Desempenho da comunicação MPI *Shared Memory* no Modelo Meteorológico BRAMS

Carlos R. de Souza[1], **Jairo Panetta**[2], **Stephan Stephany**[3]

[1]Centro de Previsão de Tempo e Estudo Climáticos (CPTEC)
Instituto Nacional de Pesquisas Espaciais (INPE) – Cachoeira Paulista, SP – Brasil

[2]Divisão de Ciência da Computação (IEC)
Instituto Tecnológico de Aeronáutica (ITA) – São José dos Campos, SP – Brasil

[3]Laboratório Associado de Computação e Matemática Aplicada (LAC)
Instituto Nacional de Pesquisas Espaciais (INPE) – São José dos Campos, SP – Brasil

carlos.souza@inpe.br, jairo.panetta@gmail.com, stephan.stephany@inpe.br

Abstract. *The regional meteorological model BRAMS is executed operationally at CPTEC/INPE in a supercomputer composed of nodes with multicore processors. Its parallel programming employs the message-passing library MPI, being the model domain divided among computational nodes and also among the cores of each node. BRAMS model uses two-sided communication with the standard non-blocking asynchronous functions. However, the recent version 3.0 of MPI supports the new shared memory one-sided communication in order to optimize communication between processes executed in the same computational node. This work evaluates the communication performance of this new functionality in the parallel execution of the BRAMS model.*

Resumo. *O modelo meteorológico regional BRAMS é executado operacionalmente no CPTEC/INPE num supercomputador composto por nós com processadores multicore. Sua programação paralela é feita com a biblioteca de comunicação por troca de mensagens MPI, sendo o domínio do modelo dividido entre nós e também internamente a cada nó, geralmente com uso da comunicação convencional bilateral assíncrona e sem bloqueio. Entretanto, a recente versão 3.0 do MPI disponibiliza a nova comunicação unilateral de memória compartilhada para otimizar a comunicação entre processos executados num mesmo nó computacional. Este trabalho avalia o desempenho de comunicação dessa nova funcionalidade na execução paralela do modelo BRAMS.*

1. Introdução

Publicada em 1994, a primeira versão da biblioteca de comunicação por troca de mensagens *Message Passing Interface* (MPI) suporta comunicação entre dois processos, denominada comunicação ponto a ponto ou *comunicação bilateral*. Esta forma de comunicação requer que os dois processos envolvidos emitam comandos MPI (Send/-Recv). Em 1997, a versão MPI 2.0 acrescentou uma nova forma de comunicação, denominada *comunicação unilateral* (*One Sided Communication*), em que apenas um processo emite comandos MPI (Put/Get) para acessar uma janela da área de memória do processo

alvo, utilizando RMA (*Remote Memory Access*). Publicada em 2015 [mpi 2015], a versão MPI 3.0 introduz uma nova forma de comunicação unilateral exclusiva para processos residentes no mesmo nó computacional, denominada *comunicação de memória compartilhada* (*Shared Memory*, ou SHM). Nessa comunicação um processo acessa diretamente a memória de outro processo residente no mesmo nó por meio de ponteiros em C ou Fortran sem a necessidade de emitir comandos MPI. Potencialmente esta nova funcionalidade exploraria memória compartilhada sem a necessidade de criar procedimentos *thread-safe*, requeridos por OpenMP.

Este trabalho avalia o desempenho da comunicação MPI SHM comparando-a com a comunicação bilateral convencional MPI quando aplicadas ao modelo meteorológico BRAMS (*Brazilian developments on the Regional Atmospheric Modeling System* [Freitas et al. 2016]), utilizando nós de memória compartilhada de uma máquina Cray.

2. O modelo meteorológico BRAMS

O BRAMS é um modelo meteorológico de previsão numérica regional de tempo, desenvolvido pelo Centro de Previsão de Tempo e Estudos Climáticos do Instituto Nacional de Pesquisas Espaciais (CPTEC/INPE) em conjunto com outros institutos de pesquisas, a partir do modelo norte-americano RAMS que foi adaptado para tratar os problemas ambientais e sistemas atmosféricos locais da América do Sul. Este modelo atualmente é mantido, atualizado e distribuído pelo CPTEC/INPE e também é utilizado em suas operações diárias de previsões de tempo e em suas atividades de pesquisa [Freitas et al. 2016]. O BRAMS é escrito em sua maior parte em linguagem Fortran-2003 e com alguns módulos escritos em linguagem C, com aproximadamente 350.000 linhas de código. Seu paralelismo é essencialmente MPI por decomposição de domínio feita no plano horizontal, ou seja, todos os níveis verticais de cada ponto do domínio horizontal estão totalmente contidos em um processo MPI. O domínio horizontal é particionado entre os processos MPI, contendo um subdomínio por processo. Em sua principal etapa, os processos entram no laço principal e integram as equações ao longo de sucessivos *timesteps*. As equações diferenciais parciais são discretizadas pelo método das Diferenças Finitas e resolvidas pelo cálculo de estênceis de 5 pontos aplicados em seu próprio subdomínio. Então, a cada iteração do laço principal (*timestep*) cada ponto de grade da malha discretizada é atualizado pela combinação linear de seu próprio valor e dos valores dos 4 pontos vizinhos. Na atualização dos pontos nas bordas de cada subdomínio, cada processo precisa dos valores das fileiras de pontos vizinhos, que foram atualizadas pelos processos vizinhos. Este problema é resolvido adicionando-se uma fileira extra de pontos de grade ao longo das bordas de cada subdomínio, constituindo a *ghost zone*. Assim surge a necessidade de comunicação entre os processos para que cada processo envie e receba de seus vizinhos os valores correspondentes às bordas necessárias para atualização de sua própria grade.

3. A comunicação unilateral por memória compartilhada

A comunicação unilateral por memória compartilhada (SHM) aplica-se a processos residentes num mesmo nó, os quais podem fazer leituras e escritas diretas numa janela da memória compartilhada [Gropp 2016]. Como um programa MPI pode utilizar vários nós computacionais, foi necessária a criação de funções específicas para identificar e mapear os processos que estão no mesmo nó. Essa identificação é necessária para particionar o comunicador global em sub-comunicadores disjuntos e específicos a cada nó de memória

compartilhada. Cada processo utiliza funções específicas do MPI SHM para criar a janela de memória compartilhada no seu sub-comunicador e para obter endereço nesta janela que corresponde à parte de cada vizinho, armazenando tais endereços em ponteiros locais a cada processo.

Na comunicação unilateral RMA, leituras e escritas são feitas através de funções específicas RMA, MPI_Put e MPI_Get, sendo os processos envolvidos não necessariamente locais a um nó, enquanto que na comunicação unilateral SHM, leituras e escritas são feitas diretamente, mas os processos tem que ser locais ao nó. A janela de memória compartilhada pode ser criada em endereços contíguos ou não contíguos de memória. Neste último caso, a parte de cada processo na janela fica mais próxima de sua parte na memória física, o que geralmente otimiza o desempenho de acesso à memória, dependendo da arquitetura de memória considerada [mpi 2015]. Neste trabalho, utiliza-se uma comunicação híbrida entre processos, que é a comunicação bilateral convencional assíncrona e sem bloqueio, no caso inter-nó, combinada com a comunicação unilateral SHM, no caso intra-nó. Isso permite portar a comunicação bilateral de códigos legados tais como o do modelo BRAMS para a comunicação unilateral SHM no caso intra-nó. [Hoefler et al. 2013]

4. Análise de desempenho computacional

Os testes de desempenho foram executados em uma máquina Cray com nós computacionais contendo duas pastilhas Intel Xeon E5-2699.v4 (*Broadwell*) de 2,2 GHz com 22 cores cada (44 por nó). O ambiente de compilação escolhido foi o conjunto de compiladores PGI 17.10.0 e a biblioteca Cray-MPICH 7.7.0. O BRAMS foi executado em paralelo utilizando-se um único ou vários nós computacionais de forma a comparar o desempenho da versão original que utiliza comunicação convencional assíncrona e sem bloqueio (denotada por "IS/IR") com o desempenho da versão denominada "híbrida", que combina as comunicações inter-nó IS/IR com comunicação intra-nó SHM. Foram utilizados até 32 nós computacionais com o número máximo de *cores* por nó (44), num total de 1.408 *cores*. A Tabela 1 mostra os tempos de execução dessas duas versões em função do número total de processos (N1), do número de processos por nó (N2) e do número de nós computacionais (N3) para uma grade de 100×100 pontos com 1.440 iterações no tempo, correspondentes a 24 horas de simulação. Os tempos correspondem à média de 5 execuções para cada caso, e nessa mesma tabela, aparecem os correspondentes desvios-padrão, *speedups* e eficiências, calculados em relação à versão sequencial correspondente. Pode-se observar, que utilizando-se um único nó, o desempenho da versão SHM degrada-se mais que aquele da versão original com o aumento do número de processos. De maneira geral, o desempenho paralelo da versão híbrida MPI foi pior do que a versão MPI original, mas podem-se observar casos, como os 64-32-2, 128-32-4 e 256-32-8, em que a versão híbrida foi mais rápida, sugerindo que essa comparação depende da granularidade do paralelismo.

5. Conclusões

O objetivo deste trabalho foi avaliar o desempenho da comunicação unilateral MPI SHM, implementada na versão híbrida MPI do código, para processos locais a um mesmo nó de memória compartilhada na execução paralela do modelo BRAMS. Aparentemente, em Fortran, a criação de uma janela de memória compartilhada comum aos processos locais gera uma contenção no acesso à memória que penaliza o desempenho paralelo

Tabela 1. Tempos de execução (s), desvios-padrão, *speedups* e eficiências das versões MPI original e híbrida (com SHM) executadas com N1 processos, utilizando N2 processos por nó e N3 nós computacionais (para cada caso, o menor tempo aparece em negrito).

[N1-N2-N3]	MPI IS/IR + SHM				MPI IS/IR			
	Tempo (s)	Desv.	*Speed Up*	Efic.	Tempo (s)	Desv.	*Speed Up*	Efic.
2-2-1	462,25	0,47	1,94	0,97	**459,46**	0,98	1,95	0,98
4-4-1	**223,89**	0,63	4,01	1,00	231,94	1,91	3,87	0,97
8-8-1	115,26	0,29	7,79	0,97	**112,11**	0,04	8,01	1,00
16-16-1	67,17	0,20	13,37	0,84	**67,13**	0,20	13,38	0,84
32-32-1	43,39	0,13	20,70	0,65	**40,27**	0,15	22,27	0,70
44-44-1	38,14	0,63	23,54	0,54	**33,15**	0,10	27,05	0,61
64-32-2	**31,61**	0,10	28,41	0,44	34,86	0,37	25,76	0,40
128-32-4	**46,93**	0,80	19,13	0,15	48,87	0,71	18,38	0,14
256-32-8	**16,41**	3,53	54,71	0,21	17,18	0,29	52,27	0,20
352-44-8	25,81	0,79	34,79	0,10	**16,62**	1,95	54,03	0,15
512-32-16	19,07	0,11	47,09	0,09	**15,36**	0,27	58,62	0,11
704-44-16	26,56	1,63	33,81	0,05	**16,64**	0,65	53,97	0,08
1024-32-32	23,17	3,76	38,77	0,04	**15,60**	1,25	57,57	0,06
1408-44-32	24,04	0,91	37,35	0,03	**17,23**	0,98	52,12	0,04

mais do que as inúmeras trocas de mensagens na comunicação MPI bilateral convencional assíncrona e sem bloqueio. Deve-se levar em conta a especificidade do compilador, da máquina e de sua arquitetura de memória e processadores, além do sistema operacional Linux, que possivelmente favorecem a comunicação convencional. Adicione-se também a possível imaturidade da implementação das funções MPI SHM, ainda não suficientemente otimizadas. Entretanto, a otimização de um código legado MPI portando-se parte do código para OpenMP para execução em nós de memória compartilhada exigiria uma custosa reprogramação de forma a garantir a condição de *thread-safe*, o que não é o caso ao portar-se o código para MPI SHM. Esses pontos foram levantados num trabalho anterior, que também identificou a penalização devida à conversão de ponteiros C para Fortran, requerida pelo MPI SHM [Souza et al. 2017].

Referências

(2015). Message Passing Interface Standard. Version 3.1.

Freitas, S. et al. (2016). The brazilian developments on the regional atmospheric modeling system (brams 5.2): An integrated environmental model tuned for tropical areas. *Geosci. Model Dev. Discuss., doi*, 10.

Gropp, W. (2016). MPI + MPI: Using MPI-3 Shared Memory as a Multicore Programming System.

Hoefler, T., Dinan, J., Buntinas, D., Balaji, P., Barrett, B., Brightwell, R., Gropp, W., Kale, V., and Thakur, R. (2013). Mpi + mpi: a new hybrid approach to parallel programming with mpi plus shared memory. *Computing*, 95(12):1121–1136.

Souza, C. R., Stephany, S., and Panetta, J. (2017). Análise do desempenho de comunicação usando a funcionalidade de memória compartilhada do MPI 3.0. *Anais do XX Encontro Nacional de Modelagem Computacional - ENMC*. http://nbcgib.uesc.br/enmc2017.