

粉丝日志

跨界的IT博客|Hadoop家族, R, RHadoop, Nodejs, AngularJS, NoSQL, IT金融

R语言实践 » OpenBlas让R的矩阵计算加速

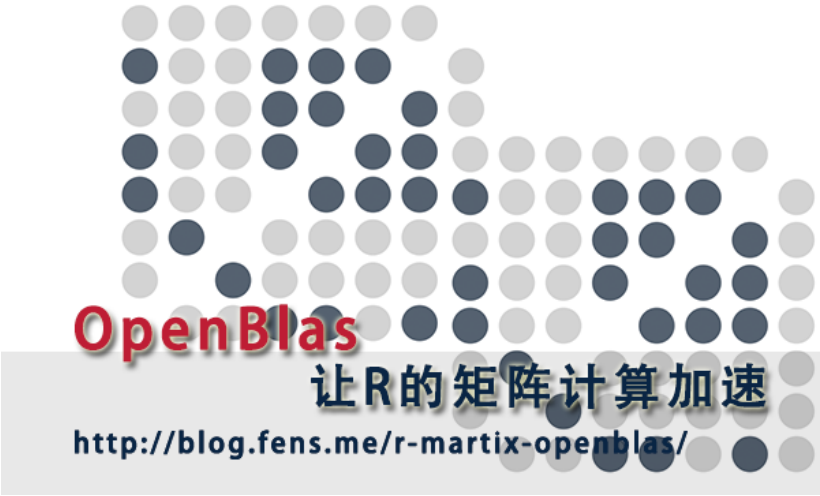
@晒粉丝 @每日中国天气

OpenBlas让R的矩阵计算加速

**Posted:** [R的极客理想系列文章](#), 涵盖了R的思想, 使用, 工具, 创新等的一系列要点, 以我个人的学习和体验去诠释R的强大。  
**Tags:** blas, cpu, matrix, openBlas, R  
**Comments:** 2 Comments

关于作者:  
张丹(Conan), 程序员Java,R,JavaScript  
weibo: @Conan\_Z  
blog: <http://blog.fens.me>  
email: bsspirit@gmail.com

转载请注明出处:  
<http://blog.fens.me/r-martix-openblas/>



前言

昨天在IBM的大会上, 又一次见到了OpenBlas主题的分享, 这次必须要试一下。我第一次了解OpenBlas是在2年前的R语言大会上, 听到了OpenBlas的各种优势, 不过听完也就过去了。2年后再来这个项目, 团队成员稳定, 项目进展顺利, 已经广泛接受, 再不尝试一下就真的落伍了。

目录

- 1. OpenBlas介绍
- 2. R和OpenBlas的安装
- 3. 让R语言加速

1. OpenBlas介绍

OpenBlas是一个开源项目, 是由 中科院软件所并行软件与计算科学实验室 发起的基于GotoBLAS2 1.13 BSD版的开源BLAS库高性能实现。

BLAS (Basic Linear Algebra Subprograms 基础线性代数程序集) 是一个应用程序接口 (API) 标准, 用以规范发布基础性代数操作的数值库 (如矢量或矩阵乘法)。该程序集最初发布于1979年, 并用于建立更大的数值程序包 (如LAPACK)。在高性能计算领域, BLAS被广泛使用。例如, LINPACK的运算成绩则很大程度上取决于BLAS中子程序DGEMM的表现。为提高性能, 各软硬件厂商则针对其产品对BLAS接口实现进行高度优化。

项目主页: <http://www.openblas.net/>

站内导航

- [R的极客理想系列文章](#)
- [从零开始nodejs系列文章](#)
- [用IT技术玩金融系列文章](#)
- [跨界知识聚会系列文章](#)
- [Hadoop家族系列文章](#)
- [AngularJS体验式编程系列文章](#)
- [RHadoop实践系列文章](#)
- [无所不能的Java系列文章](#)
- [ubuntu实用工具系列文章](#)
- [R利剑NoSQL系列文章](#)
- [MongoDB部署实验系列文章](#)
- [让Hadoop跑在云端系列文章](#)
- [自己搭建VPS系列文章](#)
- [架构师的信仰系列文章](#)
- [算法为王系列文章](#)
- [我的博客我的SEO系列文章](#)
- [创造可视化系列文章](#)
- [创业者的困境系列文章](#)
- [写作计划列表](#)
- [关于站长](#)
- [投放广告](#)

最新评论

- imshuai on [关于站长](#)
- Dan Zhang on [Nginx反向代理Websocket](#)
- Ruhi varma on [关于站长](#)
- 流民 on [Nginx反向代理Websocket](#)
- 流民 on [Nginx反向代理Websocket](#)
- Dan Zhang on [图书出版《R的极客理想-量化投资篇》](#)
- @yo Lian Hu, ENG on [图书出版《R的极客理想-量化投资篇》](#)
- 2017WOT全球软件开发技术峰会:面向数据的思维模式和R语言编程 | 粉丝日志 on [用R语言把数据玩出花样](#)
- 2017WOT全球软件开发技术峰会:面向数据的思维模式和R语言编程 | 粉丝日志 on [跨界知识聚会系列文章](#)
- 用R语言把数据玩出花样 | 粉丝日志 on [R语言解读一元线性回归模型](#)
- liheng peng on [Socket.io在聊天室](#)
- jiangsheng on [R语言构建配对交易量化模型](#)
- jiangsheng on [R语言构建配对交易量化模型](#)
- yz4d2 on [Socket.io在聊天室](#)
- 图书出版《R的极客理想-量化投资篇》 |

2. R和OpenBlas的安装

OpenBlas可以为各种语言底层，提供矩阵计算的性能提升，那么让我们把R和OpenBlas结合试试吧！

本机的系统环境：

Linux Ubuntu 14.01.1  
CPU 双核 Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz  
内存 4G

通过命令查看系统参数

```
# 操作系统
~ cat /etc/issue
Ubuntu 14.04.1 LTS \n \l

# CPU
cat /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 62
model name     : Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz
stepping      : 4
microcode     : 0x428
cpu MHz        : 2600.048
cache size     : 20480 KB
physical id    : 0
siblings       : 2
core id        : 0
cpu cores      : 2
apicid         : 0
initial apicid : 0
fpu            : yes
fpu_exception  : yes
cpuid level    : 13
wp             : yes
flags          : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat clflush mmx fxsr s
bogomips       : 5200.09
clflush size   : 64
cache_alignme  : 64
address sizes  : 46 bits physical, 48 bits virtual
power managem  :

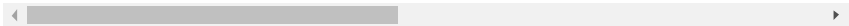
processor       : 1
vendor_id      : GenuineIntel
cpu family     : 6
model          : 62
model name     : Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz
stepping      : 4
microcode     : 0x428
cpu MHz        : 2600.048
cache size     : 20480 KB
physical id    : 0
siblings       : 2
core id        : 1
cpu cores      : 2
apicid         : 2
initial apicid : 2
fpu            : yes
fpu_exception  : yes
cpuid level    : 13
wp             : yes
flags          : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat clflush mmx fxsr s
bogomips       : 5200.09
clflush size   : 64
cache_alignme  : 64
address sizes  : 46 bits physical, 48 bits virtual
power managem  :

# 内存
~ cat /proc/meminfo
MemTotal:      4046820 kB
MemFree:       1572372 kB
Buffers:       40588 kB
```

最新文章

- 2017WOT全球软件开发技术峰会:面向数据的思维模式和R语言编程
- 图书出版《R的极客理想-量化投资篇》
- 2017微软技术暨生态大会:R语言搭建多因子体系
- 回测好，为什么实盘不靠谱？
- R语言数据科学新类型tibble
- 用R语言把数据玩出花样
- 新一代Node.js的Web开发框架Koa2
- 在Ubuntu上安装IPSEC VPN服务
- 2017CDAS中国数据分析师行业峰会:用R语言解读股利贴现模型
- 2017微软MVP:在AWS上部署免费的Shiny应用
- 在AWS上部署免费的Shiny应用
- 2017河北民族师范学院:大数据时代的变革
- 2017猎奇金融大数据:用R语言开始量化投资
- 用R语言开始量化投资
- 技术大牛如何寻找下一个风口
- 算法，如何改变命运
- Angular2新的体验
- R语言解读资本资产定价模型CAPM
- 用数据解读摩羯智投
- 2016中国软件技术大会:用R语言进行投资组合管理

```
Cached:          709684 kB
SwapCached:      0 kB
Active:          1953940 kB
Inactive:        418084 kB
Active(anon):    1621840 kB
Inactive(anon):  5732 kB
Active(file):    332100 kB
Inactive(file):  412352 kB
Unevictable:     0 kB
Mlocked:         0 kB
SwapTotal:       0 kB
SwapFree:        0 kB
Dirty:          24 kB
Writeback:       0 kB
AnonPages:      1623792 kB
Mapped:         34936 kB
Shmem:          5828 kB
Slab:           58024 kB
SReclaimable:   45252 kB
SUnreclaim:     12772 kB
KernelStack:    1512 kB
PageTables:     8980 kB
NFS_Unstable:   0 kB
Bounce:         0 kB
WritebackTmp:   0 kB
CommitLimit:    2023408 kB
Committed_AS:   2556460 kB
VmallocTotal:   34359738367 kB
VmallocUsed:     9664 kB
VmallocChunk:   34359723308 kB
HardwareCorrupted: 0 kB
AnonHugePages:  1562624 kB
HugePages_Total: 0
HugePages_Free:  0
HugePages_Rsvd:  0
HugePages_Surp:  0
Hugepagesize:   2048 kB
DirectMap4k:    28672 kB
DirectMap2M:    4296704 kB
```



首先，我们要安装R语言的运行环境，在Linux Ubuntu中一条命令就可以搞定。

```
# 安装R语言
~ sudo apt-get install r-base

#查看R语言的版本
~ R --version
R version 3.2.2 (2015-08-14) -- "Fire Safety"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under the terms of the
GNU General Public License versions 2 or 3.
For more information about these matters see
http://www.gnu.org/licenses/.
```

我们安装的R语言最新版本是3.2.2。

接下来，我们跑一个矩阵的计算，让2个3000行列的方阵相乘。

```
# 启动R
~ R

# 方阵相乘
> x <- matrix(1:(3000 * 3000), 3000, 3000)

# 计算耗时
> system.time(tmp <- x %*% x)
   user  system elapsed 
33.329   0.332  33.788
```

接下来，我们安装OpenBlas来提高计算性能。在Ubuntu中安装OpenBlas非常简单，只需要一条命令就可以搞定。

```
~ sudo apt-get install libopenblas-base
```

切换blas的计算引擎，使用openblas替换libblas。

```
~ sudo update-alternatives --config libblas.so.3
There are 2 choices for the alternative libblas.so.3 (providing /usr/lib/libblas.so.3).
```

Selection	Path	Priority	Status
* 0	/usr/lib/openblas-base/libblas.so.3	40	auto mode
1	/usr/lib/libblas/libblas.so.3	10	manual mode
2	/usr/lib/openblas-base/libblas.so.3	40	manual mode

Press enter to keep the current choice[\*], or type selection number: 0

选择0，使用openblas-base引擎。

我们重新打开R运行环境，再次执行刚才的矩阵相乘计算。

```
~ R
> x <- matrix(1:(3000 * 3000), 3000, 3000)

# 计算耗时
> system.time(tmp <- x %*% x)
   user   system elapsed 
 7.391    0.127    3.869
```

神奇的事情发生了，速度提升了4倍多。由于OpenBlas可以对矩阵计算加速，那么我们对所有矩阵操作都做一下测试吧。



3. 让R语言加速

通过互联网我找到了两个用于R语言性能测试的脚本，我们可以在自己的环境中测试一下。Benchmarks脚本的发布页，脚本代码下载。

我发现Revolution Analytics公司也用这个脚本进行了测试，并对比了Revolution企业版和R的官方发行版的区别。

下载脚本

```
~ wget http://brettklamer.com/assets/files/statistical/faster-blas-in-r/R-benchmark-25.R
--2015-09-24 12:06:05-- http://brettklamer.com/assets/files/statistical/faster-blas-in-r/R-benchmark-
Resolving brettklamer.com (brettklamer.com)... 199.96.156.242
Connecting to brettklamer.com (brettklamer.com)|199.96.156.242|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13666 (13K)
Saving to: 'R-benchmark-25.R'

100%[=====>] 13

2015-09-24 12:06:06 (203 MB/s) - 'R-benchmark-25.R' saved [13666/13666]
```

执行脚本。

```
~ R

# 运行脚本
> source("R-benchmark-25.R")
Loading required package: Matrix
Loading required package: SuppDists

R Benchmark 2.5
=====
Number of times each test is run: 3
```

```
I. Matrix calculation
-----
Creation, transp., deformation of a 2500x2500 matrix (sec): 1.103
2400x2400 normal distributed random matrix `1000` (sec): 0.812333333333333
Sorting of 7,000,000 random values (sec): 0.962666666666667
2800x2800 cross-product matrix (b = a' * a) (sec): 1.547
Linear regr. over a 3000x3000 matrix (c = a \ b') (sec): 0.828000000000001
-----
Trimmed geom. mean (2 extremes eliminated): 0.957989159036612

II. Matrix functions
-----
FFT over 2,400,000 random values (sec): 0.365333333333335
Eigenvalues of a 640x640 random matrix (sec): 1.434666666666667
Determinant of a 2500x2500 random matrix (sec): 0.895999999999998
Cholesky decomposition of a 3000x3000 matrix (sec): 0.832000000000003
Inverse of a 1600x1600 random matrix (sec): 0.724333333333334
-----
Trimmed geom. mean (2 extremes eliminated): 0.814310314522547

III. Programmation
-----
3,500,000 Fibonacci numbers calculation (vector calc)(sec): 0.776666666666661
Creation of a 3000x3000 Hilbert matrix (matrix calc) (sec): 0.269666666666667
Grand common divisors of 400,000 pairs (recursion) (sec): 0.570666666666663
Creation of a 500x500 Toeplitz matrix (loops) (sec): 0.506666666666665
Escoufier's method on a 45x45 matrix (mixed) (sec): 0.533000000000001
-----
Trimmed geom. mean (2 extremes eliminated): 0.536138937440438

Total time for all 15 tests (sec): 12.162
Overall mean (sum of I, II and III trimmed means/3) (sec): 0.747841037469598
--- End of test ---
```

我们再切换到，R语言默认的blas引擎运行一次。

```
~ sudo update-alternatives --config libblas.so.3
There are 2 choices for the alternative libblas.so.3 (providing /usr/lib/libblas.so.3).

  Selection    Path                                          Priority  Status
  -----
* 0            /usr/lib/openblas-base/libblas.so.3        40       auto mode
  1            /usr/lib/libblas/libblas.so.3              10       manual mode
  2            /usr/lib/openblas-base/libblas.so.3        40       manual mode

Press enter to keep the current choice[*], or type selection number: 1
update-alternatives: using /usr/lib/libblas/libblas.so.3 to provide /usr/lib/libblas.so.3 (libblas.so.3)
◀────────────────────────────────────────────────────────────────────────────────▶
```

选择1，切换到libblas引擎。重启R语言环境，并执行脚本。

```
~ R
> source("R-benchmark-25.R")
Loading required package: Matrix
Loading required package: SuppDists

R Benchmark 2.5
=====
Number of times each test is run: 3

I. Matrix calculation
-----
Creation, transp., deformation of a 2500x2500 matrix (sec): 1.09366666666667
2400x2400 normal distributed random matrix `1000` (sec): 0.817333333333333
Sorting of 7,000,000 random values (sec): 0.954333333333333
2800x2800 cross-product matrix (b = a' * a) (sec): 15.3033333333333
Linear regr. over a 3000x3000 matrix (c = a \ b') (sec): 7.155
-----
Trimmed geom. mean (2 extremes eliminated): 1.95463154033118

II. Matrix functions
```

```
-----
FFT over 2,400,000 random values_____ (sec): 0.3636666666666669
Eigenvalues of a 640x640 random matrix_____ (sec): 1.131
Determinant of a 2500x2500 random matrix_____ (sec): 5.061
Cholesky decomposition of a 3000x3000 matrix_____ (sec): 5.634
Inverse of a 1600x1600 random matrix_____ (sec): 4.142
-----

Trimmed geom. mean (2 extremes eliminated): 2.87278425762591

III. Programmation
-----

3,500,000 Fibonacci numbers calculation (vector calc)(sec): 0.7750000000000006
Creation of a 3000x3000 Hilbert matrix (matrix calc) (sec): 0.2596666666666665
Grand common divisors of 400,000 pairs (recursion)____ (sec): 0.6333333333333345
Creation of a 500x500 Toeplitz matrix (loops)_____ (sec): 0.5336666666666666
Escoufier's method on a 45x45 matrix (mixed)_____ (sec): 0.6479999999999996
-----

Trimmed geom. mean (2 extremes eliminated): 0.602780428790226

Total time for all 15 tests_____ (sec): 44.505
Overall mean (sum of I, II and III trimmed means/3)_ (sec): 1.5014435867612
--- End of test ---
```

从运行结果看到，用OpenBlas库在做矩阵计算时，性能优势是非常明显的。完成15个测试，OpenBlas库用时12秒，而默认的Blas库用时44秒。仅仅是切换一个底层算法库的成本，就可以让计算性能得到非常大的提升，各位R的小伙伴赶紧用起来吧。

转载请注明出处：  
<http://blog.fens.me/r-martix-openblas/>

如果觉得文章不错，请作者**喝杯咖啡！**



 微信



 支付宝

This entry was posted in [R语言实践](#)

赞助商广告(购买)



广告位

Jerome Cao  
hello大神哥，这只是改变了运行环境和底层算法，但是基本的函数库没有变，对吧？谢谢

Conan Zhang  
是的，只是底层变了。