Uso da Linguagem R em Ambientes Paralelos

Paulo Ricardo Rodrigues de Souza Júnior, Rafael Machado Sampaio, Willingthon Pavan, Carlos Amaral Hölbig

Curso de Ciência da Computação – ICEG – Universidade de Passo Fundo (UPF)
99.001-970 – Passo Fundo – RS – Brasil

{119711,105609,pavan,holbig}@upf.br

Resumo. *Uma das tecnologias que estão sendo mais utilizadas atualmente para implementar modelos de simulação do crescimento e desenvolvimento de culturas é a linguagem R. Devido a crescente complexidade destes modelos é que este artigo visa o uso da linguagem R em ambientes computacionais paralelos visando, em um segundo momento, a paralelização de modelos de simulação de culturas de plantas e doenças nestes ambientes computacionais.*

1. Introdução

A linguagem R é um projeto *open source* que está disponível para a maioria das plataformas computacionais. Além de ser uma linguagem de programação também é um ambiente para computação estatística, modelagem e visualização de dados. Trata-se se uma suíte de softwares integrados que proporcionam facilidades na manipulação de dados, no uso de funções estatísticas e na geração de gráficos [Adler, 2012; R-Project, 2012].

Os modelos de simulação do crescimento e desenvolvimento de culturas têm sido usados com sucesso ao redor do mundo na agricultura para aumentar a produtividade e reduzir custos [Pavan, 2007]. A linguagem R é uma das tecnologias que estão sendo utilizadas atualmente para implementar estes modelos. Estes modelos apresentam um constante aumento de dados tornando os problemas muito complexos e demandando um grande esforço computacional, o que eleva muito o tempo de processamento dos modelos. Para ser viável trabalhar com este grande número de dados é cada vez mais importante o uso de computação paralela.

Devido a estes fatores, este trabalho foca o uso da linguagem R em ambientes computacionais paralelos e, em um segundo momento, estudará a paralelização de modelos de simulação de culturas implementados em R utilizando os pacotes avaliados e selecionados nesta pesquisa. Na seção 2 é apresentado um estudo sobre pacotes desenvolvidos para a linguagem R que possibilitam a paralelização de seus programas; na seção 3 são apresentados testes e resultados a respeito do uso de alguns destes pacotes e, por fim, na seção 4 são apresentados os resultados desta pesquisa e os trabalhos futuros que serão desenvolvidos.

2. Pacotes Paralelos para o R

A linguagem R apresenta diversos pacotes que possibilitam a sua paralelização [Schmidberger, et al., 2009]. Existem pacotes para Clusters, Grids, para o uso em GPUs e para máquinas com processadores multicore, conforme apresentado na Tabela 1. Por

existirem diversos pacotes para programação paralela em R foi realizado um estudo para analisar alguns destes pacotes e decidir quais os mais adequados para serem utilizados na paralelização dos programas em R e em uma futura paralelização dos modelos de simulação citados nesta pesquisa. Uma lista atual destes pacotes poderá ser encontrada na página da CRAN *Task View*: *High-Performance and Parallel Computing with R* (http://cran.r-project.org/web/views/HighPerformanceComputing.html), que é a página da entidade que disponibiliza o R e seus pacotes oficiais.

Tabela 1 – Visão geral sobre computação paralela com R em clusters de computadores, ambientes multicore, Grids e GPUs

Pacote	Descrição
rpvm	Interface R para PVM (Parallel Virtual Machine)
Rmpi	Interface (Wrapper) para MPI (Message-Passing Interface)
RHadoop	Pacote que integra o R ao Hadoop
snow	**S**imple **N**etwork **o**f **W**orkstations
snowFT	Pacote de tolerância a falhas para o snow
papply	Função apply em paralelo usando MPI
foreach	Construtor foreach para R (paralelização de laços)
doMC	Foreach paralelo adaptado para o pacote multicore
doSNOW	Foreach paralelo adaptado para o pacote snow
doMPI	Foreach paralelo adaptado para o pacote Rmpi
fork	Funções em R para manipulação de múltiplos processos
multicore	Código para processamento paralelo do R em máquinas com múltiplos cores ou CPUs
gridR	Executa funções em hosts remotos, clusters ou grids
gputools	Fornece vários algoritmos de mineração de dados que são implementados usando uma mistura de linguagem CUDA da nVidia e biblioteca cublas
magma	Disponibiliza uma interface para a biblioteca híbrida GPU/CPU Magma (biblioteca de classes e métodos para o processamento paralelo de operações matriciais)
Xgrid	Funções para distribuir e coletar resultados de simulações e outras tarefas executadas em Apple Xgrid clusters

Para clusters os principais pacotes utilizados são Rmpi, Rpvm e RHadoop. O RPVM (R *for Parallel Virtual Machine*) é projetado para permitir que uma rede Unix heterogênea ou máquinas Windows sejam usadas como um único computador paralelo distribuído. O RPVM é complexo de ser utilizado por valer-se de funções de baixo nível. RMPI (R *for Message-Passing Interface*) é um sistema padronizado e portátil de transmissão de mensagens em computação paralela, fornecendo uma interface R para funções MPI de baixo nível. Desta forma, o utilizador R não precisa conhecer os detalhes das implementações de MPI. O RHadoop é um pacote que integra o R as funcionalidades do Hadoop, que é um sistema baseado no modelo *Map/Reduce* e é voltado para a manipulação de grande quantidades de dados [Adler, 2012].

Para ambientes computacionais com processadores multicore os principais pacotes são Fork, Multicore e o foreach. O pacote Fork utiliza basicamente os recursos do sistema UNIX para efetuar a paralelização. Possui uma utilização relativamente simples por ter poucas funções mas não apresenta suporte para funções de alto nível como a função `aplly`. O pacote multicore apresenta além das chamadas de funções do sistema UNIX, outras rotinas próprias. Sua utilização é mais complexa por apresentar

mais funções, porém tem suporte para funções de alto nível. O pacote foreach dá suporte para a construção do loop `foreach` (similar ao comando for) que é uma expressão que permite a iteração sobre os elementos de uma coleção de dados (em sequencial ou em paralelo), sem a utilização de um contador de ciclo explícito. Para ambientes multicore, os pacotes Multicore e foreach fornecem melhores soluções pois reunem mais recursos que o pacote Fork. A partir da versão 2.14.0 a linguagem R oferece suporte direto ao paralelismo com a disponibilização do pacote "parallel" que incorpora cópias (ligeiramente revisadas) dos pacotes multicore e snow (mas excluindo clusters MPI, PVM e NWS). Além destes pacotes para *clusters* e *multicore* há pacotes para *grids*, GPUs e pacotes para aplicações específicas, todos disponíveis no site da CRAN.

Devido às características dos modelos de simulação trabalhados no grupo de pesquisa optou-se por focar os testes em alguns pacotes para ambientes *multicore* e com o uso de GPUs. Os testes e resultados são apresentados na próxima seção.

4 Testes e Resultados

Os testes foram desenvolvidos no grupo de pesquisa ComPaDi da Universidade de Passo Fundo. O computador utilizado possui um processador Intel Core i7 920, que opera à frequência de 2.66 Ghz, com 8 MB de cache L2, 8 GB de memória RAM, sistema operacional Ubuntu 12.04 64 bits e placa de vídeo GeForce GTS250 1GB DDR3 ECS. Os *softwares* utilizados foram a linguagem R (versão 2.15.2 de 64 bits), a IDE RStudio (versão 0.97.168) e os pacotes multicore (0.1-7), foreach (1.4.0), snow (0.3-10), doSNOW (1.0.6), iterators (1.0.6), parallel (2.15.2) e gputools (0.26).

O primeiro teste (Figura 1) abordou o uso do pacote multicore com a paralelização da função `mapply` do R utilizando a função paralela equivalente `mcapply`. Este teste utilizou apenas os quatro cores físicos do computador. Na execução sequencial o tempo de execução foi de 1.407 segundos e o tempo paralelo de 0.351 segundos.

```
1  library(multicore)
2  multicore:::detectCores()
3  options(cores = 4) # usando os 4 cores físicos
4  getOption('cores')
5
6  test <- lapply(1:10,function(x) rnorm(10000))
7  system.time(x <- lapply(test,function(x) loess.smooth(x,x)))
8  system.time(x <- mclapply(test,function(x) loess.smooth(x,x)))
```

Figura 1 – Paralelização da função lapply com uso do pacote multicore

O segundo teste (Figura 2) abordou a paralelização de laços de repetições (comando `for`) por meio da função paralela equivalente `foreach` (utilizando o parâmetro `%dopar%`), disponibilizada pelo pacote foreach. Neste teste foi criada uma função estatística no R e o programa chamava esta função 100 vezes. Com o uso do laço em sequencial, o tempo de execução foi de 40.46 segundos, já em paralelo, o tempo de execução utilizando os oito processadores foi de 19.17 segundos.

O último teste (Figura 3) abordou a realização de operações matriciais (uma multiplicação de matrizes de ordem 8192) utilizando a GPU da máquina. Para isso foi utilizado o pacote gputools com a função `gpuMatMult`. O tempo de execução sequencial foi de 139.1354 segundos e o utilizando a GPU foi de 11.1562 segundos.

```
1   require(doSNOW)
2   cl<-makeCluster(8) # usando oito cores
3   registerDoSNOW(cl)
4
5   # cria uma função para rodar em cada iteração do laço
6   check <-function(n) {
7       for(i in 1:1000) {
8           sme <- matrix(rnorm(100), 10,10)
9           solve(sme)
10          }
11      }
12  times <- 100      # qtde de vezes a ser executado o laço
13  system.time(x <- foreach(j=1:times ) %dopar% check(j))
14  system.time(for(j in 1:times ) x <- check(j))
15  stopCluster(cl)
```

Figura 2 – Paralelização de laços com uso do `foreach`

```
1   library(gputools)
2   ordem = 8192
3   #inicialização das matrizes A, B e C
... ...
13  system.time(C<-A%*%B) # realiza a multiplicação de matrizes em
14  sequencial – resultado: 139.1354 segs.
15
16  system.time(gpuMatMult(A, B)) # realiza a multiplicação de
17  matrizes com a GPU – resultado: 11.1562 segs.
```

Figura 3 – Multiplicação de matrizes na GPU com uso do pacote gputools

É importante destacar que a linguagem R utiliza implicitamente para as operações de álgebra linear as rotinas da biblioteca BLAS .

5 Conclusões e Trabalhos Futuros

Com os modelos de simulação apresentando cada vez mais dados é imprescindível encontrar formas de otimizar o desempenho dos mesmos. Para tanto, a paralelização mostra ser uma das alternativas, pois pode melhorar efetivamente o tempo de execução dos programas. Os pacotes Multicore, foreach e gputools vêm ao encontro das necessidades dos modelos de simulação viabilizando a programação concorrente destes. Atualmente, como consequência da escolha destes pacotes, está se trabalhando na paralelização de dois modelos de doenças que atacam a cultura do morango e que fazem parte de uma parceria entre a UPF, Embrapa Trigo e Universidade da Flórida.

Referências

Adler, J. (2012) R in a Nutshell. 2. ed. Sebastopol: O'Reilly.

Pavan, W. (2007) Técnicas de engenharia de software aplicadas à modelagem e simulação de doenças de plantas, 2007. 182 p. Tese (Doutorado em Agronomia) - Universidade de Passo Fundo, Passo Fundo.

R-PROJECT. (2012) **R Project for Statistical Computing**. Disponível em: <http://www.r-project.org/>. Acesso em 7 Jan. 2012.

Schmidberger, M.; Morgan, M.; Eddelbuettel, D.; Yu, H.; Tierney, L.; Mansmann, U. (2009) State of the Art in Parallel Computing with R. Journal of Statistical Software, vol. 31, Issue 1, p. 1-27, 2009.