Análise da distribuição de carga em um *cluster* heterogêneo.

Josemar Souza [*], *D. Rexachs* [**], *E. Luque* [**].

(josemar@stn.com.br, d.rexachs@cc.uab.es , e.luque@cc.uab.es,)

[*] Universidade Católica do Salvador
Curso de Informática
Salvador, Bahia, Brasil

[**] Universidad Autónoma de Barcelona
Departamento de Informática
Unidad De Arquitectura de Ordenadores y Sistemas Operativos
Bellatera, Barcelona – Espanha

Resumo. Este trabalho apresenta o estudo do desempenho, eficiência e balanceamento de carga em um computador paralelo heterogêneo, com comunicação por passo de mensagem e formado por computadores interconectados através de uma rede local, com grande grau de heterogeneidade de até 40 a 1.

O paradigma de programação utilizado foi o SPMD, implementado sob o esquema Mestre/Trabalhador. A multiplicação de matrizes foi eleita como *benchmark* por suas características de escalabilidade.

Para a analise de desempenho e eficiência foram desenvolvidas 4 versões do programa de multiplicação de matrizes que cobrem os aspectos de carga balanceada e simétrica e distribuição estática e dinâmica. Através da geração de tráfego adicional de comunicação na rede, se mediu a influencia das comunicação em tempo de execução dos programas de teste.

Palavras chave: PVM, NOW, SPMD, Redes de Computadores, Sistemas Heterogêneos, Balanceamento de Carga, Computação Paralela.

1. Introdução

Apesar dos avanços da área de ciência da computação e em particular das arquiteturas computacionais, ainda há diversos problemas reais que são de difícil solução em plataformas computacionais correntes em virtude do seu alto custo. Há uma tendência crescente ao uso de sistemas de computação paralela e distribuída em uma vasta gama de aplicações. Esses sistemas são compostos por vários processadores que operam concorrentemente, cooperando na execução de uma determinada tarefa. Nas chamadas arquiteturas paralelas o objetivo principal é o aumento da capacidade de processamento, utilizando o potencial oferecido por um grande número de processadores [STE99].

Apesar de se observar um grande aumento na demanda por aplicações paralelas, a utilização do processamento seqüencial assumiu um grande crescimento no decorrer dos anos, sendo as máquinas que utilizam esta tecnologia as mais divulgadas no mercado. Podemos apontar como principal responsável pelo uso em massa desses equipamentos o baixo custo de aquisição, haja visto que um

computador verdadeiramente paralelo é muito mais caro do que computadores seqüenciais. Isto torna mais difícil o acesso de usuários de pequeno porte a um hardware originalmente paralelo.

As máquinas paralelas de baixo custo, conhecidas como NOW[1], utilizam as redes de computadores comerciais, locais e/ou remotas para paralelizar suas transações. Utiliza um *software* que permite que um conjunto heterogêneo ou homogêneo de computadores (série, paralelos ou vetoriais) seja visto como uma única máquina. Dentre outros softwares poderemos utilizar o PVM[2] [ALG94] , que é um software que permite que um conjunto heterogêneo ou homogêneo de computadores seja visto como uma única máquina, sendo a portabilidade uma de suas características principais – as bibliotecas de rotinas de comunicação entre processos são "*standard*" de fato [DON-LUQ99]. A independência de plataforma que o PVM disponibiliza é indubitavelmente interessante; um software pode ser executado em ambientes diferentes, este fato gera segurança para desenvolvedores de softwares criarem aplicações paralelas, tendo em vista a portabilidade possível.

A distribuição de carga entre os nós que compõem a máquina virtual é um fator de grande importância quando estamos analisando questões relacionadas a melhoria de desempenho em ambiente PVM. O balanceamento depende de diversos fatores como homogeneidade dos grãos de paralelização obtidos na fase de decomposição do problema, a velocidade e desempenho dos computadores presentes na máquina paralela virtual, podendo ser utilizadas máquinas homogêneas – máquinas iguais com o mesmo desempenho, idênticos processadores e quantidade de memória RAM ... ou máquinas heterogêneas – máquinas diferentes, com desempenho diferente, processadores diferentes e plataformas diferentes ... [REW97]

Outro fator de grande importância é a influencia das comunicações no desempenho de um sistema de computação paralela [SNY84].

Como máquina, para formar uma máquina paralela, podemos utilizar praticamente qualquer plataforma existente, interligadas em rede, que esteja disponível ou pouco utilizada, sendo o grande responsável pelo perfeito funcionamento do sistema o software que gerencia esses projetos. Sendo que o PVM proporciona as funções necessárias para criar, comunicar e sincronizar "*tasks*" sobre a máquina virtual, de modo que cooperem com a resolução do problema [KAI98].

Para que um programa possa ser executado por um computador paralelo com memória distribuída e passagem de mensagem uma das opções é distribuir a mesma cópia do seu código entre os

[1] *Network of Workstation*

[2] *Parallel Virtual Machine*

processadores, e os distintos conjuntos de dados a processar serão distribuídos entre as diversas máquinas. Tem-se um modelo chamado SPMD[3], que foi o que utilizamos neste trabalho.

Um dos principais fatores, senão o principal, de desempenho de um algoritmo baseado em troca de mensagens é a quantidade de mensagens geradas durante a sua execução, além do número de mensagens, o custo de comunicação é muito importante.

Com o objetivo de estudar alguns aspectos da complexidade, desempenho, eficiência e balanceamento de carga através de algoritmos utilizando comunicação por passagem de mensagem, utilizamos vários *clusters* heterogêneo - formado por máquinas com desempenho diferentes com grau de heterogeneidade de até 40 a 1.

A seguir continuaremos mostrando o ambiente e a metodologia utilizada para fazer os experimentos, as medidas e métricas de desempenho, e as diversas versões do aplicativo *benchmark* executada e exemplos dos resultados obtidos, finalmente apresentamos as conclusões e linhas de trabalhos futuras.

2. Metodologia utilizadas nos experimentos

Com o objetivo de estudar alguns aspectos da complexidade, desempenho, eficiência e balanceamento de carga através de algoritmos utilizando comunicação por passagem de mensagem, montamos três máquinas paralela virtual (Fig. 1) de baixo custo dedicadas, formada por computadores seqüenciais ligados em uma rede comercial Ethernet de 10 Mbps, com sistema operacional Linux e utilizando bibliotecas já disponíveis no PVM [CUL99].

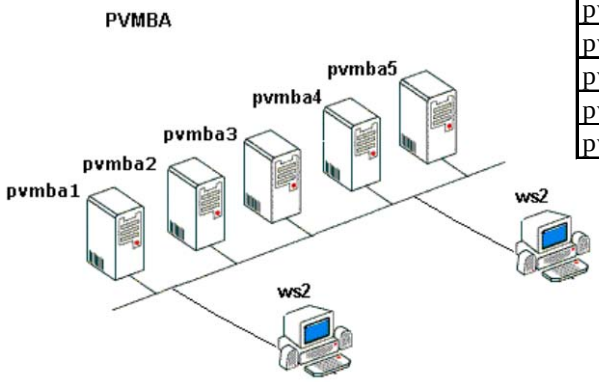
Para avaliar a influência da rede no desempenho desse Sistema de Computação Paralela utilizaremos algoritmos, que tem princípio básico de funcionamento iguais. Porém, com algumas modificações na estrutura de envio e recebimento de dados para os nós da Máquina Virtual. A diferença principal é o tamanho dos pacotes de carga enviados para cada processador (no caso dos paralelos), seguido da aplicação de comandos e funções específicos da biblioteca PVM.

O paradigma de programação utilizado foi o SPMD implementado com o esquema Mestre /Trabalhador (M/T) (Master/Worker (M/W)). O processo mestre centraliza o controle, inicia os trabalhadores, distribui o trabalho, sincroniza a comunicação e executa operações de I/O.

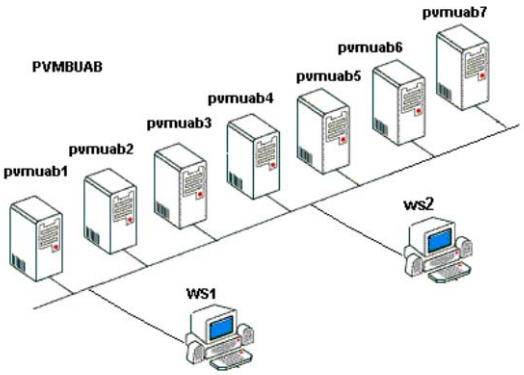
Os programas utilizados na parte experimental deste trabalho necessitam de largura de banda mínima da rede. Esta largura de banda varia ao longo do tempo segundo enviemos pequenos (linha) ou grandes pacotes (bloco). Para modificar a largura de banda disponível pela rede e o “custo de

[3] *Single Program Multiple Data*

comunicação" (latência), entre os distintos computadores do *cluster*, utilizamos um utilitário que gera trafego adicional na rede. Nos interessa estudar a influência das comunicações em tempo de execução dos programas de provas quando trabalhando com pacote pequeno ou grande.



Ambiente PVMBA-HET	
pvmba1	Intel Celeron 300 Mhz, 64 Mb
pvmba5	Pentium 200 Mhz , 128 Mb
pvmba2	Pentium 133 Mhz, 64 Mb
pvmba3	AM5x86 133 Mhz, 16 Mb
pvmba4	Intel 486 DX 4 100 Mhz, 16 Mb



Ambiente PVMUAB-HET	
pvmuab1	Pentium Celeron 433 Mhz, 32 Mb Ram
pvmuab2	Pentium Celeron 433 Mhz, 32 Mb Ram
pvmuab3	Intel 33 Mhz, 16 Mb Ram
pvmuab4	Intel 66 Mhz, 16 Mb Ram
pvmuab5	Pentium Celeron 433 Mhz, 32 Mb Ram
pvmuab6	Pentium Celeron 433 Mhz, 32 Mb Ram
pvmuab7	Pentium Celeron 433 Mhz, 32 Mb Ram

Ambiente PVMUAB_HOM	
pvmuab1	Pentium Celeron 433 Mhz, 32 Mb Ram
pvmuab2	Pentium Celeron 433 Mhz, 32 Mb Ram
pvmuab5	Pentium Celeron 433 Mhz, 32 Mb Ram
pvmuab6	Pentium Celeron 433 Mhz, 32 Mb Ram
pvmuab7	Pentium Celeron 433 Mhz, 32 Mb Ram

Fig. 1 Máquinas paralelas virtuais: PVMBA-HET , PVMUAB-HET; PVMUAB-HOM

Para medir a influência da rede no desempenho desse sistema e gerar trafego adicional na rede, para simular uma rede não dedicada (ocupação da rede), foi utilizado o utilitário *Distributed Observer® 5.0g*, analisador de protocolos baseado no Windows, © 1994-98 Network Instruments, LLC Minneapolis, MN USA (http://www.networkinstruments.com). Esse utilitário nos permite gerar trafego adicional trocando tanto o tamanho dos pacotes quanto a freqüência (PKTs/s) e monitorar a largura de banda da rede [OBS98].

O trafego adicional na rede foi gerado através do *Traffic Generator* e para monitoramento da rede foi utilizado *o Bandwith Utilization*, contido no *Distributed Observer*. O trafego adicional foi gerado pela WS1 enviando pacotes para a WS2, mostrada na Fig. 1 .

O programa multiplicação de matrizes foi escolhido como *benchmark*, por ser facilmente escalavel tanto em cômputo quanto em comunicação .

Executamos diferentes versões do programa de multiplicação de matrizes *n x n*, onde *n* é a dimensão da matriz quadrada envolvida no processo se dá conforme o esquema tradicional A*B = C (Fig. 2), com a decomposição em matrizes unidimensionais (linhas) para efeitos de envio para os nós da rede.

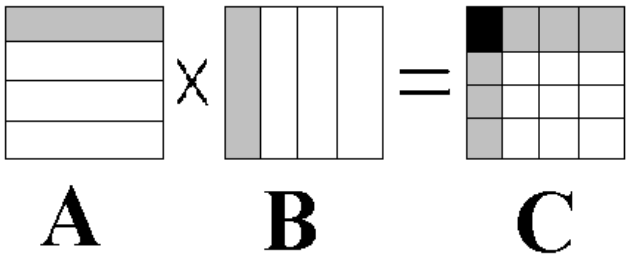


Fig. 2 – Modelo utilizado para multiplicação de matrizes

Para executar a multiplicação de matrizes em paralelo, utilizaremos M/T. No processamento da multiplicação de matriz nesse ambiente paralelo o processo mestre envia inicialmente toda a matriz B para os trabalhadores, e em seguida envia um certo número de linhas da matriz A que cada processo trabalhador executará, a contagem de tempo só é iniciada neste momento.

As maquinas utilizadas só tem capacidade de executar com alocação dinâmica de memória, matrizes de ate 800 x 800, que será o tamanho máximo utilizado nos experimentos apresentados neste artigo. Em matrizes acima de 800 X 800 utilizamos alocação estática de memória, esse tipo de alocação ocasiona aumento do tempo de execução, por isso para interpretar os resultados das métricas executando matrizes desse porte teremos que adicionar outros fatores, que não são representativos para analise que se realizam neste artigo

2.1 Medidas de Desempenho e Métricas

Medidas de desempenho, que permitam a análise do ganho obtido com o aumento do total de processadores utilizados, são necessárias. Algumas medidas utilizadas são: Tempo de Execução, *"speedup"* (ganho de desempenho) e eficiência.

Tempo de Execução – Texec de um programa paralelo é o tempo decorrido desde o primeiro processador iniciar a execução do problema (Ti) até o último terminar (Tf).

$$T_{exec} = T_f - T_i = f(T_{comp}, T_{comu}, T_{ocio})$$

O tempo de execução utilizado nas analises e mostrados nas tabelas, foram obtidos a partir dos resultados emitidos pelo mestre, sendo que a contagem de tempo é iniciada no momento em que começa a enviar as matrizes aos processos trabalhador e finaliza quando recebe o ultimo resultado.

Em um instante da execução, um processador está em fase de computação, comunicação, ou ociosidade. Representa-se por Tcomp, Tcomu e Tocio os tempos gastos por um processador em cada uma das três fases.

***Speedup* – S** obtido por um algoritmo paralelo executando sobre p processadores é a razão entre o tempo levado por aquele computador executando o algoritmo serial mais rápido (Ts) e o tempo levado pelo mesmo computador executando o algoritmo paralelo usando p processadores (Tp).

$$S = T_s / T_p$$

Eficiência – E é a fração do tempo em que os processadores estão ativos. É usada para medir a qualidade de um algoritmo paralelo.

É a razão entre o *speed-up* e a quantidade P de processadores. Esta medida mostra quanto o paralelismo foi explorado no algoritmo. Quanto maior a fração menor será a eficiência .

$$E = S / P = Ts/PTp$$

2.2 Influencia da rede: trafego adicional

Para analisar a influencia na rede, foi gerado um trafego adiciona, entre as estações de trabalho WS1 e WS2 (Vide Fig.1). Este programa nos permite trocar tamanho dos pacotes, freqüência (PKTs/s). Com o objetivo de gerar diferentes ocupações da largura de banda da rede (50%, 75%, 80% e 90%) escolhemos dois tamanhos de pacotes, grandes (1500 B) e pequenos (500 B) e as freqüências como é mostrada na Tab. 1.

Tab. 1 Diferentes ocupações da largura de banda da rede

	Pacotes pequenos	Pacotes Grandes	Pacotes pequenos	Pacotes Grandes
Utilização	50%	75%	80%	90%
Tamanho do pacote (B)	600	900	500	1500
Pacotes / seg	1000	1000	3500	3000
PKTS/s (efetivo)			2019	740
MB/s (efetivo)			1.01	1.16

2.3 Caracterização da potência das maquinas

A versão serie do programa multiplicação de matrizes será utilizado para caracterizar a potência das maquinas e analisar a capacidade computacional de cada uma das máquinas que compõem a máquina paralela virtual. Esse programa será executado em todas as maquinas que comporão a maquina paralela virtual isto nos permitirá ter os índices para calcular o speed-up, calcular o coeficiente de heterogeneidade das maquinas e o Grau de Heterogeneidade.

Coeficiente de heterogeneidade – cH(i) para a máquina "i" se define como a relação entre o tempo de execução do programa multiplicação de matrizes na maquina mais rápida e o tempo de execução do mesmo programa na maquina "i".

O Grau de heterogeneidade–gH é a razão entre o tempo levado pelo computador executando o algoritmo serial mais rápido (tR) e o tempo levado pelo computador executando o mesmo algoritmo serial mais lento (tL).

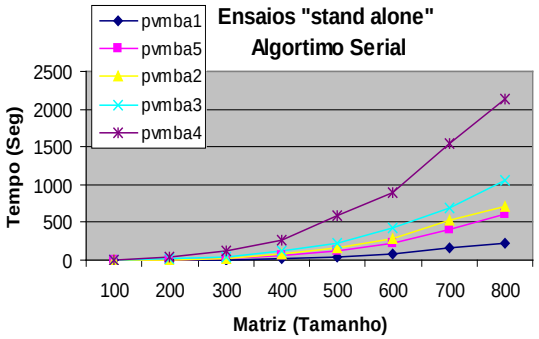
$$gH = t_R / t_L$$

Na Tab. 2. se mostram os resultados obtidos nas máquinas que compõem a maquina paralela PVMBA-HET, chamadas pvmba(x). Para cada maquina trabalhador se da o tempo de execução em segundos e o respectivo coeficiente de heterogeneidade.

Tab. 2 Algoritmo Serial executado na maquina PVMBA-HET

Alg. Serial	**Matriz (tamanho)**							
Maquina	**100**	**200**	**300**	**400**	**500**	**600**	**700**	**800**
pvmba1 M	1	3	10	27	49	83	160	226
pvmba5 T	1	5	23	62	128	231	415	600
Coef.	**1**	**0.6**	**0.43**	**0.4**	**0.38**	**0.4**	**0.39**	**0.38**
pvmba2 T	1	8	30	78	161	286	527	721
Coef.	**1**	**0.4**	**0.33**	**0.3**	**0.3**	**0.3**	**0.3**	**0.31**
pvmba3 T	1	13	46	119	230	417	691	1056
Coef.	**1**	**0.2**	**0.22**	**0.2**	**0.21**	**0.2**	**0.23**	**0.21**
pvmba4 T	4	31	115	257	586	886	1544	2143
Coef.	**0.3**	**0.1**	**0.09**	**0.1**	**0.08**	**0.1**	**0.1**	**0.11**

seg



Como pode-se observar o grau de heterogeneidade da maquina pvmba é ate 10 a 1

Na Tab. 3. se mostram os resultados obtidos nas máquinas que compõem a maquina paralela PVMUAB, chamadas pvmuab(x). Como existem varias maquinas trabalhador iguais, essas serão chamadas com 433 Mhz (Celeron). Neste caso, para cada maquina trabalhador se mostra o tempo de execução em milisegundos e os respectivos coeficientes de heterogeneidade obtidos e o coeficiente final usado para balancear a carga (cH3´ = 0,025, cH4´ = 0,075) que foi ajustado para ter em conta também o tempo de comunicação , o tempo da sobreposição (*overlap*) cômputo comunicação e a ordem de envio para as maquinas trabalhador (Tocio).

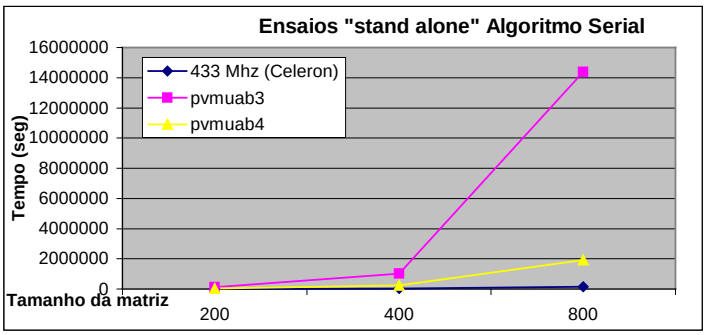
Como veremos a seguir, a experimentação com os programas que fazem uma distribuição de carga dinâmica dão coeficientes similares.

Pelo gráfico, observa-se que a máquina 433 Mhz Celeron (pvmuab1, pvmuab2, pvmuab5, pvmuab6 e pvmuab7) é a mais rápida. Esta é a maquina que utilizamos como referência para o calculo do speed-up e do coeficiente relativo de heterogeneidade.

Tab. 3 Algoritmo Serial executado na maquina PVMUAB

Alg. Serial	**Matriz (tamanho)**		
Maquina	**200**	**400**	**800**
433 Mhz (Celeron)	1853	16676	160935
cH2, cH5, cH6, cH7	**1**	**1**	**1**
pvmuab3	120940	1029458	14369134
cH3	**0.015**	**0.016**	**0.011**
pvmuab4	27486	235345	1916999
cH4	**0.067**	**0.071**	**0.084**

mseg



3. Aplicação *benchmark* e resultados experimentais

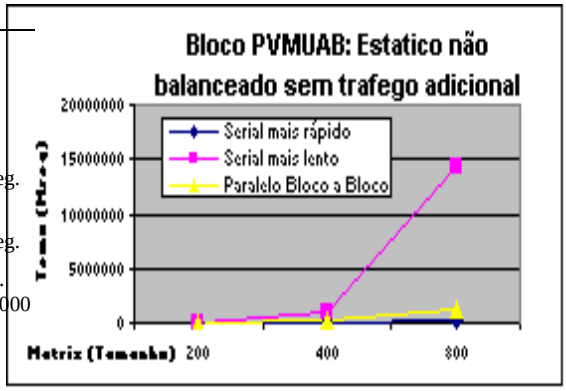
Foram desenhadas quatro versões diferentes do algoritmo de multiplicação de matrizes, com o objetivo de poder realizar os diferentes experimentos que nos permitiram analisar o comportamento dos dois sistemas heterogêneos (PVMBA-HET e PVMUAB-HET) (Fig. 1) e também foram executados na máquina homogênea PVMUAB-HOM (Fig. 1), para ter uma referência da eficiência máxima lograda com estes programas[SOU2000]. A seguir, apresentamos diferentes versões junto com alguns do resultados experimentais obtidos

As quatro versões mencionada, são descritas a seguir:

- **"bloco"** - bloco simétrico (não balanceada) assincrono estático e pacotes grandes**:** depois de enviar a matriz "B" divide as linhas da matriz "A" pelo número de trabalhadores, obtendendo o que denominamos "bloco" e lhes envia indistintamente aos trabalhadores. Neste caso, o processo mestre espera que a máquina mais lenta acabe. Utilizamos este programa porque ele distribui a carga não balanceada, pacotes grandes e execução de maneira assíncrona. Cada trabalhador tem um tempo diferente para calcular. A Tab. 4 mostra informação sobre esse experimento.

Tab. 4 Bloco PVMUAB: Estático não balanceado sem trafego adicional

Algoritmo Paralelo Bloco a Bloco (bloco)			**Matriz (tamanho)**			
Algoritmo	**Maquina**	**Tipo do ensaio**	**200**	**400**	**800**	
Algoritmo Serial	433 Mhz (Celeron)	"stand alone"	**1853**	**16676**	**160935**	Mseg.
Algoritmo Serial	pvmuab3	"stand alone"	**120940**	**1029458**	**14369134**	Mseg.
Paralelo Bloco a bloco	pvmuab	Sem trafego adicional	**20000**	**154000**	**1287000**	Seg. X 1000
		Speed-up	0.09	0.11	0.13	

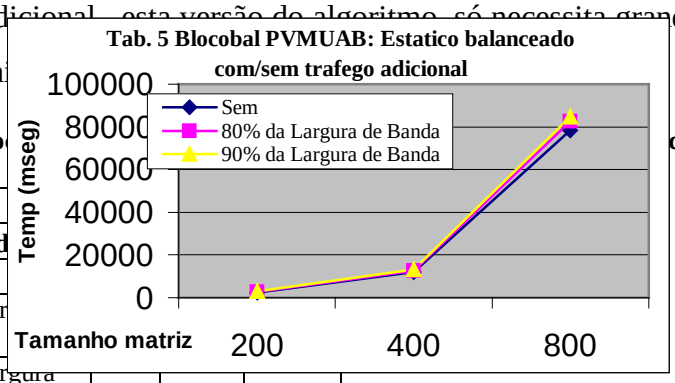


Quando existe uma maquina muito lenta o *speed-up*

resulta menor do que a unidade, pois esta limitado pela máquina mais lenta. Observamos que seria conveniente ou retirar a máquina mais lenta ou balancear a carga da maquina paralela.

Quanto a comunicação, os experimentos mostraram que quando colocamos trafego adicional, numa fase inicial é requerida uma grande largura de banda.

- **"blocobal"** bloco balanceado assincrono estático e pacote grande: depois de enviar a matriz "B" divide as linhas da matriz "A" pelo número de trabalhadores, de maneira diretamente proporcional ao coeficiente de heterogeneidade de cada maquina. Utilizamos este programa porque ele distribui a carga balanceada e de maneira assíncrona, para que o tempo de execução das diferentes máquinas sejam similar. Na Tab. 5 pode-se ver que quanto a comunicação, os experimentos mostraram pouca influencia quando colocamos trafego adicional, esta versão do algoritmo só necessita grande largura de banda da rede numa fase in grandes.



Tab. 5 blo cional

Trafego ad				
Sem				
80% da Lar de Banda				
90% da Largura de Banda	**2993**	**13351**	**84732**	mseg
Bloco sem trafego adicional	**20000**	**154000**	**1287000**	

A Fig. 3 mostra uma monitorização da rede durante a execução dos programas "bloco" e "blocobal", mostrando em "blocobal" como

Tab. 6[a]	Pvm uab1	Pvm uab2	Pvm uab3	Pvm uab4	Pvm uab5	Pvm uab6	Pvm uab7
Coeficiente	mestre	1	0.025	0.075	1	1	1
Bloco	mestre	133	133	133	133	133	135
Blocobal	mestre	195	4	14	195	196	196

Tab.6b	**Matriz (tamanho)**			
Algoritmo	**200**	**400**	**800**	
Blocobal	**2518**	**11902**	**78396**	mseg
bloco	**20**	**154**	**1287**	seg.

Tab. 6 Comparação "bloco" e "blocobal"

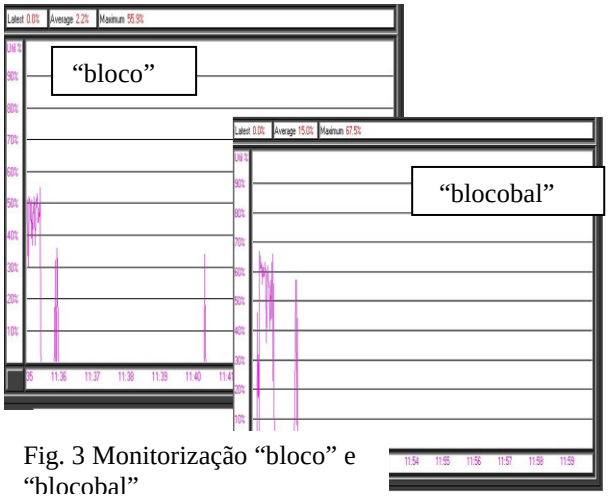


Fig. 3 Monitorização "bloco" e "blocobal"

quando se distribuem balanceado as máquinas terminam de maneira quase simultâneas.

O tempo de execução (Tab. 6b) será afetado pela exatidão dos coeficientes de heterogeneidade em relação a distribuição da carga (número de linhas) , como é mostrado na Tab. 6a. Vemos que esse programa apresenta muitas vantagens em relação ao programa "bloco";

- "**linha**" - Linha simétrica (não balanceada) síncrona estática e pacote pequeno: depois de enviar a matriz "B" envia uma linha da matriz "A" a ser calculada para cada um dos trabalhadores que compõe a máquina paralela e espera que todas enviem os resultados antes de enviar novas linhas. Obtendo o que chamamos empacotamento de dados linha a linha. Neste caso, o processo mestre espera que a máquina mais lenta acabe. Utilizamos este programa porque ele distribui a carga não balanceada, pacotes pequenos e se executa de modo sincrono

  Quando colocamos trafego adicional, numa fase inicial é requerida uma grande largura de banda, o que implica alargar o tempo de comunicação, referente a execução sem trafego adicional como é mostrado nos gráficos de monitorização da rede da Fig. 4. Na Tab. 7, vemos os tempos de execução em mseg desse algoritmo em relação ao trafego adicional da rede.

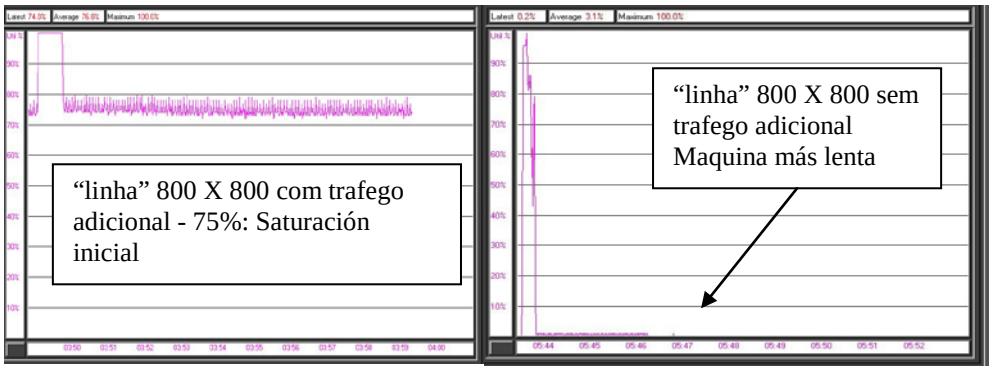
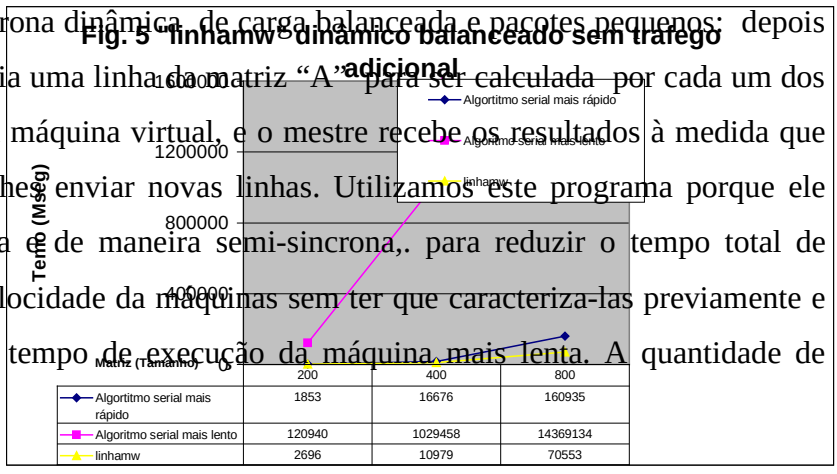


Fig. 4 Monitorização "linha"

Tab. 7 Tempo "linha"

Maquina: pvmba	**Tamanho Matriz**		
Algoritmo: Linha	**200**	**400**	**800**
sem trafego adicional	10	70	527
Linha 50%	10	56	541
Linha 75%	6	63	562
Linha 90%	28	227	1747

- "**linhamw**" Linha semi-sincrona dinâmica, de carga balanceada e pacotes pequenos: depois de enviar a matriz "B" , envia uma linha da matriz "A" para ser calculada por cada um dos trabalhadores que compõe a máquina virtual, e o mestre recebe os resultados à medida que eles calculem, tornando a lhes enviar novas linhas. Utilizamos este programa porque ele distribui a carga balanceada e de maneira semi-sincrona,. para reduzir o tempo total de execução adaptando-se à velocidade da máquinas sem ter que caracteriza-las previamente e que não seja definido pelo tempo de execução da máquina mais lenta. A quantidade de



trabalho executado por cada máquina, também serve para caracterizar e calcular seu grau de heterogeneidade, como alternativa aos algoritmos série. Como distribuímos em função da terminação individual de cada um, as máquinas terminam quase que simultaneamente e não necessitamos conhecer o grau de heterogeneidade. Concluímos que esse programa apresenta muitas vantagens em relação ao programa "linha" e "blocobal", pois nenhum trabalho prévio é alocado. Este programa também serve para calcular coeficiente de heterogeneidade em custo de comunicação.

4. Conclusões

O trabalho com *clusters* heterogêneo exige a caracterização do sistema – potência computacional (processamento e armazenamento de cada maquina) e custo de comunicação entre elas.

Por esta razão, neste trabalho experimentamos com dois sistemas heterogêneos. A caracterização foi realizada de forma sistemática na experimentação, dando-lhes os índices finais (grau de heterogeneidade) que pretendemos que seja informação útil e na medida do possível representativa.

Na escolha dos programas utilizados para a avaliação de desempenho dos *clusters* heterogêneos tomamos como base o modelo SPMD e implementado no esquema M/W. Esta seleção, alem de possuir um alto grau de coincidência com o esquema "passagem de mensagem", natural dos sistemas que estudamos, nos permitiu experimentar com a distribuição de carga, que chave nos sistemas heterogêneos.

Das alternativas de funcionamento sincrono ou assincrono na relação entre mestre e trabalhadores, junto com os padrões estático e dinâmico para a distribuição da carga de trabalho, e o conceito de carga balanceada e não balanceada, foi definido o espaço de experimentação, a qual foi incluído também a influência da largura de banda disponível para a comunicação – comportamento dos programas ante a presença de trafego adicional na rede. O trafego adicional nos permite modelar a caracterização de um *cluster* heterogêneo não dedicado.

A multiplicação de matrizes foi o programa utilizado por apresentar características facilmente escalavel (em cômputo e comunicação) assim como distintas possibilidades para o fracionamento do trabalho total.

Os resultados obtidos mostrou a influência da distribuição do trabalho sobre o desempenho. A distribuição estática é igual em uma máquina paralela com alto grau de heterogeneidade, produzindo um speed-up desfavorável. Os esquemas estáticos com carga balanceada de acordo com

a potência de cada tabulador e os esquemas dinâmicos, adaptáveis mediante dialogo semi-sincrono entre o mestre e trabalhador, permitiu obter *speed-up* próximo aos valores teórico predizíeis.
No caso do sistema fortemente heterogêneo (40 a 1), o *speed-up* conseguido apresentava uma forte correlação com o que se obtinha no sistema homogêneo resultante de excluir as máquinas lentas.
A ocupação da largura de banda da rede é baixa quando distribuímos dados e recebemos os resultados, por isso é que necessitamos gerar um grande quantidade de trafego adicional (acima de 90% de ocupação de largura de banda) para que o tempo de execução seja afetado.

A degradação lógica do speed-up, acontece porque reduzimos a largura de banda das comunicações e portanto o desempenho do *cluster*. Observamos que necessitamos dar um tratamento globalizado ao processo de distribuição da carga computacional, que incorpore uma distribuição que tenha em conta os retardas em função das larguras de banda disponível.

5. Trabalho futuro

A dificuldade referente ao manejo da heterogeneidade faz com que tenhamos que construir um modelo mais complexo para sua caracterização. Alem da potência computacional, devemos ter em conta a capacidade de comunicação entre os distintos computadores do sistema paralelo.
Essa complexa caracterização, cujos valores concretos podem variar dinamicamente, nos exigirá desenvolver uma metodologia para sua avaliação.
Esta caracterização devera ser levada em conta na distribuição de carga
Devemos ter em conta a latência de comunicação em sistemas dedicados, geograficamente separados, que requererá uma generalização dos esquemas SPMD e M/T que utilizamos. Algumas alternativas como esquema multimaster com trabalhadores de âmbito local, sob estrutura hierárquica e com esquemas mistos de balanceio de carga global e distribuição dinâmica local, serão avaliadas.

5. Bibliografia

[AKL89] Akl, Selim G. *The desing and Analysis of Parallel Algorithms*. Prentice-Hall, Inc. 1989.
[ALG94] Geist, Al. Beguelin, Adam. Dongarra, Jack. Wicheng, Jiang. Mancheck, Robert. Sunderam, Vaidy. *PVM 3 User's Guide and Reference Manual*. Prepared by the Oak Ridge National Laboratory. 1994

[CUL99] Culler, David E., Singh, Jaswinder Pal. *Parallel Computer Architecture: a hardware/software approach*. Morgan Kaufmann Publishers, Inc. 1999.

[DON-LUQ99] Jack Dongarra, Emilio Luque, Tomàs Margalef (Eds.). *Recent Advances in Parallel Virtual Machine and Message Passing Interface*. 6th European PVM/MPI User's Group Meeting. Barcelona, Spain, September 1999. Proceeding.

[KAI98] Hwang, Kai. Xu, Zhiwei. *Scalable Parallel Computing: Technology, Architecture, Programming*. WCB/McGraw-Hill. 1998

[OBS98] Network Instruments. *Observer User Manual.* Network Instruments. 1998

[REW97] El-Rewini, Hesham. Lewis, Ted G. *Distributed and Parallel Computing*. Hardbound. 1997.

[SNY84] Snyder, L. *Parallel Programing and the Poker Enviroment*. IEEE Computer Magazine. Jul-1984, pp. 27 a 36.

[SOU2000] Souza, Josemar R. de. *Influência da comunicação no desempenho de um sistema de computação paralela, baseado em rede de estações de trabalho*. Tese de Mestrado, Arquitetura de Computadores e Processamento Paralelo. Universidade Autônoma de Barcelona, - UAB. 2000

[STE99] Sterling, Thomas L. Salmon, John. Becker, Donald J. e Savaresse, Daniel F. *How to build a Beowulf: a guide to the implementation and aplication of PC clusters*. Massachusetts Institute of Technology. 1999