Setting Up a pbdR Environment

Prepared by the pbdR Core Team

Contents

1 R and RStudio

At a minimum, you will need to install R. You may also wish to install the popular IDE RStudio.

1.1 R

You can find R binaries and/or source at http://cran.r-project.org. Installation for Windows and Mac users should be very much like what you are used to. Linux and BSD users probably have access to R in their OS's software repository; you can find additional information at the above download link.

If for some reason you need/wish to build R from source, consult the R Installation and Administration manual https://cran.r-project.org/doc/manuals/R-admin.html.

1.2 Compilers

Finally, if you wish to be able to build packages from source, then you need to install some additional things.

- Windows: Install Rtools http://cran.r-project.org/bin/windows/Rtools/
- Mac:
  - First install XCode from the Mac App Store
  - Then, open XCode and choose "Preferences", "Downloads", then "Install Command Line Tools".
  - gfortran
- Linux/FreeBSD: You need C and Fortran compilers, which you may well already have installed. Consult your distribution's package manager.

1.3 RStudio (Optional)

R is supported in just about every text editor and multi-language IDE these days. However, much of the R community uses the RStudio Desktop IDE http://www.rstudio.com/products/rstudio/download/. It has binaries for Windows, Mac, and Linux. You can even install a server version running on a remote instance, and it looks/behaves much the same as the desktop version.

RStudio is probably the most popular editor used in the R community, so even if you prefer another editor/ide, it is still useful to get a sense for the tools that R users find most appealing. Installing RStudio is optional. But if you install it, you still need to install R separately as explained above.

2 R Packages

2.1 CRAN Packages

Most packages you will want to install are available on the Comprehensive R Archive Network (CRAN). To install a package named `pkg` (including automatically handling dependency resolution), start up an R session and enter

```r
install.packages("pkg")
```

Two packages which are very useful for the performance minded are:

- rbenchmark
- microbenchmark

You could install these by running:

```
install.packages(c("rbenchmark", "microbenchmark"))
```

2.2 GitHub Packages

There is a lot of exciting work on GitHub in the R world. Many people who publish their packages on CRAN maintain development versions on GitHub. So if you want the latest version, you need to install from GitHub. There are some packages that help with this effort, but to use them, you will need to first install some additional software. Follow the instructions in the Compilers subsection from the previous section if you have not already.

Once you have the necessary tools installed, you can simplify the process of installing a package from GitHub with an additional package. First run:

```
install.packages("remotes")
```

in an R session. Then to install a package from GitHub, you simply run:

```
remotes::install_github("pkg")
```

3 MPI

In addition to all of the previous requirements, you will need to have an MPI installation available in order to build and run pbdR packages.

3.1 Windows

Install MS-MPI

3.2 Mac

We recommend OpenMPI:

- From Homebrew: enter `brew install open-mpi` in a terminal
- From MacPorts: enter `sudo port install openmpi` in a terminal
- You can also build from source

3.3 Linux/FreeBSD

We recommend OpenMPI, which is likely available from your package manager:

- deb (Debian, Ubuntu): `apt-get install libopenmpi-dev`
- rpm (Fedora, Centos): `yum install openmpi-devel`
- FreeBSD: `pkg install openmpi`

You can also build from source.

4 pbdR

For more explanations of any individual package, see the pbdR packages site.

4.1 Compute

For the MPI compute tools, you will want to install:

```
install.packages('pbdMPI')
install.packages('pbdDMAT')
install.packages('pmclust')
install.packages('kazaam')
```

If you want the latest versions of all packages and their dependencies (including a few not yet on CRAN), run:

```
install_github('RBigData/pbdMPI')
install_github('RBigData/pbdSLAP')
install_github('RBigData/pbdBASE')
install_github('RBigData/pbdDMAT')
install_github('RBigData/pbdML')
install_github('RBigData/pmclust')
install_github('RBigData/kazaam')
```

4.2 I/O

For a useful, general purpose parallel I/O tool, install:

```
install_github('RBigData/pbdIO')
```

If you have netcdf4 installed, you may want:

```
install.packages('pbdNCDF4')
```

If you use the ADIOS system, then you may want:

```
install_github('RBigData/pbdADIOS')
```

4.3 Client/Server

For a basic client/server setup, install:

```
install.packages("remoter")
```

For the latest, including an MPI client/server, install:

```
install_github("RBigData/remoter")
install_github("RBigData/pbdCS")
```

4.4 Profiling

For MPI profiling, you will want:

```
install.packages("pbdPROF")
```

This will require a re-build of pbdMPI with some additional compiler flags. See the pbdMPI vignette for details.

For hardware counter profiling (NOTE: Linux only), install:

```
install_github("RBigData/pbdPAPI")
```

Finally, for a suite of profiling and vis tools that enhance pbdPROF and pbdPAPI, install:

```
install_github("RBigData/hpcvis")
```

5 Installation Problems

During the course of installation, you may run into unrecoverable issues. The pbdR team does not support MPI libraries or R core, so if you have problems during that portion of the installation phase, we probably can not directly help you. However, there are still many great resources at your disposal, maintained by those individual projects.

5.1 R and MPI

If you have problems installing or customizing R, see the *[R Installation and Administration Manual]*(http://cran.r-project.org/doc/manuals/R-admin.html) for help.

If you are having trouble installing an MPI library, such as OpenMPI or MPICH, you should see that library's official documentation.

For the remainder, we will be addressing installation issues with pbdR packages.

5.2 pbdR

This is a quick list of potential problems you could encounter when installing pbdR packages. For additional troubleshooting or installation options, each package has a vignette which may offer additional useful information.

- **Compiling pbdMPI from source**: you may be required to pass a configure argument at compile time. So for example, if you have OpenMPI installed and were installing from the command line, then you would issue the command:

  ```
  R CMD INSTALL pbdMPI_0.1-6.tar.gz  --configure-args='--with-mpi-type=OPENMPI'
  ```

  or if installing from R:

  ```
  install.packages("pbdMPI", configure.args='--with-mpi-type=OPENMPI')
  ```

  See the **pbdMPI** vignette for more details.

- **Installing on a cluster**: if you are required to install on a login node which can not execute `mpirun`, then pass the install option `--no-test-load`. So for example, if installing from the command line, then you would issue the command:

  ```
  R CMD INSTALL pbdMPI_0.1-6.tar.gz --no-test-load
  ```

  or if installing from R:

  ```
  install.packages("pbdMPI", INSTALL_opts='--no-test-load')
  ```