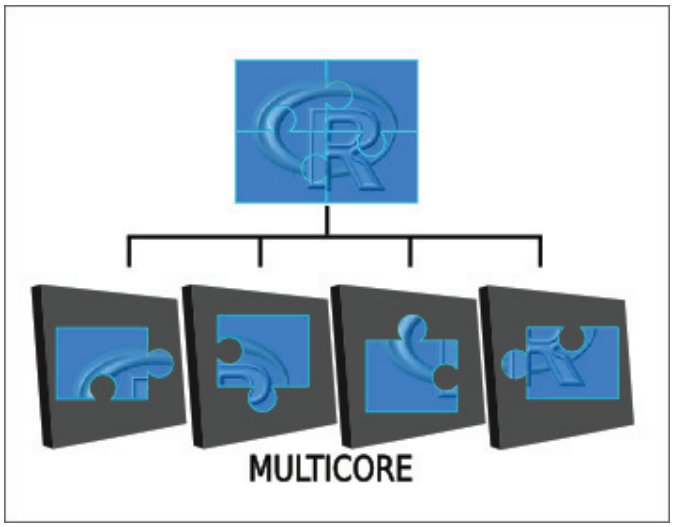
Comunicado Técnico 112

ISSN 1677-8464
Dezembro, 2012
Campinas, SP



Programação em R utilizando multicores: um estudo de caso envolvendo dados de modelos climáticos

Alan Massaru Nakai[1]
Eduardo Nespoli Ramos[2]

R é um software estatístico composto por uma linguagem e um ambiente voltados à realização de cálculos estatísticos e geração de gráficos. Trata-se de um software livre que vem sendo desenvolvido ao longo de vários anos por um grande número de colaboradores (R CORE TEAM, 2012; DALGAARD, 2008).

A sua linguagem nativa permite o desenvolvimento de novos algoritmos a partir de uma ampla variedade de funções matemáticas, estatísticas e gráficas, facilidades de entrada e saída de dados e operadores para vetores e matrizes (GENTLEMAN, 2009). Além das operações básicas, R pode ser estendido por meio da instalação de novos pacotes de funções.

Uma das limitações do R é que ele é, originalmente, um software *monothread*, ou seja, suas tarefas não são paralelizadas. Desta forma, R não aproveita a capacidade de processamento dos processadores modernos, que possuem vários núcleos de processamento.

No Laboratório de Modelagem Agroambiental (LMA) da Embrapa Informática Agropecuária, o R é amplamente utilizado na manipulação e processamento de dados geoespaciais, mais especificamente, dados de modelos de projeção climática. O processamento desses dados envolve tarefas como a conversão entre diferentes formatos, interpolações, geração de mapas, entre outras. Todas essas tarefas, que são computacionalmente custosas, são executas sob grande quantidade de dados independentes, o que caracteriza um tipo de aplicação propício para computação distribuída.

O objetivo deste trabalho é demonstrar o uso de multiprocessamento em R. Para isso, apresenta um estudo de caso envolvendo o cálculo da média móvel de séries de chuva mensal, utilizando dados de modelos de projeção climática.

O multiprocessamento em R é possível por meio de diversos pacotes, como *snow* (TIERNEY et al., 2012), *fork* (WARNES, 2012), *sprint* (UNIVERSITY OF EDINBURGH, 2012), *snowfall* (KNAUS, 2012), *multicore* (URBANEK, 2012), entre outros. Embora, a utilização desses pacotes seja semelhante em diversos aspectos, este trabalho aborda o uso do pacote *multicore*, que nas versões mais atuais do R, já vem instalada.

O restante do texto é organizado da seguinte forma: a Seção 2 introduz o problema abordado no estudo de caso; a Seção 3 apresenta o pacote *multicore*; a Seção 4 descreve a implementação distribuída do cálculo da

[1] Doutor em Ciência da Computação, Analista da Embrapa Informática Agropecuária, Campinas, SP, alan.nakai@embrapa.br
[2] Aluno do curso de Tecnologia em Informática da Unicamp, estagiário da Embrapa Informática Agropecuária, eduardo.ramos@colaborador.embrapa.br

média; a Seção 5 apresenta os resultados obtidos nos experimentos e, finalmente, na Seção 6 são apresentadas as conclusões do trabalho.

Descrição do problema

Os dados utilizados neste estudo de caso são dados de chuva mensais provenientes de modelos de projeção climática disponíveis no Intergovernamental *Panel On Cliamate Changes* (IPCC[1]). Ao todo, os dados correspondem a 26 modelos diferentes, cada qual podendo apresentar até três cenários, que variam de acordo com o nível de emissão de CO_2 considerado na simulação. Além disso, cada combinação de modelo/cenário possui pelo menos 100 anos de dados.

Frequentemente, os pesquisadores que utilizam esses dados desejam utilizar os valores da chuva mensal calculada a partir da média de vários anos. Esse método é utilizado para amenizar os efeitos da alta variabilidade que os modelos climáticos geram de um ano para outro. Para exemplificar, suponha que se deseja utilizar uma média de 10 anos em torno de 2030. Nesse caso é necessário calcular a média de chuva de 2026 a 2035, para cada mês.

Para agilizar o acesso aos dados, pode-se manter uma base de dados com as médias pré-computadas. Essa pré-computação é realizada por meio de um *script* R para o cálculo da média móvel de *k* anos, que consiste em calcular todas as médias de *k* anos consecutivos ao longo de toda a série de dados dos modelos.

O algorítimo para o cálculo da média é relativamente simples. Deve-se ler os dados de chuva mensal de todos os anos, a partir de arquivos *Comma-separeted values* (CSV) e calcular a média para todos os pontos da grade do modelo. As grades dos modelos, tipicamente, variam de 20000 a 32000 pontos.

O Código Fonte 1 apresenta o trecho de código R que realiza o cálculo da média móvel para todas as combinações de modelos e cenários de forma sequencial. Nas linhas 1 a 4, obtém-se as listas de modelos e cenários disponíveis. Os comandos nas linhas 6 a 9 iteram sobre as listas para realizar os cálculos para todos modelos e cenários. A função `MediaMovel`, na linha 12, recebe como parâmetro no nome do modelo (`modelo`), o nome do cenário (`cenario`) e o tamanho do período em anos para o qual deseja-se calcular as

```
1.  # Obter lista de modelos disponíveis
2.  lista_modelos= get_modelos()
3.  #Obter lista de cenários disponíveis
4.  lista_cenarios= get_cenarios()
5.
6.  #Iterar sobre as listas de cenários e mode-
los
7.  for(cenario in lista_cenarios)
8.    {
9.      for(modelo in lista_modelos)
10.     {
11.       #Cálculo da média
12.       MediaMovel(modelo,cenario,tamanho_ja-
nela)
13.     }
14.   }
```

Código Fonte 1. Cálculo da média sequencial.

médias (`tamanho_janela`) . A partir desses parâmetros, a função calcula as médias de chuva e salva os dados resultantes em formato *CSV*.

Nos testes realizados (Seção 5), esta versão sequencial levou aproximadamente 20 horas para calcular a média móvel de 4 anos consecutivos, para 24 modelos e 1 cenário.

Pacote *multicore*

O pacote *multicore* [5] provê funções que permitem a execução de computação paralela em R, utilizando máquinas com múltiplos núcleos de processamento. As tarefas implementadas com estas funções compartilham o estado do processo R inicial. Desta forma, a inicialização dos subprocessos é rápida, pois não é necessário iniciar novas instâncias do R.

A paralelização da computação pode ser realizada por meio das seguintes funções:

`fork`: cria um novo processo como uma cópia do processo R atual;
`mcapply`: paraleliza a execução de uma função sobre todos os elementos de um vetor;
`pvec`: paraleliza a execução de uma função sobre subdivisões de um vetor;
`parallel`: avalia uma expressão de forma assíncrona em um processo separado.

Neste trabalho, a implementação paralela do cálculo da média móvel é implementada utilizando a função *mclapply*, que permite a execução paralela de uma fun-

[1] Disponível em: <http://www.ipcc-data.org/>.

ção preexistente sobre todas as posições de um vetor de dados. A grande vantagem dessa função é permitir a paralelização de funções por meio de uma sintaxe simples, tornando transparente aspectos da criação de novos processos ao programador. A sintaxe da função é apresentada a seguir e seus principais parâmetros são descritos na Tabela 1.

Sintaxe:

```
mclapply(X, FUN, …, mc.cores)
```

A função *mclapply* retorna uma lista com o mesmo comprimento de X, no qual cada elemento é o resultado da aplicação da função FUN ao elemento correspondente de X.

Implementação da média móvel utilizando *multicore*

A implementação paralela do *script* para o cálculo da média móvel segue uma estratégia direta: cada conjunto de dados que corresponde a uma combinação de modelo/cenário é processado de forma diferente. O número máximo de processos executados simultaneamente é igual ao número de núcleos de processamento disponíveis. Quando um processo termina, um dos núcleos é liberado e inicia-se a execução de outro processo.

O Código Fonte 2 apresenta um trecho da versão paralelizada do *script* para o cálculo da média móvel. Assim como na versão sequencial, as listas de modelos e cenários são obtidas nas linhas 2 a 5. Em seguida, cria-se um data *frame*[1] (`df`) para armazenar os dados

```
1.  # Obter lista de modelos disponíveis
2.  lista_modelos= get_modelos()
3.
4.  # Obter lista de cenários disponíveis
5.  lista_cenarios= get_cenarios()
6.
7.  # gerar conjunto de
8.  df = data.frame()
9.
10. #Iterar sobre as listas de cenários e mode-
los e popular data frame
11. atual = 0
12. for(cenario in lista_cenarios)
13.   {
14.     for(modelo in lista_modelos)
15.       {
16.         df[1,atual]=modelo
17.         df[2,atual]=cenario
18.         atual=atual+1;
19.       }
20.   }
21.
22. # Disparar processos paralelos
23. mclapply(df,
24.         MediaMovelParalela,
25.         tamanho_janela,
26.         mc.cores = cores)
```

Código Fonte 2. Versão paralelizada do cálculo das médias.

que serão processados em paralelo. As linhas 12 a 14 iteram sobre as listas de modelos e cenários enquanto as linhas 16 a 18 populam df, atribuindo um modelo e um cenário para cada coluna.

Finalmente, nas linhas 23 a 26, a função mclapply é utilizada para disparar várias instâncias da função `MediaMovelParalela`, sobre os dados contidos em `df`. A função `MediaMovelParalela` é uma versão modificada da função `MediaMovel` (Código Fonte 1) que recebe como parâmetro um vetor com duas posições, contendo o nome do modelo e o nome do cenário, e o tamanho do período da média (`tamanho_janela`). Cada

Tabela 1. Parâmetros da função `mclapply`.

Parâmetro	Valor padrão	Descrição
X	Não tem	Lista que contém os dados que devem ser processados por cada processo paralelo. O tamanho da lista define o número de processos que serão necessários para o processamento.
FUN	Não tem	A função que deve ser utilizada em cada processo. A função `mclapply` irá disparar vários processos e cada um deles executará essa função sobre uma posição da lista de dados de entrada.
...	NULL	Parâmetros para a função FUN.
mc.cores	Quantidade disponível	Número de núcleos que podem ser utilizados. Caso não seja especificado, a função `mclapply` utilizará a quantidade de núcleos disponíveis.

[1] Estrutura de dados do R para armazenar tabelas. Pode ser indexado na forma dataframe[linha, coluna].

coluna de `df` é submetida a uma instância diferente dessa função, que calculará a média móvel para a combinação modelo/cenário correspondente. O parâmetro `mc.cores` define quantos núcleos de processamento serão utilizados, e, consequentemente, quantos processos serão executados simultaneamente.

Resultados

Para avaliar o uso de múltiplos núcleos de processamento no cálculo da média móvel da chuva, executamos o *script* descrito no Código Fonte 2 sobre os dados de 24 modelos e um único cenário, variando o número de núcleos utilizados de 1 a 8.

A Figura 1 mostra o tempo de execução do cálculo da média móvel com os diferentes números de núcleos de processamento. Os resultados mostram melhoras significativas no tempo de execução de 1 para 4 núcleos. O cálculo com 4 núcleos apresentou o tempo de execução aproximadamente 70% menor. Entretanto, nota-se que, a partir de 5 núcleos, o aumento do número de núcleos de processamento não apresentou melhoras significativas de desempenho. Aumentando o número de núcleos de 4 para 8, o tempo de execução diminuiu apenas 10%.

Um dos motivos para que o aumento de núcleos deixe de apresentar melhoras significativas de desempenho a partir de 5 núcleos é o fato de que as tarefas (cálculo da média da chuva) envolvem muita leitura e escrita de arquivos (I/O). Desta forma, mesmo com mais processos executando em paralelo, o tempo global de execução não melhora, pois o disco rígido torna-se o gargalo do sistema.

Essa constatação é evidenciada pelo gráfico da Figura 2, que mostra o tempo do cálculo da média móvel para cada modelo. Conforme indicado pela figura, os tempos individuais para cada modelo utilizando 8 núcleos é maior, o que acaba comprometendo o tempo total do cálculo.

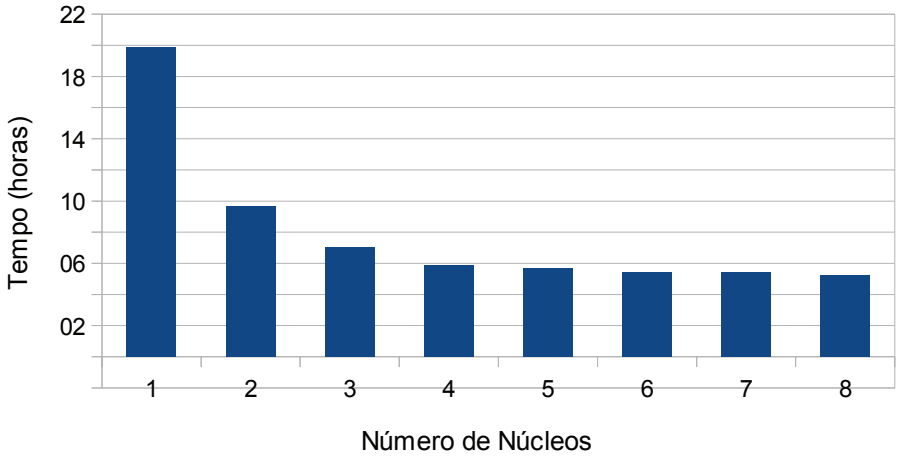


Figura 1. Comparação de tempo de execução da média móvel utilizando diferentes números de núcleos de processamento.

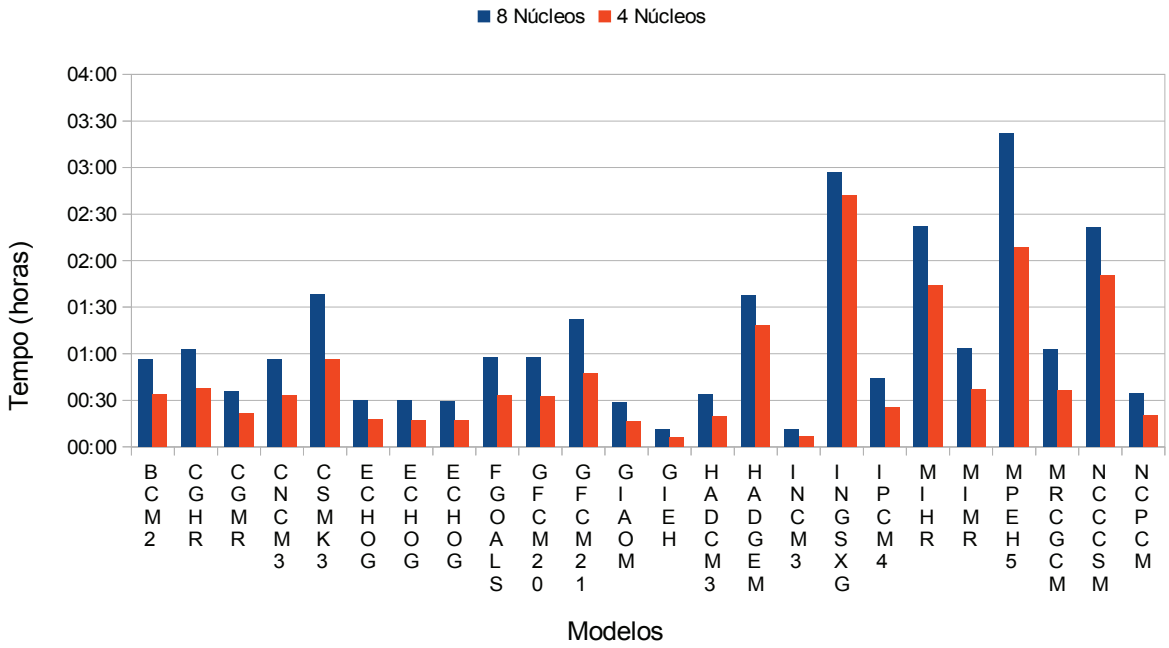


Figura 2. Comparação de tempo de execução para cada modelo utilizando 4 e 8 núcleos de processamento.

Conclusões

Este trabalho demonstrou o uso de multiprocessamento em R com o pacote *multicore*. Para isso, apresentou um estudo de caso envolvendo o cálculo da média móvel da chuva mensal de diversos modelos de projeção climática.

Os resultados mostram que, para o estudo de caso realizado, o aumento de 1 para 4 núcleos de processamento apresentou uma melhora de mais de 70% do tempo de execução. Apesar disso, pode-se concluir que o aumento do número de processadores não garante melhores desempenhos, já que outros fatores, como o acesso a disco, podem tornar-se o gargalo do sistema e influenciar negativamente no tempo de execução.

Referências

DALGAARD, P. **Introductory statistics with R**. Springer: New York, 2008. 363 p.

GENTLEMAN, R. **Data analysts captivated by R's power**. 2009. Disponivel em: <http://www.nytimes.com/2009/01/07/technology/business-computing/07program.html?pagewanted=all&_moc.semityn.www>. Acesso em: 11 set. 2011.

KNAUS, J. **Easier cluster computing (based on snow)**. 2012 . Disponível em: <http://cran.r-project.org/web/packages/snowfall/snowfall.pdf>. Acesso em: 4 out. 2012.

R CORE TEAM. **R**: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, 2012. Disponível em: <http://www.R-project.org.>. Acesso em 4 out. 2012.

TIERNEY, A.; ROSSINI, A. J.; LI, N.; SEVCIKOVA , H. **Simple network of workstations**. 2012. Disponível em: <http://cran.r-project.org/web/packages/snow/snow.pdf>. Acesso em: 4 out. 2012.

UNIVERSITY OF EDINBURGH. **Simple parallel R interface**. 2012. Disponível em: <http://cran.r-project.org/web/packages/sprint/sprint.pdf>. 2012. Acesso em: 4 out. 2012.

URBANEK, S. **Parallel processing of R code on machines with multiple cores or CPUs**. 2012. Disponível em: <http://cran.r--project.org/web/packages/multicore/multicore.pdf.>. Acesso em: 4 out. 2012.

WARNES, G. R. **R functions for handling multiple processes**. 2012. Disponível em: <http://cran.r-project.org/web/packages/fork/fork.pdf>. Acesso em: 4 out. 2012.

Comunicado Técnico, 112

Embrapa Informática Agropecuária
Endereço: Caixa Postal 6041 - Barão Geraldo
13083-886 - Campinas, SP
Fone: (19) 3211-5700
Fax: (19) 3211-5754
http://www.cnptia.embrapa.br
e-mail: cnptia.sac@embrapa.com.br

1ª edição on-line - 2012

CGPE 10176