Estudo Quantitativo do Modelo WRF de Previsão do Tempo em um Ambiente de Cluster Multi-core

Luiz C. Pinto, M. A. R. Dantas
Laboratório de Pesquisa em Sistemas Distribuídos (LaPeSD)
Departamento de Informática e Estatística (INE)
Universidade Federal de Santa Catarina (UFSC) - Florianópolis, SC, Brasil
{luigi, mario}@inf.ufsc.br

Resumo

A computação científica demanda poder computacional de alto desempenho principalmente para resolver em tempo hábil problemas conhecidos como "grandes desafios". Devido à limitação de clock, cada vez mais arquiteturas não-convencionais construídas com processadores conhecidos como commodity estão sendo utilizadas nesta tarefa, como por exemplo os ambientes multi-cluster. Atualmente, a inserção de processadores com múltiplos núcleos nas configurações de cluster cria um cenário diferenciado no que diz respeito à comunicação entre processos paralelos nestes ambientes. Nesse contexto, o presente artigo vem para ampliar a discussão e apontar possibilidades de ganho em desempenho e eficiência. Os resultados empíricos obtidos com a execução do modelo numérico de previsão do tempo WRF (Weather Research and Forecasting Model) revelaram speedup de 1.39 resultante da adequação do subsistema de comunicação entre processos às especificidades da aplicação e do cluster multi-core em foco, o que reforça a importância da análise e a pertinência deste trabalho.

1. Introdução

Configurações de alto desempenho tornaram-se imprescindíveis como ferramentas de auxílio para a resolução de problemas fundamentais principalmente das áreas científica e de engenharia, como é o caso da previsão do tempo por meio de modelos numéricos. Estes problemas são conhecidos como grandes desafios e sua solução geralmente tem enorme impacto econômico ou científico [11].

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 [19] publicada em junho de 2009, 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 de seus componentes está submetida à oferta do mercado, ou seja, à variabilidade das configurações de componentes disponíveis no mercado. Atualmente, 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. A inserção desses processadores no mercado de commodities tornou-os atraentes também para projetos de sistemas de alto desempenho.

A produção de processadores seqüenciais foi desestimulada por limitações físicas e pelo alto consumo de energia, podendo se tornar diminuta dentro de poucos anos, à medida que a tecnologia de processadores multi-core seja viabilizada para processadores many-core, com até dezenas de núcleos de processamento em um mesmo processador. Um número crescente de núcleos ou cores em um mesmo processador poderá sobrecarregar o subsis-

tema de comunicação interno a cada computador agregado. Sendo assim, a importância de sistemas de comunicação como as arquiteturas network-on-chip aumenta [8], não para a interconexão entre computadores, e sim para a interconexão dos próprios núcleos de processamento dentro do mesmo processador e internamente ao computador.

O fato é que a presença de processadores multi-core altera o comportamento do desempenho de aplicações executadas em agregados de computadores, criando um cenário diferenciado no que diz respeito à comunicação entre processos paralelos nestes ambientes. A motivação deste trabalho surgiu com os olhos voltados a esse novo contexto e evoluiu no sentido de melhor implementar e utilizar agregados de computadores multi-core para otimizar o desempenho e a eficiência na execução de aplicações paralelas científicas. Ademais, havia o interesse de aproximar-se da realidade da computação de alto desempenho em ambientes de produção.

Enfim, 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 [14], 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 [9].

Este artigo segue com os trabalhos correlatos na Seção 2. Na Seção 3, será apresentado o modelo WRF em mais detalhes. Na Seção 4, os resultados dos experimentos são apresentados e, na Seção 5, seguem 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.

Quanto às características de funcionamento e desempenho do modelo WRF, foram encontradas descrições e experimentos nos trabalhos [10, 1, 3, 13, 15, 18, 21] ajudando na caracterização apresentada na Seção 3 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 [6], o foco dá-se sobre a comunicação entre processos intra-nó (residentes no mesmo computador) baseados em MPI e apresentam uma implementação específica para este tipo de comunicação, tendo como objetivo melhor desempenho e escalabilidade. Já em [16], também é analisada a comunicação intra-nó e apresentam os ganhos obtidos com a alocação estática de processos vizinhos em núcleos de processamento de uma mesma máquina. Além disso, estes dois trabalhos utilizam apenas benchmarks específicos de rede para a coleta dos resultados.

Além de analisar o comportamento de processadores multi-core e seu impacto no desempenho, o trabalho [2] caracteriza diversas aplicações científicas, apresentando uma metodologia para validação dos resultados semelhante ao presente trabalho com relação à utilização de diferentes categorias de workload de origem científica, desde micro-benchmarks até aplicações completas.

Já o trabalho [5] apresenta modelos de avaliação de desempenho específicos para ambientes de cluster com processadores multi-core da fabricante Intel, que diferem no nível arquitetural dos processadores multi-core AMD utilizados neste trabalho. Ademais, assemelha-se no que tange às motivações e, em parte, à metodologia.

3. Modelo WRF de Previsão do Tempo

A aplicação científica em foco é o WRF (Weather Research and Forecasting Model), um modelo numérico de simulação 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, mas também envolve a 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 "grande desafio", 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 este estudo leva em consideração tão somente o modelo WRF, 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 pois existem outras etapas antes que seja de fato possível 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. Enfim, o modelo deve ser alimentado com uma condicão inicial que representa a atmosfera de uma determinada região, desde

metros até milhares de quilômetros. 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.

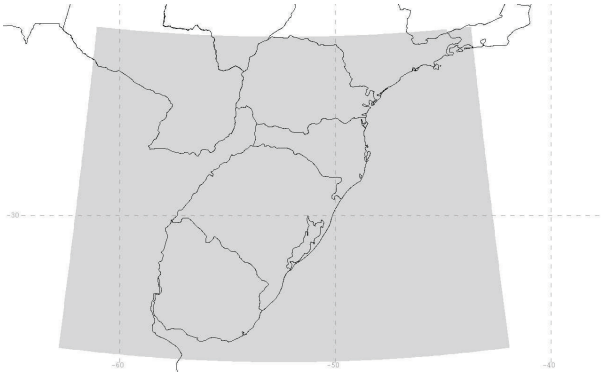

Figura 1. Região utilizada nos experimentos

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 é a redistribuição dos dados laterais 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.

4. Experimentos

Todos os experimentos estão baseados na mesma configuração do modelo WRF (especificações definidas por especialistas em meteorologia da Empresa de Pesquisa Agropecuária e Extensão Rural de Santa Catarina - EPAGRI S.A.) e conjunto de dados de entrada (do dia 29 de maio de 2008) utilizados no ambiente de produção da empresa na previsão meteorológica.

O objetivo é otimizar o desempenho do modelo WRF no ambiente de cluster recém adquirido pela empresa, antes de colocá-lo em produção com a última versão disponível do WRF. Foi realizada uma pesquisa mais aprofundada sobre a própria aplicação WRF, de relevante importância para a comunidade científica, e também uma análise mais detalhada sobre as melhores práticas e os problemas encontrados durante o processo de otimização. Nesse sentido, foi-se em busca de alternativas viáveis e eficazes a fim de tirar a máxima eficiência deste ambiente de cluster, reduzindo o tempo de execução do modelo WRF em comparação à configuração original, sem nenhuma especialização para a aplicação em questão.

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

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

Figura 2. Descrição do cluster

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

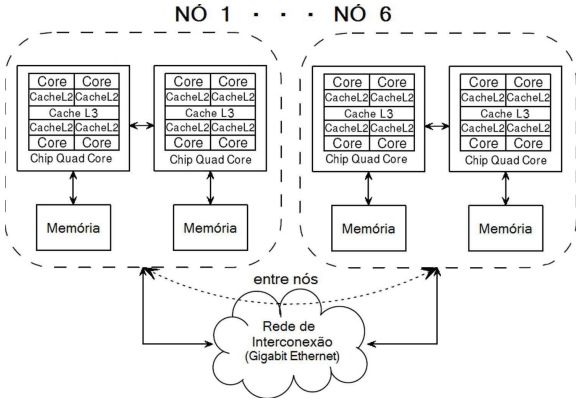

Figura 3. Esquema ilustrativo do ambiente

Todos computadores 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. Além disso, a memória virtual (swap) foi desativada.

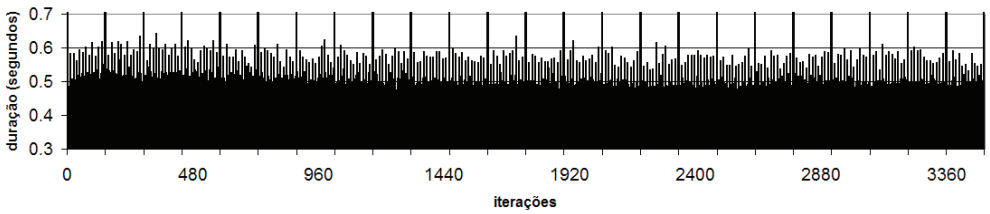

Figura 4. Duração de cada iteração com SOCK, totalizando 34min16seg

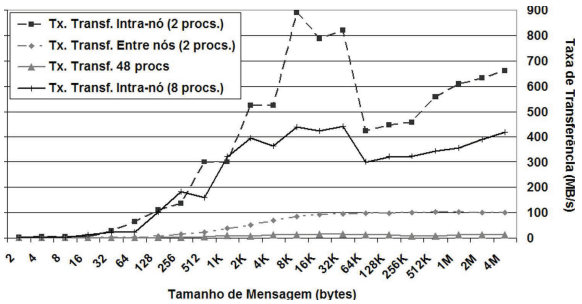

Figura 5. Latência coletada com o b_eff

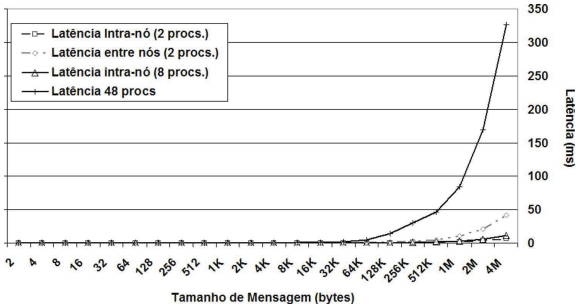

Figura 6. Taxa de transf. coletada com o b_eff

Primeiramente, foi executado o benckmark de rede b_eff [17], em versão disponível como parte do HPC Challenge Benchmark (HPCC) [12], 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, como a latência e taxa de transferência [7]. 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 SOCK), que utiliza soquetes para a comunicação entre processos.

Em resumo, as Figuras 5 e 6 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 a aplicação científica de previsão do tempo. As simulações rodam o modelo WRF 3, compilado com gfortran 4.1.2 e suporte a MPI, para previsão meteorológica de 3 dias ou 72 horas.

A Figura 4 apresenta a duração de cada iteração ao longo de toda a execução do modelo WRF com o SOCK, totalizando 34 minutos e 16 segundos de simulação. 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 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.

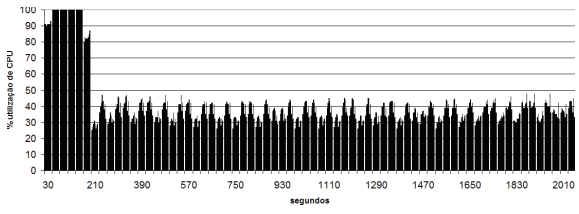

Figura 7. Utilização dos núcleos com SOCK

Por não haver distinção da localização dos processos pelo subsistema SOCK, toda operação com MPI é tratada como uma operação de comunicação por soquetes, como se estivessem em nós diferentes. Sendo assim, é gasto tempo com cópias desnecessárias da mensagem já que a interação ocorre entre processos localizados no mesmo nó, com acesso ao mesmo espaço de memória. O resultado é a subutilização dos núcleos de processamento.

Conforme [20], o desempenho de aplicações executadas em ambientes de cluster depende principalmente da escolha do modelo de programação paralela, das características da própria aplicação quanto às necessidades de computação e comunicação, e do desempenho do subsistema de comunicação. Embora as duas últimas características sejam praticamente inquestionáveis, a escolha pelo modelo de passagem de mensagem ou por um modelo híbrido para prover melhor desempenho em agregados de computadores multiprocessados ainda está em debate.

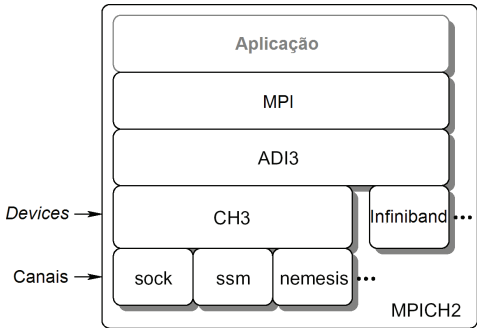

Figura 8. Esquema ilustrativo: MPICH2 [4]

MPI é uma interface padrão para prover primitivas de comunicação e sincronização necessárias à execução paralela e distribuída da aplicação. Porém, sua efetivação depende de uma biblioteca como o MPICH2. A biblioteca MPICH2 implementa a comunicação efetiva com base em camadas, como ilustra a Figura 8.

Como solução para o problema da subutilização dos processadores, optou-se por um canal de comunicação do MPICH2 que faz essa distinção, usando memória compartilhada entre processos localizados no mesmo nó, e comunicação por soquete entre nós, como o canal SSM. O tempo de execução baixou de, aproximadamente, 34 minutos para cerca de 28 minutos, uma diferença em torno de 18% em comparação ao resultado com o canal SOCK. A Figura 9 apresenta a utilização dos núcleos de processamento de um dos 6 nós do ambiente durante a execução do modelo WRF alocando todos processadores disponíveis. Percebe-se que a adoção do subsistema SSM solucionou satisfatoriamente o problema da subutilização dos processadores, pois todos os núcleos estão rodando a 100% de sua capacidade durante praticamente toda a execução.

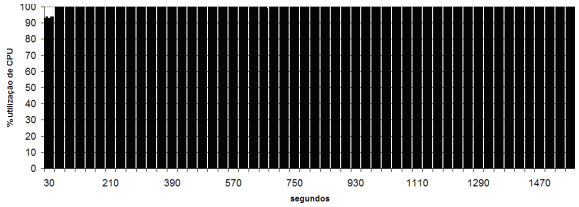

Figura 9. Utilização dos núcleos com SSM

Além do canal SSM, a Figura 8 também apresenta o canal NEMESIS, que foi desenvolvido para ser um subsistema de comunicação escalável, baseado em memória compartilhada para processos localizados em um mesmo nó do cluster e de alto desempenho. De fato, em [4], é apresentada uma avaliação de desempenho do canal NEMESIS da biblioteca MPICH2, que mostrou-se eficiente e com overhead muito pequeno, superando o desempenho do canal SSM. Em função disso, a utilização do NEMESIS surgiu como uma alternativa interessante, resultando em maior ganho de desempenho, como indica a comparação apresentada pela Figura 10.

Previsão 72 horas	**SOCK**	**SSM**	**NEMESIS**
Tempo de Execução	34min16seg	27min59seg	25min44seg
Redução de tempo	0%	18,34%	24,90%
Speedup Relativo	1	1,22	1,33

Figura 10. Comparação dos subsistemas

Portanto, os resultados empíricos indicam a adoção do canal NEMESIS, que é um subsistema híbrido de comunicação entre processos (memória compartilhada internamente ao nó e soquetes entre nós) para extrair maior desempenho e eficiência deste ambiente de cluster.

72 horas (NEMESIS)	**MTU 1500**	**MTU 9000**	**MTU 4500**
Tempo de Execução	25min44seg	25min00seg	24min36seg
Redução de tempo	0%	2,85%	4,40%
Speedup Relativo	1	1,03	1,05

Figura 11. Resultados com diversas MTU's

Além disso, o tamanho máximo das mensagens em nível de enlace, no caso a MTU (*Maximum Transmission Unit*) da rede Gigabit Ethernet, é outro aspecto que pode melhorar o desempenho do ambiente. Porém, como mostra a Figura 11, seu impacto é menor do que a escolha adequada do subsistema de comunicação. De qualquer forma, soma-se algum ganho com essa adequação.

Enfim, o objetivo em vista foi alcançado com sucesso como mostram os resultados empíricos apresentados em

OTIMIZAÇÃO	Antes	Depois
Tempo de Execução	34min16seg	24min36seg
Redução de tempo	0%	28,21%
Speedup Relativo	1	1,39

Figura 12. Resultado da otimização

resumo na Figura 12. A simulação de previsão meteorológica foi satisfatoriamente executada com uma redução significativa no seu tempo de execução. Com isso, a configuração proposta foi adotada como solução para o ambiente de produção na empresa.

5. Conclusões e Trabalhos Futuros

Este trabalho apresentou um estudo quantitativo de um ambiente de cluster equipado com processadores multi-core, resultantes da aproximação com a realidade da computação científica em ambientes de produção, cujo objetivo foi construir um sistema de alto desempenho adequado à execução de aplicações do tipo "grandes desafios", como a modelagem numérica de previsão meteorológica.

Foi realizada uma pesquisa aprofundada sobre a aplicação WRF e também uma análise detalhada sobre técnicas de otimização de clusters, buscando alternativas eficazes para reduzir o tempo de execução da aplicação. Também investigou-se o impacto da presença de processadores multi-core em ambientes de cluster, inclusive indicando boas técnicas para tirar proveito e maximizar o desempenho do ambiente de cluster. Enfim, alcançou-se o objetivo proposto com um ganho de 1.39 em speedup comparado ao desempenho do sistema original, sem adequações às especificidades da aplicação e do subsistema de comunicação do cluster.

Como trabalhos futuros, indica-se a extensão destes experimentos a ambientes de grande escala, e também a sistemas equipados com um número crescente de núcleos de processamento em um mesmo processador a fim de quantificar o overhead resultante dessa abordagem. Também indica-se a utilização de um protocolo leve de comunicação para maximização do desempenho por tratar-se de sistemas computacionais isolados de ruídos externos.

Referências

[1] A description of the advanced research wrf version 2. Technical report, January 2007.

[2] S. R. Alam, R. F. Barrett, J. A. Kuehn, P. C. Roth, and J. S. Vetter. Characterization of scientific workloads on systems with multi-core processors. *IEEE International Symposium on Workload Characterization*, pages 225–236, 2006.

[3] B. Amstrong, H. Bae, R. Eigenmann, F. Saied, M. Sayeed, and Y. Zheng. Hpc benchmarking and performance evaluation with realistic applications. *SPEC Benchmark W.*, 2006.

[4] D. Buntinas, G. Mercier, and W. Gropp. Implementation and evaluation of shared-memory communication and synchronization operations in mpich2 using nemesis communication subsystem. *Parallel Computing*, 33(9):634–644, 2007.

[5] L. Chai, Q. Gao, and D. Panda. Understanding the impact of multi-core architecture in cluster computing: A case study with intel dual-core system. *IEEE CCGRID*, 2007.

[6] L. Chai, A. Hartono, and D. Panda. Designing high performance and scalable mpi intra-node communication support for clusters. *IEEE International Conference on Cluster Computing*, pages 1–10, 2006.

[7] G. Coulouris, J. Dollimore, and T. Kindberg. *Distributed systems: Concepts and Design*. Addison Wesley, 2005.

[8] J. Flich, S. Rodrigo, J. Duato, T. Sødring, A. G. Solheim, T. Skeie, and O. Lysne. On the potential of noc virtualization for multicore chips. *International Workshop on Multi-Core Computing Systems (MuCoCoS)*, 2008.

[9] H. Jordan and G. Alaghband. *Fundamentals of Parallel Processing*. Prentice Hall, 1ª edition, 2003.

[10] D. Kerbyson, K. Barker, and K. Davis. Analysis of the wrf model on large-scale systems. *PARCO*, 2007.

[11] V. Kumar, A. Grama, A. Gupta, and G. Karypis. *Introduction to Parallel Computing*. The Benjamin/Cummings Publishing Company Inc., 1ª edition, 1994.

[12] P. Luszczek, D. Bailey, J. Dongarra, J. Kepner, R. Lucas, R. Rabenseifner, and D. Takahashi. The hpc challenge (hpcc) benchmark suite. *IEEE SC06 Conf. Tutorial*, 2006.

[13] J. Michalakes, J. Dudhia, D. Gill, T. Henderson, J. Klemp, W. Skamarock, and W. Wang. The weather research and forecast model: Software architecture and performance. *ECMWF Workshop on the Use of High Performance Computing in Meteorology*, pages 156–168, 2004.

[14] J. Michalakes, J. Dudhia, D. Gill, J. Klemp, and W. Skamarock. Design of a next-generation regional weather research and forecast model. *Towards Teracomputing, World Scientific*, pages 117–124, 1999.

[15] L. C. Pinto, L. H. B. Tomazella, and M. A. R. Dantas. Uma abordagem para composição de clusters eficientes na execução do modelo numérico wrf de previsão do tempo. *WSCAD-SSC*, 2008.

[16] H. Pourreza and P. Graham. On the programming impact of multi-core, multi-processor nodes in mpi clusters. *High Performance Computing Systems and Applications*, 2007.

[17] R. Rabenseifner and A. E. Koniges. The parallel communication and i/o bandwidth benchmarks: b_eff and b_eff_io. *Cray User Group Conference, CUG Summit*, 2001.

[18] W. Skamarock, J. Klemp, J. Dudhia, D. Gill, D. Barker, M. Duda, X. Huang, W. Wang, and J. Powers. A description of the advanced research wrf version 3. Technical report, June 2008.

[19] TOP500. Supercomputer sites. www.top500.org.

[20] R. Zamani. *Communication Characteristics of Message-Passing Applications, and Impact of RDMA on their Performance*. PhD thesis, Kingston, Ontario, Canada, 2005.

[21] R. Zamani and A. Afsahi. Communication characteristics of message-passing scientific and engineering applications. *PDCS*, pages 644–649, 2005.