R与高性能运算

简介和实例

李舰

Email: lijian.pku@gmail.com

Homepage: http://www.jliblog.com/

第四届中国R语言会议（北京会场）

2011 年 5 月

目 录

目 录

目 录

目 录

目 录

目 录

Some truths

- S语言的一个设计理念
  - Computer time is inexpensive in comparison with personnel time
  - 人的时间永远比机器的时间宝贵
- R的一些事实
  - 解释型语言
  - 数据全部读入内存
  - 单线程

提升性能的方法

- 系统升级
  - 升级硬件
  - 使用64位操作系统
  - 利用GPU
  - 租用云计算
- 开发层面的优化
  - 算法
  - 调用C或者Fortran
- 使用层面的优化
  - 充分利用R的内部机制 ——R 的基础优化
  - 大数据的处理 —— 突破内存的限制
  - 增强R的矩阵运算 —— 加速BLAS
  - 并行计算

内存限制

- 32位系统
  - 操作系统能支持的最大内存为$\frac{2^{32}}{1024^3} = 4G$
  - R在32位Windows系统中最大使用内存为$3G$
  - 通常R在内存使用超过$2G$（Windows）或$3G$（Unix）时就会报错
- 64位系统
  - 操作系统能支持的最大内存为$\frac{2^{64}}{1024^3} = 172 \times 10^8 G$
  - 32位的R在一些64位的Windows系统下最大内存也只有$4G$
  - R能使用的内存受限于硬件和操作系统
  - 由于没有64位的整型数据结构，无法定义过大的矩阵

cannot allocate vector of size

- memory.size(max = FALSE)
  - memory.size(NA)，查看系统分配给R的最大内存
  - memory.size(F)，查看当前已经使用的内存
  - memory.size(T)，查看已分配的内存
- memory.limit(size = NA)
  - memory.limit()，查看系统分配给R的最大内存
  - memory.limit(2000)，分配最大内存为2G
- Rgui –max-mem-size 2Gb

了解R的内存

- 内存划分
  - 堆内存（Heap），基本单元是“Vcells”，每个大小为8字节
  - 地址对（cons cells），最小单元一般在32位系统中是28字节、64位系统中是56字节
- R的存储模式
  - storage.mode（x）
  - 对于整型矩阵，storage.mode(x）< − "integer"
- R对象内存
  - ls()，查看当前对象
  - object.size(x)，查看对象所占用的内存

R的垃圾清理

- rm()和gc()
  - rm()清除对象的引用
  - gc()清扫内存空间，进行垃圾清理
- 程序中的垃圾清理
  - 对于长度增加的矩阵，尽量先定义一个大矩阵，然后逐步增加
  - 注意清除中间对象

compiler包

- R 2.13.0的新特性，compiler包
  - compiler包是R 2.13.0自带的一个标准包，它可以把一段R代码编译成字节码，从而在执行时提升效率。
- 示例

```
> source("rhpcbj4.R")
> require(compiler)
> la1c <- cmpfun(la1)
> y <- 1:100
> system.time(for (i in 1:1000) la1(y, is.null))

   user  system elapsed
   0.32    0.00    0.33

> system.time(for (i in 1:1000) la1c(y, is.null))

   user  system elapsed
   0.12    0.00    0.13
```

基于文件缓存的bigmemory家族

- bigmemory
  - 建立基于文件的大矩阵，实现nwitch、order等方法
- biganalytics
  - 对于bigmemory对象，实现apply、kmeans、lm、colmean等方法
- bigtabulate
  - 实现大矩阵的table、tapply等方法
- bigalgebra
  - 实现大矩阵的BLAS和LAPACK
  - 目前Linux下已能使用
- synchronicity
  - 实现Boost mutex的功能

1千万条记录60个变量的稠密Double型数据的示例

R与数据库的交互

- 常见数据库的接口
  - RODBC
  - DBI
  - ROracle, RMySQL, RPostgreSQL, RSQLite
- 使用RSQLite进行数据处理
  - 小巧轻量的关系型内存数据库
  - 通过索引优化处理大数据
  - RSQLite.extfuns包提供了扩展函数

数据库厂商的支持

- RODM
  - Oracle Data Mining (ODM) 提供的R接口，可以在R中隐式地调用ODM内置的数据挖掘算法。
  - http://cran.r-project.org/web/packages/RODM/index.html
- TeradataR
  - Teradata Warehouse Miner 和R的接口，为R提供了20个和数据仓库交互的基础函数以及44个分析函数。
  - http://downloads.teradata.com/download/applications/teradata-r/1.0

non-sql型数据库（内容待补充）

- 传统的关系型数据库为事务型的数据需求而生
- 当需求为对历史数据进行分析的时候，“关系”还有用吗？

BLAS和LAPACK

- BLAS
  - Basic Linear Algebra Subprograms，基础线性代数程序集
  - http://www.netlib.org/blas/
  - 基础的线性代数操作，向量和矩阵乘法等
  - C. L. Lawson, R. J. Hanson, D. Kincaid, and F. T. Krogh, Basic Linear Algebra Subprograms for FORTRAN usage, ACM Trans. Math. Soft., 5 (1979)
- LAPACK
  - Linear Algebra PACKage
  - 依赖BLAS，解多元线性方程式、线性系统方程组的最小平方解、计算特征向量、用于计算矩阵QR分解的Householder转换、以及奇异值分解等

Atlas

- Automatically Tuned Linear Algebra System
  - 针对特定平台进行优化
  - netlib上优化BLAS的工程，如今已经移到http://math-atlas.sourceforge.net/
  - R版本的一个实现：http://mirrors.geoexpat.com/cran/bin/windows/contrib/ATLAS/
- Linux环境下（Ubuntu 10.04 64位，下同）
  - Atlas 3.6.0，单线程
  - Atlas 3.9.25 多线程
  - sudo apt-get install libatlas3gf-base
  - sudo wajig install libatlas39_3.9.25-1_amd64.deb

MKL

- Intel提供的BLAS与LAPACK接口
  - Linux下有非商用的免费版
  - http://software.intel.com/en-us/articles/intel-mkl/
- Linux环境下编译安装
  - XiaoNan的博客有详细的安装介绍：http://www.road2stat.com/cn/r_language/optimize.html
- 最简单的安装方式：revolution-mkl
  - sudo apt-get install revolution-mkl r-revolution-revobase revolution-r

Goto BLAS

- Kazushige Goto （后藤和茂）开发的BLAS
  - 支持多线程
  - 据说是目前最快的BLAS
- Windows下安装方法
  - http://prs.ism.ac.jp/ñakama/有R版本的BLAS提供下载
- Linux下安装方法
  - 到http://www.tacc.utexas.edu注册帐号
  - apt-get install gotoblas2-helper
  - Login(register) http://www.tacc.utexas.edu/?id=402
  - vi /etc/gotoblas2-helper/gotoblas2-site.conf
  - /etc/init.d/gotoblas2-helper start

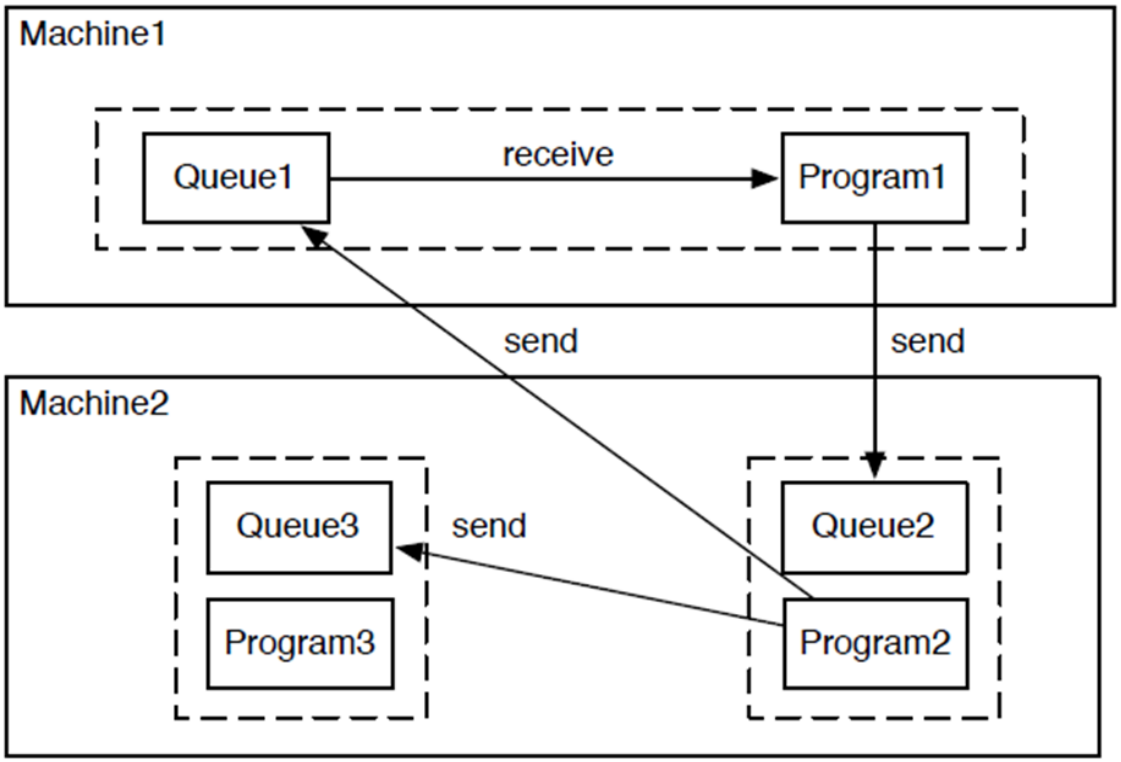
待补充

什么是并行

- 并行计算
  - 同时进行不同的计算
  - 单核多线程
  - 多核或多CPU的并行
- 显式并行
  - 由用户控制的并行，需要在算法上做专门的处理
- 隐式并行
  - 系统自动进行的并行处理

MPI

- 什么是MPI
  - Message Passing Interface，一种消息传递接口
  - 程序通过收发队列里的消息进行交互
- MPI实现机制



snow家族

- snow
  - 可以使用MPI、NWS、PVM、Sockets四种传递方式进行并行
  - 在多核或者计算机集群上实现并行计算
- snowfall
  - 对snow进行简化包装后的一个包
- http://www.wentrue.net/blog/?p=878

隐式并行

- 多线程BLAS
  - 自动多核进行代数运算
- multicore
  - 通过类似lapply的方式拆分任务
  - 自动分配到多个核计算
- doMC和foreach
  - doMC注册多核
  - foreach取代循环，分配到多核运算

示例

MapReduce

- Google的一个专利申请
  - 2010年1月获批，编号为7 650 331，名为System and method for efficient large-scale data processing（高效大规模数据处理）。是Google最引为自豪的成果之一，也是云计算最重要的核心技术之一。
- MapReduce的应用
  - Google基础应用
  - 雅虎搜索
  - Amazon的Elastic MapReduce服务
  - 开源项目Apache Hadoop

RHIPE简介

- 开源的MapReduce：Hadoop
  - Hadoop 是Google MapReduce 的一个Java实现
  - 定义Mapper，处理输入的Key-Value对，输出中间结果。定义Reducer，可选，对中间结果进行规约，输出最终结果。定义main函数。
  - 提交JOB，系统自动完成
- R和Hadoop的整合：RHIPE
  - 开源项目，将R和Hadoop集成在一起
  - 目前只有Linux和Mac OS版本

实际操作的一点建议

- 如果矩阵运算比较多
  - 更换BLAS
  - 10G数据以下使用bigmemory
  - 海量数据尝试Hadoop
- 如果数据处理比较多
  - 使用数据库，优化查询
  - 尝试并行
- 优化算法是王道

Thank you!

Email: lijian.pku@gmail.com
Homepage: http://www.jliblog.com/