REVOLUTION ANALYTICS WHITE PAPER

Advanced 'Big Data' Analytics with R and Hadoop

'Big Data' Analytics as a Competitive Advantage

Big Analytics delivers competitive advantage in two ways compared to the traditional analytical model. First, Big Analytics describes the efficient use of a simple model applied to volumes of data that would be too large for the traditional analytical environment. Research suggests that a simple algorithm with a large volume of data is more accurate than a sophisticated algorithm with little data. The algorithm is not the competitive advantage; the ability to apply it to huge amounts of data—without compromising performance—generates the competitive edge.

Second, Big Analytics refers to the sophistication of the model itself. Increasingly, analysis algorithms are provided directly by database management system (DBMS) vendors. To pull away from the pack, companies must go well beyond what is provided and innovate by using newer, more sophisticated statistical analysis.

Revolution Analytics addresses both of these opportunities in Big Analytics while supporting the following objectives for working with Big Data Analytics:

1. Avoid sampling / aggregation;
2. Reduce data movement and replication;
3. Bring the analytics as close as possible to the data and;
4. Optimize computation speed.

First, Revolution Analytics delivers optimized statistical algorithms for the three primary data management paradigms being employed to address growing size and increasing variety of organizations' data, including file-based, MapReduce (e.g. Hadoop) or In-Database Analytics.

Second, the company is optimizing algorithms - even complex ones - to work well with Big Data. Open Source R was not built for Big Data Analytics because it is memory-bound. Depending on the type of statistical analysis required, Big Data also causes issues that we'll call "Big Computations," as some algorithms require a great deal of processing capacity on their own and may not lend themselves to running in every data management paradigm. For these Big Computations, parallelism (as we've deployed with IBM Netezza and ScaleR) is important to performance and to the accuracy of the statistical analysis. Coupled with an intuitive R Development Environment from Revolution Analytics, the degree of innovation exceeds that which may be achieved through packaged analytic applications.

This paper addresses specific integration between R and Hadoop that is supported by Revolution Analytics.

Revolution Analytics and Hadoop

Traditional IT infrastructure is simply not able to meet the demands of this new "Big Analytics" landscape. For these reasons, many enterprises are turning to the "R" statistical programming language and Hadoop (both open source projects) as a potential solution to this unmet commercial need.

As the amount of data—especially unstructured data—collected by organizations and enterprises explodes, Hadoop is emerging rapidly as one of the primary options for storing and performing operations on that data. A comment from Hadoop: The Definitive Guide, Second Edition contrasts the difference between HBase and traditional DBMSs, "We currently have tables with hundreds of millions of rows and tens of thousands of columns; the thought of storing billions of rows and millions of columns is exciting, not scary."

The marriage of R and Hadoop seems a natural one. Both are open source projects and both are data driven. But there are some fundamental challenges that need to be addressed in order to make the marriage work. Revolution Analytics is addressing these challenges with its Hadoop-based development.

Iterative vs. batch processing - If we look at how most people do analytics, it is often an interactive process. Start with a hypothesis, explore and try to understand the data, try some different statistical techniques, drill down on various dimensions, etc. This is what makes R such a powerful tool, and an ideal environment for performing such analysis. Hadoop on the other hand, is batch oriented where jobs are queued and then executed, and it may take minutes or hours to run these jobs.

In-memory vs. in parallel - Another fundamental challenge is that R is designed to have all of its data in memory and programs in Hadoop (map/reduce) work independently and in parallel on individual data slices.

Revolution Analytics' Capabilities for Hadoop

Revolution has created a series of "*RevoConnectRs for Hadoop*" that will allow an R programmer to manipulate Hadoop data stores directly from HDFS and HBASE, and give R programmers the ability to write MapReduce jobs in R using Hadoop Streaming. RevoHDFS provides connectivity from tR to HDFS and RevoHBase provides connectivity from R to HBase. Additionally, RevoHStream allows MapReduce jobs to be developed in R and executed as Hadoop Streaming jobs.

Delivered in the form of free downloadable R packages, *RevoConnectRs for Hadoop* will be available in September 2011 from http://www.revolutionanalytics.com/big-analytics.

HDFS Overview

To meet these challenges we have to start with some basics. First, we need to understand data storage in Hadoop, how it can be leveraged from R, and why it is important. The basic storage mechanism in Hadoop is HDFS (Hadoop Distributed File System). For an R programmer, being able to read/write files in HDFS from a standalone R Session is the first step in working within the

Hadoop ecosystem. Although still bound by the memory constraints of R, this capability allows the analyst to easily work with a data subset and begin some ad hoc analysis without involving outside parties. It also enables the R programmer to store models or other R objects that can then later be recalled and used in MapReduce jobs. When MapReduce jobs finish executing, they normally write their results to HDFS. Inspection of those results and usage for further analysis in R make this functionality essential.

HBASE Overview

There are several layers that sit on top of HDFS that also provide additional capabilities and make working with HDFS easier. One such implementation is HBASE, Hadoop's answer to providing database like table structures. Just like being able to work with HDFS from inside R, access to HBASE helps open up the Hadoop framework to the R programmer. Although R may not be able to load a billion-row- by-million-column table, working with smaller subsets to perform ad hoc analysis can help lead to solutions that work with the entire data set.
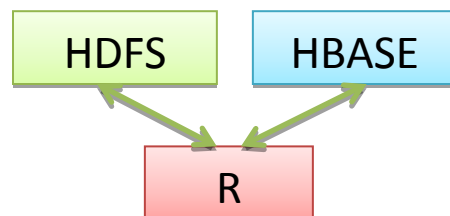

MapReduce – Data Reduction

The processing pillar in the Hadoop ecosystem is the MapReduce framework. The framework allows the specification of an operation to be applied to a huge data set, divide the problem and data, and run it in parallel. From an analyst's point of view, this can occur on multiple dimensions. For example, a very large dataset can be reduced into a smaller subset where analytics can be applied. In a traditional data warehousing scenario, this might entail applying an ETL operation on the data to produce something usable by the analyst. In Hadoop, these kinds of operations are written as MapReduce jobs in Java. There are a number of higher level languages like Hive and Pig that make writing these programs easier. The outputs of these jobs can be written back to either HDFS/HBASE or placed in a traditional data warehouse. R can then be used to do the analysis on the data.
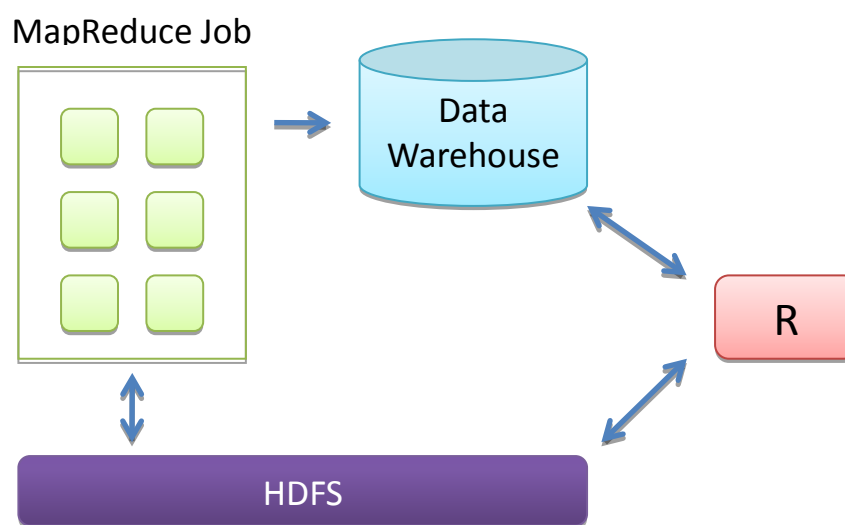
MapReduce - R

Executing R code in the context of a MapReduce job elevates the kinds and size of analytics that can be applied to huge datasets. Problems that fit nicely into this model include "pleasingly parallel" scenarios. Here's a simple use case: Scoring a dataset against a model built in R. This involves pushing the model to the Task nodes in the Hadoop cluster, running a MapReduce job that loads the model into R on a task node, scoring data either row-by row (or in aggregates), and writing the results back to HDFS. In the most simplistic case this can be done with just a Map task. This simulates the "apply" family of operators in R. Other tasks such as quantiles, crosstabs, summaries, data transformations and stochastic calculations (like Monte Carlo simulations) fit well within this paradigm. These implementations don't make any assumptions about how the data is grouped or ordered.

Visualizations of huge datasets can provide important insights that help understand the data. Creating a binning algorithm in R that is executed as a MapReduce job can produce an output that can be fed back into an R client to render such visualizations. Other more statistically challenging algorithms can also be implemented in this framework with more effort. These would include data Mining algorithms like K-Means clustering, logistic regression with small numbers of parameters and iterations, and even linear regression.
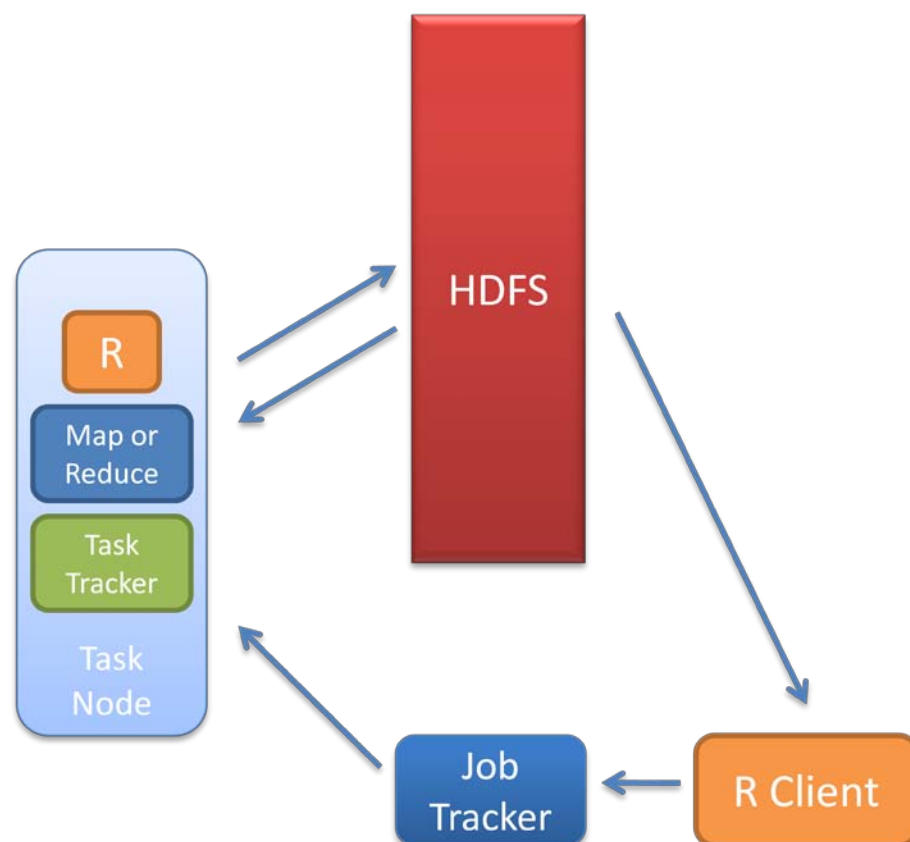
MapReduce - Hybrid

For some kinds of analysis, we can employ a hybrid model that combines using something like HIVE QL, and R. HIVE QL allows us to perform some SQL like capabilities to create naturally occurring groups where R models can be created. As an example, suppose we have some stock ticker data stored in HDFS. If we can use HIVE to partition this data into naturally occurring groups (i.e., stock ticker symbol) we could use R to create a time series model and forecast for each ticker, and do it in parallel. Another possibility might be creating a correlation matrix by using Hive and R, and feeding that into PCA or Factor Analysis routines.

Revolution has created an R package that allows creation of MapReduce jobs in R. The goal is providing a simple and usable interface that allows specification of both Map and Reduce as functions in R. This keeps the data scientist working in R, since he or she does not have to worry about the underlying Hadoop infrastructure. While it's true that the R programmer might have to rethink the approach to how algorithms can be realized and implemented, the potential benefits justify the additional effort.

Optimizing Algorithms

Finally, there is the approach of developing algorithms that have been explicitly parallelized to run within Hadoop. For example if you wanted to do a linear or logistic regression in R on a 1TB of data stored in HDFS, this requires that the algorithms themselves be implemented in way to use a distributed computing model. Revolution Analytics has a framework for developing these kinds of algorithms to be optimized within Hadoop.

Summary

The value from analysis on structured, transactional data is well understood and much of its value has been realized. Forward-looking models and other analysis that benefit from larger, more unstructured data sets (such as models of behavioral interactions) not as well understood, yet experts suggest that this new frontier of analytics holds untapped promise.

If the enterprise has an unmet business need for strategic decision making with a high degree of processing complexity using large volumes of are predominantly unstructured data and where the analysis technique is challenging, a Revolution Analytics and Hadoop combination offers significant opportunity to gain first mover advantage.

About Revolution Analytics

Revolution Analytics is the leading commercial provider of software and services based on the open source R project for statistical computing. Led by predictive analytics pioneer Norman Nie, the company brings high performance, productivity and enterprise readiness to R, the most powerful statistics language in the world. The company's flagship Revolution R Enterprise product is designed to meet the production needs of large organizations in industries such as finance, life sciences, retail, manufacturing and media.

Contact Us

Join the R Revolution at www.RevolutionAnalytics.com
Email: info@revolutionanalytics.com
Telephone: 650-646-9545
Twitter: @RevolutionR