Microsoft R Server ScaleR

Transparent Parallelism Accelerates Big Data Analytics Easily

Microsoft R Server provides computational and data size scalability through ScaleR, a library of big data analytics algorithms. Microsoft R Server ScaleR provides data scientists with a range of R algorithms that provide transparent parallelization of computations and data analysis they can easily scale to Big Data. ScaleR brings big data analytics within reach without added complexity or the need to learn new languages or parallel programming. It includes a rich set of data preparation, statistics, predictive modeling, and machine learning algorithms that accelerate Big Data Big Analytics and support systems ranging from workstations and servers, clustered systems such as Hadoop, Teradata or IBM or compute grids from Microsoft.

Comprehensive Big Data Analytics Algorithms in ScaleR

The following is a list of the big-data ready capabilities included with Microsoft R Server:

Data Preparation

- Data import: ASCII, SAS, SPSS, ODBC, HDFS
- Variable creation
- Variable transformation and recoding

- Sort / Merge / Split
- Random Sampling

Descriptive Statistics

- Min / Max / Mean
- Median and Quantiles
- Standard Deviation / Variance
- CoMicrosoft R Serverlation / Covariance/ Sum of Squares cross-product matrix
- Cross-Tabulations and marginal summaries
- Aggregation by category

Data Visualization for Big Data

- Histogram
- Line Plot / Scatter Plot
- Lorenz Curve
- ROC Curve
- Tree Visualizer

Statistical Tests

- Chi-squared Test
- Fisher's Exact Test
- Kendall's Tau Rank CoMicrosoft R Serverlation Coefficient
- Risk Ratio and Odds Ratio on two-by-two objects

Parallelized Statistical Modeling Algorithms

- Linear Regression
- Logistic Regression
- Multiple Regression
- Generalized Linear Models with all multiple exponential distributions (including Tweedie distribution) and a variety of standard and user-defined link functions
- Stepwise Regression – Linear, GLM & Logistic
- Clustering using K-Means Clustering

- Predictions for fitted models
- PMML export

Classification and Machine Learning Capabilities:

- Decision Trees
- Ensemble modeling with Decision Trees (similar to Random Forests)

Transparent Parallelism Brings Fast Execution

ScaleR algorithms enable R developers to run R scripts on massive data sets at high speeds. In conjunction with DistributedR, ScaleR transparently distributes analytics computations across all available resources – threads, cores, processors and nodes.

No Additional Languages, No Parallel Software Development

ScaleR enables R developers to easily maximize compute capability without writing any distributed applications themselves. This has two advantages over other solutions:

- No Java, Python or other programming skills are needed to harness the power of massively parallel systems including Hadoop and Teradata EDWs.
- No Parallel Programming. R developers are provided with transparent parallelism, so that they aren't slowed by the complexity of parallel program design. Parallelism is provided transparently within the Scale Algorithm set.

Available Parallel Platforms

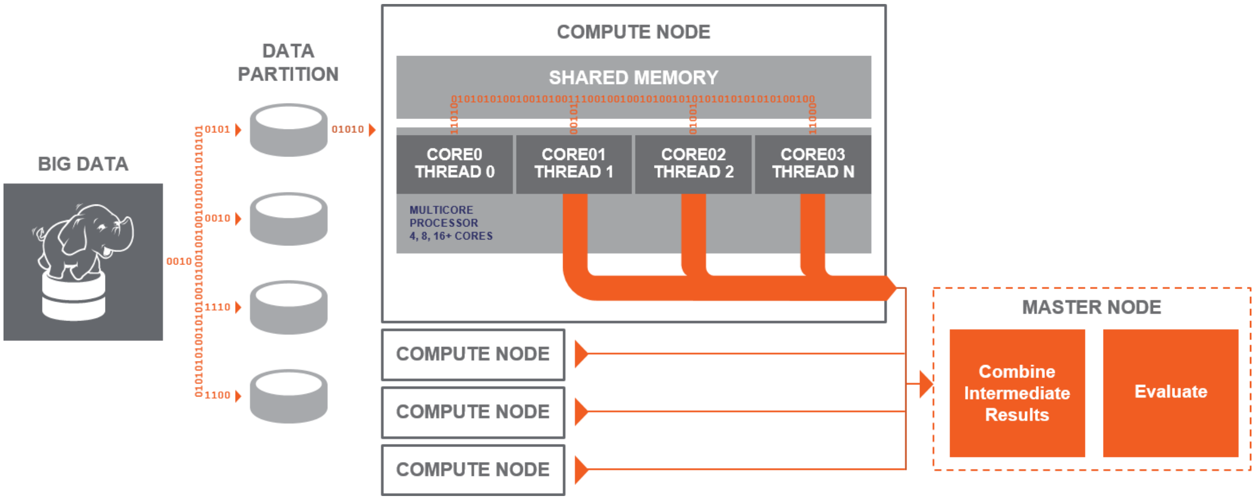
Revolution R Enteprise DistributedR brings all these Big Data algorithms distributed computing parallel platforms. Use the computing power of servers, grids, databases and Hadoop — without the need to move the data anywhere. DistributedR is supported on the following platforms:

- Teradata DB
- Hadoop clusters

This is a partial list – click here for a complete list of DistributedR supported platforms.

No Memory Barriers

Microsoft R Server ScaleR algorithms are implemented as Parallel External Memory Algorithms (PEMAs). By managing available RAM and permanent storage together, PEMAs are able to analyze data well beyond the limits of available memory.



- Compute Faster: Most ScaleR PEMA are optimized to run faster than their open source equivalents on both small and large data sets.
- Unlimited Data Size: PEMAs process data in “chunks” – moving data into memory as needed, enabling the algorithm to operate on data that far exceeds available memory.
- Fast Parallel Computation: PEMAs divide work into smaller pieces, distributing them across available cores, and nodes to dramatically accelerate modeling and machine learning.
- Efficient Analysis of Distributed Data: Storage of data in MPP EDWs and Hadoop clusters is distributed across many nodes of the compute cluster. By analyzing local data using local compute resources, data movement and consolidation is eliminated, providing optimum efficiency and speeding computation.