Introducing R:
From Your Laptop to HPC and Big Data

George Ostrouchov and Drew Schmidt

SC14

The pbdR Core Team

Wei-Chen Chen[1]
George Ostrouchov[2,3]
Pragneshkumar Patel[3]
Drew Schmidt[3]

pbdR
Programming with Big Data in R

Support

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.

[1] Department of Ecology and Evolutionary Biology
University of Tennessee, Knoxville TN, USA

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

[3] Joint Institute for Computational Sciences
University of Tennessee, Knoxville TN, USA

About This Presentation

Downloads

This presentation is available at: http://r-pbd.org/tutorial

About This Presentation

Installation Instructions

Installation instructions for setting up a pbdR environment are available:

http://r-pbd.org/install.html

This includes instructions for installing R, MPI, and pbdR.

Contents

1 Profiling and Benchmarking

2 Wrapup

Contents

1. Profiling and Benchmarking
   - Profiling R Code
   - Advanced R Profiling
   - Summary

1. Profiling and Benchmarking
   - Profiling R Code
   - Advanced R Profiling
   - Summary

Timings

Getting simple timings as a basic measure of performance is easy, and valuable.

- `system.time()` — timing blocks of code.
- `Rprof()` — timing execution of R functions.
- `Rprofmem()` — reporting memory allocation in R .
- `tracemem()` — detect when a copy of an R object is created.
- The **rbenchmark** package — Benchmark comparisons.

Performance Profiling Tools: system.time()

system.time() is a basic R utility for timing expressions

```r
x <- matrix(rnorm(20000*750), nrow=20000, ncol=750)

system.time(t(x) %*% x)
#     user  system elapsed
#    2.187   0.032   2.324

system.time(crossprod(x))
#     user  system elapsed
#    1.009   0.003   1.019

system.time(cov(x))
#     user  system elapsed
#    6.264   0.026   6.338
```

Performance Profiling Tools: Rprof()

Rprof() times the execution of all R functions:

```
Rprof(filename="Rprof.out", append=FALSE, interval=0.02,
  memory.profiling=FALSE, gc.profiling=FALSE,
  line.profiling=FALSE, numfiles=100L, bufsize=10000L)
```

```
x <- matrix(rnorm(10000*250), nrow=10000, ncol=250)

Rprof(interval=.99)
invisible(prcomp(x))
Rprof(NULL)

summaryRprof()
```

Performance Profiling Tools: Rprof()

```
$by.self
                self.time self.pct total.time total.pct
"La.svd"             0.68    69.39       0.72     73.47
"%*%"                0.12    12.24       0.12     12.24
"aperm.default"      0.04     4.08       0.04      4.08
"array"              0.04     4.08       0.04      4.08
"matrix"             0.04     4.08       0.04      4.08
"sweep"              0.02     2.04       0.10     10.20
### output truncated by presenter

$by.total
                 total.time total.pct self.time self.pct
"prcomp"               0.98    100.00      0.00     0.00
"prcomp.default"       0.98    100.00      0.00     0.00
"svd"                  0.76     77.55      0.00     0.00
"La.svd"               0.72     73.47      0.68    69.39
### output truncated by presenter

$sample.interval
[1] 0.02

$sampling.time
[1] 0.98
```

Performance Profiling Tools: Rprof()

```
$by.self
[1] self.time  self.pct   total.time total.pct
<0 rows> (or 0-length row.names)

$by.total
[1] total.time total.pct  self.time  self.pct
<0 rows> (or 0-length row.names)

$sample.interval
[1] 0.99

$sampling.time
[1] 0
```

Performance Profiling Tools: rbenchmark

rbenchmark is a simple package that easily benchmarks different functions:

```
x <- matrix(rnorm(10000*500), nrow=10000, ncol=500)

f <- function(x) t(x) %*% x
g <- function(x) crossprod(x)

library(rbenchmark)
benchmark(f(x), g(x))

#   test replications elapsed relative
# 1 f(x)          100  64.153    2.063
# 2 g(x)          100  31.098    1.000
```

1 Profiling and Benchmarking

- Profiling R Code
- Advanced R Profiling
- Summary

Other Profiling Tools

- perf
- PAPI
- MPI profiling: fpmpi, mpiP, TAU

Profiling MPI Codes with **pbdPROF**

1. Rebuild pbdR packages

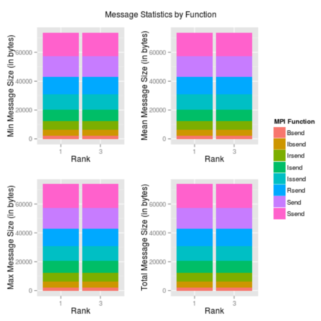
```
R CMD INSTALL pbdMPI_0.2-1.tar.gz \
    --configure-args= \
    "--enable-pbdPROF"
```

2. Run code

```
mpirun -np 64 Rscript my_script.R
```

3. Analyze results

```
library(pbdPROF)
prof <- read.prof( "output.mpiP")
plot(prof, plot.type="messages2")
```



Profiling with **pbdPAPI**

- Bindings for Performance Application Programming Interface (PAPI)
- Gathers detailed hardware counter data.
- High and low level interfaces

Function	Description of Measurement
`system.flips()`	Time, floating point instructions, and Mflips
`system.flops()`	Time, floating point operations, and Mflops
`system.cache()`	Cache misses, hits, accesses, and reads
`system.epc()`	Events per cycle
`system.idle()`	Idle cycles
`system.cpuormem()`	CPU or RAM bound*
`system.utilization()`	CPU utilization*

1 Profiling and Benchmarking
- Profiling R Code
- Advanced R Profiling
- Summary

Summary

- *Profile, profile, profile.*
- Use `system.time()` to get a general sense of a method.
- Use **rbenchmark**'s `benchmark()` to compare 2 methods.
- Use `Rprof()` for more detailed profiling.
- Other tools exist for more hardcore applications (**pbdPAPI** and **pbdPROF**).

Contents

2 Wrapup

Summary

- Profile your code to understand your bottlenecks.
- pbdR makes distributed parallelism with R easier.
- Distributing data to multiple nodes
- For truly large data, I/O must be parallel as well.

The pbdR Project

- Our website: http://r-pbd.org/
- Email us at: RBigData@gmail.com
- Our google group: http://group.r-pbd.org/

Where to begin?

- The **pbdDEMO** package
  http://cran.r-project.org/web/packages/pbdDEMO/
- The **pbdDEMO** Vignette: http://goo.gl/HZkRt

Thanks for coming!

Questions?

pbdR

Programming with Big Data in R

http://r-pbd.org/

Come see our poster on Wednesday at 5:30!