

Análise de Desempenho de um Algoritmo de Roteamento Adaptativo em Malhas 3D Sob Diversos Padrões de Tráfego.

Celso A. Saibel Santos, Carlos Augusto P. S. Martins, Edward D.M.Ordoñez, Martha X.T.Delgado, Sergio T.Kofuji¹

*Laboratório de Sistemas Integráveis
Departamento de Engenharia Eletrônica
Escola Politécnica da Universidade de São Paulo*

RESUMO

Este artigo analisa o desempenho dos algoritmos de roteamento determinístico e adaptativo em sistemas multicomputadores baseados em malhas tridimensionais, sob vários padrões de tráfego. Para realizar este estudo foi desenvolvido um simulador denominado SIM3D. Os resultados obtidos nas simulações são comparados considerando-se a latência e o *throughput*. Entre os fatores utilizados na avaliação do desempenho temos: o tipo de algoritmo de roteamento, o padrão de tráfego e o tamanho das mensagens.

ABSTRACT

The paper analyses the performance of deterministic and adaptative routing algorithms in multicomputing systems based on three dimensional meshes under various traffic patterns. The analysis was developed on a simulator named SIM3D. The results took into account latency and throughput. Some factors used on performance evaluation were: routing algorithm form, traffic pattern and message size.

¹ Pesquisadores do Laboratório de Sistemas Integráveis (LSI/EPUSP)
E-mail: (saibel, capsm, edmoreno, mxted, kofuji) @lsi.usp.br

1- Introdução

Sistemas multiprocessadores surgiram com o objetivo de alcançar um maior poder de processamento. Numa arquitetura baseada no paradigma da passagem de mensagens, a qual elimina o conceito de memória compartilhada, as tarefas concorrentes de um algoritmo paralelo são executadas assincronamente em diferentes nós de processamento, sendo a comunicação realizada somente por troca de mensagens. Os processadores estão ligados diretamente apenas à sua memória local que é fisicamente separada e logicamente privada das memórias dos outros processadores (ATHAS, 1988). A comunicação entre processos executados em cada nó ocorre pelo roteamento de mensagens através de uma rede de interconexão. Os nós desta rede multicomputadora contém um processador, um controlador de comunicação capaz de rotear as mensagens sem afetar o processamento, e um pequeno número de conexões com outros nós.

Se um sistema é usado para suportar este tipo de computação, o tempo requerido para a movimentação dos dados entre cada um dos pontos é crítico para alcançar uma bom desempenho, assim como irá determinar o nível de granularidade do paralelismo possível na execução de um programa. Os parâmetros geralmente usados como medida na avaliação do desempenho de sistemas são a latência de comunicação e o *throughput* (vazão) (GRUNWALD, 1990).

2- Topologia

Na maioria dos casos, as redes utilizam uma topologia fixa denominada múltiplos saltos (*multiple-hops*) - hipercubos e malhas, por exemplo - obrigando que se utilize uma sequência de nós intermediários numa comunicação. Duas exigências conflitantes necessariamente aparecem: acomodar um grande número de nós na estrutura, e exibir uma baixa latência na comunicação. Além disso, se o número de ligações cresce, torna-se cada vez mais difícil construir uma dada topologia numa área limitada.

Uma malha (*mesh*) n -dimensional é uma estrutura com $k_0 \times k_1 \times \dots \times k_{n-1}$ nós (ou processadores), k_i ao longo de cada dimensão, onde $k_i > 1$. Cada nó p é identificado por suas n coordenadas (Figura 1). Os nós estão ligados a um número de vizinhos que varia entre n e $2n$, dependendo das posições na malha.

Define-se caminho como uma sequência de canais a partir de um nó fonte, origem da mensagem, até o nó destino, onde a mensagem é consumida. O número mínimo de canais a serem percorridos entre dois nós quaisquer é denominado de distância.

Um ponto importante a ser abordado no estudo de uma topologia é a definição do número de canais virtuais a serem utilizados no sistema. Geralmente, o número de canais virtuais está relacionado com o esquema de prevenção de travamento (*deadlock*) adotado (se esta for a metodologia usada). Para roteamentos determinísticos, devem ser implementados um ou dois canais virtuais por canal físico para prevenção de travamentos em redes de qualquer dimensão (DALLY, 1987a). Por outro lado, roteamentos adaptativos normalmente requerem um grande número de canais virtuais para assegurar a ausência de deadlock em redes de dimensão elevada, geralmente proporcionais a 2^n (LINDER, 1991) (FELPERIN, 1991).

Contudo, o número máximo de canais virtuais pode ser limitado através de técnicas recentemente propostas (DALLY, 1993) (BERMAN, 1992), e ainda, tornado independente da dimensão do sistema, através da redução da adaptatividade do algoritmo de roteamento (CHHEM, 1992).

Estruturas escaláveis são essenciais para o desenvolvimento de arquiteturas maciçamente paralelas com centenas ou milhares de nós e sob esse ponto de vista, as redes em malhas tridimensionais abordadas neste trabalho apresentam-se como uma solução bastante adequada se a largura de biseção² é tomada como medida de custo (DALLY, 1990).

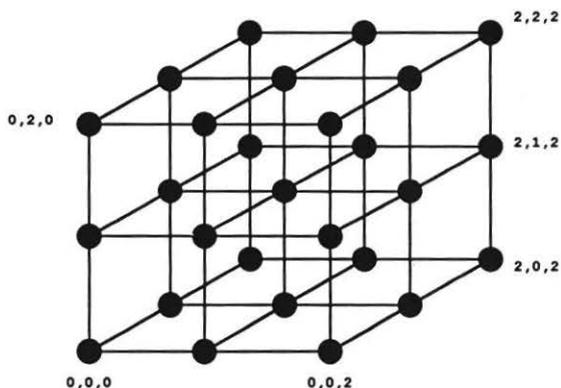


Figura 1 - Rede direta em estrutura Malha tridimensional 3x3x3.

3- A política wormhole

Quando a política de armazenar-e-repassar (*store-and-forward*) é utilizada, as mensagens são armazenadas em cada um dos nós intermediários do caminho, para só então serem enviadas para um nó adjacente de acordo com a política de roteamento adotada. Este processo é repetido até que a mensagem alcance seu destino. A principal desvantagem dessa metodologia é que a latência é dependente do comprimento do caminho.

Com o objetivo de evitar um atraso desnecessário devido ao armazenamento completo do pacote em cada nó intermediário mesmo quando o canal de saída está livre, foi proposta a política *wormhole routing* (DALLY, 1987b). Neste caso, uma mensagem é dividida em flits (*flow control digits*) para a transmissão, sendo o tamanho do flit geralmente dependente da largura do canal utilizado. Um flit é a menor unidade de informação que pode ser transferida pela rede; em contrapartida, um pacote deve conter muitos flits. Os flits *header* governam a rota dos pacotes, sendo retransmitidos assim que chegam a um nó intermediário, sem esperar que a mensagem inteira seja recebida, estabelecendo um fluxo contínuo (*pipeline*) de flits dentro da rede. Cada nó precisa apenas de um controlador de rede e alguns poucos armazenadores de flit. Se o *header* encontra um canal requisitado em uso, ele é bloqueado até que este se torne disponível. Ao invés de armazenar os flits restantes, removendo-os dos canais como no *virtual cut-through* (KERMANI, 1979), o controlador de rede os bloqueia, fazendo com que estes permaneçam na rota já estabelecida pelo flit *header*. Isto é necessário pois apenas o flit inicial contém informações de roteamento. Os canais são liberados assim que o último flit (*flit tail*) é recebido. Normalmente, os canais físicos com este método são multiplexados sobre a forma de canais virtuais.

²Largura de biseção corresponde ao número mínimo de canais que devem ser removidos, ou cortados, para que a rede seja particionada em duas sub-redes, cada uma com a metade dos nós.

Canais virtuais são entidades lógicas associadas às ligações físicas, tendo a função de distinguir as várias seqüências de dados que atravessam o mesmo canal físico (GAUGHAN, 1993). Geralmente são multiplexados sobre um canal físico de acordo com a demanda, utilizando parte da banda do canal quando necessário. Estas entidades podem ser utilizadas não só com a função de evitar travamentos pela imposição de restrições de roteamento que levem a quebra de dependências cíclicas na rede (DALLY, 1987a) (GRUNWALD, 1990) (NGAI, 1989) (REED, 1987), mas também para aumentar o *throughput* através da redução do tempo ocioso (*idle*) das ligações (DALLY, 1992) (GAUGHAN, 1993).

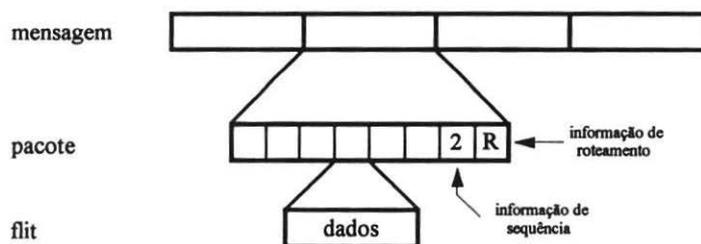


Figura 2 - Unidades de informação: uma mensagem é a unidade lógica de informação. Ela é subdividida em pacotes, a menor unidade contendo informação de roteamento. Os pacotes são novamente divididos em flits, a menor unidade física na qual o controle de fluxo é realizado.

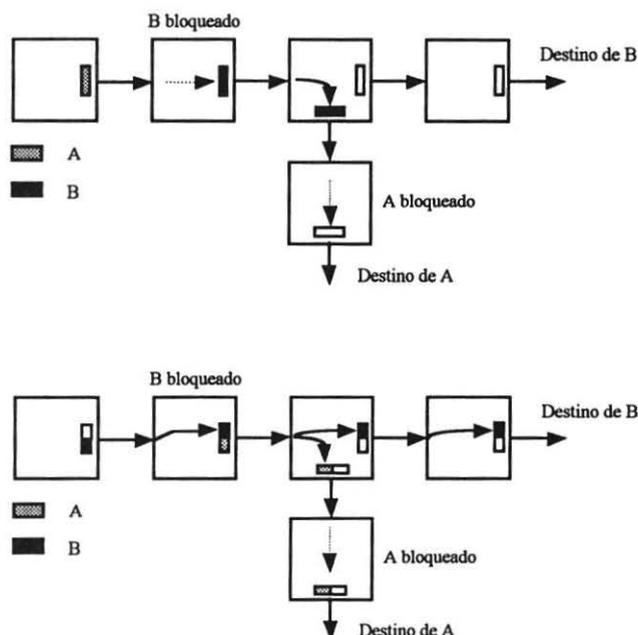


Figura 3 - Aumento do *throughput* com canais virtuais: à esquerda sem a utilização dos canais virtuais, e à direita com sua utilização.

O aumento do compartilhamento de recursos da rede resulta num razoável crescimento do *throughput*. Na situação da Figura 3, a mensagem está bloqueada no nó *n*. A mensagem B deve ser bloqueada porque a mensagem A tem a posse da ligação física. Os canais virtuais permitem que a mensagem B ultrapasse A e siga em direção a seu destino.

4- Algoritmos de roteamento

O objetivo essencial de um algoritmo de roteamento é determinar o caminho que deve ser seguido por cada mensagem na comunicação entre dois nós não diretamente conectados, garantindo as propriedades de continuidade de comunicação, tais como a ausência ou tratamento de travamentos (*deadlocks*), de bloqueios vivos (*livelocks*), e obrigando que todas as mensagens enviadas alcancem seu destino.

Uma rota é dita travada (*deadlocked*) se existe uma espera por um recurso que nunca estará disponível. No caso de bloqueio vivo (*livelocks*), a restrição passa a ser temporal: deve ser garantido que uma mensagem irá alcançar seu destino num tempo finito. As rotas indicadas devem garantir que o sistema inteiro progrida significativamente, ou seja, que nenhuma mensagem deixe de alcançar seu destino por ser repetidamente desviada.

Os algoritmos de roteamento podem ser divididos em duas classes principais, *óbvios* ou *adaptativos*. Nos algoritmos adaptativos os caminhos de roteamento são estabelecidos dinamicamente, enquanto que para os óbvios, eles são estaticamente definidos. O roteamento *determinístico* é um caso especial dos algoritmos óbvios no qual uma única rota existe entre dois nós, ou melhor, todo o caminho seguido por uma mensagem é pré-determinado por seu par fonte-destino (TALIA, 1993).

Boa parte dos multicomputadores construídos, tais como o iPSC, o NCUBE, o Ametek e o SYMULT 2010, implementam o roteamento determinístico (DALLY, 1987) (TALIA, 1993). Embora estes roteadores exijam uma lógica relativamente simples para determinar a rota de uma mensagem e garantir sua transmissão, o procedimento do algoritmo é independente da situação instantânea da rede, tornando-o intolerante a falhas e provocando congestionamentos e atrasos desnecessários em algumas situações.

Roteadores adaptativos podem melhorar o desempenho da rede através da seleção de caminhos que evitem áreas congestionadas ou regiões com falhas, baseados apenas em informações de tráfego local (condições instantâneas da rede), consequentemente diminuindo a latência de comunicação e promovendo uma distribuição bem mais uniforme de carga nos canais de ligação.

Os caminhos que podem ser selecionados de acordo com o algoritmo de roteamento utilizado são uma distinção básica dos roteadores adaptativos. Os roteadores que utilizam apenas caminhos correspondentes a distância entre o nó fonte e destino são denominados mínimos (*minimal*) e os que permitem a escolha de caminhos de comprimento maior que essa distância em busca de maior flexibilidade, permitindo movimentos na direção contrária à sua minimização (*misrouting*), não-mínimos (*nonminimal*). Vale ressaltar que o afastamento da rota mínima é temporária e que o pacote eventualmente alcança seu destino. Além disso, os pacotes são retirados da rota mínima somente se nenhum dos caminhos mínimos possíveis estiver disponível. Esses esquemas apesar da maior flexibilidade permitida, tornam complexa a tarefa de prevenir o *livelock*.

5- O algoritmo de roteamento adaptativo utilizado

O algoritmo adaptativo utilizado (SANTOS, 1993) nas simulações é bastante semelhante ao obtido por Glass e Ni (GLASS, 1992) utilizando o modelo das voltas (*turn model*), denominado roteamento negativo-primeiro (*negative-first routing*). Em nosso caso, apenas os caminhos mínimos permitidos pelas restrições do algoritmo de roteamento são utilizados. Além disso, a seleção do próximo canal a ser utilizado é feita a partir de informações de carregamento da rede (mais precisamente do carregamento dos canais). No algoritmo negativo-primeiro, as colisões são resolvidas através da seguinte sequência de ações: primeiro tenta-se enviar o pacote através de um dos canais pertencentes ao conjunto de caminhos mínimos permitidos pelas restrições entre os nós, escolhido de maneira aleatória. Se todos estiverem ocupados, o pacote será desviado (*misrouting*), percorrendo um caminho não-mínimo. Se isso não for possível, o pacote será finalmente bloqueado. Uma alternativa que será estudada é a utilização de canais virtuais compartilhando os canais físicos e, baseando-se no número de canais virtuais utilizados pelos canais, estabelece-se como canal de saída aquele com menor carga (menor número de canais virtuais utilizado).

6- Padrões de tráfego

Devido as dificuldades de obtenção dos caminhos seguidos (*traces*) correspondentes a operações reais de redes maciçamente paralelas, alguns padrões de tráfego são normalmente utilizados para as análises, cada um dos quais com um respectivo ponto de observação. Um padrão de tráfego é determinado pelo tipo de aplicação e pela forma com que os processos são mapeados na estrutura.

Num padrão uniforme de tráfego (ou aleatório), cada nó envia mensagens para um destino escolhido aleatoriamente. Assim, todos os nós têm a mesma probabilidade de serem destino para a mensagem e o tráfego gerado produz, teoricamente, um carregamento igual para os canais.

Padrões não-uniformes de tráfego acentuam ainda mais a diferença entre o desempenho de um algoritmo adaptativo e de um determinístico. Como as técnicas adaptativas tendem a melhorar a distribuição de carga entre os canais, sua atuação em situações caracterizadas por uma utilização não uniforme dos canais é muito mais pronunciada.

Padrões que produzem focos (*hotspots*) são especialmente interessantes pois reproduzem uma situação bastante usual em sistemas reais. Nestas situações, um nó em particular recebe muito mais tráfego que qualquer outro da rede. Em multicomputadores, este tráfego pode representar uma situação na qual seções críticas estão localizadas em um único nó (BOPPANA, 1993). Esta situação de tráfego pode ser implementada, por exemplo, através da operação: Destino=(k,k,n_z), onde n_z corresponde a coordenada z do nó gerador da mensagem. Isso obriga que os nós enviem mensagens sempre com destino a um dos vértices do plano 2D formado pelos nós (0,0,n_z), (0,k,n_z), (k,0,n_z), (k,k,n_z).

Outro padrão de interesse é o produzido por um tráfego local. Neste caso, as mensagens geradas por um nó estão sempre dentro de uma região bem determinada da rede. Por exemplo, numa estrutura com $N = 8^3$ nós, o endereço do nó destino, (x,y,z), produzido por outro nó qualquer, (i,j,k), seria definido como:

$$\{(x,y,z) \mid i-2 \leq x \leq i+2; j-2 \leq y \leq j+2; k-2 \leq z \leq k+2\}$$

Portanto, está se restringindo o espaço de transação de mensagens a uma vizinhança, gerando conseqüentemente uma localidade de dados.

7- Limites de desempenho

• Limites de comunicação

Sejam L , B e D o tamanho do pacote em flits, a banda do canal e a distância entre dois nós numa rede. Define-se latência da rede como o intervalo de tempo decorrido desde que o flit inicial (*header*) entrou na rede no nó fonte até que o flit final (*tail*) seja liberado no canal destino. Para uma política tipo *wormhole*, pode-se estimar a latência por (NI, 1993):

Latência = $(L_f/B)D + L/B$, onde L_f é o tamanho de cada flit. Deste modo, se $L_f \ll L$, D não afeta significativamente a latência.

Vale lembrar que a análise anterior é feita a partir da suposição de que não ocorrem contenções, ou seja, não inclui o tempo de bloqueio das mensagens caso ocorram colisões.

• Limites de latência e *throughput* (vazão)

Como era de se esperar, algumas restrições físicas estabelecem limites para o desempenho de uma arquitetura paralela. Os limites relacionados à latência média das mensagens e à taxa máxima de mensagens liberadas (consumidas no destino) de topologias regulares como as malhas 3D, podem ser determinados a partir das características construtivas da estrutura.

Algumas considerações iniciais devem ser feitas para simplificar a análise; o problema tratável matematicamente. Desta forma, assume-se um padrão de tráfego uniforme, uma operação independente entre os nós (injeções independentes de mensagens) e um controle de fluxo tipo *wormhole*.

O limite inferior da latência média, ou seja, desprezando possíveis conflitos, corresponde a:

Latência = distância (fonte, destino) + tamanho da mensagem

Com um tráfego uniforme na rede, a distância média percorrida, $D_{média}$, de uma malha n -dimensional, com k nós por dimensão, é dada por (NGAI, 1989):

$$D_{média} = \frac{n}{3} \left(k - \frac{1}{k} \right) \cong \frac{n}{3} k \quad , \text{ para valores grandes de } k$$

Se for considerado, por exemplo, um tráfego uniforme numa rede malha $6 \times 6 \times 6$, com mensagens de tamanho médio 48 flits, a latência média encontrada deve ser igual ou superior a 54 ciclos. Vale lembrar que essa análise é válida para k par.

Para a taxa média de injeção por nó (*throughput*) da rede, o limite é imposto pela largura de banda da biseção. Para uma malha n -dimensional com $N = k^n$, e k par, com canais unidirecionais em ambas as direções, a largura de biseção é dada por $k^{n-1} = \frac{2N}{k}$. A biseção,

como definida anteriormente, pode ser vista como um plano com $n-1$ dimensões, perpendicular aos eixos nas n dimensões originais, dividindo a malha em duas partes iguais, cada uma com a metade dos nós. Para um tráfego uniforme, devido a escolha aleatória dos destinos, pode-se supor que qualquer mensagem com nó fonte em um dos lados do corte terá uma possibilidade de 50% de ter como destino um nó da outra metade. Assim, a taxa média de injeção por nó vale (NGAI, 1989):

$$q \cdot \frac{N}{2} \cdot \frac{1}{2} \leq \frac{2N}{k} \text{ ou seja } \quad q \leq \frac{8}{k}$$

Este valor de q corresponde ao limite superior da taxa de injeção média por nó em flits/ciclo, o qual é independente da dimensão do sistema. Para uma malha 3D 6x6x6, o limite de injeção por nó imposto pela largura de bisecção é de $2/3$, ou seja, em cada 3 ciclos em média, cada nó envia dois novos flits à rede, considerando um padrão de tráfego uniforme.

8- Descrição do simulador e seus principais parâmetros

Para estudar o comportamento do sistema sob diferentes formas de roteamento e padrões de tráfego, foi utilizado um simulador denominado SIM3D. O programa é capaz de simular estruturas em malha 3D com um número máximo de 16x16x16 nós (16 em cada dimensão). Cada nó $NÓ(x,y,z)$ é constituído por um elemento de comunicação, COM, e por um elemento de processamento, PROC. O elemento de comunicação é responsável pelo roteamento das mensagens que chegam (ou saem) do nó através dos 7 canais de entrada (6 correspondentes aos pares de canais unidirecionais conectando os nós adjacentes em cada uma das dimensões e 1 à ligação com seu processador local) e 12 de saída (6 correspondentes aos canais adjacentes e 6 para seu processador local). Assume-se que cada mensagem é composta por diversos flits, sendo cada flit composto por 5 bytes (1 byte de header e 4 bytes de dados). Para sinalizar o início e o término da transmissão de uma mensagem, são utilizados dois flits, um de cabeçalho (**header**) e o outro de fim de comunicação (**tail**). O flit cabeçalho contém informações sobre o destino da mensagem, enquanto o de finalização carrega dados estatísticos que serão utilizados no levantamento das características do sistema nas diferentes situações propostas. O simulador apresenta uma boa flexibilidade, permitindo variação do número de nós em cada dimensão, do tamanho dos flits (mensagens), do número de canais virtuais e principalmente, do algoritmo de roteamento para as mensagens.

Para comparar o algoritmo adaptativo com determinístico das dimensões ordenadas³, foram consideradas as seguintes características para o sistema: todos os canais têm a mesma largura de banda. Os **buffers** dos canais de entrada e dos de saída têm largura de um flit (5 bytes). Os processadores geram mensagens de tamanho fixo, ou com valor médio e desvio, conforme uma distribuição Erlang, a qual representa com alguma precisão situações práticas (NGAI, 1989) (PRITSKER, 1986). O intervalo entre mensagens varia segundo uma distribuição exponencial negativa, de acordo com os níveis de tráfego desejados. Para cada simulação foram observadas as seguintes características: latência média de comunicação e intervalo de tempo entre a liberação de mensagens. Foram considerados os padrões de tráfego descritos anteriormente.

Estuda-se também a variação do número de canais virtuais para as situações de tráfego propostas, verificando o comportamento dos algoritmos para cada caso.

³ No algoritmo de roteamento ordenado por dimensões, essas são percorridas numa única sequência previamente definida.

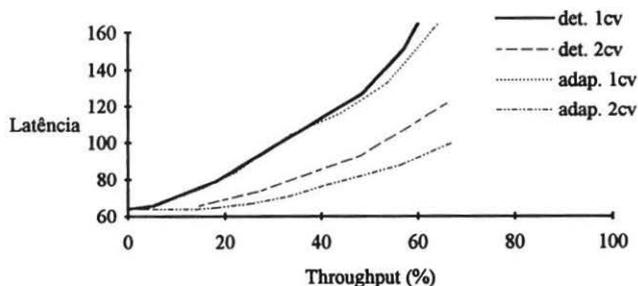


Gráfico 1 - padrão de tráfego local: Latência x Throughput para os algoritmos determinístico e adaptativo com 1 ou 2 canais virtuais - mensagens de 48 flits.

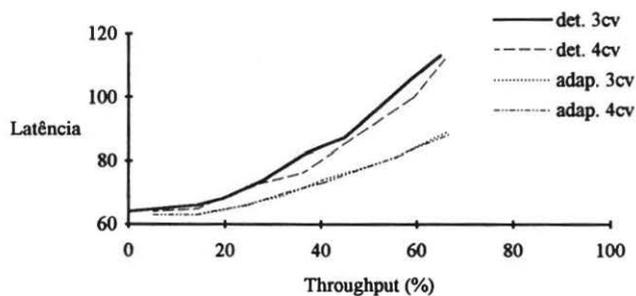


Gráfico 2 - padrão de tráfego local: Latência x Throughput para os algoritmos determinístico e adaptativo com 3 ou 4 canais virtuais - mensagens de 48 flits.

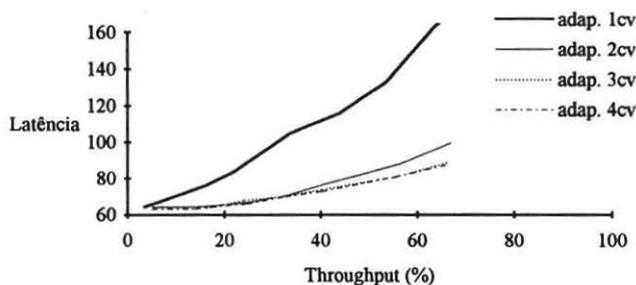


Gráfico 3 - Efeito dos canais virtuais num padrão de tráfego local com roteamento adaptativo - mensagens de 48 flits.

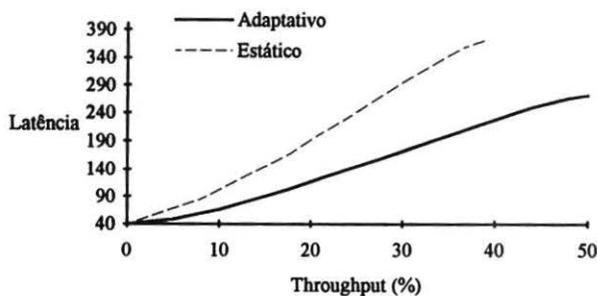


Gráfico 4 - padrão de tráfego *hot spot* : Latência x Throughput para os algoritmos determinístico e adaptativo - mensagens de 16 flits.

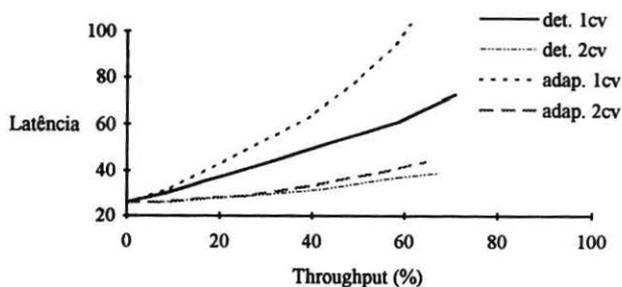


Gráfico 5 - padrão de tráfego uniforme: Latência x Throughput para os algoritmos determinístico e adaptativo com 1 ou 2 canais virtuais - mensagens de 16 flits.

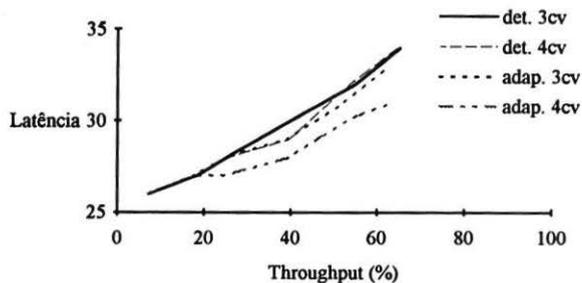


Gráfico 6 - padrão de tráfego uniforme: Latência x Throughput para os algoritmos determinístico e adaptativo com 3 ou 4 canais virtuais - mensagens de 16 flits.

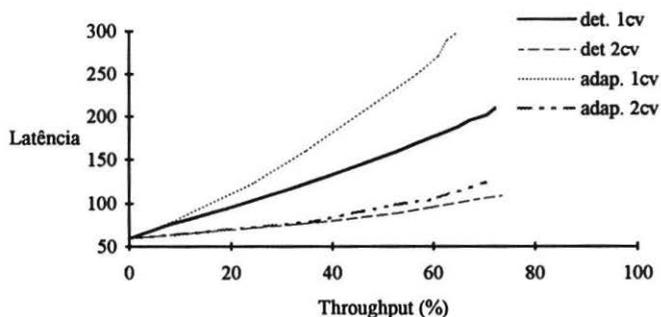


Gráfico 7 - padrão de tráfego uniforme: Latência x Throughput para os algoritmos determinístico e adaptativo com 1 ou 2 canais virtuais - mensagens de 48 flits.

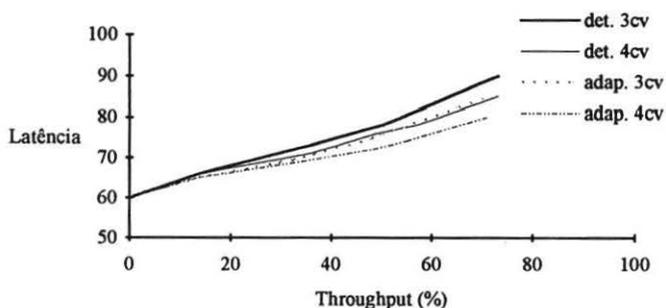


Gráfico 8 - padrão de tráfego uniforme: Latência x Throughput para os algoritmos determinístico e adaptativo com 3 ou 4 canais virtuais - mensagens de 48 flits.

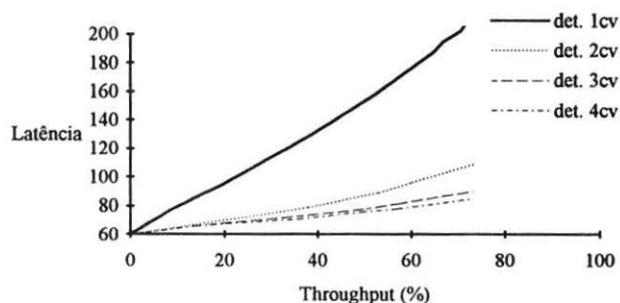


Gráfico 9 - Efeito dos canais virtuais num padrão de tráfego uniforme com roteamento determinístico - mensagens de 48 flits

9- Análise dos resultados

Como pode ser observado nos gráficos 1 e 2, o algoritmo de roteamento adaptativo apresenta um desempenho superior ao do roteamento determinístico num padrão de tráfego local, o que é ainda mais acentuado quando são utilizados dois ou três canais virtuais. Considerando-se que boa parte das aplicações buscam uma localidade de dados na distribuição das tarefas, pode-se dizer que este padrão de tráfego representa uma boa parte das situações práticas, e que nesse caso a técnica de roteamento adaptativa apresenta melhores resultados quanto à latência e ao throughput.

Devemos observar também que a colocação de canais virtuais nem sempre é efetiva. A variação de desempenho com a inclusão do primeiro canal virtual é bastante acentuada, enquanto que para o terceiro e quarto canais virtuais adicionados o ganho em termos de desempenho não é satisfatório. Mais ainda, a diferença entre as curvas para três e quatro canais virtuais, conforme mostra o gráfico 3, demonstram que a partir da inclusão do quarto canal virtual não há ganho de desempenho, pelo menos para este padrão de tráfego.

A superioridade do roteamento adaptativo é ainda mais acentuada para um padrão de tráfego "hot spot", já descrito anteriormente. A disponibilidade de rotas alternativas para as mensagens de todos os nós direcionadas a pontos específicos, ressaltam ainda mais as vantagens do algoritmo adaptativo em relação ao determinístico. A latência das mensagens para um mesmo throughput sob o roteamento determinístico em dimensões ordenadas é quase o dobro da observada no algoritmo adaptativo, comprovando a sua ineficiência para os casos em que o tráfego não está suficientemente balanceado pelos canais.

Para a padrão de tráfego uniforme, como era de se esperar (BOPPANA, 1993), o algoritmo adaptativo apresenta um desempenho inferior ao determinístico quando não são usados canais virtuais, como mostrado nos gráficos 4 e 7. Porém, deve ser lembrado que este padrão representa o pior caso de tráfego, o qual dificilmente é encontrado em situações reais. A inclusão de um canal virtual entretanto, aproxima bastante o desempenho dos dois algoritmos e quando são utilizados três canais virtuais, o desempenho do roteamento adaptativo é superior, mesmo quando o tamanho da mensagem varia de 16 para 48 flits. Novamente, observa-se que a inclusão do quarto canal virtual não apresenta nenhum ganho de desempenho para ambos os algoritmos, como mostra o gráfico 9.

O desempenho do algoritmo adaptativo em termos de latência e throughput é superior para os padrões de tráfego local com qualquer número de canais virtuais e para o padrão "hot spot". Para o padrão de tráfego uniforme, o desempenho do algoritmo determinístico é bem superior, quando não são utilizados canais virtuais. Porém, para dois canais virtuais o desempenho é bastante semelhante, e para um número maior que dois canais virtuais o desempenho do algoritmo adaptativo já se torna superior.

10- Conclusão e trabalhos futuros

A utilização da técnica de roteamento adaptativa mostrou um possível ganho de desempenho para a rede de interconexão, o que pode ser traduzido como um melhor desempenho da máquina.

A inclusão de canais virtuais provou ser uma técnica eficiente para aumentar o throughput da rede com pequeno aumento no número de componentes, apenas alguns buffers e lógica de controle.

Um ponto interessante para futuros estudos é a influência da política de saída no desempenho do algoritmo. Esta política determina entre os vários canais de saída permitidos à mensagem, qual será utilizado. Neste trabalho, foi utilizada a política de menor carga, entretanto outras formas podem ser consideradas, como por exemplo escolha aleatória.

Outro ponto interessante é estudar o efeito da inclusão de canais virtuais na ligação entre o dispositivo de comunicação e o processador, que pode afetar a taxa máxima de mensagens injetadas na rede em cada ciclo.

11- Referências Bibliográficas

- ATHAS, W.C.; SEITZ, C.L. Muticomputers: Message-Passing Concurrent Computers. **IEEE Computer**. v.21, n.8, p.9-24, Aug. 1988.
- BERMAN, P.; et al. Adaptive deadlock- and livelock-free routing with all minimal paths in torus networks. **SYMPOSIUM ON PARALLEL ALGORITHMS AND ARCHITECTURES, 4., 1992**. Proceedings. ACM, p.3-12, 1992.
- BOPPANA, R.V.; CHALASANI, S. A comparison of wormhole routing algorithms based on adaptivity. **Computer Architecture News**. v.21, n.2, p.351-60, May 1993.
- CHIEN, K.; KIM, J.H. Planar-adaptive routing: Low-cost adaptive networks for multiprocessors. **Computer Architecture News**. v.20, n.2, p.268-77, May 1992.
- DALLY, W.J.; SEITZ, C.L. Deadlock-free message routing in multiprocessor interconnection networks. **IEEE Transactions on Computers**. v.36, n.5, p.547-53, May 1987.
- DALLY, W.J. **A VLSI architecture for concurrent data structures**. Norwell, Kluwer Academic Publishers, 1987.
- DALLY, W.J. Performance Analysis of k-ary n-cube interconnection networks. **IEEE Transactions on Computers** v.39, n.6, p.775-85, June 1990.
- DALLY, W.J. Virtual-channel flow control. **IEEE Transactions on Parallel and Distributed Systems**. v.3, n.2, p.194-205, Mar. 1992.
- DALLY, W.J.; AOKI, H. Deadlock-free adaptive routing in multicomputer networks using virtual channels. **IEEE Transactions on Parallel and Distributed Systems**. v.4, n.4, p.466-75 Apr. 1993.
- FELPERIN, S.A.; et al. Routing techniques for massively parallel communication. **Proceedings of the IEEE**. v.79, n.4, p.488-503, Apr. 1991.
- GAUGHAN, P.T.; YALAMANCHILI, S. Adaptive routing protocols for hypercube interconnection networks. **IEEE Computer**. v.26, n.5, p.12-23, May 1993.
- GLASS, C.J.; NI, L.M. The turn model for adaptative routing. **Computer Architecture News**. v.20, n.2, p.278-87, May 1992.
- GRUNWALD, D.C. Circuit switched multicomputers and heuristic load placement. Urbana, 1990. 216p. **(PhD)Thesis** - Department of Computer Science, University of Illinois at Urbana-Champaign. /June 1990, revised/

- KERMANI, P.; KLEINROCK, L. Virtual cut through: A new computer communication switching technique. **Computer Networks**. v.3, p.267-86, 1979.
- LINDER, D.H.; HARDEN, J.C. An adaptive and fault-tolerant wormhole routing strategy for k-ary n-cubes. **IEEE Transactions on Computers**. v.40, n.1, p.2-12, Jan. 1991.
- NGAI, J. A framework for adaptive routing in multicomputers networks. Pasadena, 1989. 169p. (Phd)Thesis - Computer Science Department, California Institute of Technology. (Caltech-CS-TR-89-09).
- NI, L.M.; MCKINLEY, P.K. A survey of wormhole routing techniques in direct networks. **IEEE Computer**. v.26, n.2, p.62-76, Feb. 1993.
- PRITSKER, A.A.B. **Introduction to simulation and SLAM II**. 3.ed., Systems Publishing, 1986.
- REED, D.A.; FUJIMOTO, R.M. **Multicomputer networks: message-based parallel processing**. Cambridge, The MIT Press, 1987. (Scientific Computation Series).
- SANTOS, C.A.S. et al. Um algoritmo de roteamento adaptativo em estruturas mesh n-dimensionais. In: **SIMPÓSIO BRASILEIRO DE ARQUITETURAS DE COMPUTADORES / PROCESSAMENTO DE ALTO DESEMPENHO, 5.**, Florianópolis, 1993. **Anais**. Florianópolis, UFSC, 1993. v.1, p.49-58.
- TALIA, D. Message-routing systems for transputer-based multicomputers. **IEEE Micro**, p.62-72, June 1993.