

PARALELIZAÇÃO DA COMPONENTE DE SUPERFÍCIE DO MODELO HIDROLÓGICO DISTRIBUÍDO DA BIOSFERA

PARALLELIZATION OF SURFACE COMPONENT OF DISTRIBUTED BIOSPHERE HYDROLOGICAL MODEL

Nelson Leonardo Vidaurre Navarrete¹
Jônatan Dupont Tatsch²

Resumo: Modelos hidrológicos distribuídos acoplados a modelos de superfície são uma ferramenta numérica importante para avaliação de impactos das mudanças climáticas e do uso da terra. A maior disponibilidade de dados em alta resolução espacial implica em um aumento no custo computacional e de armazenamento de dados pelos modelos. A computação paralela e distribuída fornece uma alternativa eficiente para reduzir o tempo computacional tornando viável a realização de cenários, tendo em vista o grande número de simulações necessárias para a calibração dos parâmetros do modelo. Portanto, o objetivo deste trabalho foi reduzir o tempo de simulação do Modelo Hidrológico Distribuído da Biosfera (DBHM) através da paralelização do Modelo Simples da Biosfera (SiB2), que corresponde à sua componente de superfície do DBHM. O modelo de paralelização foi desenvolvido para máquinas com memória distribuída, baseado em técnicas de decomposição de domínio e faz uso da biblioteca MPI para passagens de mensagens. A independência horizontal nesta parte do modelo viabiliza uma decomposição unidimensional ou bidimensional com balanceamento de carga adequado. Os nossos testes com até 64 processadores mostraram uma boa escalabilidade. A subsequente paralelização do DBHM tornará viável a simulação do ciclo hidrológico em bacias de maior escala, em alta resolução e em um tempo computacionalmente aceitável.

Palavras-chave: Computação paralela. Modelagem hidrológica. Modelo da superfície terrestre.

Abstract: Distributed hydrological models coupled with surface models are an important numerical tool for assessing impacts of climate change and land use. The greater availability of data in high spatial resolution implies an increase in computational cost and data storage by the models. Parallel and distributed computing provides an efficient alternative to reduce computational time making feasible the realization of scenarios, considering the large number of simulations necessary for the calibration of the model parameters. Therefore, the objective of this work was to reduce the simulation time of the Biosphere Distributed Hydrological Model (DBHM) through the parallelization of the Simple Biosphere Model (SiB2), which corresponds to its DBHM surface component. The parallelization model was developed for distributed-memory machines, based on domain decomposition techniques and makes use of the MPI library for message passages. The horizontal independence in this part of the model enables a one-dimensional or two-dimensional decomposition with adequate load balancing. Our tests with up to 64 processors showed good scalability. The subsequent parallelization of the DBHM will make feasible the simulation of the hydrological cycle in larger scale basins, in high resolution and in a computationally acceptable time.

Keywords: Parallel computing. Hydrological modeling. Land surface model.

¹ Doutor em Matemática Aplicada (USP), Pós-doutorando do Programa de Pós-Graduação em Meteorologia da Universidade Federal de Santa Maria (UFSM). E-mail: nelsonvn@gmail.com.

² Doutor em Meteorologia (USP), Professor do Programa de Pós-Graduação em Meteorologia da Universidade Federal de Santa Maria (UFSM). E-mail: jdtatsch@gmail.com

1 INTRODUÇÃO

O tempo e o clima influenciam profundamente a disponibilidade de recursos hídricos e energéticos para a sociedade. As atividades humanas, alteram os padrões de uso da terra, causando mudanças na composição da atmosfera e nas propriedades da superfície terrestre que conseqüentemente alteram os fluxos de água, energia e CO₂ entre a superfície e a atmosfera (FOLEY *et al*, 2005). O ciclo hidrológico é o mecanismo natural de renovação da água entre os componentes do sistema terrestre e os modelos hidrológicos são as ferramentas de modelagem numérica que representam os processos hidrológicos, como por exemplo: a infiltração, o escoamento superficial, subterrâneo e a evapotranspiração. As simulações para cenários de uso da terra e de mudanças climáticas fornecem uma ferramenta analítica para a tomada de decisões sobre a gestão dos recursos hídricos, ambientais (REFSGAARD & ABBOT 1996) e para avaliação dos impactos das mudanças da legislação ambiental brasileira.

Entre os modelos hidrológicos distribuídos acoplados a modelos de superfície utilizados para estudar as relações entre a variabilidade climática e as atividades humanas podemos citar o modelo Hidrológico Distribuído da Biosfera (DBHM) (TANG *et al*, 2006, TANG *et al*, 2008), utilizado para avaliar os efeitos da variabilidade climática (precipitação e da temperatura do ar) e da irrigação sobre os processos hidrológicos. O DBHM foi aperfeiçoado e adaptado para as condições subtropicais para avaliar os efeitos das mudanças no uso da terra no clima e no ciclo hidrológico na Bacia Hidrográfica do Rio de Mogi-Guaçu no Sudeste do Brasil (TATSCH 2012).

O DBHM é um modelo espacialmente distribuído, determinístico e contínuo no tempo, desenvolvido inicialmente para aplicação em bacias de escala continental. O DBHM integra processos hidrológicos e processos de transferência solo-vegetação-atmosfera na escala de bacia hidrográfica (TANG *et al*, 2006). O DBHM incorpora o Modelo Simples da Biosfera Revisado (SiB2, SELLERS *et al*, 1996a) e o Modelo Hidrológico Distribuído Baseado na Geomorfologia da bacia (GBHM, Yang *et al*, 1998), permitindo o uso de: dados de satélites para descrição da fenologia e do estado da vegetação; Modelos

Digitais de Elevação do Terreno (MDET) para descrever as características geomorfológicas da bacia. O SiB2 resolve o balanço hídrico em cada ponto de grade do modelo, determinando a evapotranspiração (ET), a precipitação efetiva, os escoamentos superficial, sub-superficial e a drenagem para o aquífero. O SiB2 inclui a representação de ambos mecanismos de geração de escoamento, Hortoniano e Dunniano, e um reservatório de água subterrânea. O modelo também simula os fluxos de energia (calor sensível, saldo de radiação, albedo, radiação da onda longa emitida pela superfície) e de CO₂, o que permite avaliar os *feedbacks* biogeoquímicos e biogeofísicos de interação entre a vegetação e a atmosfera (BONAN 2008). Adicionalmente, o submodelo de superfície permite determinar a eficiência do uso da água na escala de bacia hidrográfica, um índice crítico para explorar a adaptação da vegetação as condições climáticas.

O DBHM baseia-se nas características geomorfológicas da bacia para simular o escoamento das vertentes até a rede de drenagem (incluindo a interação aquífero-rio, TOWNER 1975) e o escoamento nos canais dos rios, ambos descritos por um modelo de onda cinemática unidimensional (LIGHTHILL 1955). A discretização das sub-bacias é feita segundo o esquema de Pfafstetter (VERDIN 1999). As sub-bacias são discretizadas em intervalos de escoamento (IE) (YANG *et al*, 1998) que representam pontos de grades com distâncias similares ao exutório de cada sub-bacia. Uma seção transversal de rio é atribuída a cada IE e escoamento total de água é acumulado para a seção de rio e propagado nos IE de cada sub-bacia seguindo a rede de drenagem até a foz da bacia.

O desempenho do DBHM depende da disponibilidade de dados com alta resolução espacial. Entretanto, o aumento da resolução espacial implica em maior custo computacional e armazenamento de dados. Desta forma, a computação paralela e distribuída é necessária para reduzir o tempo computacional e distribuir o problema, de forma que os dados possam ser contidos na memória (APOSTOLOPOULOS & GEORGAKAKOS 1997; DOWD & SEVERENCE 1998, p. 4; WIKINSON 1999, p. 5) e o número de simulações

necessárias para a calibração dos parâmetros do modelo sejam realizadas em tempo viável. Portanto o objetivo deste trabalho foi reduzir o tempo de simulação do DBHM através da paralelização do SiB2 que corresponde à sua componente de superfície do DBHM.

2 MATERIAIS E MÉTODOS

2.1 Instrumentalização do modelo

Uma característica fundamental que todo modelo deve ter é a capacidade de calcular o tempo consumido pelas suas principais etapas. Isto é mais relevante durante o desenvolvimento e aperfeiçoamento do modelo, uma vez que a análise destes tempos permite identificar os processos mais custosos e realizar modificações visando obter um melhor desempenho do mesmo. Por estes motivos, decidiu-se incorporar ao DBHM o módulo *times*, que permite tomar e acumular os tempos, de relógio de parede (*wall-clock time*) e em segundos, de até 70 etapas diferentes, gerando ao final da execução um relatório dos mesmos. O tempo transcorrido em cada etapa é calculado por meio da sub-rotina Fortran 90 padrão *system_clock*, tomando o tempo imediatamente antes de iniciar cada etapa, imediatamente após a conclusão da mesma e subtraindo ambos os tempos. A tomada dos tempos pode ser habilitada ou desabilitada por meio de uma variável lógica que é lida de um arquivo de configuração.

A Tabela 1, montada a partir do relatório gerado pelo módulo *times*, mostra as quatro etapas do modelo DBHM que consomem mais tempo, para uma simulação da bacia de Mogi-Guaçu durante um período de 14 anos. Pode-se observar que aproximadamente 50% do tempo total de simulação corresponde ao modelo SiB2 (*sib2_grid*) e 29% à desagregação temporal das forçantes atmosféricas do modelo da escala diária para horária (*downscale*). Devido a estes tempos e ao fato do SiB2 ser executado de forma independente em cada ponto da malha, decidiu-se paralelizar as chamadas do SiB2 e realizar a etapa da desagregação temporal como uma etapa de pré-processamento.

Tabela 1 – Tempos (s) das etapas que consomem mais tempo no DBHM.

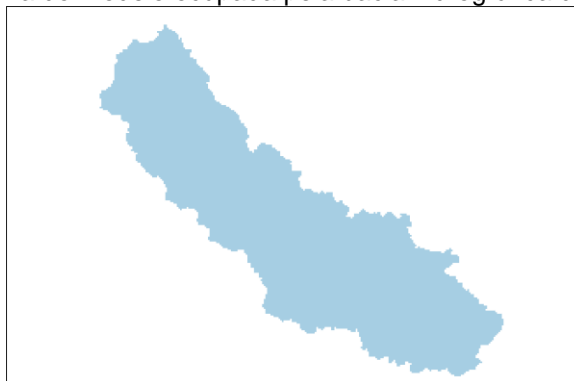
Etapa	Tempo
sib2_grid	12865,42 (51,0%)
downscale	7229,98 (28,7%)
sib_river	2077,29 (8,2%)
wr_bal	1734,38 (6,9%)
Total	25201,71 (100,0%)

2.2 Representação de malhas regulares

Como a etapa do SiB2 pode ser executada de forma independente e em qualquer ordem em cada ponto da malha, não é necessário se ter uma malha estruturada durante esta etapa; é suficiente, por exemplo, ter uma lista com os pontos a serem processados e os valores dos campos necessários para seu processamento. Adicionalmente, como se pode observar da Figura 1, em geral a bacia não ocupa todo o domínio espacial da malha e assim é necessário utilizar uma máscara para indicar quais pontos pertencem ou não à bacia. Menos da metade dos pontos da malha pertencem à bacia para o exemplo da bacia de Mogi-Guaçú (Tabela 2), portanto menos da metade da memória alocada pelo modelo é efetivamente utilizada.

Assim se criou uma estrutura de dados que permita descrever malhas irregulares, considerando apenas os pontos que pertencem à bacia.

Figura 1 – Área da malha do modelo ocupada pela bacia hidrográfica do Rio Mogi Guaçú.



Fonte: elaborado pelos autores, 2016.

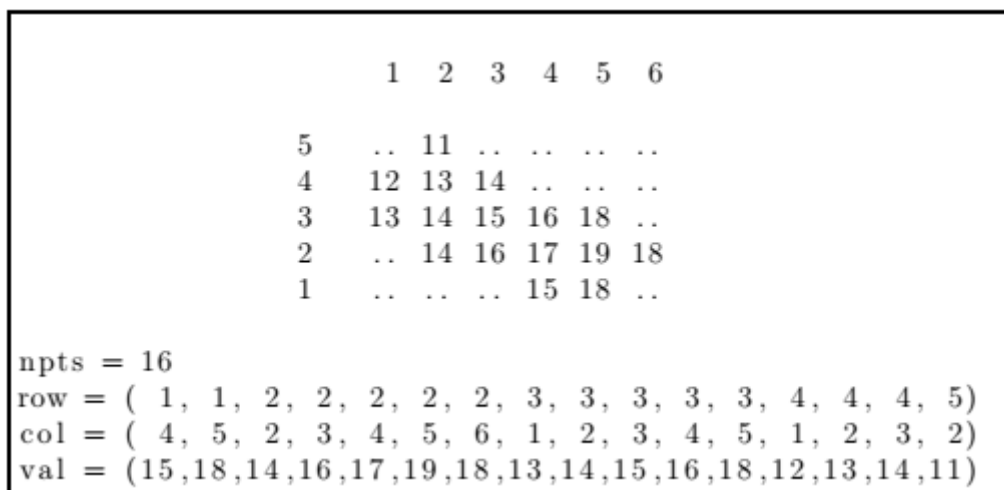
Este descritor de malhas deve garantir uma conversão rápida e simples de uma malha estruturada para uma malha nesta nova disposição (Figura 2) e vice-versa.

Tabela 2 – Dimensões da malha e quantidade de células dentro da bacia hidrográfica do Rio Mogi-Guaçu.

linhas	colunas	Total de células	células dentro da bacia hidrográfica
230	270	62100	18367 (29,6 %)

Fonte: elaborado pelos autores, 2016.

Figura 2 – Exemplo minimalista do descritor de malha e de campo *val* (valores).



Fonte: elaborado pelos autores, 2016.

Esta nova estrutura deve ser favorável à sua paralelização, isto é, à sua decomposição entre várias máquinas. Baseado em métodos para armazenar matrizes esparsas (BARRET *et al.*, 1994), o descritor de malhas proposto é composto pelo número *npts* que indica o número de pontos na bacia e pelos vetores *row* e *col* que armazenam a posição da fila e coluna dos mesmos. Os valores de campos nesses pontos podem ser armazenados utilizando vetores, como mostrado na Figura 2.

As vantagens desta nova estrutura são evidentes. Primeiro, os vetores *row* e *col* somente necessitam ser declarados uma vez. Segundo, o vetor *val* armazena valores do campo apenas para pontos que pertencem à bacia. Terceiro, a varredura dos pontos é simplificada, sem a necessidade de verificar se o ponto está contido ou não na área da bacia hidrográfica (Figura 3). Quarto, a conversão entre a estrutura de dados nova e a original é imediata. Por último, a nova estrutura de dados é simples de paralelizar (ver Figura 4).

Figura 3 – Varredura de uma malha utilizando a nova estrutura de dados.

```
do k = 1, npts
  call do_some_work(col(k), row(k), val(k))
end do
```

Fonte: elaborado pelos autores, 2016.

Figura 4 – Algoritmo de decomposição do domínio.

```
! All variables are integers
q = npts / nrank
r = mod(npts, nrank)

p2 = nmax
do k = nrank - 1, 0, -1
  p1 = p2 - q + 1
  if (r > 0) then
    p1 = p1 - 1
    r = r - 1
  end if
  call create_desc(p1, p2, row(p1), pcol(p1))
  p2 = p1 - 1
end do
```

Fonte: elaborado pelos autores, 2016.

2.3 Sub-rotinas de comunicação

Após a paralelização da chamada do SiB2, a integração do DBHM ocorreu da seguinte forma: o processo mestre ou root executa as etapas do modelo até atingir a chamada do SiB2. Neste ponto, todos os processos sabem quais pontos da malha devem processar. Entretanto, como o processo root é o único que tem acesso aos dados de entrada, é necessário que este comunique aos outros processos os campos necessários antes da execução do SiB2. De forma análoga, ao finalizar a etapa do SiB2, os resultados são enviados ao processo root para a realização das demais etapas da integração do modelo. Aproximadamente 70 variáveis (malhas) devem ser comunicadas de formas diferentes de acordo com a escala de tempo: única, anual, mensal, diária e a cada iteração. Em geral as comunicações dividem-se em duas categorias: o processo root envia dados aos outros processos ou o processo root recebe dados dos outros processos. A Figura 5 mostra o algoritmo da primeira categoria de comunicação enquanto a Figura 6 mostra o algoritmo da segunda.

Figura 5 – Algoritmo de comunicação: *root* envia dados aos outros processadores.

```
if (myrank = root) then    ! ROOT sends data
  do k = 1, nrank -1
    call fill_buffer(buf)
    call mpi_send(buf, k)
  end do
else    ! RANKS receive data
  call mpi_recv(buf, root)
end if
```

Fonte: elaborado pelos autores, 2016.

Figura 6 – Algoritmo de comunicação: *root* recebe dados dos outros processadores.

```
if (myrank = root) then    ! ROOT receives data
  do k = 1, nrank -1
    call mpi_irecv(buf, k)
  end do
  do k = 1, nrank -1
    call mpi_waitany(status)
    q = status(mpi_source,1)
  end do
else    ! RANKS send data
  call fill_buffer(buf)
  call mpi_send(buf, root)
end if
```

Fonte: elaborado pelos autores, 2016.

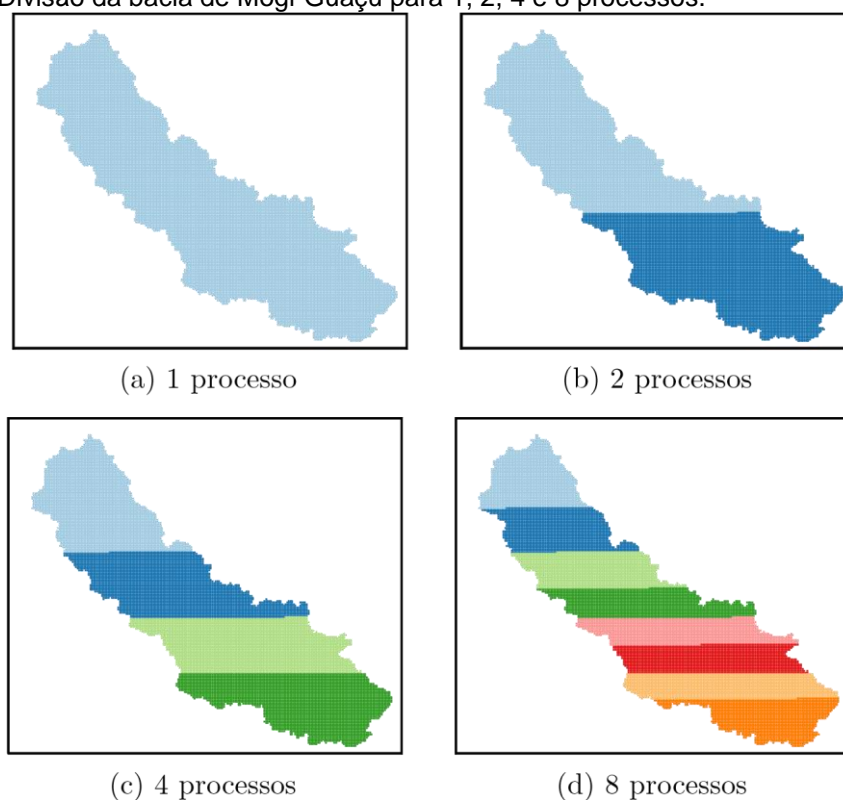
2.4 Instrumentalização do DBHM em paralelo

Por último, o módulo *times* foi ampliado de forma que cada processo possa medir de forma local os tempos consumidos pelas suas etapas e armazená-los em uma tabela. Uma vez concluída a simulação, cada processo envia a sua tabela local com os tempos para o processo *root*, que gera um relatório dos tempos de cada etapa. De forma análoga à versão sequencial, a tomada dos tempos pode ser habilitada ou desabilitada por meio de uma variável lógica lida pelo processo *root* de um arquivo de configuração e comunicada a todos os processos.

2.5 Testes de desempenho

Foram realizados dois testes de desempenho do modelo. O primeiro teste (teste1) consistiu na simulação do pDBHM (DBHM paralelizado) com dados horários em precisão dupla para a bacia hidrográfica do Rio Mogi-Guaçu no período de 1995 a 1996. A malha possui 230 linhas, 270 colunas e um total de 62100 pontos ou células, das quais 18367 correspondem à bacia, tal como indicado na Tabela 2. A simulação foi executada em um computador com processador Intel i7-3770 (3.40GHz) e memória RAM de 16Gb. O compilador utilizado foi o Intel Ifort 16.0.1 e foram realizadas simulações com 1, 2, 4 e 8 processos. A Figura 7 mostra as regiões da bacia onde cada processo executa a etapa do SiB2 e os resultados deste teste são apresentados nas Tabelas 3 e 4.

Figura 7 – Divisão da bacia de Mogi-Guaçu para 1, 2, 4 e 8 processos.



Fonte: elaborado pelos autores, 2016.

O segundo teste (teste 2) consistiu em realizar novamente o teste 1 porém utilizando agora o computador *cf01* do Laboratório de Matemática Aplicada (LabMAP) do Instituto de Matemática e Estatística (IME) da Universidade de São Paulo (USP). Este computador possui quatro processadores AMD Opteron 6376 (2,3GHz/3,2GHz) com 16 núcleos cada,

totalizando 64 núcleos, e 256 Gb de memória RAM. Foram realizadas simulações com 1, 2, 4, 8, 16, 32 e 64 processos. Os resultados do teste 2 são apresentados nas Tabelas 5, 6 e 7 e Figuras 8 e 9, descritas a seguir.

3 RESULTADOS

Em relação ao teste 1, a Tabela 3 mostra os tempos consumidos pelas principais etapas do modelo, quando executado em um único processo. O tempo consumido pelo SiB2 (*sib2_grid*) corresponde aproximadamente a 70% do tempo total, enquanto que a etapa *sib_river* (propagação da água na rede de drenagem) e a leitura de dados consumiram em torno de 11% do tempo de simulação. Destaca-se também o surgimento de novas etapas, como a comunicação entre os processos, que para o caso de um único processo é zero; e a cópia local, que consiste na conversão dos campos da estrutura original para a nova e vice-versa.

Tabela 3 – Tempos (s) das etapas que consomem mais tempo no pDBHM, para um processo. Entre parênteses apresenta-se o tempo relativo em %.

Etapa	Tempo	
Pré-processamento	1,43	(0,03%)
Leitura de dados	599,83	(11,04%)
<i>sib2_grid</i>	3773,87	(69,47%)
<i>sib_river</i>	607,20	(11,18%)
<i>wr_bal</i>	277,62	(5,11%)
Comunicação	0,00	(0,00%)
Cópia local	69,10	(1,27%)
Total	5432,03	(100,00%)

Fonte: elaborado pelos autores, 2016.

A Tabela 4 mostra os tempos gastos pelas principais etapas, para 1, 2, 4 e 8 processos. O SiB2 escala razoavelmente bem para 2 e 4 processos, enquanto para 8 processos o desempenho sofre uma queda. Uma possível explicação para a queda de desempenho com 8 processos pode ser o fato do computador ter quatro núcleos (*cores*) físicos, cada um destes precisa realizar o trabalho de dois processos. Ocorreu um aumento no tempo de comunicação, já que conforme o número de processos aumenta o processo root precisa

enviar/receber dados aos/dos outros processos. Também houve uma diminuição do tempo de cópia local, pois quanto maior o número de processos, menor a quantidade de dados processados por cada um deles e conseqüentemente menor a quantidade de dados convertida pelo processo *root* da estrutura original para a nova e vice-versa.

Para confirmar esta suspeita foi realizado o teste 2. A Tabela 5 mostra a porcentagem do tempo total de simulação do maior tempo empregado pelos processos nas principais etapas do modelo. Destaca-se a porcentagem do tempo gasto pelo SiB2, que diminuiu de 75,5% (para um processo) para 5,2% (com 64 processos). Observa-se também que as porcentagens dos tempos gastos pelas etapas *sib_river* e *wr_bal* aumentaram conforme o aumento do número de processos. Isto é esperado já que essas etapas não são executadas em paralelo. Conforme esperado, a porcentagem do tempo de comunicação aumentou conforme o aumento do número de processos.

A Tabela 6 mostra os tempos totais de simulação para 1, 2, 4, 8, 16, 32 e 64 processos. A Figura 8 mostra o *speedup* ($S(n)$) definido como o tempo gasto por um processo dividido pelo tempo gasto por n processos. Observou-se um desempenho insatisfatório, porque unicamente a etapa do SiB2 foi paralelizada.

Tabela 4 – Tempos (s) das principais etapas do pDBHM, para 1, 2, 4 e 8 processos. Valores em azul correspondem à etapa executada em paralelo (SiB2) e em preto às executadas unicamente pelo processo *root*. Com corresponde à comunicação e nproc corresponde ao número de processos.

nproc	Tempo total	Leitura	sib2_grid	sib_river	wr_bal	Com.	Cópia
1	5432,0 (100)	599,8	3773,9 (100)	607,2	277,6	0,0	69,1
2	3582,7 (66)	516,1	1872,6 (50)	608,2	278,7	66,0	37,7
	3581,6 (66)	0,0	1877,3 (50)	0,0	0,0	55,3	0,0
4	3051,6 (56)	542,8	1197,2 (32)	684,4	313,4	115,8	26,9
	3050,5 (56)	0,0	956,0 (25)	0,0	0,0	95,5	0,0
	3050,5 (56)	0,0	1113,7 (30)	0,0	0,0	104,9	0,0
	3050,5 (56)	0,0	1036,6 (27)	0,0	0,0	91,2	0,0
8	2815,2 (52)	572,0	745,6 (20)	761,1	356,4	162,1	18,9
	2813,4 (52)	0,0	773,6 (20)	0,0	0,0	131,0	0,0
	2813,4 (52)	0,0	730,0 (19)	0,0	0,0	140,9	0,0

2813,4 (52)	0,0	725,8 (19)	0,0	0,0	144,2	0,0
2813,4 (52)	0,0	730,0 (19)	0,0	0,0	132,5	0,0
2813,4 (52)	0,0	741,6 (20)	0,0	0,0	136,7	0,0
2813,4 (52)	0,0	759,7 (20)	0,0	0,0	139,1	0,0
2813,4 (52)	0,0	766,8 (20)	0,0	0,0	137,1	0,0

Fonte: elaborado pelos autores, 2016.

Tabela 5 – Porcentagem máxima do tempo total consumidos pelas principais etapas para 1, 2, 4, 8, 16, 32 e 64 processos. Valores em azul correspondem à etapa executada em paralelo (SiB2) e em preto às executadas unicamente pelo processo *root*.

Etapa/nproc	1	2	4	8	16	32	64
<i>sib2_grid</i>	75,5	60,5	41,9	29,3	17,8	10,0	5,2
<i>sib_river</i>	15,3	21,7	33,4	40,7	48,2	53,5	55,8
<i>wr_bal</i>	4,5	6,7	9,7	12,5	14,7	16,1	17,0
Leitura de dados	2,6	3,9	5,5	7,4	8,3	8,7	9,0
Comunicação	0,0	1,8	3,9	5,4	7,1	8,2	10,1
Cópia local	1,4	1,1	0,8	0,5	0,3	0,2	0,1

Fonte: elaborado pelos autores, 2016.

Tabela 6 – Tempo total de simulação para 1, 2, 4, 8, 16, 32 e 64 processos.

nproc	Tempo (s)
1	10328,0
2	7886,1
4	4741,5
8	4202,3
16	3749,3
32	3883,5
64	3805,0

Fonte: elaborado pelos autores, 2016.

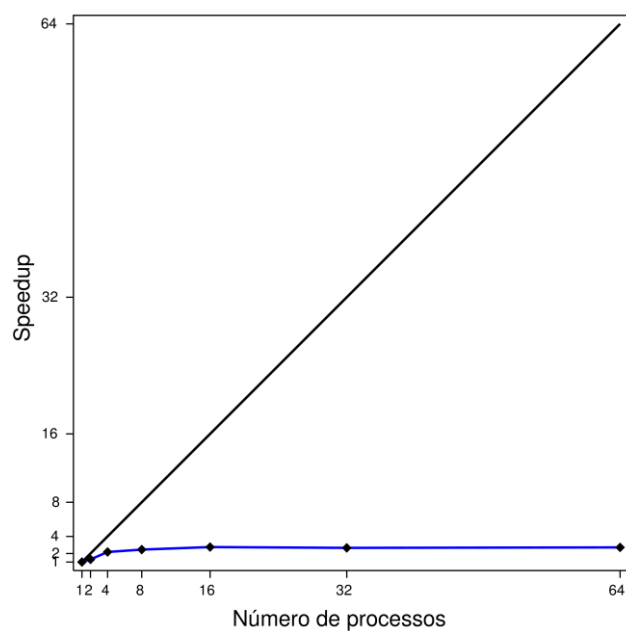
Finalmente, a Tabela 7 mostra os tempos gastos pelo SiB2, para 1, 2, 4, 8, 16, 32 e 64 processos. A Figura 9 mostra o *speed-up* relativo desta etapa. Neste caso, observa-se um escalonamento satisfatório, o que indica que o SiB2 foi paralelizado de forma eficiente para esta aplicação.

Tabela 7 – Tempo máximo consumido pelo SiB2 para 1, 2, 4, 8, 16, 32 e 64 processos.

nproc	Tempo (s)
1	7797.0
2	4774.1
4	1985.0
8	1232.0
16	669.2
32	387.7
64	196.2

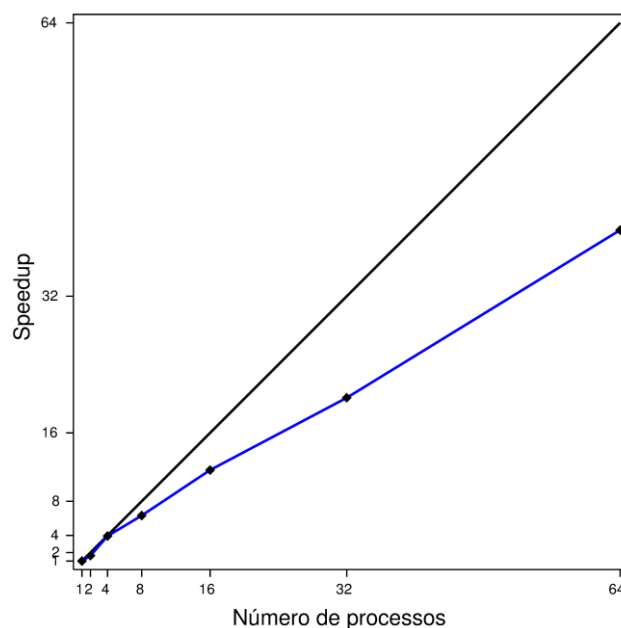
Fonte: elaborado pelos autores, 2016.

Figura 8 – *Speed-up* do pDBHM.



Fonte: elaborado pelos autores, 2016.

Figura 9 – *Speed-up* relativo da etapa *sib2_grid* do pDBHM.



Fonte: elaborado pelos autores, 2016.

4 CONCLUSÕES

O descritor de malhas irregulares proposto mostrou-se eficiente e favorável à decomposição do domínio entre várias máquinas. Os resultados obtidos indicaram um escalonamento satisfatório, indicando que a execução em paralelo do submodelo de superfície SiB2 foi realizado de forma eficiente. Uma continuação deste trabalho será o aperfeiçoamento do DBHM por meio da paralelização da etapa de propagação de água na rede de drenagem (*sib_river*).

5 AGRADECIMENTOS

O primeiro autor agradece ao Laboratório de Matemática Aplicada (LabMAP) do IME-USP pelo fornecimento dos recursos computacionais e à CAPES pelo apoio via uma bolsa institucional PNPd do Programa de Pós-graduação em Meteorologia da Universidade Federal de Santa Maria (UFSM). Os autores agradecem à FAPERGS (Proj. 1988-2551/13-3) e ao CNPQ (Proj. 461130/2014-9) pelo financiamento dos projetos de pesquisa.

REFERÊNCIAS

- APOSTOLOPOULOS, T. K.; GEORGAKAKOS, K. P. Parallel computation for streamflow prediction with distributed hydrologic models. **Journal of Hydrology**, v. 197, p. 1–24, 1997.
- BARRET, R, et al. **Templates for the solution of linear systems: building blocks for iterative methods**, SIAM, 1994.
- BONAN, G. B. Forests and climate change: Forcings, feedbacks, and the climate benefits of forests. **Science**, v. 320, p. 1444-1449, 2008.
- DOWD, K.; SEVERANCE, C. **High Performance Computing**, 2, ed, O'Reilly, 1998,
- FOLEY, J. A. et al. Global consequences of land use. **Nature**, v. 309, p. 570–574, 2005.
- LIGHTHILL, M. J.; WHITHAM, G. B. On kinematic waves, i, flood movement in long rivers. **Proceedings of the Royal Society A**, v. 229, p. 281–316, 1955.
- REFSGAARD, J. C.; ABBOTT, M. B. The role of distributed hydrological modelling in water resources management. In: **Distributed Hydrological Modelling**, Kluwer Academic Publishers, 1996, cap.
- SELLERS, P. J. et al. A revised land surface parameterization (SiB2) for atmospheric GCMs, Part I: Model formulation. **Journal of Climate**, v. 9, p. 676-705, 1996.
- TANG, Q. A.; OKI, T.; KANAE, S. A Distributed Biosphere-Hydrological Model (DBHM) for Large River Basins. **Annual Journal of Hydraulic Engineering**, v. 50, p. 37-42, 2006.
- TANG, Q. et al. Hydrological cycles change in the yellow river basin during the last half 20th century. **Journal of Climate**, v. 21, p. 1790-1806, 2008.
- TATSCH, J. D. **Controle do uso da terra no clima e no regime hidrológico da bacia do rio Mogi-Guaçu, SP**, Tese (Doutorado) – Universidade de São Paulo, 2012.
- TOWNER, G. D. Drainage of groundwater resting on a sloping bed with uniform rainfall. **Water Resources Research**, v. 11, n. 1, p. 144-147, 1975.
- VERDIN, K.; VERDIN, J. A topological system for delineation and codification of the earth's river basins. **Journal of Hydrology**, v. 218, 1999.
- WANG, H. et al. A common parallel computing framework for modeling hydrological processes of river basins. **Parallel Computing**, v. 37, 2011.
- WILKINSON, B.; ALLEN, M. **Parallel programming, Techniques and applications using networked workstations and parallel computers**. Prentice Hall, 1999.
- YANG, D.; HERATH, S.; MUSIAKE, K. Development of a geomorphology-based hydrological model for large catchments. **Annual Journal of Hydraulic Engineering**, v. 42, p. 169-174, 1998.