Utilização Eficiente do Modelo WRF de Previsão do Tempo em um Ambiente de Cluster Multi-core

Luiz C. Pinto e Mário A. R. Dantas

Laboratório de Pesquisa em Sistemas Distribuídos (LaPeSD)
Departamento de Informática e Estatística (INE)
Universidade Federal de Santa Catarina (UFSC)
Campus Trindade, 88040-900, Florianópolis, SC, Brasil
{luigi, mario}@inf.ufsc.br

Abstract. The solution to a grand challenge problem, such as modelling numerical weather prediction, requires high computational power provided by high performance configurations. Unconventional computer architectures have been increasingly adopted, e. g. multi-clusters, on account of physical limitation for higher processor clocks. Moreover, cluster configurations with multi-core processors introduce a diverse scenario for communicating parallel processes. On this matter, the article intends to augment discussion as well as to point possible gains in eficiency and performance by suiting MPI communication subsystem to particularities of multi-core clusters. In conclusion, empirical results collected with WRF execution (Weather Research and Forecasting Model) revealed decrease of near 20% on execution time, confirming the importance of this adjustment and the relevance of this work.

Keywords: distributed and parallel systems, high performance computing, cluster computing, multi-core processors

Palavras-chave: sistemas paralelos distribuídos, computação de alto desempenho, clusters ou agregados de computadores, processadores multi-core

1 Introdução

Configurações de alto desempenho tornaram-se imprescindíveis como ferramentas de auxílio para a resolução de problemas conhecidos como grandes desafios, principalmente nas áreas científica e de engenharia [1], como é o caso da previsão do tempo por meio de modelos numéricos.

Há cerca de quinze anos, eram utilizadas quase que exclusivamente máquinas massivamente paralelas (*massively parallel machines* ou MPP), soluções proprietárias e de alto custo financeiro, para suprir a demanda por alto desempenho. No entanto, com o acesso facilitado a um crescente poder de processamento em computadores de menor porte, a agregação destes computadores mostrou-se como uma alternativa viável às MPP's, tanto do ponto de vista financeiro como da capacidade computacional.

Pouco mais de uma década após seu surgimento, os agregados de computadores (ou *clusters*) tornaram-se muito populares na comunidade acerca da computação de alto desempenho (HPC) pois podem atingir configurações massivamente paralelas de forma distribuída. Hoje em dia, os clusters representam a maior fatia das soluções adotadas. Na lista do TOP500 [2] publicada em novembro de 2008, 410 dos 500 supercomputadores são classificados como clusters, ou seja, uma fatia de 82%.

Apesar da consolidação dos ambientes de cluster como solução para prover alto desempenho, a escolha dos computadores que o compõe está submetida à variabilidade do mercado, ou melhor, à variabilidade das configurações de componentes disponíveis no mercado. Estão acessíveis como *commodity*, por exemplo, taxas de transferência da ordem de megabytes por segundo com redes de interconexão Gigabit Ethernet, surgindo como uma alternativa de baixo custo quando se pensa na construção de um cluster. Além disso, o mercado de computadores recentemente sofreu uma mudança significativa com o lançamento dos processadores multi-core, que oferecem suporte nativo a processamento paralelo.

Com efeito, a inserção desses processadores no mercado de *commodities* tornou-os atraentes também para projetos de clusters de alto desempenho, sendo que sua utilização nesses ambientes já é fato. Por outro lado, surge um cenário distinto no que diz respeito à comunicação entre os processos da aplicação paralela, assunto que ainda carece de aprofundamento.

O presente trabalho pretende ampliar a discussão sobre esse novo cenário, apontando melhores práticas para a utilização eficiente de clusters equipados com processadores multi-core. Nesse sentido, foi realizado um estudo de caso com a aplicação WRF [3], um modelo numérico de previsão do tempo, visando a otimização do desempenho destes sistemas paralelos distribuídos que, em última instância, traduz-se em uma redução do tempo de execução da aplicação em foco [4].

Este artigo segue com trabalhos correlatos na Seção 2. Na Seção 3, será apresentado o modelo WRF em mais detalhes. Já na Seção 4, os resultados dos experimentos são apresentados, seguidos pela Seção 5 com as conclusões e indicações de trabalhos futuros.

2 Trabalhos Correlatos

Os trabalhos relacionados com este trabalho de pesquisa abrangem dois aspectos: a caracterização e avaliação de desempenho da aplicação em estudo e também o impacto da tecnologia multi-core no desempenho de clusters.

Em trabalhos como [5-9] são descritos e avaliados aspectos relativos ao funcionamento e desempenho do modelo WRF, bem como o caracterizam em função de diversas métricas, inclusive quanto à comunicação entre os processos da aplicação em ambientes de cluster.

Outros trabalhos levam em consideração aspectos relativos à tecnologia multi-core e seu impacto no desempenho de clusters em suas análises. Em [10] e [11], o foco concentra-se na comunicação intra-nó (processos residentes no mesmo computador) baseados em MPI, indicando a importância da otimização desse tipo de comunicação e inclusive apresentando soluções. O trabalho em [12] também analisa a comunicação

intra-nó e apresenta os ganhos obtidos com a alocação estática de processos vizinhos em núcleos de processamento de uma mesma máquina. Porém, estes dois trabalhos utilizam apenas benchmarks específicos de rede para a coleta dos resultados, e não aplicações completas. Já o trabalho em [13], por sua vez, apresenta resultados satisfatórios e possibilidades de ganho em desempenho com a utilização processadores multi-core em clusters, inclusive com a avaliação do modelo WRF como aplicação completa.

3 Modelo WRF

A aplicação em foco é o WRF (*Weather Research and Forecasting Model*), um modelo numérico para previsão meteorológica do tempo em mesoescala que vem se tornando cada vez mais importante entre a comunidade desta área de atuação, tanto em ambientes operacionais como em ambientes científicos de pesquisa atmosférica. Seu desenvolvimento é um esforço conjunto de um consórcio de importantes agências governamentais, em sua maioria estadunidenses, e também envolve diversos cientistas da comunidade científica mundial, o que dinamiza e intensifica as atividades em prol de uma aplicação que ofereça os últimos avanços em pesquisas da área.

O modelo WRF é uma aplicação *grand challenge*, cuja demanda é pela redução do seu tempo de execução para que, por exemplo, seja possível aumentar a resolução aplicada ao conjunto de dados ou mesmo a região escolhida para a simulação atmosférica. Deve ficar esclarecido que o presente estudo leva em consideração tão somente o modelo WRF, em sua versão mais recente (versão 3), não contabilizando as operações de pré ou pós-processamento inclusas no projeto WRF. Sendo assim, reduzir o tempo de execução do modelo numérico torna-se ainda mais importante quando se tem em mente que existem outras etapas no processo, a fim de efetivamente auxiliar os meteorologistas na previsão do tempo.

O conjunto de dados de entrada do modelo WRF é uma matriz tridimensional que representa a atmosfera de uma determinada região, desde metros até milhares de quilômetros, com diversas informações como, por exemplo, a topografia da região em foco e dados de observatórios para alimentar a simulação com uma condição inicial.

A Figura 1 apresenta a extensão da área territorial utilizada nos experimentos. A região abrange completamente os estados de Santa Catarina, Rio Grande do Sul e Paraná, e o Uruguai; e em parte, os estados de São Paulo, Rio de Janeiro, Minas Gerais e Mato Grosso do Sul, e ainda, parte do Paraguai e da Argentina. Em todos os experimentos, o domínio utilizado é de 100 por 126, com uma resolução de 15 km, o que representa uma área territorial de 12.600 quilômetros quadrados. Na vertical, o domínio é de 37, totalizando 466.200 elementos na matriz.

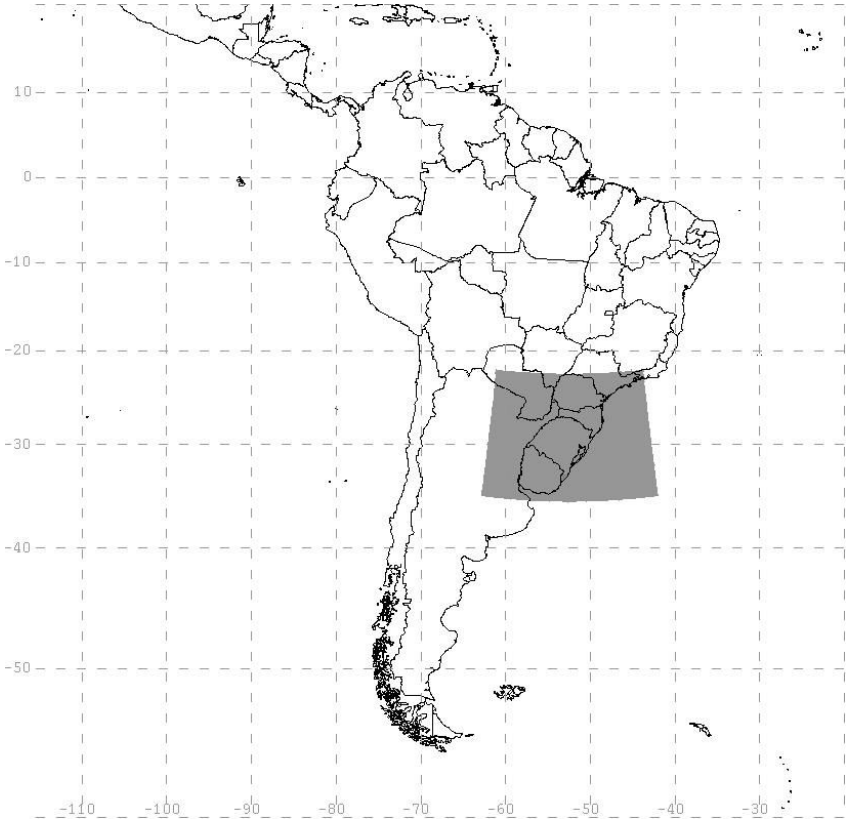

Fig. 1. Extensão territorial utilizada nos experimentos.

Na execução paralela do modelo WRF, cada processo recebe uma sub-matriz do conjunto de dados de entrada, de tamanho aproximadamente igual, que diminui com o aumento do número de processos. A principal atividade que demanda comunicação entre os processos é a redistribuição dos dados laterais, nos quatro limites lógicos de cada sub-matriz, ocorrendo a cada iteração com mensagens entre 10 e 100 kilobytes.

Cada iteração avança o tempo de simulação em 75 segundos, totalizando 3456 iterações na previsão para 72 horas. Além disso, a cada 12 iterações (ou 15 minutos de simulação) ocorre uma iteração de radiação física, que se soma ao tempo de processamento da iteração ordinária, e a cada 145 (ou 3 horas de simulação) ocorrem picos por causa da geração do arquivo de saída.

Essas e outras configurações seguem as especificações adotadas pelos especialistas em meteorologia da EPAGRI S.A. (Empresa de Pesquisa Agropecuária e Extensão Rural de Santa Catarina). Esta é uma empresa pública, responsável pela previsão meteorológica do tempo no estado de Santa Catarina.

4 Experimentos

A Figura 2 apresenta a configuração do ambiente em nível de componentes.

Ambiente	INFO / SISTEMA
# Nós	6
# Cores por nó	8
Interconexão	Gigabit Ethernet
	3Com Switch 3812
MTU	1500
Modelo do processador	64-bit AMD Opteron 2350
# Cores por processador	4
Veloc. do processador	2 Ghz
Tecnologia Manuf.	65nm
Cache L1 (I/D)	64KB/64KB
Cache L2	512KB
Cache L3	2MB
DRAM	8GB
Velocidade DRAM	1000Mhz

Fig. 2. Configuração do ambiente computacional.

Já a Figura 3 apresenta um esquema ilustrativo do ambiente em avaliação.

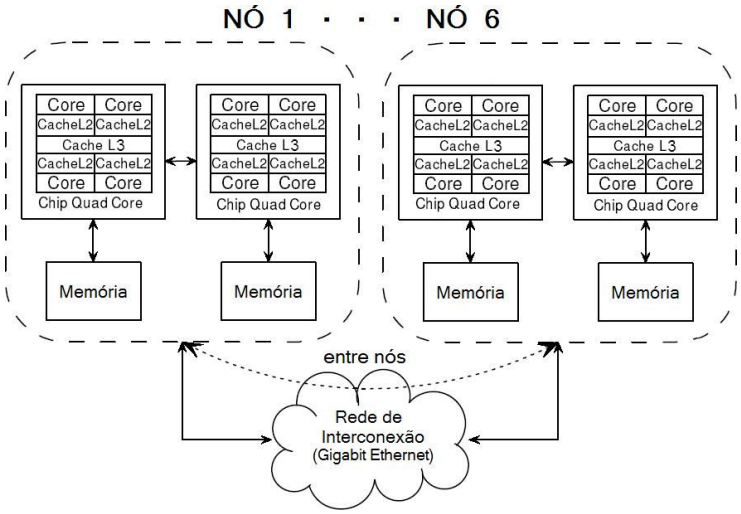

Fig. 3. Esquema ilustrativo do ambiente em avaliação.

Todos os computadores são equipados com processadores AMD Opteron [14] e rodam Debian Linux kernel 2.6.18-6-amd64. O ambiente está isolado de ruídos externos e dedicado aos experimentos, não operando quaisquer outros serviços, exceto a configuração mínima necessária à execução dos experimentos com a

biblioteca MPICH2 (versão 1.0.6p1) [15, 16]. Além disso, tomou-se o cuidado de desativar a utilização da memória virtual (*swap*).

Primeiramente, foi executado o benckmark de rede *b_eff* [17], em versão disponível como parte do HPC Challenge Benchmark (HPCC) [18], adaptada para coletar dados de mensagens entre 2 bytes e 4 megabytes. Isto foi feito para capturar características específicas da comunicação entre processos de interesse primário [19], como a latência e taxa de transferência.

Para tanto, foram coletados dados da comunicação de 2 processos em um mesmo nó e em nós distintos, 8 processos em um mesmo nó, e todos os 48 núcleos de processamento disponíveis no ambiente se comunicando. A comunicação foi mediada pelo subsistema CH3:SOCK do MPICH2 (doravante denominado MPICH-SOCK), que utiliza soquetes para a comunicação entre processos.

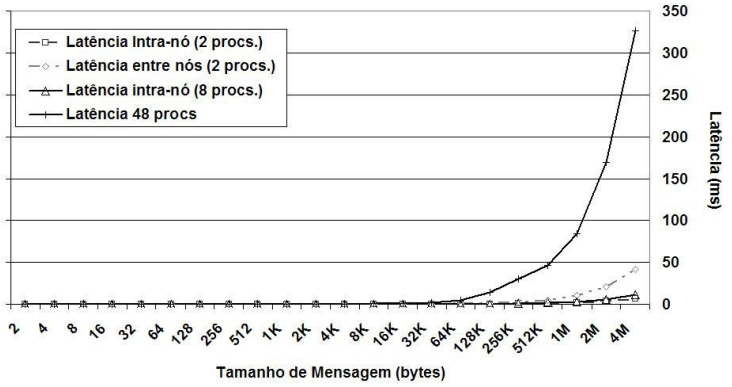

Fig. 4. Latência coletada com o b_eff.

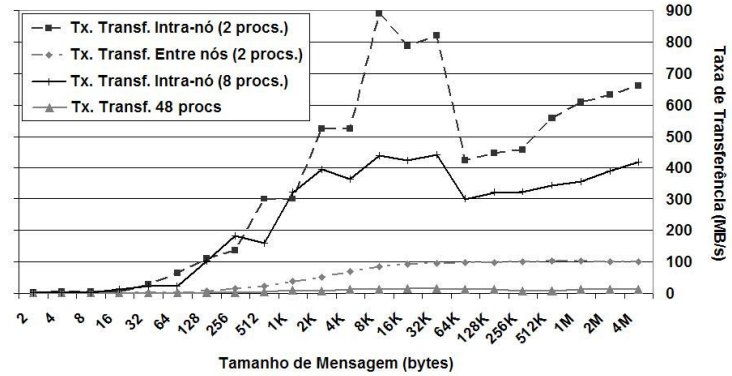

Fig. 5. Taxa de transferência coletada com o b_eff.

Em resumo, as Figuras 4 e 5 mostram duas características importantes. Quando apenas 2 processos se comunicam, o desempenho como um todo da comunicação

intra-nó é notadamente superior do que entre diferentes nós, embora essa diferença em termos de latência e taxa de transferência não seja tão discrepante como poderia se esperar. Além disso, os resultados da comunicação com mais processos mostram que o desempenho de 48 processos, ou seja, um por núcleo de processamento em todos os nós, é aproximadamente 6 vezes menor (descrescimento linear) do que o desempenho da comunicação de todos núcleos de processamento em um único nó, embora neste último caso os processos se comuniquem internamente, não sendo necessário, portanto, o acesso à rede de interconexão Gigabit Ethernet.

Feita esta caracterização, vamos aos resultados dos experimentos com o modelo de previsão do tempo. As simulações apresentadas a seguir rodam o modelo WRF 3, compilado com gfortran 4.1.2 e suporte a MPI.

Todos os experimentos são realizados com base na mesma configuração do modelo WRF (relativa aos componentes de física) e mesmo conjunto de dados de entrada (do dia 29 de maio de 2008) utilizados no ambiente de produção da empresa na previsão meteorológica, conforme apresentados na seção anterior.

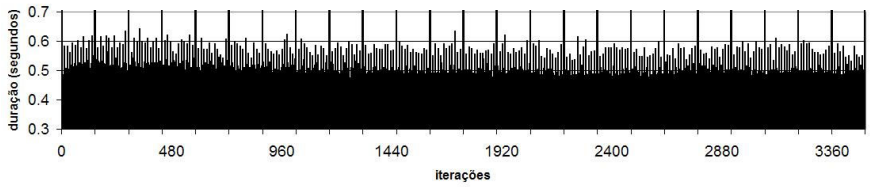

Fig. 6. Duração de cada iteração com MPICH-SOCK, totalizando 34min16seg.

A Figura 6 apresenta a duração de cada iteração ao longo de toda a execução do modelo WRF com o MPICH-SOCK. Os 48 processadores disponíveis no ambiente estão alocados a processos da aplicação. Os picos que extrapolam o eixo Y do gráfico referem-se ao agrupamento dos dados resultantes para a geração do arquivo de saída do modelo que duram cerca de 8 segundos neste caso. Porém, como ilustra a Figura 7, os processadores estão subutilizados nesta configuração com MPICH-SOCK. Durante a maior parte da execução do modelo, percebe-se que os núcleos de processamento não utilizam nem 50% de sua capacidade de processamento.

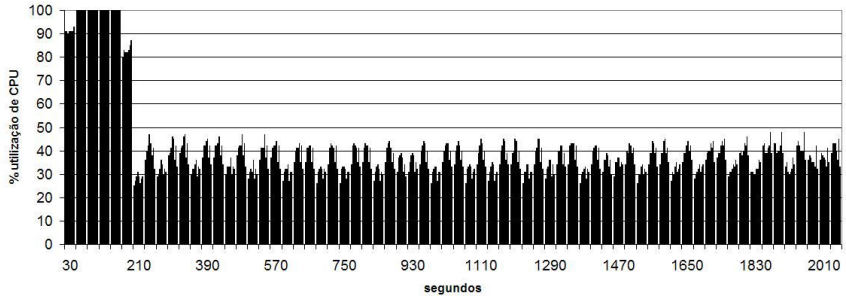

Fig. 7. Utilização dos oito núcleos de um mesmo nó com MPICH-SOCK.

Por não haver distinção entre processos localizados no mesmo nó ou em nós distintos por parte do MPICH-SOCK, toda operação com MPI é tratada como uma

operação de comunicação comum, como se estivessem em nós diferentes. Sendo assim, é gasto tempo com a criação de soquetes para a interação entre os processos localizados no mesmo nó, desnecessariamente, além de envolver o sistema operacional neste processo. O resultado é a subutilização dos núcleos de processamento na execução da aplicação.

Como solução para este problema, optou-se por um subsistema de comunicação do MPICH2 que faz essa distinção, usando memória compartilhada entre processos localizados no mesmo nó, e comunicação por soquete entre nós, como é o caso do subsistema CH3:SSM (doravante chamado MPICH-SSM).

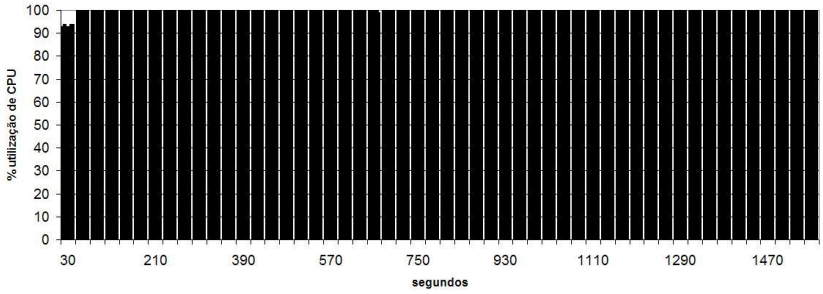

Fig. 8. Utilização dos oito núcleos de um mesmo nó com MPICH-SSM.

A Figura 8 apresenta a utilização dos núcleos de processamento de um dos 6 nós do ambiente durante a execução do modelo WRF alocando todos os processadores disponíveis. Percebe-se que o MPICH-SSM solucionou satisfatoriamente o problema da subutilização dos processadores, já que os dados mostram todos núcleos rodando a 100% de sua capacidade durante praticamente toda a execução do modelo.

Além disso, a Figura 9 ilustra o tempo de execução de cada iteração, sendo que o tempo total baixou de aproximadamente 34 minutos para cerca de 28 minutos, uma diferença em torno de 18% em comparação à execução com o MPICH-SOCK. Vale ressaltar que os picos que extrapolam o eixo Y do gráfico, relativos à geração do arquivo de saída do modelo, duram cerca de 6 segundos com o MPICH-SSM.

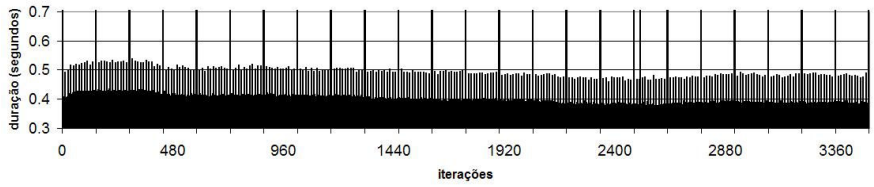

Fig. 9. Duração de cada iteração com MPICH-SSM, totalizando 27min59seg.

Com base nos resultados empíricos, a melhor opção para extrair maior desempenho e eficiência deste ambiente de cluster com processadores de múltiplos núcleos é o MPICH-SSM. Desta forma, resulta em um menor tempo de execução do modelo WRF na simulação para 3 dias ou 72 horas, que é a janela de previsão meteorológica utilizada pela empresa em suas operações de produção.

5 Conclusões e Trabalhos Futuros

Com base nos resultados coletados neste estudo empírico, apresentados em resumo na Figura 10, o subsistema MPICH-SSM possibilitou uma redução de mais de 18% no tempo de execução do modelo WRF para a simulação de 3 dias ou 72 horas.

Previsão 72 horas	MPICH-SOCK	MPICH-SSM
Tempo Execução (48 procs.)	34min16seg	27min59seg
Redução de tempo	0%	18,34%
Speedup Relativo	1	1,22

Fig. 10. Resultados da simulação para 3 dias antes e depois do processo de otimização.

Portanto, os resultados empíricos indicam a adoção de um subsistema híbrido de comunicação entre processos (memória compartilhada internamente ao nó e soquetes entre nós) como a melhor opção para extrair maior desempenho e eficiência desta configuração, utilizada nas operações de produção da empresa responsável pela previsão meteorológica no Estado de Santa Caratarina.

Enfim, este trabalho apresentou um estudo quantitativo resultante de uma aproximação com um ambiente de produção em que fica ressaltada a importância da adequação do subsistema de comunicação MPI à especificidade da configuração a fim de maximizar sua eficiência. Além disso, foi possível discutir aspectos envolvidos no processo de maximização do desempenho destes sistemas paralelos distribuídos, bem como investigar os fatores redutores do desempenho e o impacto da presença de processadores com múltiplos núcleos em ambientes de cluster.

Como trabalhos futuros, indica-se a extensão destes experimentos a ambientes de grande escala, a fim de confrontar características relativas à escalabilidade, e também a sistemas equipados com um número crescente de núcleos de processamento concorrentes em um mesmo processador a fim de quantificar o overhead resultante dessa abordagem.

Agradecimentos. Esta pesquisa foi desenvolvida com a colaboração da CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pela bolsa de estudo e da EPAGRI S.A. (Empresa de Pesquisa Agropecuária e Extensão Rural de Santa Catarina), que gentilmente cedeu seu ambiente para a execução dos experimentos.

Referências Bibliográficas

1. Kumar, V., Grama A., Gupta A., Karypis G.: Introduction to Parallel Computing. The Benjamin/Cummings Publishing Company Inc. (1994)
2. TOP500 Project, http://www.top500.org
3. Michalakes J., Dudhia J., Gill D., Klemp J., Skamarock W.: Design of a Next-generation Regional Weather Research and Forecast Model. In: Towards Teracomputing (Proceedings of the Eighth ECMWF Workshop on the Use of Parallel Processors in Meteorology), pp. 117--124. World Scientific. (1999)
4. Jordan H., Alaghband G.: Fundamentals of Parallel Processing. Prentice Hall. (2003)

5. Michalakes J., Dudhia J., Gill D., Henderson T., Klemp J., Skamarock W., Wang W.: The Weather Research and Forecast Model: Software Architecture and Performance. In: ECMWF Workshop on the Use of High Performance Computing in Meteorology, pp. 156--168. (2004)
6. Zamani R., Afsahi A.: Communication Characteristics of Message-Passing Scientific and Engineering Applications. In: International Conference on Parallel and Distributed Computing and Systems (PDCS), pp. 644--649. (2005)
7. Kerbyson D., Barker K., Davis K.: Analysis of the Weather Research and Forcasting (WRF) model on large-scale systems. In: Parallel Computing: Architectures, Algorithms and Applications (Parco). (2007)
8. Amstrong B., Bae H., Eigenmann R., Saied F., Sayeed M., Zheng Y.: HPC Benchmarking and Performance Evaluation With Realistic Applications. In: SPEC Benchmark Workshop. (2006)
9. Skamarock W., Klemp J., Dudhia J., Gill D., Barker D., Duda M., Huang X., Wang W., Powers J.: A Description of the Advanced Research WRF Version 3. Technical report, National Center for Atmospheric Research. (2008)
10. Chai L., Hartono A., Panda D.: Designing High Performance and Scalable MPI Intra-node Communication Support for Clusters. In: IEEE International Conference on Cluster Computing. IEEE Computer Society. (2006)
11. Chai L., Gao Q., Panda D.: Understanding the Impact of Multi-Core Architecture in Cluster Computing: A Case Study with Intel Dual-Core System. In: IEEE International Symposium on Cluster Computing and the Grid, pp. 471--478. IEEE Computer Society. (2007)
12. Pourreza H., Graham P.: On the Programming Impact of Multi-core, Multi-Processor Nodes in MPI Clusters. In: High Performance Computing Systems and Applications. (2007)
13. Pinto L., Tomazella L., Dantas M.: Uma Abordagem para Composição de Clusters Eficientes na Execução do Modelo Numérico WRF de Previsão do Tempo. In: Workshop em Sistemas Computacionais de Alto Desempenho – WSCAD-SSC. (2008)
14. AMD: AMD Opteron™ processor product data sheet. Technical report, 23932. (2007)
15. Message Passing Interface Forum: MPI-2: Extensions to the Message-Passing Interface. Technical report. (2003)
16. Gropp W., Lusk E., Thakur R.: Using MPI-2: Advanced Features of the Message-Passing Interface. MIT Press. (1999)
17. Rabenseifner R., Koniges A.: The Parallel Communication and I/O Bandwidth Benchmarks: b_eff and b_eff_io. In: Cray User Group Conference, CUG Summit. (2001)
18. Luszczek P., Bailey D., Dongarra J., Kepner J., Lucas R., Rabenseifner R., D. Takahashi D.: The HPC Challenge (HPCC) Benchmark Suite. In: IEEE SC06 Conference Tutorial. (2006)
19. Coulouris G., Dollimore J., Kindberg T.: Distributed systems: Concepts and Design. Addison Wesley. (2005)