Tight Coupling of R and Distributed Linear Algebra for High-Level Programming with Big Data

D. Schmidt[1], G. Ostrouchov[1,2], W-C. Chen[2], P. Patel[1]

November 12, 2012

http://r-pbd.org

[1]Remote Data Analysis and Visualization Center, University of Tennessee, Knoxville, TN
[2]Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN

Contents

What is R? (1 of 2)

What is R? (1 of 2)

1. High-level DSL.

What is R? (1 of 2)

1. High-level DSL.
2. Free as in "beer", free as in "speech" (GPL).

What is R? (1 of 2)

1. High-level DSL.
2. Free as in "beer", free as in "speech" (GPL).
3. A C program (mostly): 52% C 26% Fortran 22% R

What is R? (1 of 2)

1. High-level DSL.
2. Free as in "beer", free as in "speech" (GPL).
3. A C program (mostly): 52% C 26% Fortran 22% R
4. Highly extensible, with over 4000 user-contributed packages.

What is R? (2 of 2)

What is R? (2 of 2)

1. *lingua franca* for analytics.

What is R? (2 of 2)

1. *lingua franca* for analytics.
2. Dialect of S (Bell Labs).

What is R? (2 of 2)

1. *lingua franca* for analytics.
2. Dialect of S (Bell Labs).
3. Syntax designed for people thinking about data.

What is R? (2 of 2)

1. *lingua franca* for analytics.
2. Dialect of S (Bell Labs).
3. Syntax designed for people thinking about data.
4. Functional programming paradigms, lazy evaluation, and lexical scoping semantics, and 2 official OOP systems.

Who Uses R? Industry

Who Uses R? Industry

Google, Pfizer, Merck, Bank of America, Shell[a], Oracle[b], Facebook, bing, Mozilla, okcupid[c], ebay[d], kickstarter[e], the New York Times[f]

---

[a] https://www.nytimes.com/2009/01/07/technology/business-computing/07program.html?_r=0

[b] http://www.oracle.com/us/corporate/features/features-oracle-r-enterprise-498732.html

[c] http://www.revolutionanalytics.com/what-is-open-source-r/companies-using-r.php

[d] http://blog.revolutionanalytics.com/2012/09/using-r-in-production-industry-experts-share-their-experiences.html

[e] http://blog.revolutionanalytics.com/2012/09/kickstarter-facilitates-50m-in-indie-game-funding.html

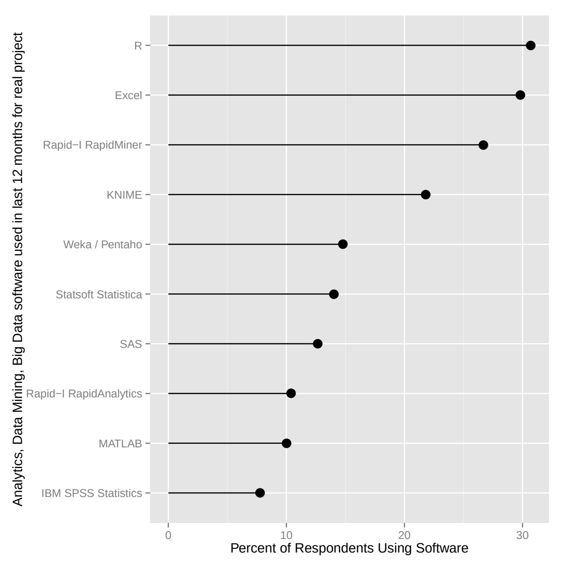
[f] http://blog.revolutionanalytics.com/2012/05/nyt-charts-the-facebook-ipo-with-r.html

Who Uses R? Data Miners (KDnuggets)

Who Uses R? Data Miners (KDnuggets)

May 2012 responses for: "What Analytics, Data mining, Big Data software you used in the past 12 months for a real project (not just evaluation)" [a]

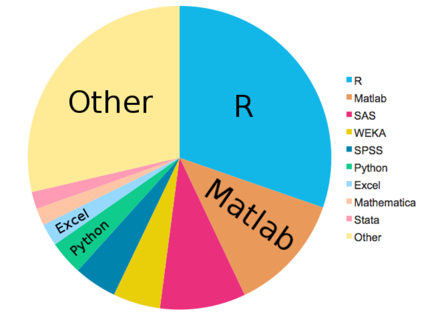
[a] *http://www.kdnuggets.com/2012/05/top-analytics-data-mining-big-data-software.html*



Who Uses R? Data Miners (Kaggle)

Who Uses R? Data Miners (Kaggle)

A third of all Kaggle competitors use R:[a]



and 50% of *winners* used R[b]

[a] http://www.meetup.com/R-Users/events/16946398/

[b] http://www.revolutionanalytics.com/news-events/news-room/2011/Revolution-Analytics-Fuels-Data-Science-Competition.php

Who Users R? Academia

Who Users R? Academia

The Journal of Statistical Software (JSS) was named a rising star in computer science by Science Watch for September and November 2011[a]:

> *The boundary between Computer Science and Statistics is vague — especially in the computational area. So providing a publication and quick distribution medium for data analysis software along with reproducible applications —* **for R packages in particular** *— is the main contribution.*

[a] http://archive.sciencewatch.com/inter/jou/2011/11decJofStatSoft/

Problems with R

Problems with R

1. Slow.

Problems with R

1. Slow.
2. If you don't know what you're doing, it's *really* slow.

Problems with R

1. Slow.
2. If you don't know what you're doing, it's *really* slow.
3. Performance improvements usually for small machines.

Problems with R

1. Slow.
2. If you don't know what you're doing, it's *really* slow.
3. Performance improvements usually for small machines.
4. Very ram intensive.

Problems with R

1. Slow.
2. If you don't know what you're doing, it's *really* slow.
3. Performance improvements usually for small machines.
4. Very ram intensive.
5. No big data.

R and Parallelism

R and Parallelism

1. Mostly serial.

R and Parallelism

1. Mostly serial.
2. When it isn't, it's overwhelmingly not distributed.

R and Parallelism

1. Mostly serial.
2. When it isn't, it's overwhelmingly not distributed.
3. Parallelism is mostly explicit.

R and Parallelism

1. Mostly serial.
2. When it isn't, it's overwhelmingly not distributed.
3. Parallelism is mostly explicit.

R does not scale.

Bridging the Gap

Bridging the Gap

Computer Scientists:

Bridging the Gap

Computer Scientists:

- Great scalable libraries
- Weak analytics

Bridging the Gap

Computer Scientists:

- Great scalable libraries
- Weak analytics

Data Scientists:

Bridging the Gap

Computer Scientists:

- Great scalable libraries
- Weak analytics

Data Scientists:

- Great analytical methods
- Weak computing

Bridging the Gap

Computer Scientists:

- Great scalable libraries
- Weak analytics

Data Scientists:

- Great analytical methods
- Weak computing

Working on similar problems. . .

Bridging the Gap

Computer Scientists:

- Great scalable libraries
- Weak analytics

Data Scientists:

- Great analytical methods
- Weak computing

Working on similar problems. . .

Can't we all just get along?

Programming with Big Data in R (pbdR)

Programming with Big Data in R (pbdR)

Goal: Bring HPC to a wider audience of data scientists.

Programming with Big Data in R (pbdR)

Goal: Bring HPC to a wider audience of data scientists.

Our solution:

- Series of free R packages.

Programming with Big Data in R (pbdR)

Goal: Bring HPC to a wider audience of data scientists.

Our solution:

- Series of free R packages.
- Enables big data analytics.

Programming with Big Data in R (pbdR)

Goal: Bring HPC to a wider audience of data scientists.

Our solution:

- Series of free R packages.
- Enables big data analytics.
- Distributed dense linear algebra + R sugar.

Programming with Big Data in R (pbdR)

Goal: Bring HPC to a wider audience of data scientists.

Our solution:

- Series of free R packages.
- Enables big data analytics.
- Distributed dense linear algebra + R sugar.
- Identical to R's syntax via OOP.

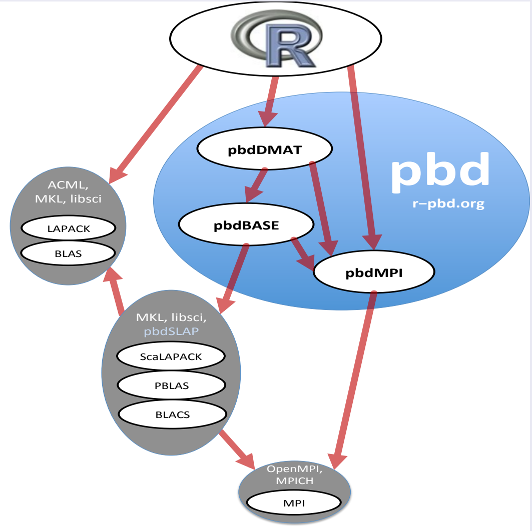
Programming with Big Data in R (pbdR)

Goal: Bring HPC to a wider audience of data scientists.

Our solution:

- Series of free R packages.
- Enables big data analytics.
- Distributed dense linear algebra + R sugar.
- Identical to R's syntax via OOP.
- Powered underneath by MPI, ScaLAPACK, PBLAS, BLACS, LAPACK, BLAS

Programming with Big Data in R (pbdR)



Example Syntax: Linear Algebra

Example Syntax: Linear Algebra

$$
\begin{aligned}
x &:= \log\left(|x|\right) \\
xtx &:= x^T x \\
xtx.inv &:= (xtx)^{-1} \\
ans &:= chol(xtx.inv) \\
&= LL^T
\end{aligned}
$$

Example Syntax: Linear Algebra

$$
\begin{aligned}
x &:= \log\left(|x|\right) \\
xtx &:= x^T x \\
xtx.inv &:= (xtx)^{-1} \\
ans &:= chol(xtx.inv) \\
&= LL^T
\end{aligned}
$$

R/pbdR

```r
x <- log(abs(x))
xtx <- t(x) %*% x
xtx.inv <- solve(xtx)
ans <- chol(xtx.inv)
```

Example Syntax: Sugar

Example Syntax: Sugar

Drop row 1, extract columns 2-5

Example Syntax: Sugar

Drop row 1, extract columns 2-5

R/pbdR

```
x <- x[-1, 2:5]
```

PCA Benchmark

PCA Benchmark

- Principal Components Analysis (PCA) on random normal data.

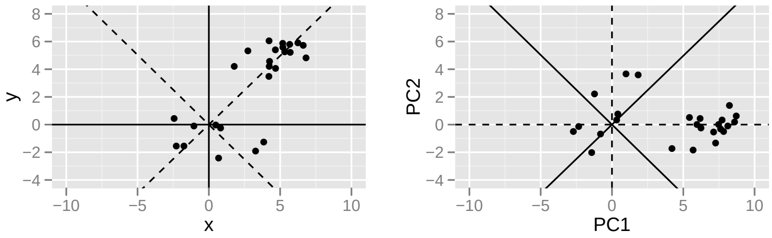
PCA Benchmark

- Principal Components Analysis (PCA) on random normal data.
- Measure time to compute PCA.

PCA?

PCA?

PCA = centering + scaling + SVD + Rotation



PCA Code

PCA Code

Fortran

```fortran
      CALL PDLACPY('N', M, N,
     $ X, IX, JX, DESCX, CPX,
     $ IX, JX, DESCX)

      CALL PDGESVD('N', 'V',
     $ M, N, CPX, IX, JX,
     $ DESCX, S, U, IU, JU,
     $ DESCU, VT, IVT, JVT,
     $ DESCVT, WORK, LWORK,
     $ INFO)

      CALL PDGEMM('N', 'N',
     $ M, N, K, 1.0D1, X, IX,
     $ JX, DESCA, VT, IVT,
     $ JVT, DESCVT, 0.0D1, Z,
     $ IZ, JZ, DESCZ)
```

PCA Code

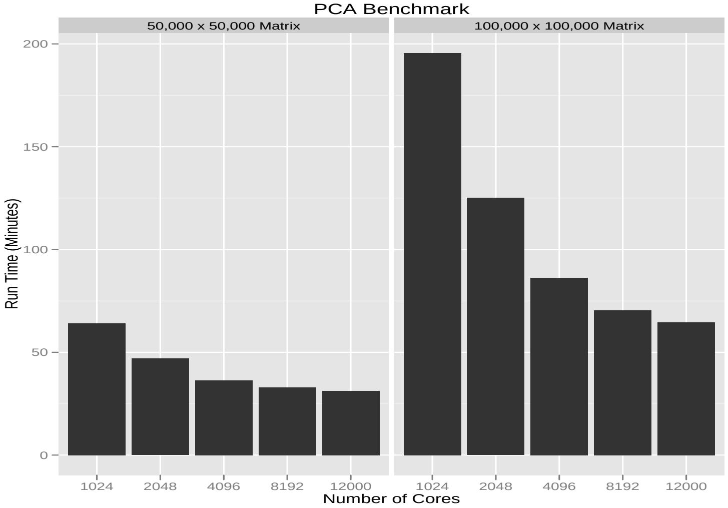
Fortran

```fortran
      CALL PDLACPY('N', M, N,
     $ X, IX, JX, DESCX, CPX,
     $ IX, JX, DESCX)

      CALL PDGESVD('N', 'V',
     $ M, N, CPX, IX, JX,
     $ DESCX, S, U, IU, JU,
     $ DESCU, VT, IVT, JVT,
     $ DESCVT, WORK, LWORK,
     $ INFO)

      CALL PDGEMM('N', 'N',
     $ M, N, K, 1.0D1, X, IX,
     $ JX, DESCA, VT, IVT,
     $ JVT, DESCVT, 0.0D1, Z,
     $ IZ, JZ, DESCZ)
```

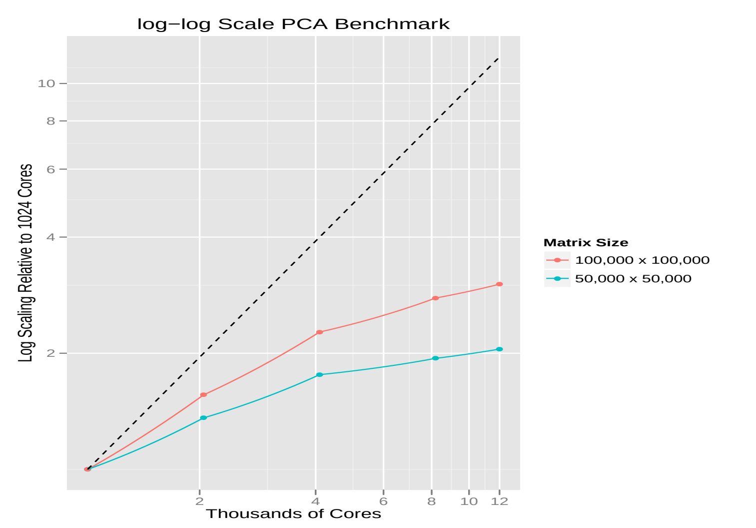
R/pbdR

```r
z <- prcomp(x,
    scale=TRUE)
```

Benchmarks


PCA Benchmark
50,000 x 50,000 Matrix
100,000 x 100,000 Matrix
Run Time (Minutes)
0
50
100
150
200
1024
2048
4096
8192
12000
1024
2048
4096
8192
12000
Number of Cores

Benchmarks


log−log Scale PCA Benchmark
Log Scaling Relative to 1024 Cores
10
8
6
4
2
2
4
6
8
10
12
Thousands of Cores
Matrix Size
100,000 x 100,000
50,000 x 50,000

Coming Soon

1. Linear models (linear least squares problems)
2. Package demos
3. pbdR inside VisIt
4. Parallel NetCDF reader
5. Parallel Model-Based Clustering

Future Work

1. Generalized linear models (Newton-Raphson method).
2. Sparse linear algebra (PETSc)
3. Parallel SVM
4. ADIOS reader
5. pbdR inside UV-CDAT

Thanks!

Questions?