Sup~~er~~R

Chuanhai Liu

Department of Statistics, Purdue University

Year of the Monkey 2016

Table of Contents

1. Introduction

Chuanhai Liu

Department of Statistics, Purdue University

2016

SupR: Why?

R is apparently one of the most popular statistical computing systems for data analysis by statisticians.

A user-friendly, efficient, and BigData-capable software like R is in great need.

SupR is intended to be built as a software that

- enables data analysts to do Big Data Analysis after their mastering “20” more (super) R functions, and
- is as efficient as possible.

SupR: How?

Build a big data computing system with

1. a R-style front-end by maintaining the existing R syntax and its internal basic data structures
2. a Java-like multithreading, which would be the key to the success of big data analysis
3. a Spark-like distributed/cluster computing
4. a built-in simple Distributed File System, which, to some extent, represents a kind of cluster-wide namespace

A private pre-release

While there is still much to do, a private pre-release is available at

`http:www.stat.purdue.edu/∼chuanhai/SupR`

The most important is perhaps the proof of concept.

SupR: Multithreading — A parallel EM example //SKIPME

- Start a single SupR session/process

```
$ SupR
...
Welcome to monkeyR, a pre-release version of SupR
>
```

- Start a graphics thread:

```
> new.thread(X11())
```

- Start a Maximization-thread:

```
> new.thread(..., start=TRUE)
```

- Start a pre-specified number of `n` Expectation-threads:

```
> for(i in 1:n) new.thread(..., start=TRUE)

# Watch the graphics output while waiting for the result

# Any student should be able to do this
```

SupR: Cluster computing — a distributed LR example //SKIP

- Start a master session on some node machine

```
$ SupR "-e master()"
```

- Start a worker session on each of selected node machines

```
$ SupR "-e worker()"
# Multiple workers on each node and multiple executors in each
# worker are allowed.
```

- Start a driver session on some node machine

```
$ SupR
...
> start.driver()
> distribute(...)  # create distributed data
> SS = map.reduce(...)  # compute suff.  stat.
> result = gauss.sweep(SS, ...)

# Any student should be able to do this
```

SupR: Some examples of software development

- Real data analysis for doing scicence: Develop tools for analyzing big data of complex structures
- mi package: a SupR package for handling missing data in big data problems
- im package for the best possible scientific inference (Martin and Liu, 2015)
- mlearn packages (machine-learning, deep-learning)
- pbayes packages (partially specified Bayes)
- Application-specific packages: you name it?

References

1. John M. Chambers (1998) *Programming with Data: A guide to the S language*, Springer, New York.
2. Holden Karau, Andy Konwinski, Patrick Wendell, and Matei Zharia (2015). *Learning Spark: Lightning-fast data analysis*, O'Reilly, Bejing.
3. Ryan Martin and Chuanhai Liu (2016) *Inferential Models: Reasoning with uncertainty*, Chapman & Hall, New York.
4. Hadley Wickham (2014) *Advanced R*, Chapman & Hall, New York.