The State of HPC in the Open Source R Ecosystem

Drew Schmidt

November 12, 2016

Support and Disclaimer

This material is based upon work supported by the National Science Foundation Division of Mathematical Sciences under Grant No. 1418195.

Speaker Bio

- M.S. in mathematics.
- Former statistics consultant.
- Former full-time university researcher.
- Now a miserable grad student.
- Prolific complainer on twitter.

Goals of This Talk

- Convince you that R has a legitimate place in HPC.
- Give a broad overview of the R package landscape.
- Make some very safe predictions.

Contents

Types

- `logical` ("boolean")
- `integer` (32-bit int)
- `numeric` (double)
- `complex` (double complex)
- `character` (string)

Also `raw` and `external pointer`

Data Structures

- Vectors (matrices, n-dim arrays)
- Lists (arrays of pointers)
- Dataframes (lists with constraints)
- Environments (hash tables?!)
- That's it.

Happy Opposite Day!

```
T
## [1] TRUE
F
## [1] FALSE

T <- FALSE
F <- TRUE

T
## [1] FALSE
F
## [1] TRUE
```

Odd Conventions

- . has no semantic meaning (except when it does
  - `t.test()`
  - `t.data.frame()`
- A *package* is installed in a *library*.

Package or Library?

- I wrote a library.
- I put that library into a package.
- I installed the package . . . into a library.
- I load the package with `library()` ???

BOOM

Part Programming Language, Part Data Analysis Package

"*R is a shockingly dreadful language for an exceptionally useful data analysis environment.*"
— Tim Smith, from **aRrgh: a newcomer's (angry) guide to R**.

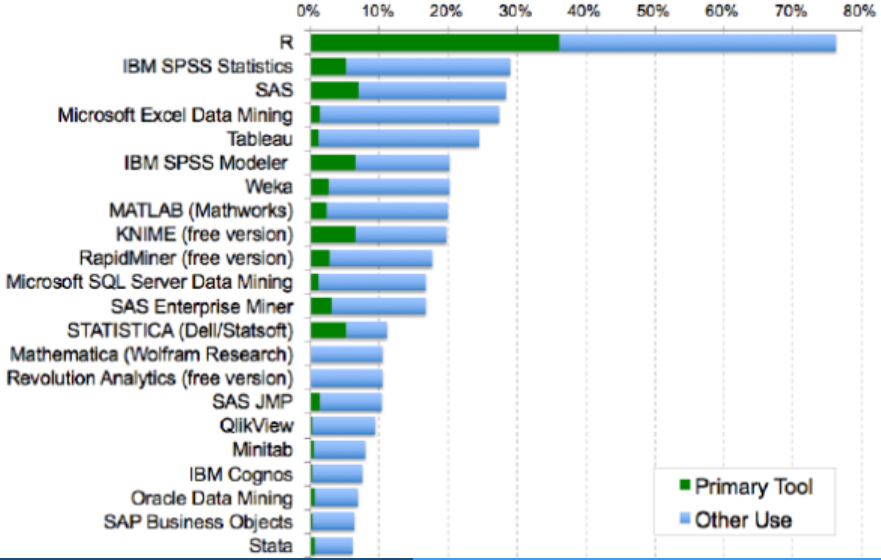
IEEE Spectrum's 2014 Ranking of Programming Languages

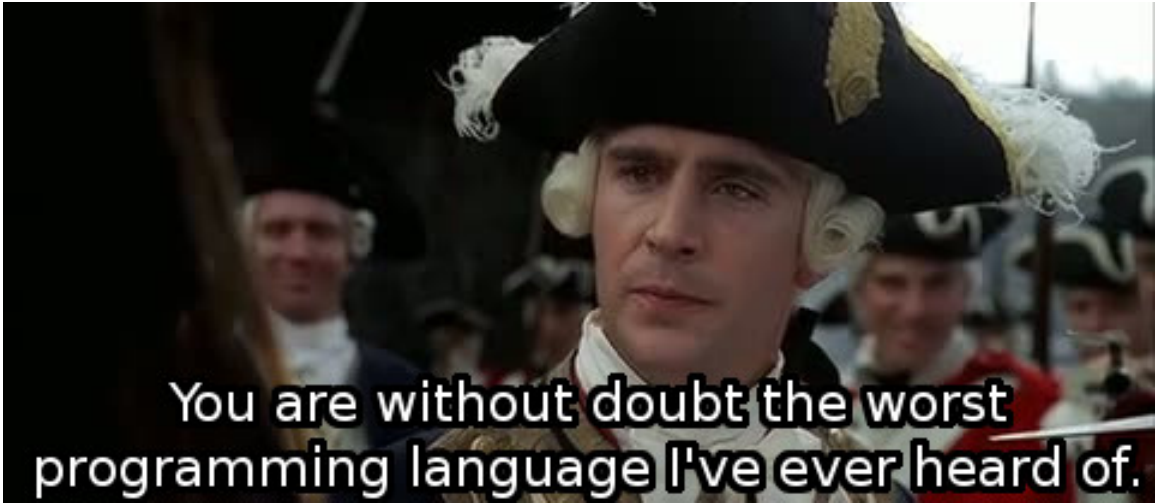
Language Rank	Types	Spectrum Ranking
1. Java		100.0
2. C		99.3
3. C++		95.5
4. Python		93.4
5. C#		92.4
6. PHP		84.7
7. Javascript		84.4
8. Ruby		78.8
9. R		74.2
10. MATLAB		72.9

IEEE Spectrum's 2016 Ranking of Programming Languages

Language Rank	Types	Spectrum Ranking
1. C		100.0
2. Java		98.1
3. Python		98.0
4. C++		95.9
5. R		87.9
6. C#		86.7
7. PHP		82.8
8. JavaScript		82.2
9. Ruby		74.5
10. Go		71.9

Rexer 2015 data scientist survey




You are without doubt the worst
programming language I've ever heard of.


But you HAVE heard of me!

Computer Scientists Hate It!

Looks Weird Is Useful

Esoteric language topples industry standards using one weird trick. Click to learn its stunning secrets.

LEARN THE TRUTH NOW

Why use R at all?

- Most diverse set of statistical methods available.
- Rapid prototyping.
- CRAN (and increasingly GitHub) packages.
- *Awesome* community.
- Syntax is designed for analysis of data.

2 A Little History

- Statistics, Data Science, Big Data, and So On
- Enter R

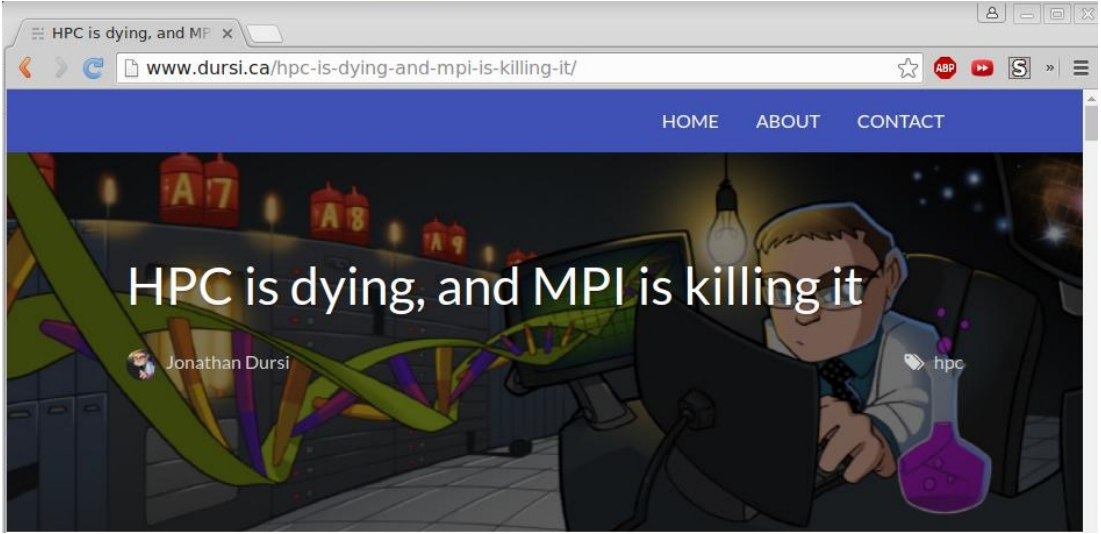
2 A Little History

- Statistics, Data Science, Big Data, and So On
- Enter R

HPC: Not Just for PDE'S Anymore!

- R's use in HPC.
- No traditional HPC...
- Lots of interesting work

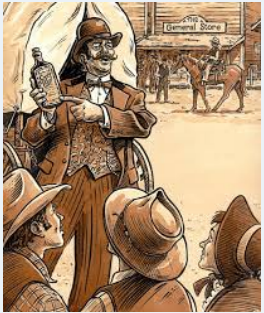
About Traditional HPC. . .



Changing Landscape of HPC

- "non-traditional" HPC: everybody but physics.
- What kind of software do they need?
- Can we leverage any existing HPC stuff?

Problems with "Big Data" Software



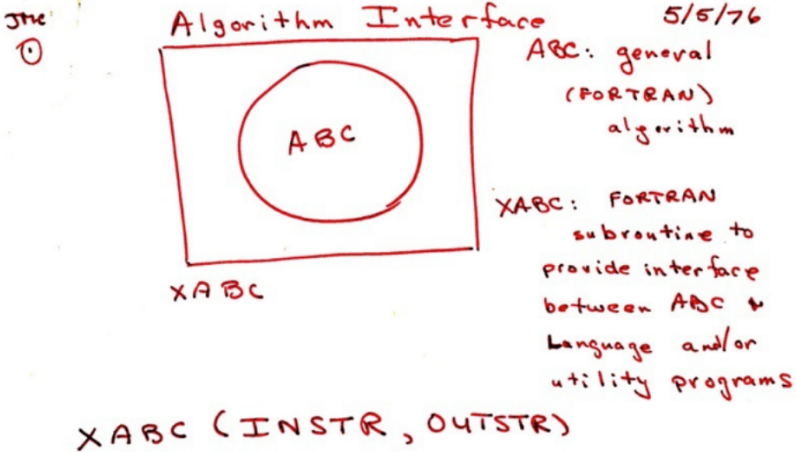
- *Many* frameworks; what do they all do?
- Don't always play nice with HPC systems.
- Often not as "high level" as advertised.
- Almost exclusively batch!

Data Analysis Is An Interactive Activity

Data analysis is an interactive activity[a]

[a] Data analysis is an interactive activity

Data science in action



http://datascience.la/john-chambers-user-2014-keynote/

3 Packages

- Advanced Compute Packages
- HPC Packages
- Hadoop and Applications
- Ok, So What?

Where to Begin?

- Many packages of varying scope and quality.
- 1 core package (parallel)
- Over 100 contributed packages
  https://cran.r-project.org/web/views/HighPerformanceComputing.html
- Even more on GitHub.

2015	future remoter
2014	
2013	
2012	pbdMPI pbdDMAT HiPLARM
2011	parallel
2010	
2009	gputools multicore foreach
2008	bigmemory Rcpp
2007	ff
2006	
2005	
2004	
2003	snow
2002	Rmpi

Out of Core Packages

- ff, bigmemory and friends
- R is very "copy happy"
- Many statisticians don't know about things like XSEDE.
- Others hear "Linux" and run away screaming.
- Bizarrely, cloud computing is changing this.

Rcpp

- Rcpp
- RcppArmadillo, RcppEigen
- RcppParallel
- . . .

Accelerator Packages

- gputools, Magma, HiPLARM, a few others.
- Accessibility mostly from things like nvblas and Intel MKL.

Distributed Packages

- Rmpi
- snow
- pbdMPI and friends

Remote Evaluation Packages

- rzmq, pbdZMQ
- remoter, future

Hadoop et al Packages

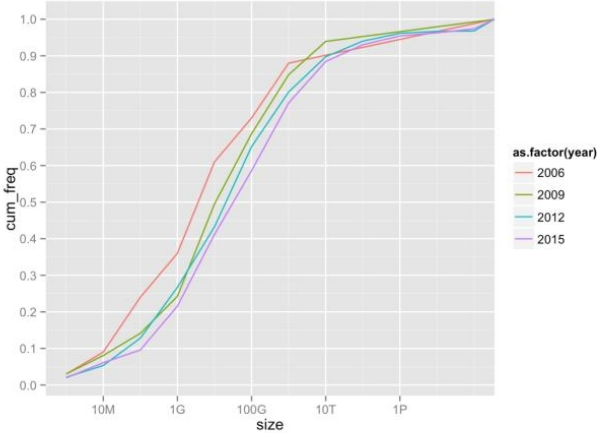
- RHadoop, RHIPE
- SparkR
- sparklyr
- h2o

"Applications"

- dplyr and data.table
- caret
- randomForest
- xgboost

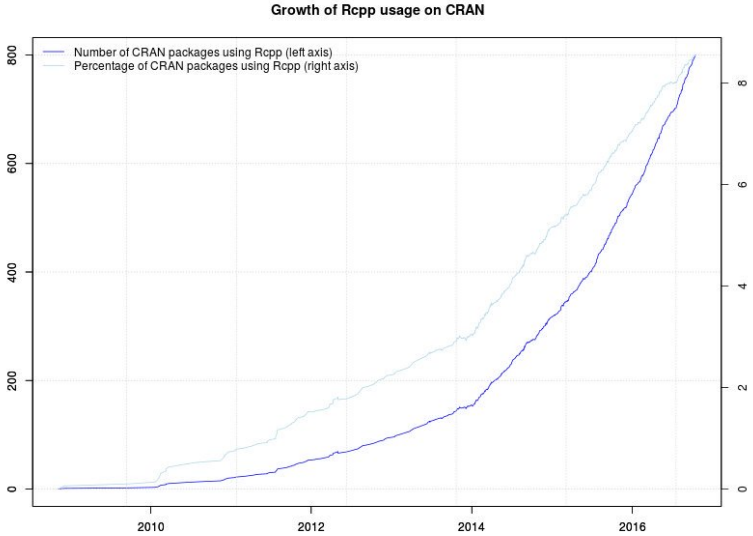
Is the R community using this stuff?

- Short answer: yes.
- Long answer: mostly single-node parallelism.
- Hard truth: in addition to hype and buzzwords — fear and distrust



BIG RAM IS EATING BIG DATA – SIZE OF DATASETS USED FOR ANALYTICS

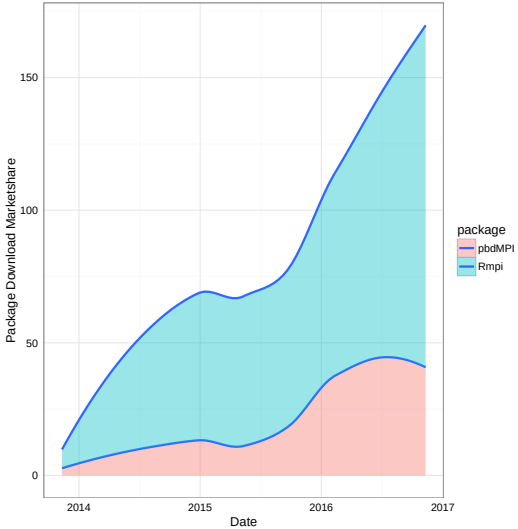
NOVEMBER 18, 2015 EDUCATION SZILARD PAFKA 1 COMMENT 3



Source https://twitter.com/eddelbuettel/status/787740983433854977

4 A Closer Look at HPC and R

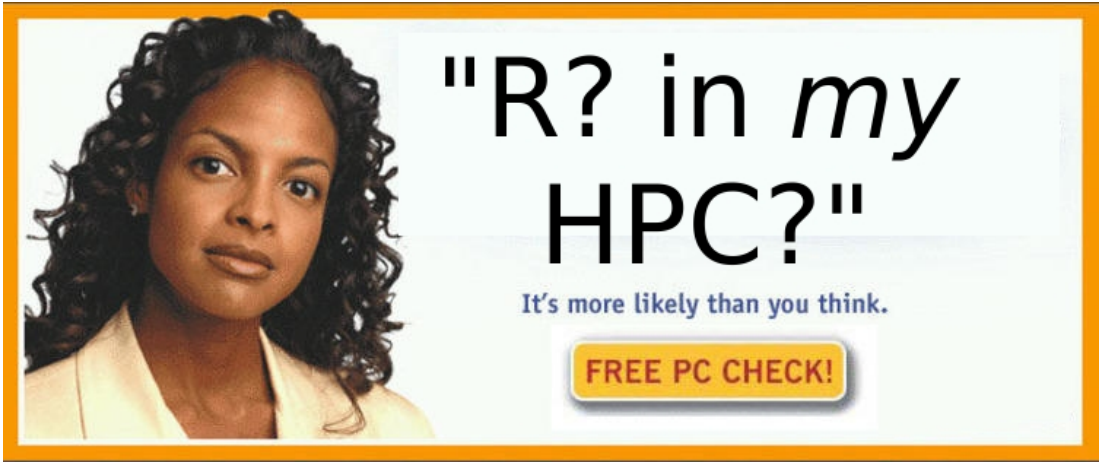
HPC may be dying, but we're behind the times

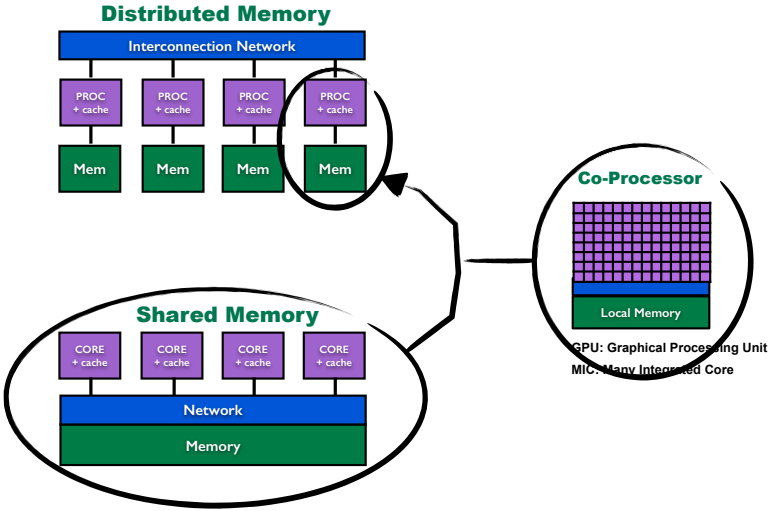


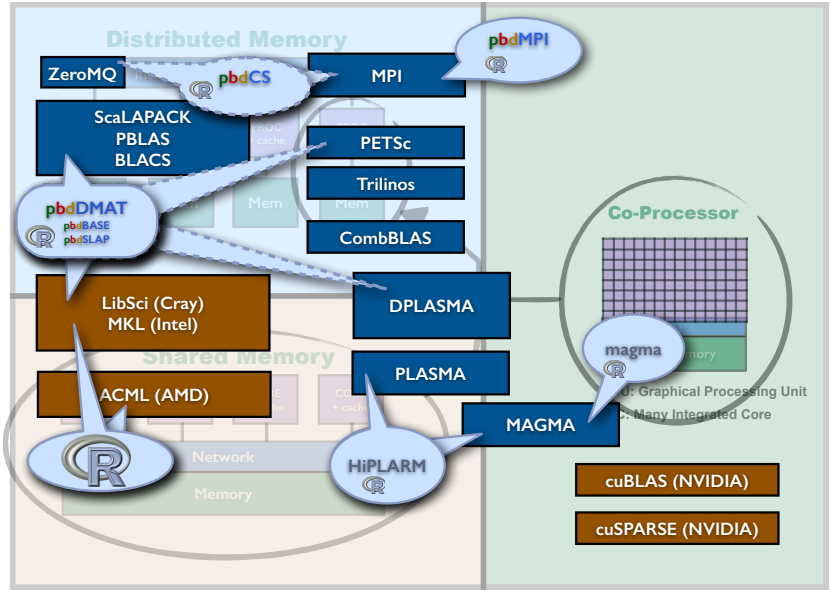
"OLCF Researchers Scale R to Tackle Big Science Data Sets"

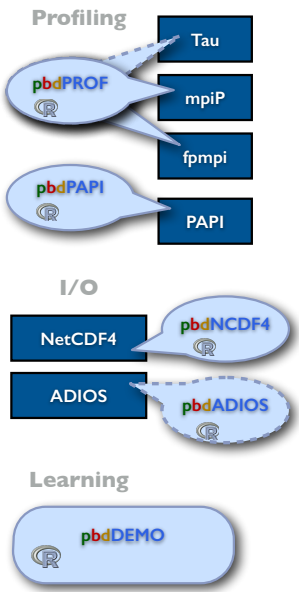
- A problem that takes several hours on Apache Spark [was analyzed] in less than a minute using R on OLCF high-performance hardware.
- "*. . . for situations where one needs interactive near-real-time analysis,* **the pbdR approach is much better**."

https://www.hpcwire.com/2016/07/06/olcf-researchers-scale-r-tackle-big-science-data-sets/


"R? in *my* HPC?"
It's more likely than you think.
FREE PC CHECK!


Distributed Memory
Interconnection Network
PROC + cache
PROC + cache
PROC + cache
PROC + cache
Mem
Mem
Mem
Mem
Co-Processor
Local Memory
GPU: Graphical Processing Unit
MIC: Many Integrated Core
Shared Memory
CORE + cache
CORE + cache
CORE + cache
CORE + cache
Network
Memory


Distributed Memory
ZeroMQ
pbdCS
MPI
pbdMPI
ScaLAPACK
PBLAS
BLACS
PETSc
Trilinos
CombBLAS
pbdDMAT
pbdBASE
pbdSLAP
Mem
Mem
LibSci (Cray)
MKL (Intel)
DPLASMA
Co-Processor
magma
Shared Memory
PLASMA
ACML (AMD)
MAGMA
Network
HiPLARM
Memory
cuBLAS (NVIDIA)
cuSPARSE (NVIDIA)
Released
Under Development


Profiling
Tau
pbdPROF
mpiP
fpmpi
pbdPAPI
PAPI
I/O
NetCDF4
pbdNCDF4
ADIOS
pbdADIOS
Learning
pbdDEMO

5 Concluding Remarks

The Future?

- Better dplyr backends.
- More threading + accelerator usage in packages (Rcpp + RcppParallel).
- Astronomical amounts of buzz in the Haddop/Spark-and-friends space — will ultimately hurt us in the MPI space.

∼Thanks!∼

Questions?

Email: wrathematics@gmail.com

GitHub: https://github.com/wrathematics

Web: http://wrathematics.info

Twitter: @wrathematics