

Otimização Computacional do Modelo BRASIL-SR

Jefferson Gonçalves de Souza¹, Celso Luiz Mendes¹, Rodrigo Santos Costa²

Instituto Nacional de Pesquisas Espaciais (INPE) – São José dos Campos – SP – Brasil

Laboratório Associado de Computação e Matemática Aplicada - LAC¹

Centro de Ciência do Sistema Terrestre – CCST²

jefferson.souza@inpe.br, celso.mendes@inpe.br, rodrigo.costa@inpe.br

Abstract. *This paper describes the improvement of computational performance of the BRASIL-SR model. This computational code, which is used for the estimate of incident surface solar radiation, was optimized with an application of OpenMP directives and code modifications for the implementation of data output (I/O) operations using NetCDF format files. After these implementations, it was verified that the speedup of the code was further improved with the I/O optimizations, with a more stable efficiency, which shows a better use of the processors in the underlying supercomputer.*

Resumo. *Este artigo apresenta a melhoria de desempenho computacional do modelo BRASIL-SR. Este código computacional, utilizado para a estimativa da irradiação solar incidente na superfície, foi otimizado com a aplicação de diretivas de OpenMP e modificações do código para implementação de operações de entrada e saída de dados (E/S) utilizando arquivos em formato NetCDF. Após estas implementações, verificou-se que o ganho de desempenho do código foi ainda maior com um novo tratamento de E/S, gerando uma eficiência mais estável, o que mostra uma melhor utilização dos processadores no supercomputador empregado.*

1. Introdução

Diante da preocupação em atender a demanda energética mundial sem aumentar as emissões de gases do efeito estufa, diversos países têm buscado soluções relacionadas com a inserção de fontes limpas em suas matrizes energéticas. A geração solar tem crescido vertiginosamente no Brasil, principalmente depois dos déficits hídricos registrados nos últimos anos em função da variabilidade climática. Condições de estiagem afetam a geração de energia elétrica, já que os reservatórios podem vir a atingir níveis muito baixos; isso também levanta questões relativas ao conflito do uso da água, que também é utilizada no abastecimento e na agricultura. Há uma necessidade iminente da diversificação da matriz elétrica brasileira, o que minimizaria problemas relacionados à segurança hídrica, energética e alimentar. Também é importante que esta diversificação ocorra no sentido da inserção de fontes renováveis, com baixa emissão de gases.

O potencial solar brasileiro é considerado elevado, uma característica de regiões tropicais. Além disso, a baixa variabilidade intra-anual permite que as regiões brasileiras com menor capacidade de geração produzam mais energia que as regiões de maior potencial da Alemanha, por exemplo.

Antes da implantação de empreendimentos solares, é necessário conduzir estudos de avaliação de viabilidade e potencial do recurso. Utilizar dados de estações meteorológicas é a fonte mais segura para se obter o conhecimento do potencial local de energia solar, entretanto, pesquisas indicam que os erros de interpolação de dados observados em estações de superfície afastadas em mais de 30km entre si, são superiores aos erros de estimativas produzidas por modelos de transferência radiativa, como pode ser observado na Figura 1. Assim, a utilização de códigos computacionais que realizam estimativas precisas a respeito do potencial solar no Brasil é o melhor modo de se obter dados confiáveis.

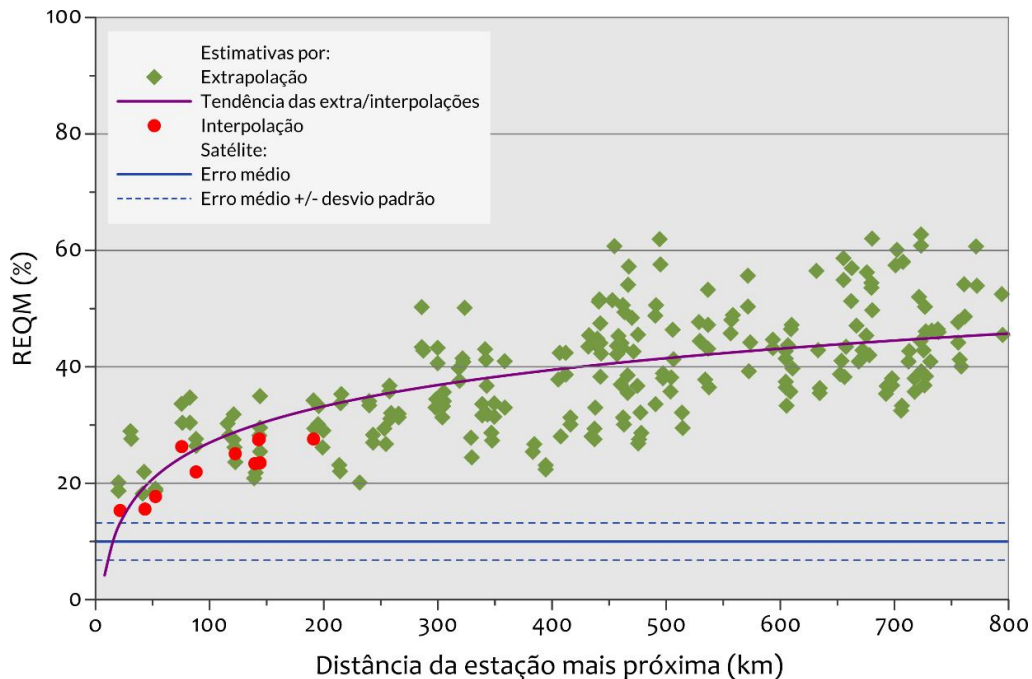


Figura 1. Incerteza típica nos dados interpolados de irradiância solar e dados obtidos através de modelos satelitais.

Fonte: Martins, 2011

Assim, a utilização de códigos computacionais (modelos) que realizam estimativas precisas a respeito do potencial solar é o melhor modo de se obter dados confiáveis. Vários modelos foram desenvolvidos de modo a obter estimativas da irradiação solar como (Dave e Canosa, 1974) que utilizaram harmônicos esféricos, (Liou, 1976) utilizando ordenadas discretas, (Lenoble, 1985) com as equações de Monte Carlo e Diferenças Finitas. Uma alternativa aos modelos citados é a utilização de métodos que geram resultados aproximados como o método de dois fluxos (Two-Stream) (Lenoble, 1985).

O modelo utilizado neste trabalho, BRASIL-SR, é um modelo físico para obtenção de estimativas da radiação solar incidente na superfície, que combina a utilização da aproximação de dois fluxos na solução da equação de transferência radiativa com o uso de informações climatológicas e parâmetros determinados a partir de imagens de satélites. Contudo, com um custo de execução serial de quase 27 horas em computadores atuais, o modelo se torna inviável para, por exemplo, realizar previsões de curtíssimo prazo ou mesmo para realização de estimativas diárias da irradiação solar. Este tipo de informação tem uso em diversos setores, tais como no planejamento de operações de sistemas

fotovoltaicos visando um melhor aproveitamento do recurso, gerenciamento de despacho em linhas de transmissão e realização de manutenção preventiva em painéis solares com uma menor perda na geração de energia, sendo inclusive de uso estratégico.

Assim, neste trabalho, buscou-se a melhoria do desempenho computacional relacionado ao tempo de processamento do modelo de transferência radiativa BRASIL-SR, que devido ao seu alto custo computacional, possui um elevado tempo de processamento em processadores convencionais. Na primeira fase deste estudo, foi realizada uma análise do desempenho da versão original do modelo – que é executada em modo serial – com o objetivo de verificar quais são os processos mais custosos computacionalmente. Posteriormente, foi utilizada a inserção das diretivas de OpenMP, permitindo a criação e o gerenciamento automático de *threads*, nas regiões importantes do código, alteração dos dados de entrada e saída (E/S), para o formato NetCDF, e a utilização da técnica de bloqueio para tentar reduzir ainda mais o tempo de processamento.

2. Metodologia

O modelo BRASIL-SR foi desenvolvido com base no modelo alemão IGMK Forschungszentrum (Stuhlmann, 1990) e posteriormente adaptado para as condições brasileiras pelo LABREN/CCST/INPE e a Universidade Federal de Santa Catarina – UFSC (Pereira, 1996), sendo o principal recurso deste grupo a quantificação do recurso energético solar. Neste trabalho, foram utilizadas 665 imagens do mês de abril de 2016, com resolução temporal de 30 minutos e resolução espacial de 0.03° em longitude e 0.05° de latitude, o que corresponde aproximadamente a 3km x 5km do ponto nadir do satélite.

O código, contendo mais de 8300 linhas, está escrito em linguagem Fortran 90 e foi separado em 7 módulos a fim de facilitar qualquer tipo de aperfeiçoamento ou alteração de algum cálculo do modelo. No módulo 1 é calculada a transmitância em céu claro, no módulo 2 a transmitância em céu parcialmente nublado, no módulo 3 é realizado o cálculo das Irradiações Global, Difusa e Direta, no módulo 4 da Irradiação Par, no módulo 5 a integral diária da irradiação Par, no módulo 6 a integral diária das Irradiações Global, Direta, Difusa e Plano Inclinado e no módulo 7 as médias mensais das irradiações Global, Direta, Difusa, Par e Plano Inclinado (Pereira, 2017).

O modelo está sendo executado no supercomputador Santos Dumont, localizado em Petrópolis, RJ, no Laboratório Nacional de Computação Científica – LNCC. Para a compilação dos códigos do modelo é necessário carregar as variáveis de ambiente e a inclusão no path dos programas e bibliotecas de forma modular utilizando o aplicativo “*module*”. Foi utilizada a biblioteca NetCDF 4.4.1 e o compilador Intel ifort 16.0.2 em ambiente computacional RedHat Linux 6.4 (LNCC, 2017).

Deste supercomputador, será utilizado os processadores Intel Xeon com 12 núcleos por unidade de processamento e cada nó computacional têm duas unidades de processamento tendo assim um total de 24 núcleos por nó. No supercomputador há um total de 12.096 núcleos multi-core, não sendo contabilizados os nós computacionais com os processadores Intel Xeon Phi e GPU NVidia. A capacidade de armazenamento principal é de 1,7 Pbytes, com sistema secundário de 640 Tbytes, e sistema operacional Linux (LNCC, 2018).

O modelo, em sua forma original, realiza suas operações de E/S em formato binário e texto. Também foi avaliada a implementação destas operações a partir de arquivos em formato NetCDF (Network Common Data Form) que é uma especificação desenvolvida pela fundação UNIDATA e oferecida sob a forma de bibliotecas utilitárias, visando o armazenamento de grandes volumes de dados. Além disso, um dos objetivos do formato é o acesso eficiente a pequenos subconjuntos de grandes conjuntos de dados, fazendo uso do acesso direto em vez de acesso sequencial, sendo muito mais eficiente quando a ordem em que os dados são lidos é diferente da ordem em que foram gravados (UNIDATA, 2018).

A paralelização do modelo foi realizada utilizando as diretivas de OpenMP (Open Multi-Processing), que é uma interface de programação multi processo de memória compartilhada em múltiplas plataformas (OpenMP, 2018). Esta forma de paralelização foi escolhida devido a dois fatores principais: (a) permite a paralelização gradativa do código sequencial original, através de alterações sucessivas e independentes em trechos do código; e (b) grande popularização atual de processadores multi-núcleo (multi-core), com custo moderado, que podem ser facilmente empregados pelas instituições interessadas em executar o modelo no futuro próximo. Uma paralelização mais ampla, para ambientes de memória distribuída, através do uso de um paradigma de troca de mensagens (por exemplo com MPI), é uma possibilidade viável para desenvolvimentos futuros.

O OpenMP foi desenvolvido pelo grupo OpenMP Architecture Review Board (ARB) – que é formado pelos maiores fabricantes de programas e componentes eletrônicos do mundo (OpenMP, 2017). O OpenMP trabalha com arquitetura *multicore* e *multithread*, conseguindo assim executar simultaneamente duas ou mais tarefas utilizando *threads*. Cada *thread* possui sua própria pilha de execução que compartilha o mesmo espaço de memória com outros *threads* do mesmo processo. Por utilizar diretivas, chamadas de sentinelas, que identificam a área que será paralelizada, o OpenMP se torna uma ferramenta muito importante quando se tem um código complexo onde há a necessidade de paralelização, com diretivas simples existem poucas mudanças necessárias no código, fazendo com que até códigos mais antigos tenham uma implementação com o OpenMP de maneira mais fácil do que utilizando MPI.

Para se paralelizar um código com OpenMP o início da área a ser paralelizada começa com uma diretiva do tipo *!\$omp parallel*, como pode ser observado abaixo.

```
!$omp parallel  
    write(*,*)"Olá"  
!$omp end parallel
```

A diretiva *!\$omp private()* é importante quando se precisa especificar uma variável que terá seu uso específico em cada *thread*. Assim, durante a execução a variável, caso seja um índice de um laço, terá um valor (posição) em cada *thread*, garantindo que o OpenMP não utilize uma posição que está sendo utilizada em outro *thread*.

Quando no código existe uma região onde há a necessidade de ser executada por um *thread* por vez, podemos utilizar a diretiva *!\$omp critical*, assim quando um *thread* estiver executando a região crítica os outros *threads* irão bloquear a execução quando alcançarem essa região, e cada uma executará a região de acordo com a ordem de chegada.

A paralelização, neste trabalho, foi realizada em cada um dos 7 módulos que compõem o modelo. O primeiro módulo é o mais custo computacionalmente tanto na versão original como na versão com dados em NetCDF, utilizando apenas 1 thread, o tempo de processamento é maior que 60 mil segundos. E o módulo 7 é o que tem menor tempo de processando, um pouco mais de 10 segundos nas duas versões do modelo.

Nos dois primeiros módulos a paralelização ocorreu no laço principal que é o responsável por chamar todas as funções que calculam as transmitâncias. Os módulos 3 e 4 foi paralelizado a região onde são chamadas as funções responsáveis por calcular as irradiações. E nos módulos 5, 6 e 7 foi paralelizado a região onde são realizadas as integrações diárias e as médias mensais das irradiações solares.

Para realizar a paralelização dos módulos foi utilizado a diretiva *!\$omp parallel do private(i,j)* como pode ser observado na Figura 2. A otimização dos laços *DO* foi realizada invertendo para que a linha seja percorrida e junto os valores próximos possam ser acessados e trazidos para a memória *cache* melhorando assim o acesso a memória e melhorando o tempo de processamento.

```

!$omp parallel do private(I,J)
DO J=1,1180
DO I=1,1784
  IF(XALT(I,J).GT.-98.0) THEN
    if (DMACC(I,J) < SSIW(I,J))then
      SSIW(I,J) = DMACC(I,J)
    endif
  CALL TRANSMIT(XLAT(J),XALB(I,J),XUMI(I,J),XALT(I,J),SSIW(I,J),TCLEAR(I,J),TDIR(I,J))
  CALL TRANSMIT(XLAT(J),XALB(I,J),XUMI(I,J),XALT(I,J),SSIW(I,J),TCLLOUD(I,J),TTDIR)
  ELSE
    TCLEAR(I,J) = -1.0
    TCLLOUD(I,J) = -1.0
    XTDIR(I,J) = -1.0
  ENDIF
END DO
END DO
!$omp end parallel do

```

Figura 2. Código do módulo 1 com a diretiva de OpenMP

Como os arquivos de E/S estão em formatos binário e texto fazendo com que a leitura e escrita consumam um tempo de processamento grande, foi realizada a conversão desses arquivos para NetCDF, melhorando assim o tempo de processamento e facilitando a visualização dos dados dos arquivos, já que os arquivos em NetCDF contém um *dataset* com informações de dimensão, variáveis e atributos, todas possuindo um nome e um número de identificação pelos quais podem ser referenciadas. Na Figura 3 podemos observar como é o *dataset* do NetCDF e como os dados ficam disposto nos arquivos em NetCDF. No *dataset* é definido o tamanho das matrizes e o tipo de informação que existe dentro do NetCDF.

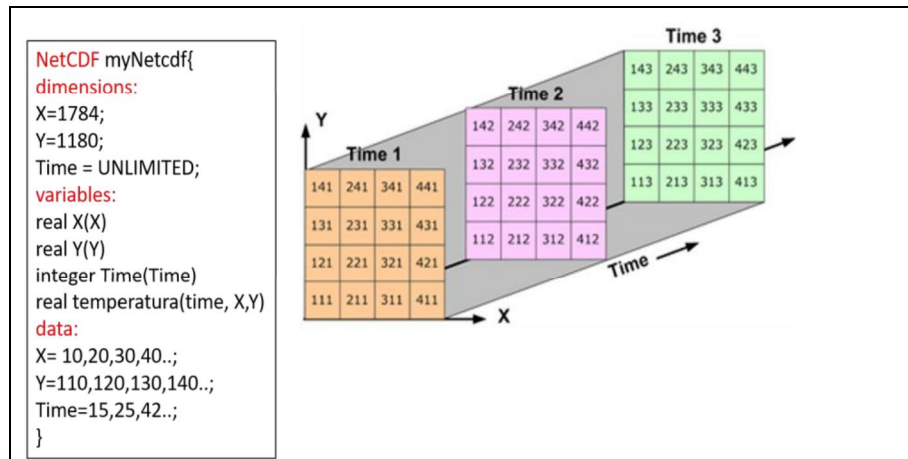


Figura 3. Dataset do NetCDF e disposição dos dados

Neste trabalho foi utilizada a técnica de blocagem, para tentar melhorar ainda mais os resultados de tempo de execução do modelo BRASIL-SR. A técnica de blocagem se baseia em agrupar os elementos adjacentes da matriz, onde as colunas adjacentes serão lidas ao mesmo tempo. Esse método é eficiente desde que a quantidade de dado lido seja capaz de ficar armazenado em *cache*. Assim, ao criar um bloco, deve se garantir que o bloco seja o maior possível minimizando o *overhead* dos novos laços e ser pequeno o bastante para ser armazenado na *cache*.

Com a blocagem, pequenos trechos são criados e alocados na memória *cache*. Os *strideI* e *strideJ* são os responsáveis pelo tamanho do bloco a ser obtido e que será armazenado na *cache* e a função *min* é utilizada para verificar se o laço chegou ao final.

Para realizar os testes com blocagem no modelo, tendo que a resolução das imagens de satélite é de 1784x1180, foi escolhido o máximo divisor comum entre eles, gerando assim matrizes quadradas de tamanho 4x4, como pode ser observado na Figura 4.

```

ni = 1784
nj = 1180
stridei = 4
stridej = 4
do kj=1,nj,stridej
do ki=1,ni,stridei
do J=kj,min(nj,kj+stridej-1)
do I=ki,min(ni,ki+stridei-1)
CALL TRANSMIT(XLAT(J),XALB(I,J),XUMI(I,J),XALT(I,J),SSIW(I,J),TCLEAR(I,J),TDIR(I,J))
CALL TRANSMIT(XLAT(J),XALB(I,J),XUMI(I,J),XALT(I,J),SSIW(I,J),TCLCLOUD(I,J),TTDIR)
ELSE
TCLEAR(I,J) = -1.0
TCLCLOUD(I,J) = -1.0
XTDIR(I,J) = -1.0
ENDIF
enddo
enddo
enddo
enddo
    
```

Figura 4. Código do modelo utilizando blocagem

Para avaliar o desempenho do modelo BRASIL-SR são utilizadas duas medidas, o ganho de desempenho e a eficiência. O ganho de desempenho é a relação entre o tempo da execução sequencial e o tempo da execução paralela. A equação para se obter essa medida é $S_p = T_i/T_p$ onde o T_i é o tempo da execução sequencial e o T_p é o tempo da execução paralela. A eficiência é a fração de tempo que os processadores estão sendo utilizados em processamento e é calculada levando em conta o número de processadores p e o ganho de desempenho, através da equação $n_p = S_p/p$, sendo que a eficiência ideal é 1. (Pacheco,1996)

3. Resultados e Discussões

Depois do processo de inserção das diretivas de OpenMP, foi verificado a existência de uma chamada de arquivo nos dois primeiros módulos. Foi inserida a diretiva `!$omp critical` para que aquela região fosse executada com apenas um `thread` por vez. Visto que a região estava ficando serial com a diretiva, o seu conteúdo foi alocado em um vetor e passado por parâmetro pelas funções.

Na Figura 6, podemos observar que o tempo de processamento melhorou à medida que a quantidade de `threads` foi aumentando no modelo BRASIL-SR original. Com um ganho de desempenho máximo de 10,9 e uma eficiência moderada (0,45), nota-se que o modelo não tem uma eficiência paralela muito boa. Isto é devido, em parte, à influência dos trechos seriais originais do modelo que não puderam ser cobertos pelas diretivas OpenMP inseridas, e assim limitaram o ganho de desempenho da paralelização, ou ainda aos efeitos dos processos de leitura/escrita de dados.

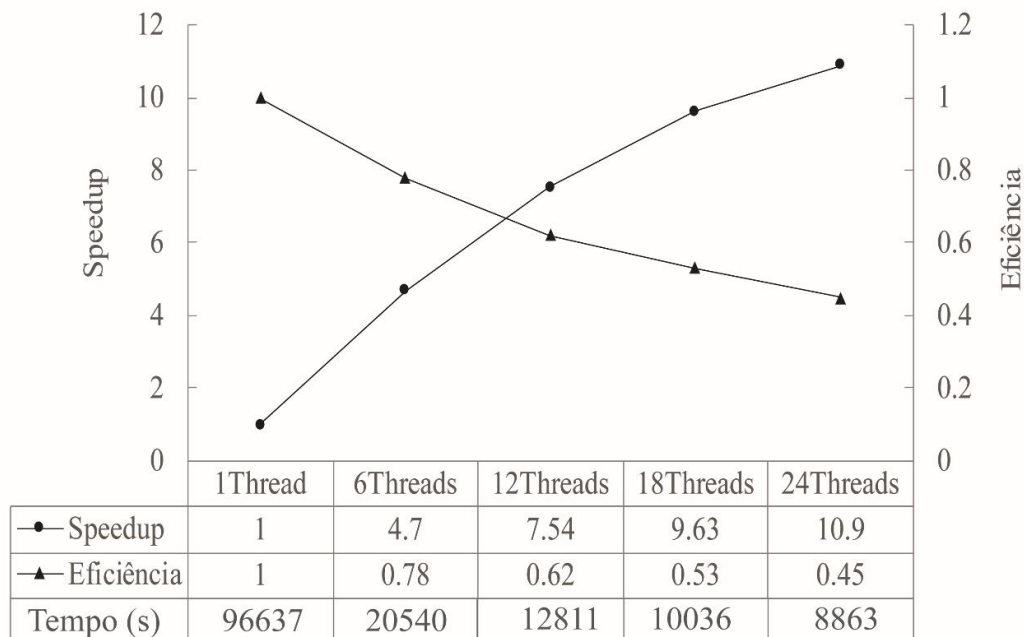


Figura 6. Tempo (segundos), Ganho de desempenho e Eficiência do Modelo BRASIL-SR

Com a versão do modelo contendo os dados em NetCDF, o tempo total de processamento diminuiu ainda mais. Na Figura 7, podemos observar a melhora do desempenho e a eficiência, até 12 *threads*, se mantendo constante e a eficiência com 24 *threads* melhorando em relação ao modelo original.

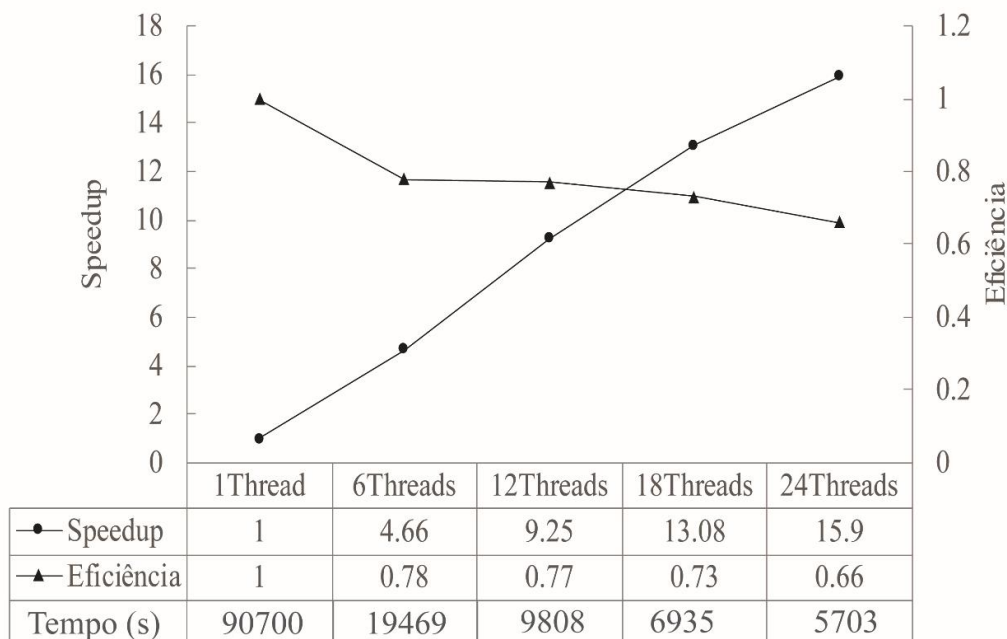


Figura 7. Tempo (segundos), Ganho de Desempenho e Eficiência do Modelo BRASIL-SR utilizando dados de E/S em formato NetCDF

Em comparação com o modelo original, podemos observar que o ganho de desempenho e a eficiência no modelo com dados em NetCDF se mostraram melhores.

Com os testes realizados, podemos afirmar que o modelo com E/S em formato NetCDF e utilizando 24 *threads* tem uma melhora de 37% no tempo comparado com sua versão serial, e melhora de 41% se comparado com o código original serial. O tempo de processamento de E/S sem NetCDF e com 24 *threads* é de 4444 segundos e com NetCDF é de 2728 segundos, mostrando uma melhora em 61% no tempo de E/S. O ganho de desempenho e eficiência também melhoraram com essa alteração. A possível razão para esta melhora é que os arquivos em NetCDF têm acesso direto podendo ser recuperados sem passar por outros dados na sequência, sendo importante quando a execução do modelo é realizada utilizando mais de um *thread*. Além das imagens de satélite e dos dados climatológicos de entrada, os dados de saída somam 4100 imagens.

A técnica de blocagem foi utilizada no modelo original e na versão com dados em NetCDF, os tempos obtidos ficaram muito próximos ou maiores em relação aos tempos obtidos nas Figuras 7 e 8. Neste teste foi definindo valores de *stride* igual 4, mas o modelo não apresentou melhoras, (veja Tabela 1). A utilização de diferentes valores de *stride* para a blocagem é um caminho a ser potencialmente explorado em estudos futuros.

Tabela 1. Tempo (segundos) da execução com a técnica de blocagem

| | 1T | 6T | 12T | 18T | 24T |
|------------------------|--------------|--------------|--------------|--------------|-------------|
| Modelo Original | 96742 | 20591 | 12168 | 10519 | 9427 |
| Modelo NetCDF | 91135 | 19869 | 9246 | 7168 | 6131 |

4. Conclusão

Neste artigo, foram apresentados resultados iniciais da paralelização do modelo de radiação solar BRASIL-SR, através de OpenMP. Este paradigma de paralelização foi escolhido, principalmente, para permitir aos usuários atuais do modelo a exploração de processadores multo-core, amplamente disponíveis atualmente. Com isto, a utilização do modelo poderia atender satisfatoriamente aos requisitos operacionais de tempo de execução hoje existentes. A extensão deste trabalho para ambientes de memória distribuída, via paralelização com MPI, é uma possibilidade para trabalhos futuros.

A paralelização do modelo BRASIL-SR, utilizando OpenMP, mostrou uma redução considerável no seu tempo de processamento com 24 *threads*. Posteriormente, serão utilizados mais *threads*, num nó computacional com maior número de núcleos, a fim de avaliar mais extensivamente a escalabilidade do desempenho e eficiência do modelo. Além disso, a implementação relativa às operações de E/S em formato NetCDF mostrou uma melhora adicional, com uma melhor eficiência paralela.

Nos testes realizados com blocagem, em que não obtivemos uma melhora no tempo de processamento, mostra-se necessária uma maior exploração com diferentes valores de *stride*, de modo a variar a quantidade de dados que são efetivamente armazenados em *cache*. Simultaneamente, deve ser conduzida uma concreta medição das taxas de acerto/erro em *cache*, de modo a caracterizar precisamente quais valores de *stride* são mais vantajosos.

Esses testes apresentados são parte do trabalho que está sendo realizado para a redução do tempo de processamento do modelo BRASIL-SR, e uma implementação futura contemplará a exploração da técnica de vetorização de laços, dado que boa parte dos cálculos realizados no modelo são efetuados dentro de laços

5. Referências

Dave, J. V.; Canosa, Z. A direct solution of the radiative transfer equation: application to atmospheric models with arbitrary vertical non-homogenities. *Journal of Atmospheric Science*, 31,1089-1101, 1974.

Laboratório Nacional de Computação Científica (LNCC), “Manual Supercomputador Santos Dumont”, Disponível: http://sdumont.lncc.br/support_manual.php?pg=support# Acesso: Janeiro/2018

Lenoble, J. Radiative transfer in scattering and absorbing atmospheres: standard computational procedures. Virginia: A. Deepak Publishing, 1985.

- Liou, K. N. On the absorption, reflection and transmission of solar radiation in cloudy atmospheres. *Journal of Atmospheric Science*, 33,798-805, 1976.
- Martins, F. R., Pereira, E. B., “Estudo Comparativo da confiabilidade de estimativas de irradiação solar para o sudeste brasileiro obtidas a partir de dados de satélite e por interpolação/extrapolação de dados de superfície”. *Revista Brasileira de Geofísica*, São Paulo, v.29, p.265-276, 2011
- OpenMP, “Parallel Programming in Fortran 95 using OpenMP”, Disponível: <https://www.openmp.org/wp-content/uploads/OpenMP-WelcomeGuide.pdf> Acesso: Janeiro/2017
- OpenMP, “Parallel Programming in Fortran 95 using OpenMP”, Disponível: https://www.openmp.org/wp-content/uploads/F95_OpenMPv1_v2.pdf Acesso: Janeiro/2018
- Pacheco, Peter S., “Parallel Programming With MPI”. Morgan Kaufmann Publisher Inc. San Francisco, CA, USA, p.418, 1996, ISBN 1-55860-339-5
- Pereira, E.B., Martins, F.R, Gonçalves, A.R., Costa, R.S., Lima, J.L., Rütther, R., Abreu, S.L., Tiepolo, G.M., Pereira, S.V. and Souza, J.G., “Atlas Brasileiro de Energia Solar”, 2.ed. São José dos Campos: INPE, 2017. 88p. ISBN 978-85-17-00090-4, 2017
- Pereira, E.B., Abreu, S.L., Stuhlmann, R., Rieland, M. and Colle, S. “Survey of The Incident Solar Radiation In Brazil by use of Meteosat Satellite Data”, *Solar Energy*, Phoenix, v. 57, n.2, p.125-132, 1996.
- Stuhlmann, R., Rieland, M. and Raschke, E. “An Improvement of the IGMK model to derive total and difuse solar radiation at the surface from satellite data”, *J. Applied Meteorology*, v29, n. 7, p.586-603, 1990.
- UNIDATA, “NetCDF 4.6.1”, Disponível: https://www.unidata.ucar.edu/software/NetCDF/docs/NetCDF_introduction.html Acesso: Janeiro/2018