Scalable linear algebra with the **nzmatrix** package

Mario E. Inchiosa[1,*], Cezary Dendek[1], Przemysław Biecek[1]

1. Netezza Corporation
* Contact author: minchiosa@netezza.com

Keywords: Linear algebra, large matrices, ScaLAPACK, high performance computing, in-database analytics

R is an excellent platform for computing with matrices: it is easy to use, and it efficiently supports many useful matrix operations and linear algebra computations, for example via the **Matrix** package. *R* even provides partial access to the ScaLAPACK library of linear algebra routines for distributed-memory high performance computing, via the **RScaLAPACK** package. However, all matrix data feeding into and out of a calculation using these packages must fit within *R*'s memory. The **bigmemory** and **ff** packages lift the memory restriction, but they provide virtually no linear algebra support.

We will present a new package, **nzmatrix,** which enables the *R* user to perform a wide range of linear algebra computations and manipulations on very large matrices using fully distributed and parallelized memory, processing, and database storage. The package supports distributed matrix addition, subtraction, multiplication, inversion, transposition, linear equation solving, linear least squared problems, QR decomposition, LU decomposition, Cholesky decomposition, SVD decomposition, eigenvalues and eigenvectors, reshaping, reduction, and many other functions. **Nzmatrix** scales to hundreds of gigabytes of RAM, hundreds of processing cores, and hundreds of terabytes of database storage, and beyond.

The *R* environment serves as the front end for performing interactive or scripted operations with **nzmatrix** matrices in a manner very similar to that used for native *R* matrices. Matrices are transparently stored in a Netezza Performance Server (NPS) database. **Nzmatrix** provides user-friendly *R* wrapper functions for in-database stored procedures that access MPI, PBLAS, and ScaLAPACK routines for matrix-oriented high performance computing. Matrix data transfer directly from the database to the distributed matrix engine and back. Distributed matrices, in whole or in part, can be converted to native *R* matrices, and vice versa.