Uma Abordagem para Composição de Clusters Eficientes na Execução do Modelo Numérico WRF de Previsão do Tempo

Luiz C. Pinto, Luiz H. B. Tomazella, 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, tomazella, mario}@inf.ufsc.br

Resumo

A resolução de problemas conhecidos por grand challenge, *como é o caso da previsão do tempo por meio de modelos numéricos, demandam computação de alto desempenho. Apesar da consolidação dos clusters como solução para prover alto desempenho, a escolha dos computadores que o compõe está submetida à variabilidade das configurações disponíveis no mercado. De fato, a inserção de processadores multi-core em ambientes de cluster cria um cenário distinto no que diz respeito à comunicação entre processos. Nesse contexto, propõe-se uma abordagem em que alguns núcleos de processamento não são alocados a processos da aplicação, com o intuito de construir clusters econômicos mas também eficientes, interconectados por Gigabit Ethernet em alternativa a redes de interconexão como Myrinet e Infiniband. Experimentos com o modelo numérico de previsão do tempo WRF (Weather Research and Forecasting Model) e o algoritmo de granularidade fina IS do NAS Parallel Benchmarks, revelaram redução de mais de 20% no tempo de execução. Portanto, os resultados empíricos indicam um ganho expressivo no desempenho de um mesmo cluster quando configurado segundo a abordagem proposta, provando a pertinência deste trabalho.*

1. Introdução

Computadores de alto desempenho, também conhecidos como supercomputadores, tornaram-se imprescindíveis como ferramenta de auxílio ou para a resolução de problemas conhecidos por *grand challenge*, principalmente das áreas científica e de engenharia [16], 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, em inglês), 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 (*clusters*, em inglês) tornaram-se muito populares na comunidade acerca da computação de alto desempenho (ou *high performance computing*, em inglês) pois podem atingir configurações massivamente paralelas de forma distribuída. Hoje em dia, os *clusters* representam a maior fatia das soluções adotadas. Vide, por exemplo, a lista dos 500 supercomputadores mais rápidos do mundo, conhecida por TOP500 [20], cuja atualização ocorre a cada seis meses. Em junho deste ano de 2008, dos 500 supercomputadores, 400 deles são classificados como *clusters*, ou seja, uma fatia de 80%.

Apesar da consolidação dos *clusters* 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 computadores disponíveis no mercado. De fato, o mercado de computadores recentemente sofreu uma mudança significativa com o lançamento dos processadores *multi-core*, que oferecem suporte nativo a processamento paralelo. Também como *commodity*, as taxas de transferência da ordem de megabytes por segundo proporcionadas pelas redes de interconexão Gigabit Ethernet surgem como uma alternativa de baixo custo quando se pensa em construir um *cluster*.

Portanto, o presente trabalho tem como objeto de estudo a composição eficiente de *clusters* de alto desempenho que, embora interconectados por redes Gigabit Ethernet, apresentam um desempenho diferenciado em função da disponibilidade de múltiplos núcleos de processamento em cada computador agregado. Nesse sentido, será apresentada uma proposta que visa a maximização do desempenho destes

sistemas paralelos distribuídos, traduzida em redução do tempo de execução de um modelo numérico de previsão do tempo como o WRF [22].

Como fruto do trabalho de pesquisa e experimentação desenvolvido, os resultados indicam que esta abordagem é pertinente para a composição de *clusters* de pequeno e médio porte eficientes e de menor custo financeiro, pois o tempo de execução foi reduzido em mais de 20%, surgindo como uma alternativa à utilização de redes de interconexão como Myrinet [4] e Infiniband [7].

Este artigo segue com os trabalhos correlatos na Seção 2. Na Seção 3, será apresentada a proposta e os conceitos e tecnologias envolvidos. Já na Seção 4, os resultados dos experimentos são apresentados, seguidos pela Seção 5 com as conclusões e trabalhos futuros. Alguns agradecimentos são dedicados na Seção 6 e, em seguida, são apresentadas as referências bibliográficas.

2. Trabalhos Correlatos

Os trabalhos relacionados com este trabalho de pesquisa abrangem os seguintes assuntos: caracterização e avaliação de desempenho das aplicações utilizadas nos experimentos ou impacto da tecnologia *multi-core* e de processadores de rede dedicados no desempenho de *clusters*.

Em trabalhos como [30], [14] e [2] são avaliados aspectos relativos ao desempenho do WRF em ambientes de *cluster*, ao mesmo tempo em que o caracterizam em função de diversas métricas, inclusive sobre a comunicação entre os processos distribuídos. Outros trabalhos têm como objetivo identificar os padrões de comunicação dos algoritmos do NPB, como em [28], [27], [15], [29], [19] e [11], concentrando-se em características de comunicação do NPB baseados em MPI, como tipos de comunicação utilizados pelos algoritmos, tamanho das mensagens, quantidade, volume e freqüência de operações de comunicação.

O presente trabalho se relaciona com [17], [5] e [23], pois, dentre outros aspectos, avaliam o impacto da utilização de processadores dedicados para o processamento relativo à comunicação, embora com o uso de dispositivos de interconexão como Myrinet e Infiniband. Estes estudos se dão em nível de aplicação, com o uso de algoritmos largamente utilizados no meio científico como, por exemplo, o NAS Parallel Benchmark (NPB). Outros trabalhos levam em consideração aspectos relativos à recente tecnologia multi-core e seu impacto no desempenho de *clusters* em suas análises. Em [8], o foco concentra-se na comunicação intra-nó dos processos utilizando MPI e em [24] também são consideradas as vantagens de tal tecnologia em nível de compartilhamento de recursos, sendo que ambos fazem uso de benchmarks de comunicação apenas.

3. Abordagem Proposta

Devido à ausência de um processador de rede, todo processamento decorrente da comunicação via Ethernet é atribuída a um processador principal, isto é, concorrendo pelos mesmos processadores utilizados pela aplicação. No caso de um único processador por computador, a aplicação deve necessariamente parar sua execução para dar lugar ao processamento relativo à comunicação, decorrente da sua própria necessidade de interação entre os processos distribuídos, o que causa uma grande perda de desempenho.

Diferentemente, redes de interconexão mais eficientes como Myrinet e Infiniband, baseadas no modelo VIA [10], são equipadas com processadores especializados e dedicados ao processamento da comunicação, localizados nas placas de rede em cada computador e no dispositivo de interconexão (*switch*). Além disso, o fluxo de comunicação é desviado do sistema operacional e a transmissão dos dados é feita via DMA para a placa de rede, enquanto que na tecnologia Ethernet é necessário envolver o sistema operacional e algum processador principal no processo de comunicação. A Figura 1 ilustra essas diferenças.

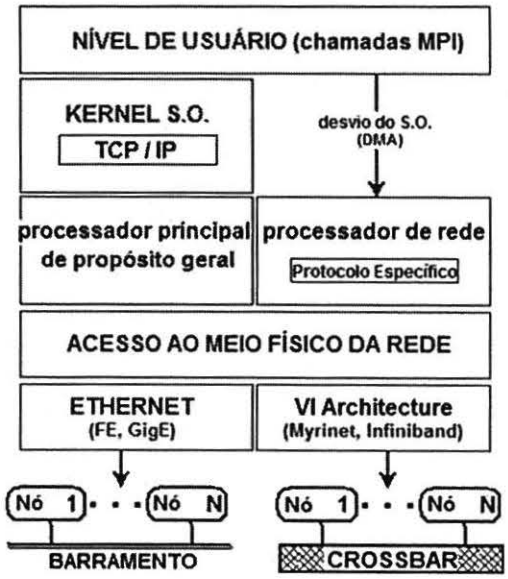

Figura 1. Fluxo dos modelos Ethernet e VIA.

Ainda mais com o advento dos processadores *multi-core*, computadores com múltiplos núcleos estão se consolidando como uma tecnologia de prateleira (*commodity*). Quando tais computadores são agregados via Ethernet para a composição de um *cluster*, surge a possibilidade de melhorar o desempenho do sistema através da sobreposição do processamento da aplicação e do processamento decorrente da comunicação entre os processos da aplicação. Porém, quando todos os núcleos são alocados para a execução

da aplicação, ocorrem inúmeras trocas de contexto entre os processos dos diferentes níveis, já que a efetivação da comunicação não se dá em nível de aplicação. Conforme [12], uma troca de contexto de um processo para outro tipicamente requer de centenas a milhares de ciclos de processamento, provocando um *overhead* considerável.

Enfim, a abordagem proposta sugere que não sejam alocados aos processos da aplicação todos os núcleos de processamento disponíveis em cada computador, de modo que os núcleos então ociosos em relação à aplicação podem se ocupar com as tarefas decorrentes de sua execução, como o processamento de pacotes e protocolos de comunicação. Portanto, em função desse paralelismo, espera-se que o tempo gasto com o processamento decorrente da comunicação entre processos tenha seu impacto reduzido, traduzindo-se em um menor tempo de execução da aplicação. Esta é a idéia norteadora deste estudo.

4. Experimentos

Serão avaliadas três configurações de *cluster* distintas no que diz respeito à distribuição dos processos e o número de núcleos de processamento utilizados por computador agregado. Além disso, experimentos em um único computador multiprocessado com até 8 núcleos de processamento serão apresentados a titulo de comparação, ampliando a análise em nível de arquiteturas paralelas e de processadores.

INFO / SISTEMA	N8xP2	N8xP1	N4xP2	CMP-SMP
Interconexão	Gigabit Ethernet (GigE)			Crossbar
	3Com Switch 3812			HyperTransport
MTU	1500			N/A
Modelo do processador	32-bit Intel Xeon (DP)			64-bit AMD Opteron 2350
Veloc. do processador	2.66 GHz			2 Ghz
Tecnologia Manuf.	130nm			65nm
# Cores por soquete	1			4
# Cores por nó	2	2	2	8
# nós	8	8	4	1
# Cores Ativo/Ocioso	*16/0*	*8/8*	*8/0*	*variável*
	POR CORE			
Cache L1 (I/D)	12KB/8KB			64KB/64KB
Cache L2	512KB			512KB
Cache L3	-			2MB
DRAM	512MB	1GB	512MB	1GB
Velocidade DRAM	533MHz			1000Mhz
BogoMIPS	~ 5300			~ 4000

Figura 2. Ambientes de experimentação

De antemão, alguns termos devem ser definidos. Um ***core*** ou **núcleo** é a unidade atômica de processamento dos sistemas. Um **soquete** contém um ou mais *cores*. Um **nó** é uma máquina independente com um ou mais soquetes, que compartilham recursos como a memória principal e o acesso à rede de interconexão. Um ***cluster***, **sistema** ou **ambiente** é um conjunto de nós interconectados. No entanto, note que o sistema CMP-SMP não é um *cluster*, e sim um nó multiprocessado com dois processadores *quad-core*.

A nomenclatura dos sistemas seguem um padrão conforme o número de nós (N) e o número de núcleos de processamento ativos por nó (P). Por exemplo, o sistema N8xP1 é composto por 8 computadores (N = 8), cada um com 2 núcleos de processamento, porém apenas 1 núcleo é alocado para a aplicação (P = 1). O segundo núcleo de cada computador está, portanto, ocioso em relação à aplicação.

Os sistemas baseados em processadores Xeon [13] rodam Linux kernel 2.6.8.1 e os sistemas baseados em processadores Opteron [1] rodam Linux kernel 2.6.22.8, todos com suporte a SMP ativado. Além disso, todos os sistemas estão isolados de ruídos externos e dedicados aos experimentos, isto é, não estão operando quaisquer outros serviços, exceto a configuração mínima necessária à execução dos experimentos com a biblioteca MPICH.

É importante ressaltar que as aplicações usadas nestes experimentos são baseadas exclusivamente em MPI para a extração do paralelismo, embora haja trabalhos [26, 6] que indiquem um modelo de programação híbrida, com OpenMP para paralelismo intra-nó e MPI entre nós, como o meio mais eficiente de utilizar agregados de computadores multiprocessados. Porém, como o foco deste trabalho está em analisar a sobreposição de computação e comunicação, levou-se em consideração apenas a comunicação MPI por soquete entre os processos, seja entre nós ou intra-nó. Além disso, programação paralela com MPI deve continuar importante por razões de portabilidade e até mesmo pela enorme quantidade de aplicações baseadas em MPI.

Em nível de aplicação, serão utilizados os seguintes *workloads* na avaliação de desempenho da proposta:

1. ***b_eff:*** *micro-benchmark* de rede, utilizado para capturar características específicas da comunicação entre processos de interesse primário, como latência e taxa de transferência [9];

2. ***NAS Parallel Benchmarks (NPBv2.3):*** conjunto de algoritmos de *benchmark* consolidados na avaliação de computadores paralelos, utilizado para capturar o impacto da proposta tendo em vista aplicações de diversas granularidades;

3. ***Weather Research and Forecasting Model (WRFv2):*** aplicação de modelagem númerica da previsão do tempo, largamente utilizada em ambientes de produção, utilizada para avaliar a proposta em vista de uma aplicação completa.

A utilização de categorias distintas de *workload*, desde *micro-benchmarks* até aplicações completas, serve como reforço na validação dos resultados.

LATÊNCIA (ms)	# processos	modo de comunic.	0	1K	8K	32K	64K	128K	1M	4M
N8xP2	2	uni-direcional	0.0153	0.0173	0.0250	0.0557	0.1552	0.3029	2.6355	12.5876
	2	bi-direcional	0.0168	0.0207	0.0304	0.0998	0.3246	0.7457	5.8836	23.6697
	16	bi-direcional	0.0469	0.0576	0.2327	1.0482	2.3646	5.0371	40.4980	162.1403
N8xP1	2	uni-direcional	0.0342	0.0561	0.1281	0.3371	0.6215	1.1798	9.0810	35.8172
	2	bi-direcional	0.0261	0.0381	0.1211	0.4015	0.8061	1.5252	11.7471	44.2522
	8	bi-direcional	0.0289	0.0417	0.1300	0.4360	0.8788	1.7858	12.7988	51.4637
N4xP2	2	uni-direcional	0.0153	0.0173	0.0250	0.0557	0.1552	0.3029	2.6355	12.5876
	2	bi-direcional	0.0168	0.0207	0.0304	0.0998	0.3246	0.7457	5.8836	23.6697
	8	bi-direcional	0.0337	0.0463	0.1992	0.7589	1.6351	3.2785	24.2303	99.6214

Figura 3. Latência coletada com o b_eff

TX. TRANSF. (MB/s)	# processos	modo de comunic.	0	1K	8K	32K	64K	128K	1M	4M
N8xP2	2	uni-direcional	0.1173	54.0247	292.4233	528.6114	421.2688	422.7265	378.9459	371.8689
	2	bi-direcional	0.1143	47.3600	272.5599	301.3550	206.6780	178.6099	156.8393	171.6436
	16	bi-direcional	0.0396	16.3430	33.6024	30.9186	27.1158	25.5238	25.6144	25.2320
N8xP1	2	uni-direcional	0.0548	16.6795	63.5115	97.2331	104.8752	110.8938	115.5708	117.1184
	2	bi-direcional	0.0802	26.8313	67.4022	77.7124	81.6810	85.7770	88.7027	95.0442
	8	bi-direcional	0.0611	21.8215	59.9733	74.0701	75.1796	74.3785	79.4108	80.0891
N4xP2	2	uni-direcional	0.1173	54.0247	292.4233	528.6114	421.2688	422.7265	378.9459	371.8689
	2	bi-direcional	0.1143	47.3600	272.5599	301.3550	206.6780	178.6099	156.8393	171.6436
	8	bi-direcional	0.0519	19.6169	39.4759	46.1811	43.6014	42.9530	41.3949	44.3636

Figura 4. Taxa de transferência coletada com o b_eff

4.1. Benchmark de rede: *b_eff*

Para caracterizar a taxa de transferência e a latência dos ambientes de *cluster* (exceto o sistema CMP-SMP), foi executado o algoritmo de benchmark de comunicação *b_eff* [25], que possui uma versão disponível como parte do HPC Challenge Benchmark [18]. No entanto, esta versão testa a taxa de transferência e latência apenas para mensagens de tamanho 8 e 2.000.000 bytes. Sendo assim, adaptamos o algoritmo *b_eff* para avaliar as características de comunicação para uma gama de tamanhos de mensagens.

Com base nos resultados coletados, conforme as Figuras 3 e 4, verifica-se que as configurações N4xP2 e N8xP1 apresentam resultados de latência e taxa de transferência semelhantes para a comunicação entre dois processos, tanto uni-direcional como bi-direcional, pois estão localizados no mesmo nó e os processos utilizam, portanto, o barramento interno do computador como rede de interconexão. Já no ambiente N8xP1, a comunicação entre dois processos é cerca de duas vezes pior porque a comunicação se dá através da rede Gigabit Ethernet, já que os processos estão localizados em computadores distintos.

Porém, vale ressaltar que em uma aplicação com 8 ou 16 processos, por exemplo, é provável que mais do que dois processos se comuniquem simultaneamente, possivelmente todos. Por isso, foram coletados dados referentes ao pior caso, em que todos os processos se comunicam ao mesmo tempo e nos dois sentidos, a fim de melhor caracterizar os ambientes em termos de latência e taxa de transferência.

Neste caso, os resultados são bem diferentes em comparação à situação em que apenas dois processos se comunicam. O ambiente de *cluster* N8xP1 passa a apresentar o melhor desempenho tanto em latência como em taxa de transferência, e o ambiente N8xP2 apresenta os piores valores em ambas as métricas.

A maior demanda por uma característica de comunicação ou outra dependerá da aplicação paralela, porém, essa caracterização mais ampla do desempenho da comunicação entre processos, através da taxa de transferência e latência de todos processos se comunicando simultaneamente, permite revelar vantagens e desvantagens não percebidas quando apenas 2 processos se comunicam.

4.2. NAS Parallel Benchmarks

O NPB é um conjunto de oito algoritmos de benchmark. Cinco deles são denominados kernel benchmarks (EP, FT, IS, CG e MG) e os outros três são considerados aplicações simuladas de benchmark (SP, BT e LU) [3]. A compilação foi configurada igualmente para todos os sistemas, utilizando-se a diretiva de otimização O3. Este trabalho abrange tão somente os algoritmos EP, FT e IS, pois estes algoritmos apresentam granularidades bem distintas. O conceito de granularidade está associado à razão do tempo gasto com computação em relação ao tempo gasto com comunicação entre os processos distribuídos.

O algoritmo EP (*Embarrassingly Parallel*) é um gerador de números aleatórios que apresenta alta demanda por computação e necessidade de comunicação desprezível.

NAS.EP.B	N8xP2 (16 proc.)	N8xP1 (8 proc.)	N4xP2 (8 proc.)
Tempo de execução	25,9	47,7	47,3
Redução de tempo	0%	-84,2%	-82,7%
Speedup Relativo	1	0,54	0,55

NAS.FT.B	N8xP2 (16 proc.)	N8xP1 (8 proc.)	N4xP2 (8 proc.)
Tempo de execução	57,5	53,2	82,7
Redução de tempo	0%	7,48%	-43,8%
Speedup Relativo	1	1,08	0,70

NAS.IS.B	N8xP2 (16 proc.)	N8xP1 (8 proc.)	N4xP2 (8 proc.)
Tempo de execução	4,23	3,23	4,60
Redução de tempo	0%	23,6%	-8,75%
Speedup Relativo	1	1,31	0,92

Figura 5. Resultados do NAS EP, FT e IS

Sua demanda por comunicação limita-se à distribuição de dados em um momento inicial e ao reagrupamento dos resultados parciais ao final de sua execução. Portanto, o algoritmo EP é de **granularidade grossa**.

De fato, os resultados reafirmam essas características do algoritmo EP, pois com 16 processadores alocados à aplicação, seu tempo de execução diminui quase pela metade, como mostra a Figura 5, enquanto que não há uma diferença significativa entre os resultados com 8 processos, obtidos pelos sistemas N8xP1 e N4xP2.

Já o algoritmo FT (*FFT 3D PDE*), que resolve equações diferenciais parciais através de FFT's 1D em série, apresenta alta demanda por computação assim como por comunicação, segundo um padrão de todos para todos perfeitamente balanceado. Como os autores do NPB tomaram o cuidado de agregar as mensagens em nível de aplicação para minimizar o custo de comunicação, o resultado é um algoritmo de **granularidade média**, com um grande volume de dados transmitidos embora através de mensagens grandes, a maioria da ordem de megabytes, em uma freqüência baixa.

Apesar dessa baixa freqüência das operações de comunicação, pode-se apontar um ganho leve no desempenho do algoritmo FT quando o *cluster* é composto segundo a abordagem proposta. Já a configuração N4xP2 resultou em uma enorme perda de desempenho em relação às configurações N8xP2 e N8xP1, o que revela certa fragilidade do subsistema de memória e de E/S desses computadores. Vale ressaltar que, embora a necessidade de comunicação com maior taxa de transferência cresça proporcionalmente ao desempenho do processador, esse aumento não é significativo com o crescimento do número de processadores usados na execução do algoritmo FT [28].

O algoritmo IS (*Integer Sort*) é um ordenador de inteiros. Predominam comunicações de redução e de todos para todos não-balanceadas com muitas mensagens pequenas (até poucos kilobytes) e grandes (da ordem de megabytes), mas poucas mensagens de tamanho mediano. A freqüência das mensagens é alta e o volume total de dados transmitidos é considerável. Por isso, a granularidade do algoritmo IS é considerada menor do que dos algoritmos EP e FT, caracterizando-se como de **granularidade fina**.

Mais uma vez, o desempenho do sistema N4xP2 mostrou-se o pior dentre as três configurações. Por outro lado, o *cluster* configurado segundo a abordagem proposta, o sistema N8xP1 com 8 processos executando a aplicação, obteve uma redução de mais de 20% em comparação com o tempo obtido pelo sistema N8xP2, com 16 processos, na execução do algoritmo de granularidade fina IS.

Portanto, com base nos resultados obtidos com o NPB, é possível apontar que o sistema N8xP1, configurado segundo a abordagem proposta, teve sucesso em minimizar o overhead decorrente do processo de comunicação entre os processos. Exceto para o algoritmo de granularidade grossa, houve ganho de desempenho em relação à configuração N8xP2, na qual todos os 16 núcleos de processamento disponíveis foram alocados a processos da aplicação.

4.3. Aplicação de previsão do tempo: WRF

A aplicação WRF (*Weather Research and Forecasting Model*) é um sistema de previsão meteorológica baseado em modelos numéricos, ativamente desenvolvida por um consórcio de agências governamentais, como o NCAR (*National Center for Atmospheric Research*), em parceria com a comunidade científica dos Estados Unidos e internacional [22]. É amplamente utilizado tanto na previsão de tempo operacional como na pesquisa atmosférica.

O WRF utiliza uma matriz tri-dimensional para a representação da atmosfera, desde metros até milhares de quilômetros, com diversas informações como de topografia e dados de observatórios para alimentar a simulação com uma condição inicial. As simulações aqui apresentadas tomam como entrada o conjunto de dados da forma que é utilizado no ambiente de produção da EPAGRI S.A., que realiza a previsão do tempo para o estado de Santa Catarina.

Na execução paralela do modelo WRF, cada processo recebe uma sub-matriz de tamanho aproximadamente igual, que diminui com o aumento do número de processos. Quanto à comunicação, a redistribuição dos dados laterais, nos quatro limites lógicos de cada sub-matriz, é a principal atividade, ocorrendo a cada iteração com mensagens entre

10 e 100 kilobytes [14]. Cada iteração avança o tempo de simulação em 75 segundos, totalizando 96 e 576 iterações nas previsões para 2 e 12 horas, respectivamente. Além disso, a cada 20 iterações ocorre uma iteração de radiação física, que se soma ao tempo de processamento da iteração.

2 horas	N8xP2 (16 proc.)	N8xP1 (8 proc.)	N4xP2 (8 proc.)
Tempo de execução (min:seg)	24:12	18:03	24:50
Redução de tempo	0%	25,4%	-2,6%
Speedup Relativo	1	1,34	0,98

12 horas	N8xP2 (16 proc.)	N8xP1 (8 proc.)	N4xP2 (8 proc.)
Tempo de execução (h:mm:ss)	2:01:07	1:33:49	2:01:54
Redução de tempo	0%	22,5%	-0,7%
Speedup Relativo	1	1,29	0,99

Figura 6. Resultados do WRF no cluster

Conforme a Figura 6, as previsões de tempo para 2 e 12 horas, apresentaram uma redução maior do que 20% no tempo de execução. Os resultados obtidos com o WRF mostram um ganho de desempenho quando o *cluster* é configurado segundo a proposta, atingindo um *speedup* relativo de 1,29 em comparação à previsão do tempo para 12 horas no ambiente de *cluster* original, com 16 processos.

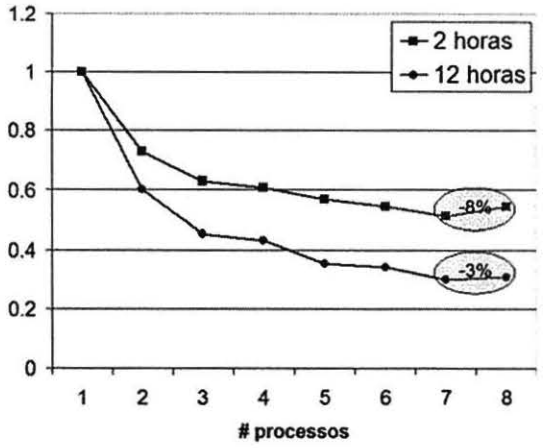

Figura 7. Speedup do WRF no sist. CMP-SMP

Em adição aos experimentos em ambientes de *cluster*, a Figura 7 apresenta os resultados obtidos em uma única máquina multiprocessada, com dois processadores AMD *quad-core*. Pode-se observar que, embora a interação entre os processos se dê sem necessidade de acesso à rede, o tempo de execução aumenta quando todos os processadores disponíveis são alocados a processos da aplicação. Esta constatação vem para reforçar as indicações dos resultados anteriores no sentido de que a abordagem proposta é realmente pertinente, oferecendo melhores eficiência e desempenho, traduzidos em um menor tempo de execução.

5. Conclusões e Trabalhos Futuros

Como indicação do estudo empírico desenvolvido nesta pesquisa, foi possível comprovar o sucesso da abordagem proposta para compor *clusters* eficientes para a execução do modelo numérico WRF de previsão de tempo, oferecendo uma alternativa para ganhar desempenho em *clusters* interconectados via Gigabit Ethernet sem a necessidade de adquirir, por exemplo, rede de interconexão mais eficiente. Vale ressaltar que a abordagem aproveita-se da tendência atual com relação a computadores multiprocessados.

Embora apenas 8 núcleos tenham sido alocados para a aplicação, houve ganho de desempenho em relação à configuração de *cluster* no qual todos os 16 núcleos de processamento disponíveis foram alocados a processos da aplicação. O ganho expressivo no desempenho do modelo WRF, com uma redução de mais de 20% no seu tempo de execução, também foi verificado para o algoritmo de granularidade fina IS do NAS Parallel Benchmark. Por outro lado, a abordagem se mostrou ineficiente para o algoritmo de granularidade grossa EP.

Portanto, a eficiência da abordagem dependerá da aplicação paralela, porém, a caracterização das capacidades de comunicação de cada ambiente com o benchmark de rede *b_eff*, através da taxa de transferência e latência de todos processos se comunicando simultaneamente, também revelou vantagens do *cluster* configurado segundo a abordagem, não percebidas quando apenas 2 processos se comunicam.

Ademais, reduzir o tempo de execução do modelo numérico torna-se ainda mais importante quando se tem em mente que todo o processo de previsão de tempo, que inclui o modelo WRF [21], é composto de outras etapas que não foram consideradas nesse trabalho. Há, por exemplo, a necessidade de um pós-processamento dos dados gerados pelo modelo para permitir a vizualização, que efetivamente auxilia os meteorologistas na previsão do tempo.

Como trabalhos futuros, indica-se a investigação mais aprofundada das causas que permitiram o ganho de desempenho apresentado pelo *cluster* configurado segundo a abordagem proposta. Além disso, pretende-se mensurar a potencialidade da proposta em comparação aos ganhos rela-

tivos a um mesmo cluster interconectado com uma rede de interconexão mais eficiente, como Myrinet ou Infiniband.

6. Agradecimentos

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

Referências

[1] AMD. Amd opteron(tm) processor product data sheet. Technical report, Publication 23932, 2007.

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

[3] D. H. Bailey, H. Barszcz, J. T. Barton, D. S. Browning, R. L. Carter, L. Dagum, R. A. Fatoohi, P. O. Frederickson, T. A. Lasinski, R. S. Schreiber, H. D. Simmon, V. Venkatakrishnan, and S. K. Weeratunga. The nas parallel benchmarks. *International Journal of Supercomputer Applications*, 5(3):63–73, 1991.

[4] N. Boden, D. Cohen, R. Felderman, A. Kulawik, C. Seitz, J. Seizovic, and W. Su. Myrinet: A gigabit-per-second local area network. *IEEE Micro*, 15(1):29–36, 1995.

[5] R. Brightwell and K. Underwood. An analysis of the impact of mpi overlap and independent progress. *International Conference on Supercomputing*, 2004.

[6] F. Cappello and D. Etiemble. Mpi versus mpi+openmp on the ibm sp for the nas benchmarks. *Supercomputing*, 2000.

[7] D. Cassiday. Infiniband architecture. *Hot Chips 12*, 2000.

[8] 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*, 2006.

[9] G. Coulouris, J. Dollimore, and T. Kindberg. *Distributed systems: Concepts and Design*. Addison Wesley, 4ª edition, 2005.

[10] D. Dunning, G. Regnier, G. McAlpine, D. Cameron, B. Shubert, F. Berry, A. M. Merritt, E. Gronke, and C. Dodd. The virtual interface architecture. *IEEE Micro*, 18(2):66–76, 1998.

[11] A. Faraj and X. Yuan. Communication characteristics in the nas parallel benchmarks. *Parallel and Distributed Computing and Systems*, 2002.

[12] J. L. Hennessy and D. A. Patterson. *Computer Architecture - A Quantitative Approach*. Morgan Kaufmann Publishers, 3ª edition, 2003.

[13] Intel®. Intel® xeon® processor with 533 mhz fsb at 2ghz to 3.20ghz datasheet. Technical report, Publ. 252135, 2004.

[14] D. Kerbyson, K. Barker, and K. Davis. Analysis of the weather research and forcasting (wrf) model on large-scale systems. 2007.

[15] J. Kim and D. Lilja. Characterization of communication patterns in message-passing parallel scientific application programs. *Communication, Architecture, and Applications for Network-Based Parallel Computing*, pages 202–216, 1998.

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

[17] M. Lobosco, V. S. Costa, and C. L. de Amorim. Performance evaluation of fast ethernet, giganet and myrinet on a cluster. *International Conference on Computational Science*, pages 296–305, 2002.

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

[19] R. Martin. *A Systematic Characterization of Application Sensitivity to Network Performance*. PhD thesis, Berkeley, 1999.

[20] H. Meuer, E. Strohmaier, J. Dongarra, H. D. Simon, U. of Mannheim, and U. of Tennessee. Top500 supercomputing sites (www.top500.org), 2008.

[21] 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*, 2004.

[22] 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.

[23] L. C. Pinto, R. P. Mendonça, and M. A. R. Dantas. The impact of interconnection networks and applications granularity to compound cluster configurations. *IEEE Symposium on Computers and Communications*, 2008.

[24] 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.

[25] 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.

[26] R. Rabenseifner and G. Wellein. Communication and optimization aspects of parallel programming models on hybrid architectures. *International Journal of High Performance Computing Applications*, 17(1):49–62, 2003.

[27] J. Subhlok, S. Venkataramaiah, and A. Singh. Characterizing nas benchmark performance on shared heterogeneous networks. *IEEE International Parallel and Distributed Processing Symposium*, pages 86–94, 2002.

[28] Y. Sun, J. Wang, and Z. Xu. Architetural implications of the nas mg and ft parallel benchmarks. *Advances in Parallel and Distributed Computing*, pages 235–240, 1997.

[29] T. Tabe and Q. Stout. The use of mpi communication library in the nas parallel benchmarks. Technical report. Technical Report CSE-TR-386-99, University of Michigan, 1999.

[30] R. Zamani and A. Afsahi. Communication characteristics of message-passing scientific and engineering applications. *International Conference on Parallel and Distributed Computing and Systems (PDCS)*, pages 644–649, 2005.