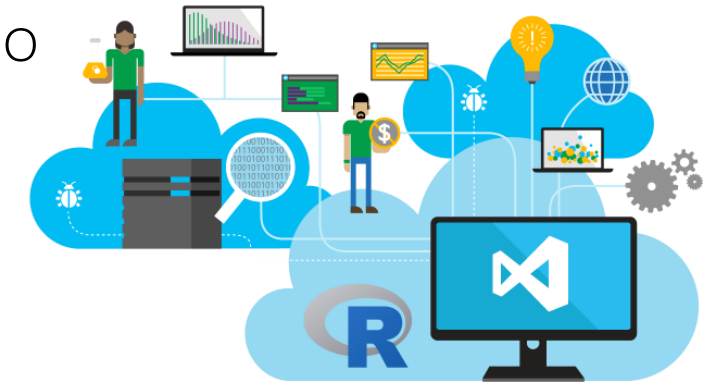
DeployR

- Web service - API integration
- Compatible with array of tools
- Abstract usage of R without knowing it



DevelopR

- R IDE based on Visual studio
- Rstudio for linux Users
- Client Based



Microsoft

Microsoft R Server

ConnectR

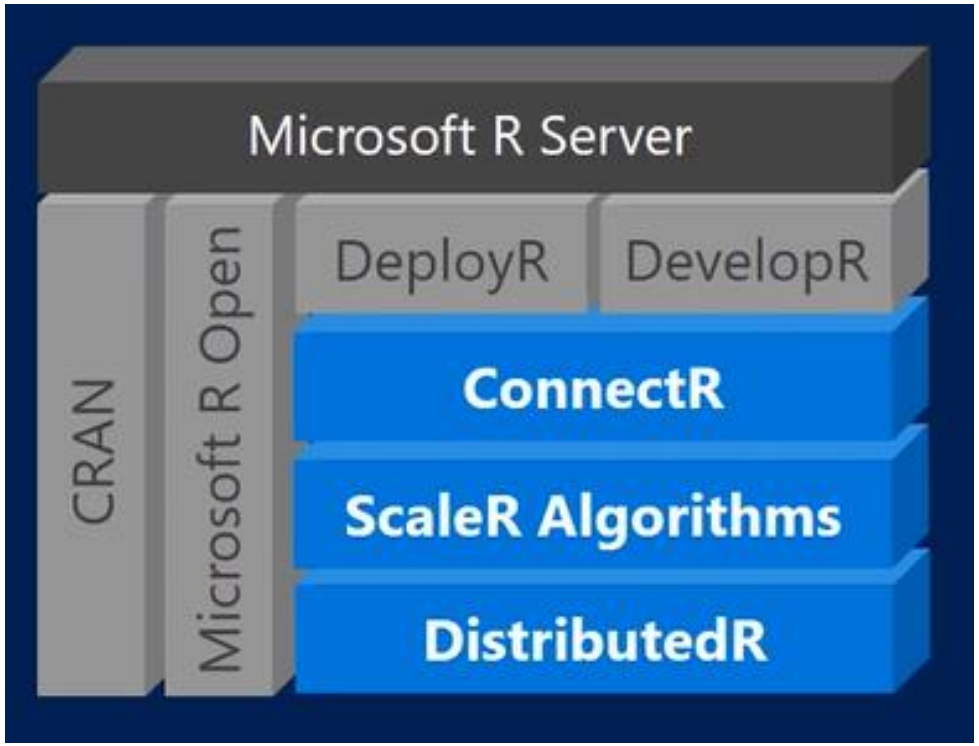
- Serier of connectors for consistent access to scaleR algorithms

DistributedR

- Normalization layer for ScaleR algoritms (SQLServer,Win, Lin, TeraData, Hadoop, HDI)

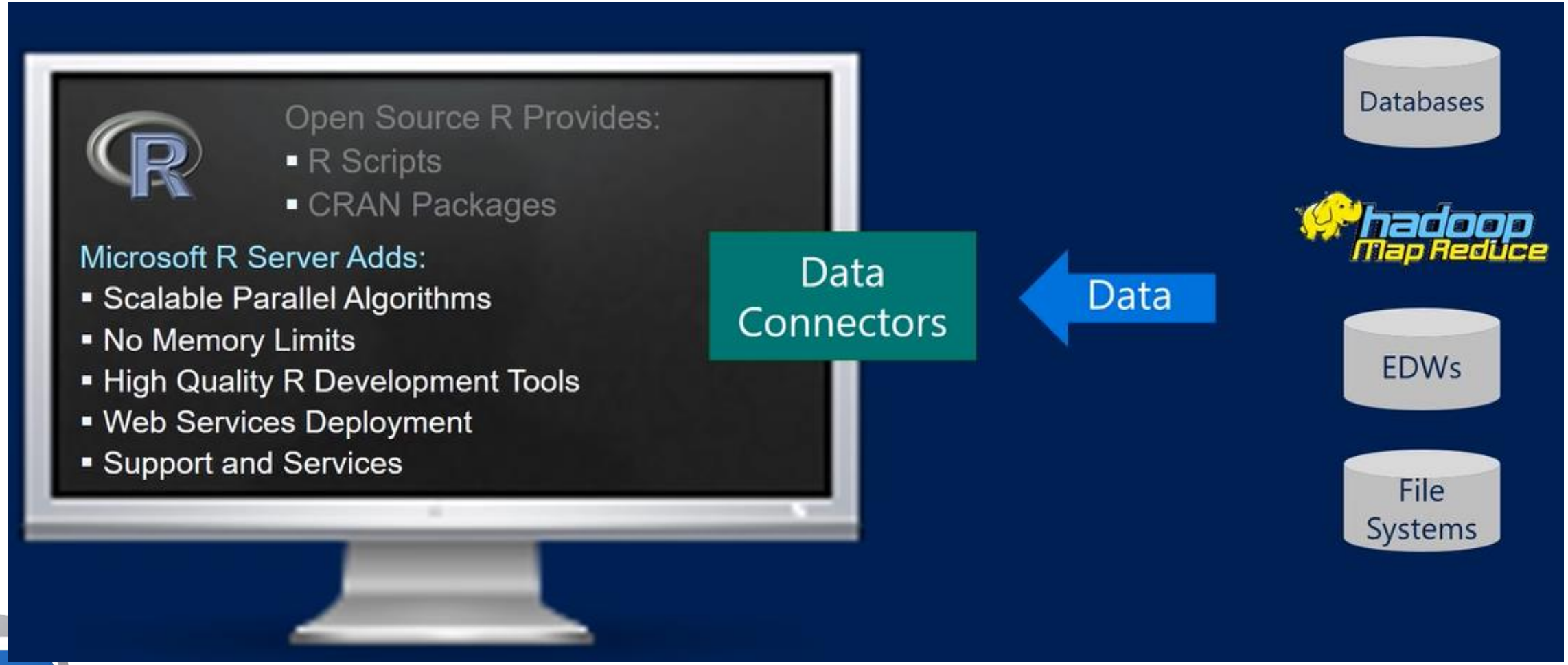
ScaleR

- Typical statistical approaches refactored for parallel computation
- Block-wise computation; No In-Memory constraints



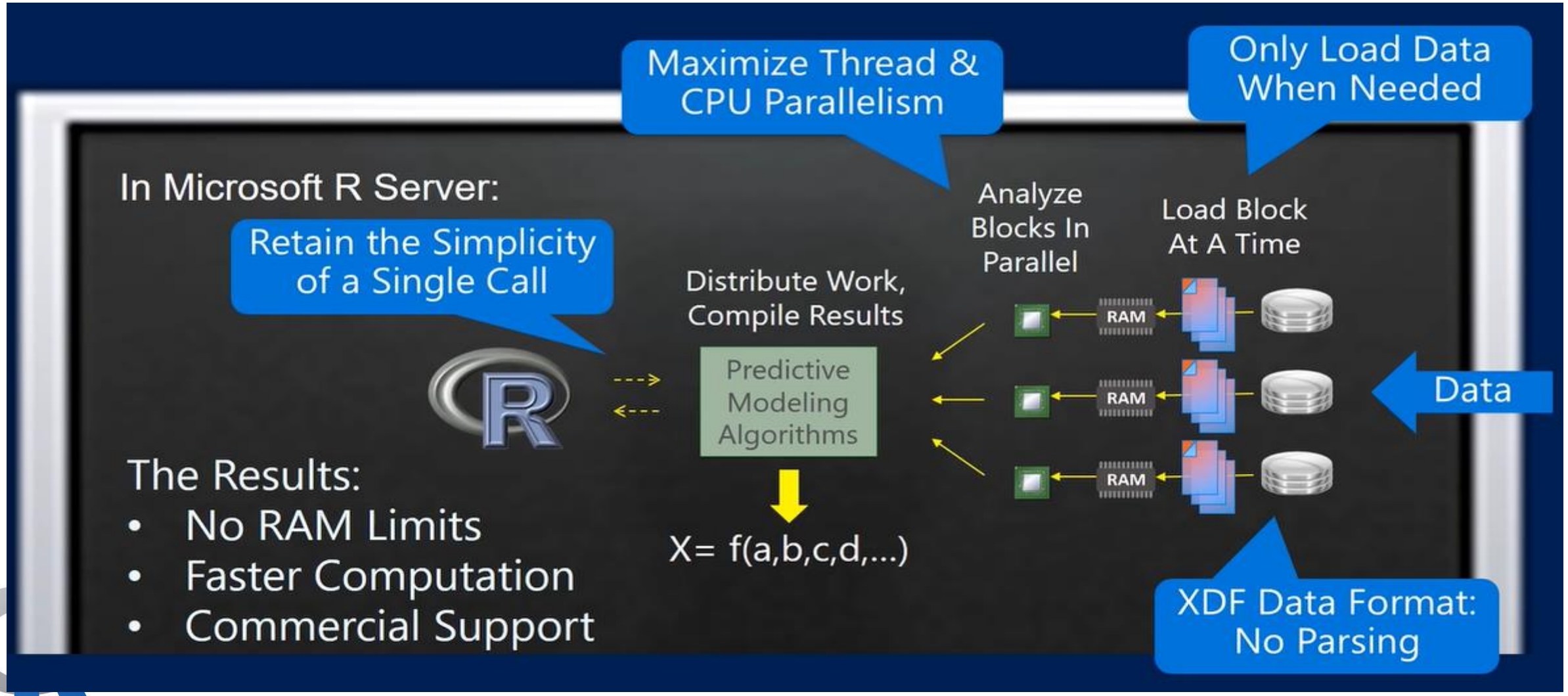
Microsoft

Microsoft R Server - architecture



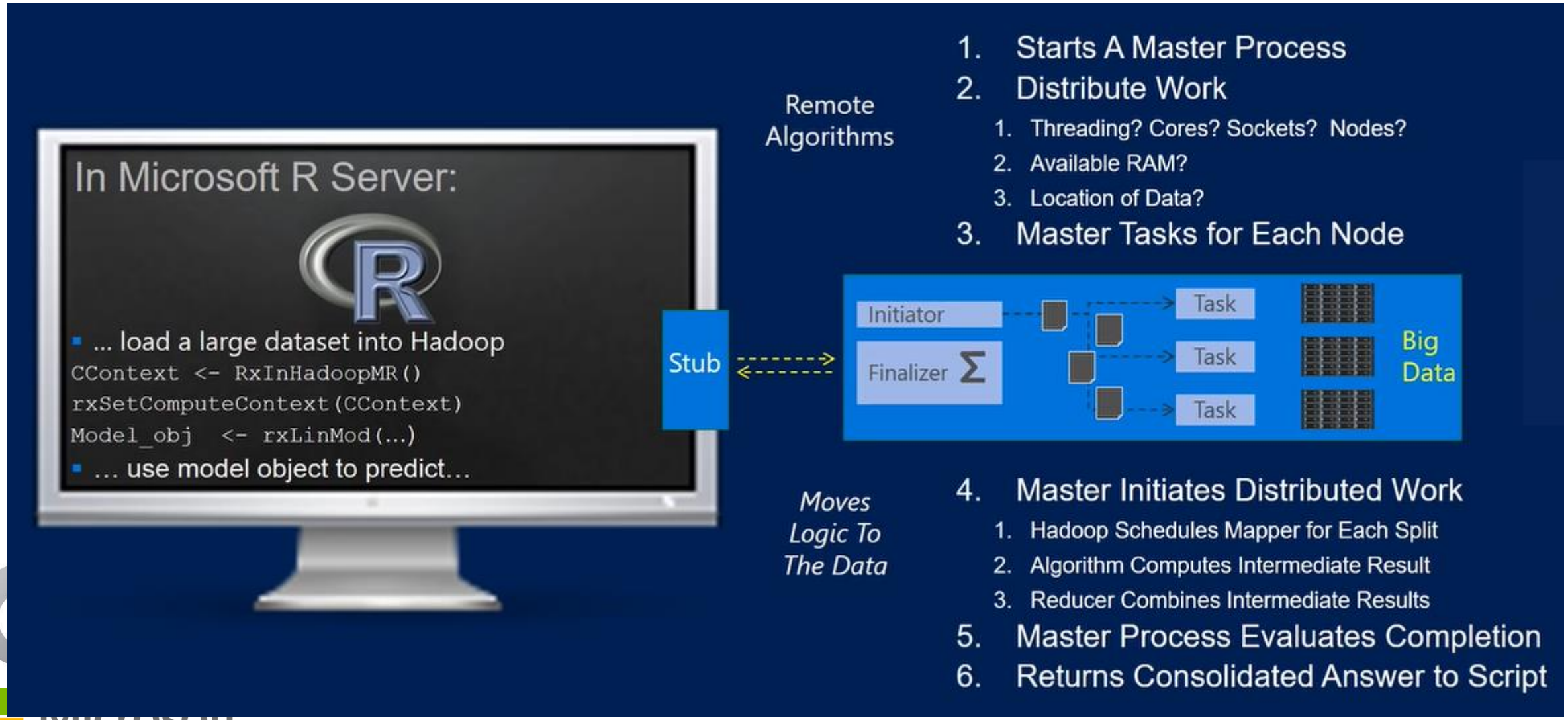
Microsoft

Parallelizing data process



Microsoft

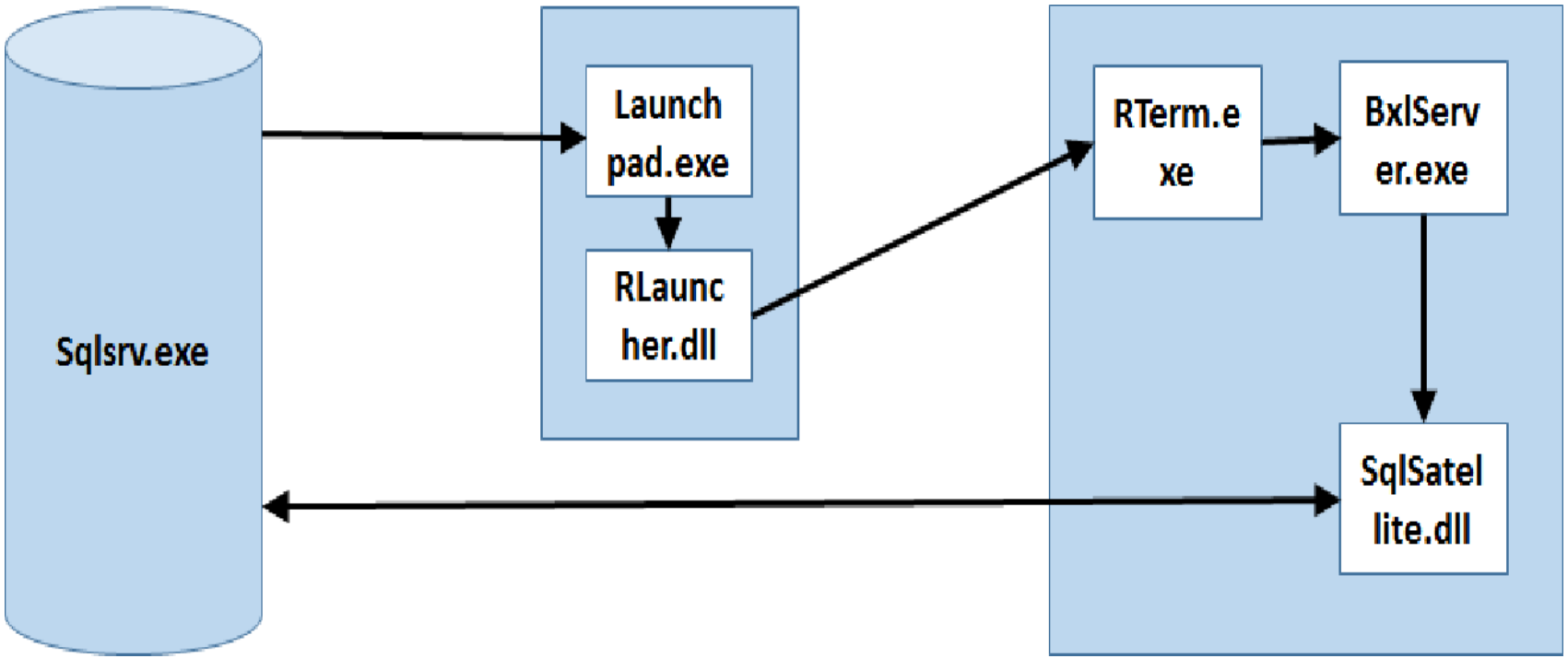
How does parallelization work



For Client R / Server R – Remote execution

Microsoft

Communication between R and SQL Server



ScaleR algorithms

Data Preparation

- Data import – Delimited, Fixed, SAS, SPSS, OBDC
- Variable creation & transformation
- Recode variables
- Factor variables
- Missing value handling
- Sort, Merge, Split
- Aggregate by category (means, sums)

Descriptive Statistics

- Min / Max, Mean, Median (approx.)
- Quantiles (approx.)
- Standard Deviation
- Variance
- Correlation
- Covariance
- Sum of Squares (cross product matrix for set variables)
- Pairwise Cross tabs
- Risk Ratio & Odds Ratio
- Cross-Tabulation of Data (standard tables & long form)
- Marginal Summaries of Cross Tabulations

Statistical Tests

- Chi Square Test
- Kendall Rank Correlation
- Fisher's Exact Test
- Student's t-Test

Sampling

- Subsample (observations & variables)
- Random Sampling

Predictive Models

- Sum of Squares (cross product matrix for set variables)
- Multiple Linear Regression
- Generalized Linear Models (GLM) exponential family distributions: binomial, Gaussian, inverse Gaussian, Poisson, Tweedie. Standard link functions: cauchit, identity, log, logit, probit. User defined distributions & link functions.
- Covariance & Correlation Matrices
- Logistic Regression
- Classification & Regression Trees
- Predictions/scoring for models
- Residuals for all models

Variable Selection

- Stepwise Regression

Simulation

- Simulation (e.g. Monte Carlo)
- Parallel Random Number Generation

Cluster Analysis

- K-Means

Classification

- Decision Trees
- Decision Forests
- Gradient Boosted Decision Trees
- Naïve Bayes

Combination

- rxDataStep
- rxExec
- PEMA-R API Custom Algorithms

Microsoft

R code in SQL Server as T-SQL

```
EXECUTE sp_execute_external_script
   @language = N'R'
  ,@script = N'
              library(e1071);
              irismodel <-naiveBayes(iris_data[,1:4], iris_data[,5]);
              trained_model <- data.frame(payload = as.raw(serialize(irismodel, connection=NULL)));'
  ,@input_data_1 = N'select "Sepal.Length", "Sepal.Width","Petal.Length","Petal.Width","Species" from iris_data'
  ,@input_data_1_name = N'iris_data'
  ,@output_data_1_name = N'trained_model'

WITH RESULT SETS ((model VARBINARY(MAX)));
```

R code in SQL Server using Scale R algorithms

```
EXECUTE sp_execute_external_script
    @language = N'R'
   ,@script = N'require("RevoScaleR");
                irisLinMod <- rxLinMod(Sepal.Length ~ Sepal.Width + Petal.Length + Petal.Width + Species,
                       data = iris_rx_data);
                trained_model <- data.frame(payload = as.raw(serialize(irisLinMod, connection=NULL)));'
   ,@input_data_1 = N'select "Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width", "Species" from
                         iris_rx_data'
   ,@input_data_1_name = N'iris_rx_data'
   ,@output_data_1_name = N'trained_model'

WITH result SETS ((model VARBINARY(MAX)));
```

RevoScaleR Code

```
#####~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#
#
#      LOADING DATA (small sample)
#      178 MB
#      8.4Mio Rows
#
#####~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


ptm <- proc.time()
#inFile <- file.path(rxGetOption("sampleDataDir"), "AirlineDemoSmall.csv")
inFile <- file.path(rxGetOption("sampleDataDir"), "airsample.csv")
rxTextToXdf(inFile = inFile, outFile = "airline.xdf",  stringsAsFactors = T, rowsPerRead = 200000, overwrite=TRUE)
proc.time() - ptm
# ~ 22 seconds!
# - 42 Chunks per 200.000 Rows; Total: 8.400.000 Rows

##################################
#   EXPLORING DATA (small sample)
##################################


rxGetInfo(data="airline.xdf", getVarInfo = TRUE, numRows = 5)

#Histograms by day of week
ptm <- proc.time()
rxHistogram( ~ ArrDelay|DayOfWeek, data = "airline.xdf")
proc.time() - ptm

#summary
rxSummary( ~ ArrDelay, data = "airline.xdf")


rxSort(inData="airline.xdf", outFile = "sortFlights.xdf", sortByVars="ArrDelay",  decreasing = TRUE,overwrite=TRUE)
# ~ 4 Seconds!
mostflights5 <- rxGetInfo(data = "sortFlights")
mostflights5
top5f <- as.data.frame(mostflights5[[5]])
topOA <- unique(as.vector(top5f$ArrDelay))
topOA



##################################
#   Linear Model with ReportProgress!
##################################

# Linear Model using rxLinMod
sampleDataDir <- rxGetOption("sampleDataDir")
airlineDemoSmall <- file.path(sampleDataDir, "AirlineDemoSmall.xdf")

    ineLinMod <- rxLinMod(ArrDelay ~ CRSDepTime, data = airlineDemoSmall,
```

R code in SQL Server as T-SQL to generate graphs

```
DECLARE @RScript nvarchar(max)
DECLARE @SQLScript nvarchar(max)

SET @RScript = N'library(plotly)
                 library(ggplot2)
                 library(htmlwidgets)
                 #setwd("C:/DataTK/HTML")
                  image_file <- tempfile()
                  jpeg(filename = image_file, width = 500, height = 500)
                  df <- InputDataSet
                  d <- df[sample(nrow(df), 10), ]
                  p <- plot_ly(d, x = OrderQty, y = DiscountPct, text = paste("OrderQty: ", OrderQty),
                          mode = "markers", color = OrderQty, size = OrderQty)
                  saveWidget(as.widget(p), "index.html")
                  OutputDataSet <- data.frame(data=readBin(file(image_file, "rb"), what=raw(), n=1e6))'


SET @SQLScript = N'SELECT
                       ps.[Name]
                      ,AVG(sod.[OrderQty]) AS OrderQty
                      ,so.[DiscountPct]
                      ,pc.name AS Category
                  FROM  Adventureworks.[Sales].[SalesOrderDetail] sod
                  INNER JOIN Adventureworks.[Sales].[SpecialOffer] so
                  ON so.[SpecialOfferID] = sod.[SpecialOfferID]
                  INNER JOIN Adventureworks.[Production].[Product] p
                  ON p.[ProductID] = sod.[ProductID]
                  INNER JOIN Adventureworks.[Production].[ProductSubcategory] ps
                  ON ps.[ProductSubcategoryID] = p.ProductSubcategoryID
                  INNER JOIN Adventureworks.[Production].[ProductCategory] pc
                  ON pc.ProductCategoryID = ps.ProductCategoryID
                  GROUP BY ps.[Name],so.[DiscountPct],pc.name'

EXECUTE sp_execute_external_script
@language = N'R',
@script = @RScript,
@input_data_1 = @SQLScript
WITH RESULT SETS ((Plot varbinary(max)))
```

Microsoft

Benefits of R integration

- Based on Open source R
- Different versions available (Open, Client and Server)
- Distributed workloads, multi-threading and parallelization
- Interoperable (Windows, Linux, MacOS) with different flavors (Hadoop, Teradata, HDInsight)
- Faster model prediction and model deployment
- No „in-memory“ constraints, less data movement, less bottlenecks in performance, no data size limitations
- Hybrid topologies, agile development, stable platform for data operationalization, investment protection (SLA, Terms and agreements)
- R Code is available in SSMS environment
- Community and commercial support
- R Language is growing in popularity

Bunch of demos

SQL Server vNext (CTP1 and above)

Microsoft R Server 9.0 (MRS9)

- SQL Server vNext (CTP1)
- R - 3.3.2
- RevoScaleR (9.0.3)
- MicrosoftML (1.0.0)

Quick recap: RevoScaleR (9.0.1)

Importing functions and computation context
rxImport
rxDataStep
rxGetInfo
rxSetInfo
rxGetVarInfo
rxSetVarInfo
rxGetVarNames
rxCreateColInfo
rxCompressXdf
RxXdfData
RxTextData
RxOdbcData
RxSqlServerData
rxOpen
rxClose
rxReadNext
rxSetFileSystem
rxGetFileSystem
rxNativeFileSystem
rxSetComputeContext
rxGetComputeContext
RxHadoopMR
RxInSqlServer
RxComputeContext
RxLocalSeq
RxLocalParallel
RxForeachDoPar
rxInstalledPackages
rxFindPackage

Data Manipulation
xDataStep
rxFactors
rxGetFuzzyDist
rxGetFuzzyKeys
rxSplit
rxSort
rxMerge
rxExecuteSQLDDL

Data Visualization
rxHistogram
rxLinePlot
rxLorenz
rxRocCurve

Descriptive /cross-tab Statistics
rxQuantile
rxSummary
rxCrossTabs
rxCube
rxMarginals
as.xtabs
rxChiSquaredTest
rxFisherTest
rxKendallCor
rxPairwiseCrossTab
rxRiskRatio
rxOddsRatio

Analysis and Predictive statistics
rxLinMod
rxLogit
rxGlm
rxCovCor
rxDTree
rxBTrees
rxDForest
rxPredict
rxKmeans
rxNaiveBayes
rxCov
rxCor
rxSSCP
rxRoc

Shorter list of functions

Quick recap: RevoScaleR (9.0.1)

	Task				Scalability			Description
	Predict categories		Predict values					
RevoScaleR functions / learner	**Binary classification**	**Multiclass classification**	**Regression**	**Other**	**rows**	**columns**	**CPU / threads**	
rxLinMod			Yes		100 Mil.	10 K	Multi	Linear model for regression model (analog to lm()) with stepwise
rxLogit	yes				100 Mil.	10 K	Multi	Logistic regression (similar to glm with family = binomial())
rxGlm	yes		Yes		10 Mil	5 K	Multi	Generalized linear regression models (with specified family)
rxDTree	Yes	Yes	Yes		100 Mil.		Multi	(recursive partitioning and regression) decision trees
rxBTrees	Yes	yes	Yes		100K	1K	Multi	Stohastic gradient Boosting DT (gaussian - Regression; bernoulli - Binary; Multinominal - multiclass)
rxDForest	Yes	Yes	Yes				Multi	Random forests (gaussian - Regression; bernoulli - Binary; Multinominal - multiclass)
rxKmeans				Classification		<10 K		Lloyd based K-means classification
rxNaiveBayes	Yes	Yes??			100Mil		Multi	Naive Bayes learner (class and probability)

All functions / learners work with XDF data formats

Microsoft

MicrosoftML (1.0.0)

- New fast and accurate learners (Sentiment analysis, Customer Churn, Loadn risk prediction, demand prediction)
- Text Classification (Sentiment Analysis, Classification of Support ticket, Complaint book,. etc)
- DNN with GPU Accelleration (Retail image matching, medical image classification, metal/iron/steel industry control check (with live stream)
- High-Dimensional Categorical data (regression predictions with a lot of productIDs, customerIDs, Web Analytics (click through predictions))

Microsoft

Algorithms in MicrosoftML

	Task				Scalability			Description
	Predict categories		Predict values					
MicrosoftML Learner	Binary classification	Multiclass classification	Regression	Other	rows	columns	CPU / threads	
rxFastLinear	Yes		Yes		1 Bil.	1 Bil.	Multi	Fast Linear (SDCA) with L1 & L2
rxLogisticRegression	Yes	Yes				100 Mil		Logistic regression with L1 & L2
rxNeuralNet	Yes	Yse	Yes		unlimited	10 Mil	Multi / GPU	Neural Network / GPU-accelerated NET# DNN with convolutions
rxFastTree	Yes		Yes			50 K	Multi	Boosted Decision Tree
rxFastForest	Yes		Yes			50 K	Multi	Random Forest
rxOneClassSvm				Anomaly/Reduction		1 K	Single	Anomaly detection / reduction / unbalanced binary classification

All learners work with XDF data formats

Microsoft

Questions?


tomaz.kastrun@gmail.com


@tomaz_tsql


/in/tomaztsql


http://github.com/tomaztk

https://mvp.microsoft.com/PublicProfile/5002196