Providing scalable, transparent access to database-resident data for efficient enterprise advanced analytics

Mark Hornick*, Denis Mukhin, Patrick Aboyoun, Vaishnavi Sashikanth

Oracle

*Contact author: mark.hornick@oracle.com

**Keywords:** database, transparency, R, scalability, SQL

Data analysts and statisticians, performing advanced analytics on enterprise data, are often affected by reduced efficiency due to three key pain points associated with accessing database-resident data: access latency, paradigm switching, and ability to scale to large datasets in their tool of choice, i.e., R.

Access latency occurs when analysts cannot directly access database data, but must request data extracts, typically from IT organizations. This impedes progress until the requested data is delivered. Enterprise costs, sometimes hidden, resulting from access latency include exporting, moving, and storing these data extracts, along with the associated backup, recovery, and security risks.

Paradigm switching involves changing from the R language and environment to a SQL-based language and/or environment for data access. Here, R users specify SQL to formulate queries to process or filter data in the database, and then pull data into the R environment for further processing using R.

When dealing with large enterprise-level data sets, R often does not scale, limited by the memory of the machine where R executes and not able to leverage multiprocessor machines without special packages or programming. With packages such as foreach, snow, Rmpi, R users may need to restructure their scripts or be aware of the underlying hardware configuration to enable parallelism.

These pain points can be minimized, if not eliminated, by providing transparent access to database tables from R. In our talk, we explain how Oracle R Enterprise [1] provides access from base R functionality to Oracle Database tables and views, allowing direct access to database-resident data from R. Knowledge of SQL or data tools is not required, eliminating paradigm shift. Using the database as a compute engine enables scaling to data sets much larger than available RAM and to leverage multi-processors for parallel query execution, as well as user-controlled data parallel execution.

Through the Oracle R Enterprise transparency layer, R users access data stored in the database as virtual data frames, matrices, vectors, etc. Base R functions performed on, e.g., ore.frame – an S4 class wrapping a database table or view with data.frame-like methods, are overloaded to generate SQL. This SQL is transparently sent to Oracle Database for execution. This transparency is complemented with the ability to embed R scripts in Oracle Database for execution using database server-resident R engines, which can be spawned in parallel. Overall functionality is expanded to include a wide range of statistical functions and machine learning algorithms executed in Oracle Database from R via Oracle SQL extensions.

In this talk, we introduce the ORE package, and the underlying architecture and high level design for Oracle R Enterprise transparency, embedded R execution, and enhanced statistical capabilities.

**References**