

Introduction to Renjin

Zürich R meetup
February 6, 2017

Maarten-Jan Kallen
@mj_kallen

BeDataDriven B.V.
Netherlands





BEDATADRIVEN.COM

Introduction to Renjin

- What is Renjin?
- Why Renjin?
- Example of improving performance
- Compatibility with GNU R



What is Renjin?



Elements of the GNU R project

- John Chambers in *Software for data analysis* (2008):
 - the language (i.e. syntax)
 - the evaluator (i.e. interpreter)
 - the management of memory for objects (i.e. garbage collection)
- Plus:
 - the packaging system (i.e. libraries)

Elements of the Renjin project

- John Chambers in *Software for data analysis* (2008):
 - ~~the language (i.e. syntax)~~
 - the evaluator (i.e. interpreter)
 - ~~the management of memory for objects (i.e. garbage collection)~~
- Plus:
 - the packaging system (i.e. libraries)

How does Renjin differ from GNU R?

- the core of the interpreter is written in Java, not C or Fortran
- it uses Java's built-in garbage collector
- the packaging system is based on Maven's approach to artifact management, not a home-grown solution

Renjin is an alternative interpreter (“engine”) for the R programming language designed to run in the Java Virtual Machine (JVM).



Other interpreters

- **FastR** is also implemented in Java, but in its current version is built on top of *Graal* which is a “Polyglot Runtime for the JVM”
- **Rho** (formerly known as CXXR) is a rewrite in C++ of the GNU R interpreter
- **TERR** (“TIBCO Enterprise Runtime for R”) is a closed-source implementation in C++
- **pqR** (“pretty quick R”) is a fork of GNU R

Why Renjin?



Why did we create Renjin?

- compile everything to pure Java bytecode to run an R interpreter in the JVM of a Platform-as-a-Service provider such as Google App Engine or Microsoft Azure App Service
- improve performance for specific use-cases where GNU R ran out of memory
- because it's fun

Some other goals with Renjin

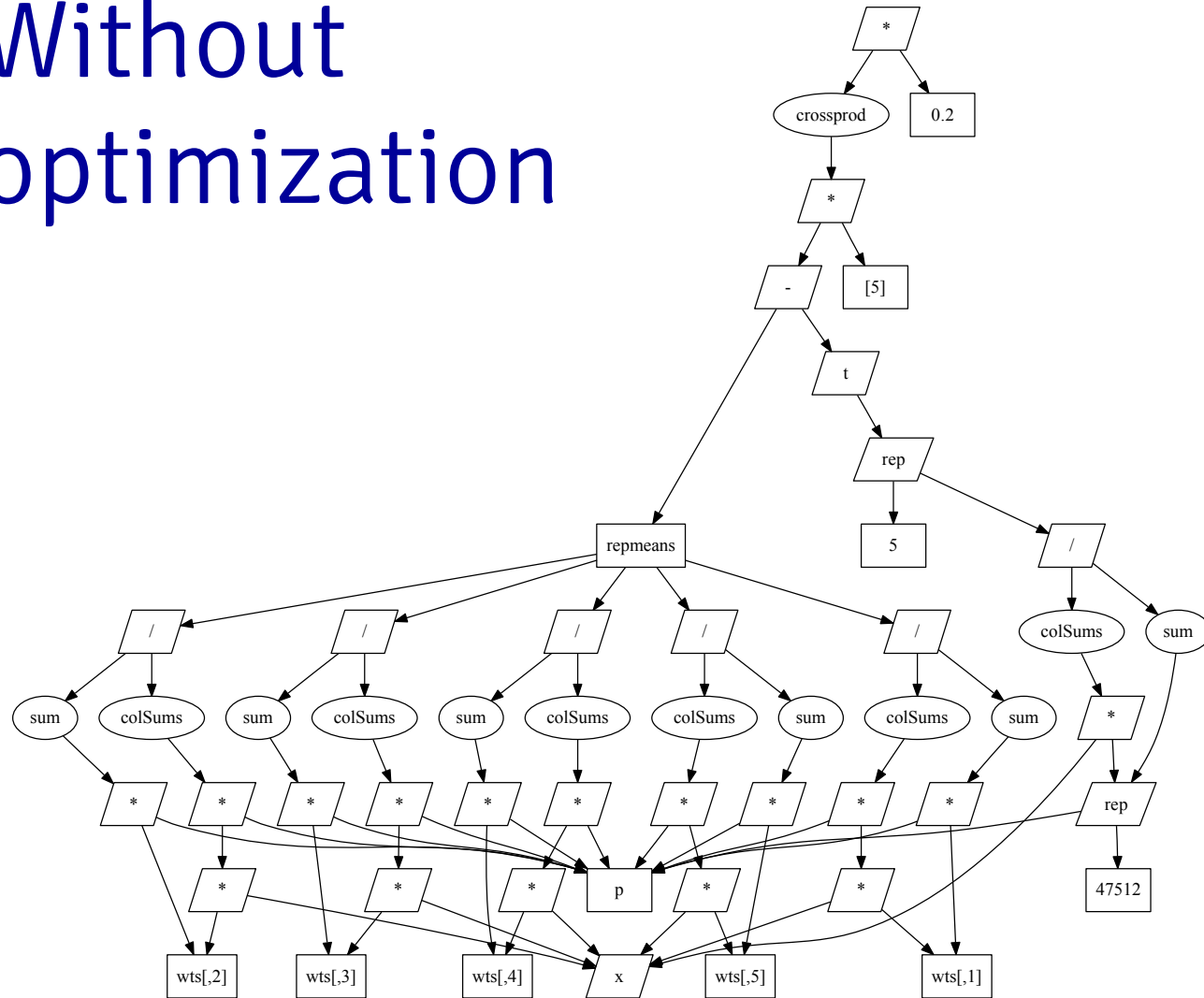
- better performance without the need to rewrite your R code using
 - packages such as *data.table*, *bigmemory*, *sqldf* and others, and
 - additional functions such as *anyNA* which tend to solve only one particular problem
- better integration with enterprise-class tooling and systems
- better dependency management

Two examples of performance improvement using deferred evaluation

Example 1: R as a Query Language

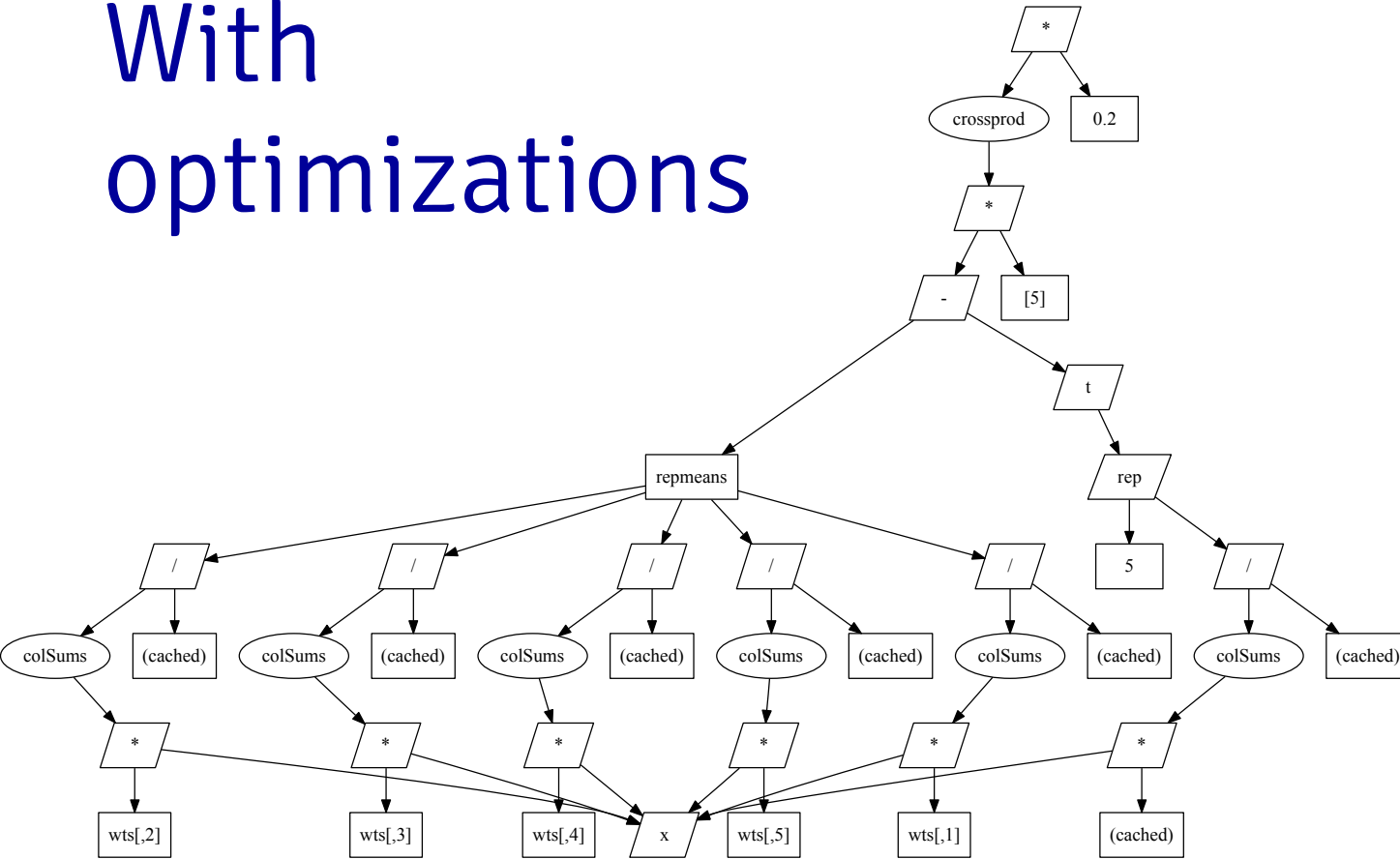
- research by Hannes Mühleisen to consider an R program as a declaration of intent, much like a SQL statement
- apply common database query optimizations to the execution graph of an R program
- case study using the American Community Survey data with the *survey* package

Without optimization



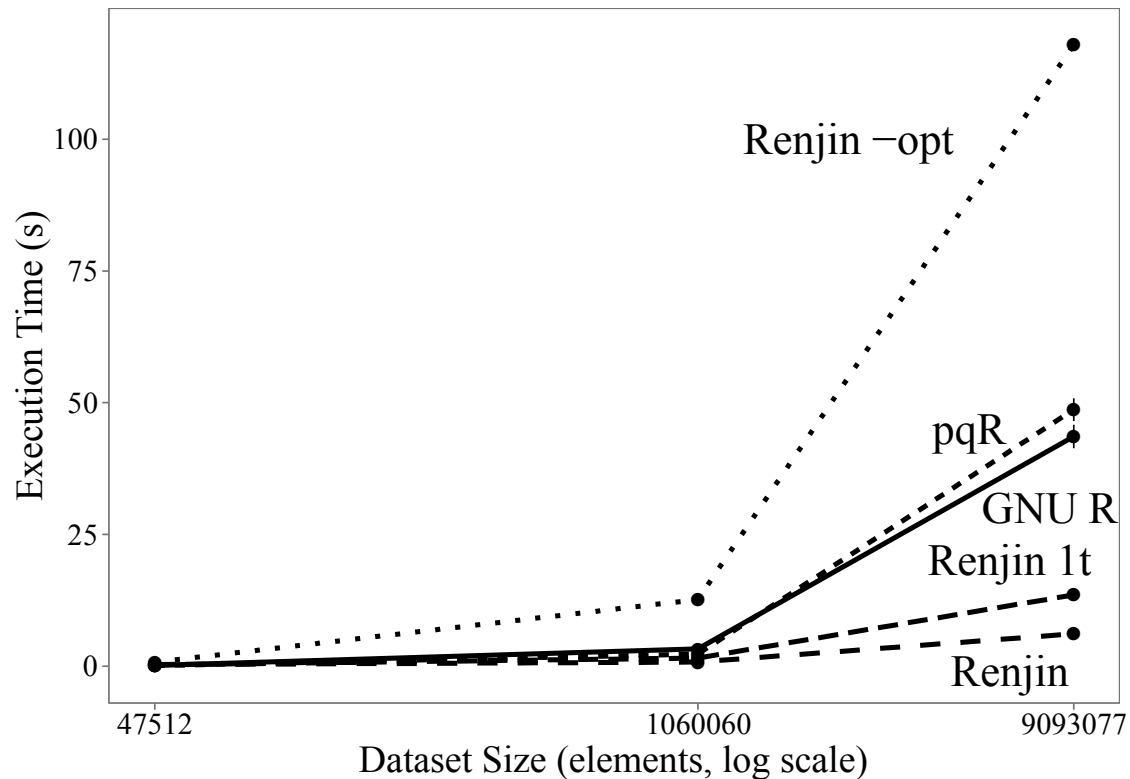
Execution graph analyzed by the interpreter through deferred evaluation.

With optimizations



- Optimizations:
- Selection push-down
 - Expression result caching
 - Identity removal
 - Vectorized/specialized operators
 - Parallel execution using worker threads

Case study results



R becomes much faster with the optimizations

Example 2: anyNA(x)

- `anyNA()` was introduced in R 3.0 as `any(is.na(x))` is inefficient and slow for large `x`
- GNU R solution: mash into one function, farm implementation out to C
- But: introduces yet another oddly named function (`paste0` anyone?) and doesn't solve similar cases:
 - `Y <- is.na(x); any(y)`
 - `any(is.na(x) | is.na(y))`
 - `all(!is.na(x))`

anyNA(x) in Renjin

```
anyNA.default <- function(x, recursive = FALSE) {  
  if (isTRUE(recursive)) x <- unlist(x)  
  any(is.na(x))  
}
```

- due to deferred evaluation, `is.na(x)` is never ‘materialized’ and `any(y)` is aware that `y` is the result of `is.na(x)`, not just any logical vector.
- all cases are automatically optimized.

Compatibility with GNU R

What about compatibility?

- Renjin is close to 100% compatible with R's base, stats and methods packages
- no support for graphics
- all tests in about 25% of all CRAN packages pass in Renjin, close to 50% of packages have at least one test passing
- you can check the compatibility of your favorite package(s) at <http://packages.renjin.org>
- no integration with RStudio

To finish

- visit the project website at renjin.org and sign up to receive the Renjin newsletter
- follow me on Twitter: twitter.com/mj_kallen
- send us an email at info@renjin.org