A Quick Guide for the pbdBASE Package

Version 2.0

Drew Schmidt[1], Wei-Chen Chen[2], George Ostrouchov[1,2], Pragneshkumar Patel[1]

[1]Remote Data Analysis and Visualization Center
University of Tennessee,
Knoxville, TN, USA

[2]Computer Science and Mathematics Division,
Oak Ridge National Laboratory,
Oak Ridge, TN, USA

Contents

Acknowledgement **ii**

Abstract **1**

1. Introduction **1**

1.1. Installation 1

1.2. Indented Audience 2

References **3**

This publication was typeset using LaTeX.

Acknowledgement

Ostrouchov, Schmidt, and Patel were supported in part by the project "NICS Remote Data Analysis and Visualization Center" funded by the Office of Cyberinfrastructure of the U.S. National Science Foundation under Award No. ARRA-NSF-OCI-0906324 for NICS-RDAV center. Chen and Ostrouchov were supported in part by the project "Visual Data Exploration and Analysis of Ultra-large Climate Data" funded by U.S. DOE Office of Science under Contract No. DE-AC05-00OR22725.

This work used resources of National Institute for Computational Sciences at the University of Tennessee, Knoxville, which is supported by the Office of Cyberinfrastructure of the U.S. National Science Foundation under Award No. ARRA-NSF-OCI-0906324 for NICS-RDAV center. This work also used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. This work used resources of the Newton HPC Program at the University of Tennessee, Knoxville.

We thank our colleague, Ed D'Azevedo from the Computational Mathematics Group, Computer Science and Mathematics Division, Oak Ridge National Laboratory (ORNL), for his discussions and illuminating advice using ScaLAPACK and distributed matrix computation.

We also thank Brian D. Ripley, Kurt Hornik, Uwe Ligges, and Simon Urbanek from the R Core Team for discussing package release issues and helping us solve portability problems on different platforms.

Abstract

With the size of data ever growing, the use of multiple processors in a single analysis becomes more and more a necessity. The Programming Big Data (pbd) project attempts to address the R language's current shortcomings in parallel distributed computations. The **pbdBASE** package for R provides a distributed matrix datatype and low-level methods for this data type, including extraction via `[`, as well as `NA` removal. Further, the package contains a set of BLACS, PBLAS, and ScaLAPACK wrappers. In addition to performance improvements through parallelism, use of this system with more than one processor allows the user to break R's local memory barrier, namely the requirement that a vector be indexed by a 32-bit integer, by only storing subsets of the vector on each processor.

1. Introduction

The Programming with Big Data: BASE system, the R (R Core Team 2012) package **pbdBASE** (Schmidt *et al.* 2012a), is a (mostly) implicitly parallel foundational infrastructure to support higher level pbd packages, such as **pbdDMAT** (Schmidt *et al.* 2012b). Much of what it does is meant to live behind the scenes of packages further up the chain of the pbd ecosystem, and is largely targeted at developers. However, it does offer some essential functionality for all users.

In many ways, the **pbdBASE** package serves the pbd project in much the same way as R's **base** package serves it. The principal goal of the **pbdBASE** package is to provide distributed classes (presently, a distributed dense matrix class), and many low-level functions for interacting with these classes. Many of these functions are wrappers of and for the distributed matrix algebra libraries BLACS, PBLAS, and ScaLAPACK. (Blackford *et al.* 1997) A set of S4 methods for R's linear algebra functions using these wrappers is provided by a separate package, **pbdDMAT**.

Updates and bug releases for this and other **pbd** projects may, especially while in infancy, be much more frequent than CRAN releases. So for up to date packages, as well as evolving information about the **pbd** project, see the website "Programming with Big Data in R" at http://r-pbd.org/.

1.1. Installation

The **pbdBASE** package is available from the CRAN at `http://cran.r-project.org`, and can be installed via a simple

Installing pbdBASE

```
install.packages("pbdBASE")
```

This assumes only that you have MPI installed and properly configured on your system. If the user can successfully install the package's two principal dependencies, **pbdMPI** (Chen *et al.* 2012a) and **pbdSLAP** (Chen *et al.* 2012c) (each available from the CRAN), then the installation for **pbdBASE** should go smoothly. If you experience difficulty installing either these packages, you should see their documentation.

1.2. Intended Audience

The **pbdBASE** package is a dependency of **pbdDMAT**, and so anyone who wishes to use the latter package must first install **pbdBASE**. However, much of the use of **pbdBASE** is intended only for extremely advanced users and developers.

1.3. Terminology

Before beginning, we will make frequent use of concepts from the Single Program/Multiple Data (SPMD) paradigm. If you are entirely unfamiliar with this approach to parallelism, or if you are unfamiliar with the **pbdMPI** package, then you are strongly encouraged to read the vignette (Chen *et al.* 2012b) contained in the **pbdMPI** package, as well as examine and digest its many examples in order to better understand what follows.

A concise explanation of SPMD is that it is an approach to parallel, distributed programming in which one program is written, and each processor runs that same program, though that program locally will often be interacting with different data. This, in contrast to the manager/worker paradigm where one processor, the manager, is in charge of its workers, each of whom swear fealty to the manager. So in SPMD, each processor believes itself to be the manager, the one in charge. As a colleague, Dr. Russell Zaretzki put it, "it's like academia."

Throughout the remainder, we will be discussing distributed data objects such as matrices, and wish to do so with some standardized terminology. A matrix is of course a rectangular collection of numbers. A *distributed matrix* then is just a matrix which has been decomposed in some fashion so that each processor only owns a piece of the "whole" matrix. The "whole" matrix (which need not ever actually exist, except theoretically, at any time), rather than pieces of it distributed among the processors, will be referred to as a/the *global* matrix. Loosely speaking, the global matrix is what we are really thinking of when we deal with the distributed matrix.

In the SPMD paradigm, each processor, though only owning a piece of the whole (henceforth referred to as the *local matrix* or *submatrix*, relative to that processor), will call functions on that matrix exactly as one would with an ordinary, non-distributed matrix on a single processor. The difference for the user is minimal; all the "heavy lifting" which explicitly handles the distributed nature of the object is performed in the background.

Matrices, distributed or otherwise, have dimensions — that is, lengths of the number of rows and the number of columns in the rectangle. The global matrix has a *global dimension*, and this is a global value, i.e., this value does not vary from processor to processor. Every processor agrees as to the size of the "full" matrix, otherwise we would have anarchy. However, the local matrices, in practice, will differ from processor to processor, and so too should their *local dimensions*. A local dimension, as the name implies, is the dimension of the submatrix, relative to a particular processor.

References

Blackford LS, Choi J, Cleary A, D'Azevedo E, Demmel J, Dhillon I, Dongarra J, Hammarling S, Henry G, Petitet A, Stanley K, Walker D, Whaley RC (1997). *ScaLAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA. ISBN 0-89871-397-8 (paperback). URL http://netlib.org/scalapack/slug/scalapack_slug.html/.

Chen WC, Ostrouchov G, Schmidt D, Patel P, Yu H (2012a). "pbdMPI: Programming with Big Data – Interface to MPI." R Package, URL http://cran.r-project.org/package=pbdMPI.

Chen WC, Ostrouchov G, Schmidt D, Patel P, Yu H (2012b). "A Quick Guide for the pbdMPI package." R Vignette, URL http://cran.r-project.org/package=pbdMPI.

Chen WC, Schmidt D, Ostrouchov G, Patel P (2012c). "pbdSLAP: Programming with Big Data – Scalable Linear Algebra Packages." R Package, URL http://cran.r-project.org/package=pbdSLAP.

R Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.r-project.org/.

Schmidt D, Chen WC, Ostrouchov G, Patel P (2012a). "pbdBASE: Programming with Big Data – Core pbd Classes and Methods." R Package, URL http://cran.r-project.org/package=pbdBASE.

Schmidt D, Chen WC, Ostrouchov G, Patel P (2012b). "pbdDMAT: Programming with Big Data – Distributed Matrix Algebra Computation." R Package, URL http://cran.r-project.org/package=pbdDMAT.