

# 突破 R 内存瓶颈的若干技术

寇强

kouqiang@mail2.sysu.edu.cn

华南统计科学研究中心

2014 年 5 月 25 日



# 自我介绍

- 寇强
- 微博: @Gossip\_useR
- 华南统计科学研究中心成员
- 信息学博士在读 (博一), 串联质谱的数据分析和软件开发
- 生物本科, 大三开始自学编程

# 干货时间

# 相关技术和实现

- Hashing
  - ▶ **digest**: create cryptographic hash digests of R objects
- Memoization
  - ▶ **memoise**: memoise functions
  - ▶ **R.cache**: fast and light-weight memorization of objects and results
- Key/value storage
  - ▶ **stashR**: a set of tools for administering shared repositories
  - ▶ **filehash**: simple key-value database
  - ▶ **cacher**: tools for caching and distributing statistical analyses
- Disk
  - ▶ **bigmemory**: shared memory and memory-mapped files
  - ▶ **SOAR**: memory management in R by delayed assignments
  - ▶ **ff**: memory-efficient storage of large data on disk and fast access
  - ▶ **mvbutils**: workspace organization, code and documentation editing
  - ▶ **track**: track objects
- Database
  - ▶ **dplyr**: a grammar of data manipulation
  - ▶ **SciDB**: R interface to SciDB

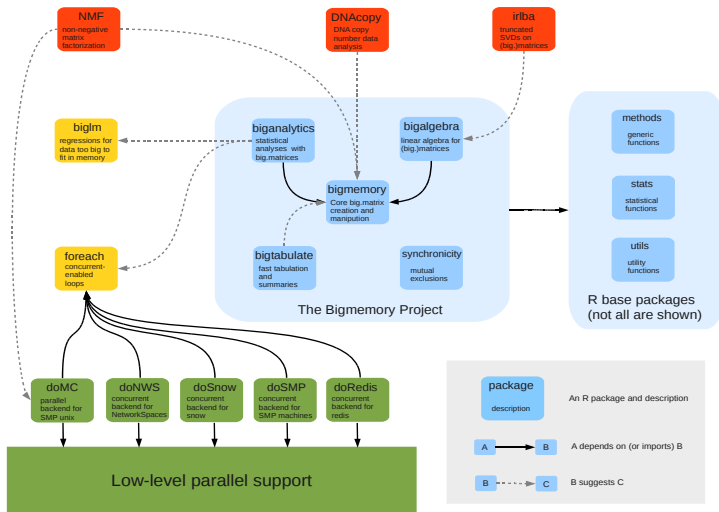
# 相关技术和实现

- Hashing
  - ▶ **digest**: create cryptographic hash digests of R objects
- Memoization
  - ▶ **memoise**: memoise functions
  - ▶ **R.cache**: fast and light-weight memorization of objects and results
- Key/value storage
  - ▶ **stashR**: a set of tools for administering shared repositories
  - ▶ **filehash**: simple key-value database
  - ▶ **cacher**: tools for caching and distributing statistical analyses
- Disk
  - ▶ **bigmemory**: shared memory and memory-mapped files
  - ▶ **SOAR**: memory management in R by delayed assignments
  - ▶ **ff**: memory-efficient storage of large data on disk and fast access
  - ▶ **mvbutils**: workspace organization, code and documentation editing
  - ▶ **track**: track objects
- Database
  - ▶ **dplyr**: a grammar of data manipulation
  - ▶ **SciDB**: R interface to SciDB

# bigmemory

- 一系列扩展包:  
**bigmemory**, **bigmemory.sri**, **bigtabulate**, **biganalytics**,  
**synchronicity**, **bigalgebra**
- 2010 年 John M Chambers Statistical Software Award
- <http://www.bigmemory.org/>
- 核心数据结构由 C++ 实现

# The bigmemory Project



<http://www.bigmemory.org/>

# bigmemory

```
A <- matrix(data = 0, 5000, 5000)
x <- sample(1:5000, size = 5000, replace = TRUE)
y <- sample(1:5000, size = 5000, replace = TRUE)
for (i in 1:5000) {
  A[x[i], y[i]] <- 1
}
library(bigmemory)
library(biganalytics)
options(bigmemory.typecast.warning = FALSE)
B <- big.matrix(5000, 5000, init = 0)
for (i in 1:5000) {
  B[x[i], y[i]] <- 1
}
B

## An object of class "big.matrix"
## Slot "address":
## <pointer: 0x26a5690>
```

```
typeof(B)
```

```
## [1] "double"
```

```
class(B)
```

```
## [1] "big.matrix"
```

```
## attr(,"package")
```

```
## [1] "bigmemory"
```

```
lsos()
```

##	Type	Size	Rows	Columns
## A	matrix	200000200	5000	5000
## x	integer	20040	5000	NA
## y	integer	20040	5000	NA
## B	big.matrix	664	5000	5000

# 内存共享

```
desc <- describe(B)
dput(desc, file = "B.big.matrix")
colsum(B, 1:10)

## [1] 0 1 0 2 1 1 2 0 0 0
```

另一个 R 会话中：

```
dput(desC, file = "B.big.matrix")
C <- attach.big.matrix(desC)
colsum(C, 1:10)

## [1] 0 1 0 2 1 1 2 0 0 0
```

```
desc

## An object of class "big.matrix.descriptor"
## Slot "description":
## $sharedType
## [1] "SharedMemory"
##
## $sharedName
## [1] "e675ab11-0431-46c2-9024-6e33564974ec"
##
## $totalRows
## [1] 5000
##
## $totalCols
## [1] 5000
##
## $rowOffset
```

```
library(bigmemory)
library(biganalytics)
D <- filebacked.big.matrix(5000, 5000, init = 0,
  backingfile = "matrix.example",
  descriptorfile = "matrix.example.desc")
D
```

```
## An object of class "big.matrix"
## Slot "address":
## <pointer: 0x3534e00>
```

```
list.files()
```

```
## [1] "matrix.example"
## [2] "matrix.example.desc"
```

## 12 GB 航空数据读入

```
x <- read.big.matrix("airline.csv", header = TRUE,  
  backingfile = "airline.bin",  
  descriptorfile = "airline.desc",  
  type = "integer")
```

- **read.table()**

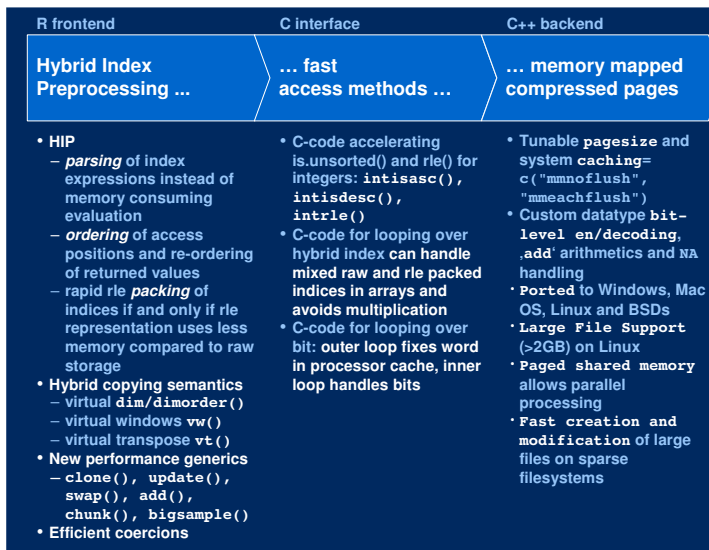
- ▶ Memory overhead
- ▶ 不能内存共享
- ▶ Data frame

- **read.big.matrix()**

- ▶ 几乎没有 memory overhead
- ▶ 内存共享
- ▶ 只能是矩阵

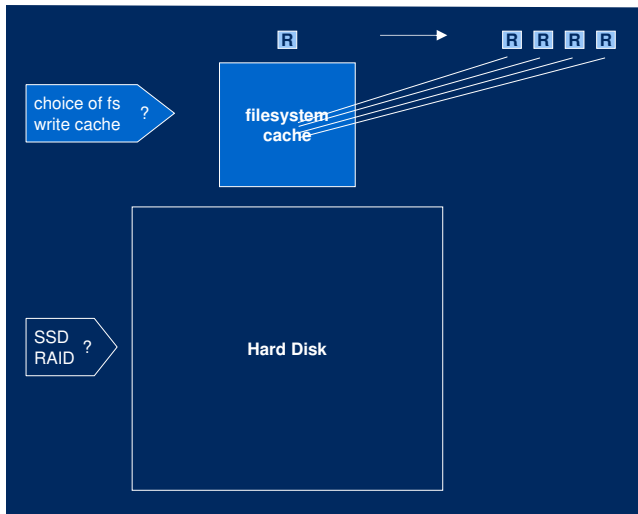
ff

- “fast access files”
- ffd: **ff** 中的 data frame
- **ffbase**: Basic statistical functions for package **ff**
- binary files
- <http://r-forge.r-project.org/projects/ff/>



```
library(ff)
ff.obj1 <- ff(vmode = "double", length = 100)
## "ff2dde53acb8c9.ff" in "/tmp/RtmpZ5qpC2"
R.obj1 <- double(100)
ff.obj2 <- ff(vmode = "double", length = 10000)
R.obj2 <- double(10000)
lsos()
```

```
##           Type  Size  Rows Columns
## R.obj2    numeric 80040 10000      NA
## ff.obj1 ff_vector  2904   100      NA
## ff.obj2 ff_vector  2904 10000      NA
## R.obj1    numeric   840   100      NA
```



Oehlschlagel (2010) Managing large datasets in R – ff examples and concepts

# dplyr

- A grammar of data manipulation
- Hadley Wickham, Romain Franco is
- 语法简洁
- 数据库支持: sqlite, mysql, postgresql, Google's bigquery
- 相关扩展包:  
**RMySQL, ROracle, RJDBC, biglm, RSQLite, DBI**

## 循环版:

```
models <- as.list(rep(NA, 24 *
  24))
dim(models) <- c(24, 24)
deseas <- array(NA, c(24, 24, 72))
dimnames(deseas) <- dimnames(ozone)
for (i in seq_len(24)) {
  for (j in seq_len(24)) {
    mod <- deseasf(ozone[i,
      j, ])
    models[[i, j]] <- mod
    deseas[i, j, ] <- resid(mod)
  }
}
```

## 非循环版:

```
models <- apply(ozone, 1:2, deseasf)
resids_list <- lapply(models, resid)
resids <- unlist(resids_list)
dim(resids) <- c(72, 24, 24)
deseas <- aperm(resids, c(2, 3,
  1))
dimnames(deseas) <- dimnames(ozone)
```

## dplyr 版:

```
models <- dlply(ozonedf, .(lat,
  long), deseasf_df)
deseas <- ldply(models, resid)
```

来自 <http://www.loyhome.com>

# dplyr 操作数据库

```
library(dplyr)

my_db <- src_sqlite("my_db.sqlite3", create = T)
list.files()

## [1] "my_db.sqlite3"

data("hflights", package = "hflights")
hflights_sqlite <- copy_to(my_db, hflights, temporary = FALSE,
  indexes = list(c("Year", "Month", "DayofMonth"),
    "UniqueCarrier", "TailNum"))

hflights_sqlite()

## Caching hflights db at /tmp/RtmpvVHzrI/hflights.sqlite

## src:  sqlite 3.7.17 [/tmp/RtmpvVHzrI/hflights.sqlite]
## tbls: hflights, sqlite_stat1
```

# dplyr 操作数据库

```
hflights_sqlite <- tbl(hflights_sqlite(), "hflights")
hflights_sqlite

## Source: sqlite 3.7.17 [/tmp/RtmpvVHrI/hflights.sqlite]
## From: hflights [227,496 x 21]
##
##   Year Month DayOfMonth DayOfWeek DepTime
## 1  2011     1           1           6    1400
## 2  2011     1           2           7    1401
## 3  2011     1           3           1    1352
## .. ... ..
## Variables not shown: ArrTime (int), UniqueCarrier
##   (chr), FlightNum (int), TailNum (chr),
##   ActualElapsedTime (int), AirTime (int),
##   ArrDelay (int), DepDelay (int), Origin (chr),
##   Dest (chr), Distance (int), TaxiIn (int),
##   TaxiOut (int), Cancelled (int),
##   CancellationCode (chr), Diverted (int)
```

<http://cran.rstudio.com/web/packages/dplyr/vignettes/databases.html>

# SciDB

- An array database
- Michael Ralph Stonebraker
- Designed for scientific analysis and data management
- Start in March 2008, latest version 14.3
- Open-source
- User-friendly interface to R and Python
- Cases: 1000 genomes browser



Photo from Wikipedia

# 在 R 中使用 SciDB 数据

```
library("scidb")
scidbconnect()
mRNA <- scidb("mRNA")
mRNA

## SciDB 1-D array: 600 obs. of 109 variables.

mRNA[] [1:7, 1:5]

##      TCGA_barcode COL10A1  BUB1  ATAD2  CENPA
## 0 TCGA-04-1331      5.786 4.260 5.173 6.565
## 1 TCGA-04-1332      8.612 4.336 6.107 5.957
## 2 TCGA-04-1335      4.102 3.829 4.901 5.095
## 3 TCGA-04-1336      3.799 3.631 6.915 5.413
## 4 TCGA-04-1337      7.784 3.489 4.122 4.291
## 5 TCGA-04-1338      6.212 3.995 6.248 5.722
## 6 TCGA-04-1341      6.101 4.343 6.742 6.284
```

# 将 R 对象映射到 SciDB

```
data(lung)
A = lung[1:2000, ]
library(sciadb)
sciadbconnect()
X = as.sciadb(A)
X

## A reference to a 2000x56 SciDB array

t(X)

## A reference to a 56x2000 SciDB array

sum(X)

## [1] 6.997e-13
```

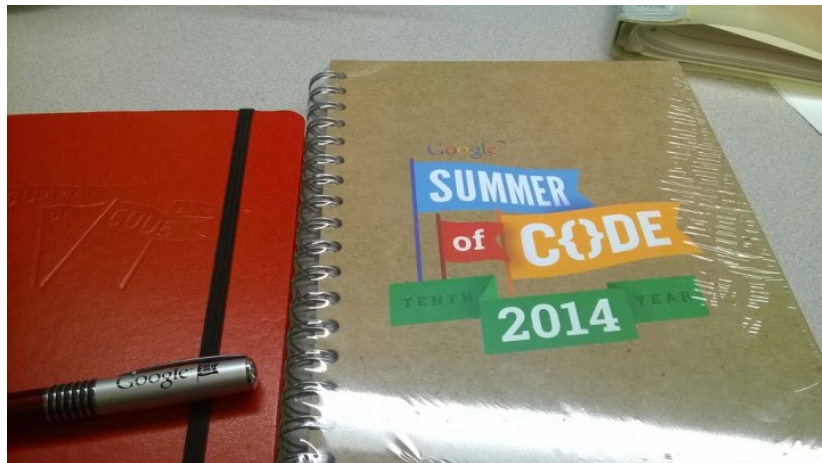
# 在 SciDB 中使用 R 函数

```
store(  
  r_exec(build(<z:double>[i=1:1000,10,0],0),  
    'expr=x<-runif(1000);  
      y<-runif(1000);list(sum(x^2+y^2<1)/250)');  
test);  
  
{inst,n} expr_value_0  
{0,0} 3.172  
{0,1} 3.144  
{0,2} 3.156  
.....  
  
r_exec(apply(test,t,double(expr_value_0)),  
  'expr=list(sqrt(mean(t)))');  
  
{inst,n} expr_value_0  
{0,0} 1.7732  
{1,0} 1.77374
```

- **bigmemory**: 获奖作品，值得信赖，只有矩阵
- **ff**: data frame 支持，缺乏良好文档
- **dplyr**: 大师出品，语法简洁
- **SciDB**: 相对于企业版，社区版严重缩水，glm 都只在企业版里

# 吹水时间

# Google Summer of Code





- MLPACK: a scalable c++ machine learning library
- Armadillo for matrix input and output
- <http://www.mlpack.org/index.html>
- R binding? RcppMLPACK?



OPEN SOURCE

CONTRIBUTE

RESOURCES

GRAPHLAB.COM

## GraphChi

OPEN SOURCE

Disk-based large-scale graph computation

- GraphChi: Disk-based large-scale graph computation
- <http://graphlab.org/projects/graphchi.html>
- R binding? RcppGraphChi?

Application	Input graph	Graph size	Comparison	GraphChi on Mac Mini (SSD)	Ref
Pagerank - 3 iterations	twitter-2010	1.5B edges	Spark, 50 machines, 8.1 min	13 min	1
Pagerank - 100 iterations	uk-union	3.8B edges	Stanford GPS (Pregel), 30 machines, 144 min	581 min	2
Web-graph Belief Propagation (1 iter.)	yahoo-web	6.7B edges	Pegasus, 100 machines, 22 min	27 min	3
Matrix factorization (ALS), 10 iters	Netflix	99M edges	GraphLab, 8-core machine, 4.7 min	9.8 min	4
Triangle counting	twitter-2010	1.5B edges	Hadoop, 1636 machines, 423 mins	55 min	5

<https://github.com/GraphChi/graphchi-cpp/blob/master/README.md>

## pqR - a pretty quick version of R

pqR is a new version of the R interpreter. It is based on R-2.15.0, distributed by the R Core Team (at [r-project.org](http://r-project.org)), but improves on it in many ways, mostly ways that speed it up, but also by implementing some new features and fixing some bugs.

One notable improvement is that pqR is able to do some numeric computations in parallel with each other, and with other operations of the interpreter, on systems with multiple processors or processor cores.

**The current stable release: pqR-2014-02-23**

- <http://www.pqr-project.org/>
- <https://github.com/radfordneal/pqR>
- 升级到和 3.1.0 兼容?

## GPUMLib 0.2.1

GPU Machine Learning Library

Main Page	Modules	Classes	Files	Examples	
-----------	---------	---------	-------	----------	--

- <http://gpumlib.sourceforge.net/>
- R binding?

## Thrust: Code at the speed of light

Thrust is a parallel algorithms library which resembles the C++ Standard Template Library (STL). Thrust's **high-level** interface greatly enhances programmer **productivity** while enabling performance portability between GPUs and multicore CPUs. **Interoperability** with established technologies (such as CUDA, TBB, and OpenMP) facilitates integration with existing software. Develop **high-performance** applications rapidly with Thrust!

- <https://github.com/thrust/thrust>

## DMTCP: Distributed MultiThreaded CheckPointing

[Home](#)[Downloads](#)[FAQ](#)[SF project page](#)[Browse Source](#)[Demo](#)[Supported Apps](#)[Parallel Computing](#)

### About DMTCP:

DMTCP (Distributed MultiThreaded Checkpointing) is a tool to transparently checkpoint the state of multiple simultaneous applications, including multi-threaded and distributed applications. It operates directly on the user binary executable, without any Linux kernel modules or other kernel modifications.

Among the applications supported by DMTCP are Open MPI, MATLAB, Python, Perl, and many programming languages and shell scripting languages. Starting with release 1.2.0, DMTCP also supports [GNU screen](#) sessions, including vim/cscope and emacs. With the use of TightVNC, it can also checkpoint and restart X Window applications, as long as they do not use extensions (e.g.: no OpenGL, no video). See the [QUICK-START](#) file for further details.

- <http://dmtcp.sourceforge.net/>
- 急需测试，尤其是分布式系统下的 R

# DMTCP in python

```
import dmtcp
...
def my_ckpt(<args>):
    # Pre processing
    my_pre_ckpt_hook(<args>)
    ...
    # Create checkpoint
    dmtcp.checkpoint()
    # Checkpoint image has been created
    ...
    if dmtcp.isResume():
        # The process is resuming from a checkpoint
        my_resume_hook(<args>)
        ...
    else:
        # The process is restarting from a previous
```

## **akima: Interpolation of irregularly spaced data**

Linear or cubic spline interpolation for irregular gridded data

Version: 0.5-11  
Depends: R ( $\geq 2.0.0$ )  
Published: 2013-09-16  
Author: Fortran code by H. Akima R port by Albrecht Gebhardt aspline function by Thomas Petzoldt  
interp2xyz, enhancements and corrections by Martin Maechler  
Maintainer: Albrecht Gebhardt <albrecht.gebhardt@uni-klu.ac.at>  
License: [ACM](#) | file [LICENSE](#)  
NeedsCompilation: yes  
Materials: [README](#) [ChangeLog](#)  
In views: [NumericalMathematics](#), [Spatial](#)  
CRAN checks: [akima results](#)

# R bug tracking system

**Bugzilla – Main Page**

Home | New | Browse | Search |  **Search** [?] | Reports | New Account | Log In | Forgot Password

## Welcome to the R bug tracking system

- [Enter a new bug report](#)
- [Show open bugs new-to-old](#)
- [Search existing bug reports](#)



[File a Bug](#)[Search](#)[Open a New Account](#)

**Quick Search**

[Quick Search help](#) | [Install the Quick Search plugin](#)

[Bugzilla User's Guide](#) | [Release Notes](#)

Thank you!

Thank you!

寇强

kouqiang@mail2.sysu.edu.cn

华南统计科学研究中心