

# Contents

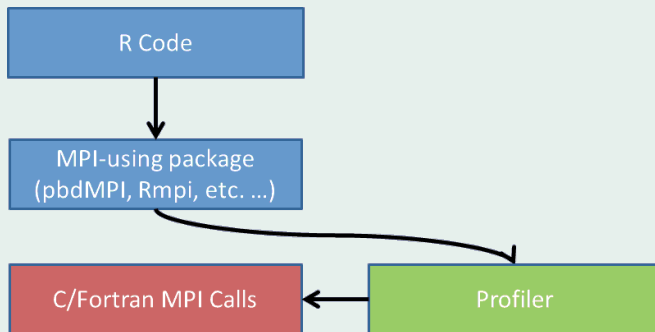
- 1 Profiling
  - Profiling
  - Installing pbdPROF
  - Example

## Introduction to **pbdPROF**

- Successful Google Summer of Code 2013 project.
- Available on the CRAN.
- Enables profiling of MPI-using R scripts.
- **pbdR** packages officially supported; can work with others. . .
- Also reads, parses, and plots profiler outputs.

## How it works

MPI calls get hijacked by profiler and logged:



## Introduction to **pbdPROF**

- Currently supports the profilers **fpmpi** and **mpiP**.
- **fpmpi** is distributed with **pbdPROF** and installs easily, but offers minimal profiling capabilities.
- **mpiP** is fully supported also, but harder to install.

## Installing **pbdPROF**

- 1 Build **pbdPROF**.
- 2 Rebuild **pbdMPI** (linking with **pbdPROF**).
- 3 Run your analysis as usual.
- 4 Interactively analyze profiler outputs with **pbdPROF**.

## Build pbdPROF

```
R CMD INSTALL pbdPROF_0.2-1.tar.gz
```

- The above installs **fpm**.
- **mpi** can be used if you have a system installation available.
- See package vignette for more details and troubleshooting.

## Rebuild pbdMPI

```
R CMD INSTALL pbdMPI_0.2-2.tar.gz  
  --configure-args="--enable-pbdPROF"
```

- Any package which explicitly links with an MPI library must be rebuilt in this way (**pbdMPI**, **Rmpi**, ...).
- Other **pbdR** packages link with **pbdMPI**, and so do not need to be rebuilt.
- See **pbdPROF** vignette if something goes wrong.

## Example Script

my\_svd.r

```
1 library(pbdMPI, lib.loc="~/R/prof", quietly=TRUE)
2 library(pbdDMAT, quietly=T)
3 init.grid()
4
5 n <- 1000
6 x <- ddmatrix("rnorm", n, n)
7
8 asdf <- La.svd(x)
9
10
11 finalize()
```



## Example Script

Run example with 4 ranks:

```
$ mpirun -np 4 Rscript my_svd.r
mpiP:
mpiP: mpiP: mpiP V3.4.0 (Build Feb 14 2014/13:55:39)
mpiP: Direct questions and errors to
      mpip-help@lists.sourceforge.net
mpiP:
Using 2x2 for the default grid size

mpiP:
mpiP: Storing mpiP output in [./R.4.28812.1.mpiP].
mpiP:
```

## Read Profiler Data into R

Interactively (or in batch) Read in Profiler Data

```
1 library(pbdPROF)
2 prof.data <- read.prof("R.4.28812.1.mpiP")
```

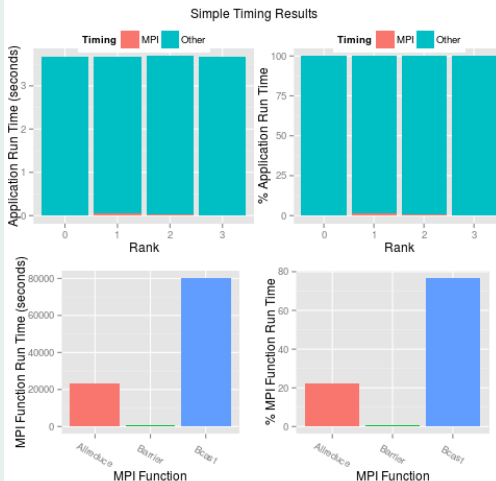
Partial Output of Example Data

```
> prof.data
An mpip profiler object:
[[1]]
  Task AppTime MPITime MPI.
1     0    3.68 0.00816 0.22
2     1    3.68 0.04890 1.33
3     2    3.69 0.03850 1.04
4     3    3.68 0.00904 0.25
5     *   14.70 0.10500 0.71

[[2]]
  ID Lev File.Address Line_Parent_Funct MPI_Call
1   1   0 1.400699e+14      [unknown] Allreduce
2   2   0 1.400699e+14      [unknown] Allreduce
```

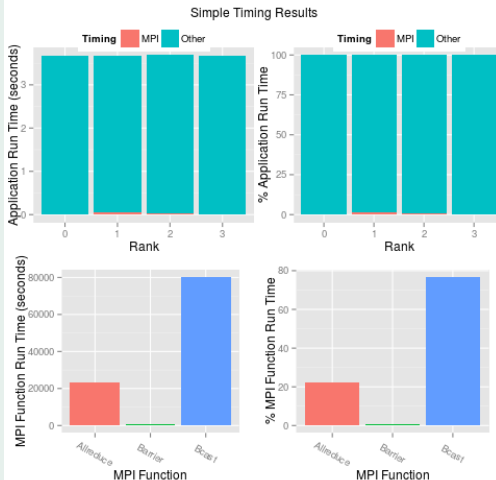
## Generate plots

```
1 plot(prof.data)
```



## Generate plots

```
1 plot(prof.data, plot.type="stats1")
```



## Generate plots

```
1 plot(prof.data, plot.type="messages1")
```

