

ANÁLISE DO DESEMPENHO DE REDES DE INTERCONEXÃO PARA MÁQUINAS PARALELAS

Gerson Bronstein *
NCE/UFRJ

Adriano Joaquim de Oliveira Cruz **
NCE/UFRJ IM/UFRJ

Otto Carlos Muniz Bandeira Duarte ***
COPPE/UFRJ EE/Depto. Eletrônica

RESUMO

O desempenho de sistemas de processamento paralelo com dezenas, ou até centenas de processadores depende do mecanismo de comunicação entre processadores utilizado. As redes de interconexão surgem como uma solução atraente, pois apresentam custo e desempenho satisfatórios quando comparadas a outras soluções. Este trabalho apresenta os resultados obtidos, através de simulação, para o desempenho de diversas estruturas de elementos comutadores (chaves) utilizados na construção de redes de interconexão. Estes resultados são discutidos e analisados e servirão para a definição da estrutura de chave a ser utilizada no MULTIPLUS, uma máquina paralela de alto desempenho que está sendo desenvolvida no Núcleo de Computação Eletrônica (NCE) da UFRJ.

ABSTRACT

The performance of parallel systems with tens, or even hundreds of processors highly depends on the communication between these processors. Interconnection networks are an attractive solution since they present low cost and good performance when compared to other solutions. This paper presents simulation results of several switching element structures used in interconnection networks. Based on an analysis and discussion of these results, it is chosen an internal structure for the switching element to be used in the MULTIPLUS interconnection network. MULTIPLUS is a high performance parallel machine under development at the Núcleo de Computação Eletrônica (NCE), the computer center of the Federal University of Rio de Janeiro.

AFILIAÇÃO

Núcleo de Computação Eletrônica - UFRJ
Caixa Postal 2324
CEP 20001 - Rio de Janeiro - RJ
Tel: (021) 290-3212 r. 292/328

* Engenheiro Eletrônico pela UFRJ (1986). Mestrando em Engenharia Elétrica - COPPE/UFRJ. Analista de Sistemas do NCE/UFRJ. Áreas de Interesse: Arquitetura de Computadores, Processamento Paralelo e Redes de Interconexão.

** Ph.D. em Computação pela Universidade de Southampton (1988). Analista de Sistemas do NCE/UFRJ. Professor Assistente do Instituto de Matemática - UFRJ. Áreas de Interesse: Arquitetura de Computadores, Processamento Paralelo e Redes de Interconexão.

*** D. Ing. (ENST/INRIA - 1985). Professor Adjunto da COPPE/Elétrica e EE/Depto. de Eletrônica. Áreas de Interesse: Protocolos de Comunicação, Transmissão de Dados via Satélite e Avaliação de Desempenho.

1 - INTRODUÇÃO

A demanda por computadores de alto desempenho tem aumentado a cada dia e hoje estas máquinas são utilizadas nas mais diversas áreas como análise estrutural, prospecção de petróleo, meteorologia, etc. Pode-se melhorar o desempenho destas máquinas aumentando-se a velocidade dos dispositivos e componentes ou implementando-se arquiteturas mais avançadas e eficientes. Porém, as limitações impostas pelas leis da física indicam que os esforços para o aumento do desempenho dos computadores devem se concentrar no desenvolvimento de novas arquiteturas e na exploração do paralelismo das aplicações [Menal]. Para que este paralelismo possa ser bem explorado, é necessário que os processadores possam se comunicar de forma eficiente. Sendo a rede de interconexão a responsável pela comunicação entre processadores, o desempenho global de uma máquina paralela é fortemente dependente do desempenho da rede que por sua vez é fortemente dependente do desempenho da chave.

Diversos trabalhos tratam da classificação de redes de interconexão [Hwan85][Sieg89][Feng81][Alma89]. Para a máquina paralela que esta sendo desenvolvida no NCE, o MULTIPLUS [Aude90], optou-se pela utilização de uma rede de interconexão multi-estágio com topologia n-cubo invertido, modo de operação assíncrono, com controle distribuído e comutação por pacotes.

Com relação a chave, existem diversas formas de implementação: sem *buffer*, com *buffer* na entrada, com *buffer* na saída, etc. No trabalho apresentado por Tamir e Frazier [Tami88], é feito um estudo de diversas estruturas de chave e são analisadas as principais vantagens e desvantagens de cada uma delas.

Neste trabalho são apresentados e discutidos os resultados de desempenho de redes de interconexão, obtidos através de simulação, para três das estruturas de chave analisadas em [Tami88]. A análise destes resultados auxiliará na escolha da estrutura de chave a ser utilizada no MULTIPLUS [Aude90].

O conteúdo do trabalho esta organizado da maneira que se segue. A arquitetura do MULTIPLUS é brevemente descrita na seção 2. Nas seções 3 e 4 são descritos, respectivamente, os modelos de simulação funcional e analítico. A análise dos resultados obtidos com o simulador funcional, bem como uma comparação dos resultados dos dois modelos são apresentados na seção 5. Finalmente, a seção 6 apresenta as conclusões obtidas.

2 - A ARQUITETURA DO MULTIPLUS

O MULTIPLUS [Aude90] é um multimicroprocessador de alto desempenho voltado para o processamento científico. A sua arquitetura (fig. 1) é modular sendo capaz de suportar até 256 nós de processamento (NP). Cada NP é constituído de um microprocessador RISC baseado na arquitetura SPARC [SUN87], um *cache* de instruções e um *cache* de dados com 64 KBytes cada, um co-processador de ponto flutuante, 16 MBytes de memória que representam uma região do espaço global de endereçamento e *hardware* de suporte à gerência de memória. Até oito nós de processamento podem ser ligados através de um barramento duplo formando um *cluster* de processamento, que são interconectados através de uma rede de interconexão do tipo n-cubo invertido.

No MULTIPLUS, um processador pode acessar a memória de quatro maneiras diferentes: no *cache* local (*cache hit*), nos 16 MBytes de memória fisicamente locais ao NP (acesso local ao NP), na memória pertencente ao *cluster* ao qual

o NP pertence (acesso local ao *cluster*) ou na memória não pertencente ao *cluster* ao qual o NP pertence (acesso remoto).

Além disto, os barramentos são especializados: em um deles, tem-se apenas leitura de instruções e no outro, apenas leitura e escrita de dados. As instruções e os dados de apenas leitura são sempre *cacheáveis*. Já os dados que podem ser alterados são *cacheáveis* apenas no *cluster* que contém o seu endereço. Com isto, só é necessário manter a coerência de *cache* dentro dos *clusters*.

A arquitetura de E/S do MULTIPLUS é distribuída. A cada *cluster* estão associados dois processadores de E/S. Um deles é orientado a caracter e processa operações de E/S com terminais e impressoras. O outro é orientado a bloco e processa operações de E/S com discos e fitas.

O sistema operacional será uma extensão do Plurix [Fall89], compatível com o UNIX System V da AT&T e capaz de suportar multiprocessamento.

3 - MODELAGEM FUNCIONAL DE REDES DE INTERCONEXÃO

Nesta seção é apresentado o modelo utilizado no simulador funcional. A simulação funcional tenta reproduzir (por software) o mais fielmente possível, o funcionamento real do sistema. Os resultados obtidos serão tão mais acurados quanto maior for o nível de detalhamento do modelo. Em geral, quanto mais detalhado o modelo, mais complexo ele é, existindo então um compromisso entre o nível de detalhamento e a complexidade. A simulação funcional apresenta a vantagem de poder utilizar distribuições não uniformes com a mesma simplicidade com que utiliza distribuições uniformes.

Dentre as diversas estruturas de chave analisadas em [Tami88], serão simuladas as seguintes: chave com *buffer* simples em cada entrada (fig. 2.a), chave com *buffer* simples em cada saída (fig. 2.b) e chave com *buffer* duplo em cada entrada e multiplex na saída (fig. 2.c). Estas estruturas foram escolhidas pois representam 3 classes distintas e não estão claras as diferenças no desempenho de cada uma delas. As demais foram descartadas porque apresentam desvantagens em relação as três estruturas escolhidas (ver [Tami88]).

Modelagem da Rede e da Chave

As redes de interconexão simuladas tem uma topologia n-cubo invertido composta de k estágios e $M = 2^k$ entradas e saídas. Cada estágio possui $M/2$ chaves 2×2 com *buffers* internos, organizados segundo a política *first-in first-out* (FIFO). Todos os *buffers* possuem o mesmo tamanho.

A rede opera no modo de comutação por pacotes e as mensagens tem tamanho fixo e igual a um pacote. Todas as operações se processam sincronamente e ocorrem em múltiplos inteiros de T_c , definido como ciclo de operação.

As mensagens possuem um campo de endereço que permite o seu roteamento através da rede até o destino. A cada ciclo de operação, as mensagens são transmitidas do estágio i para o estágio $i + 1$, até alcançarem o destino. Existem três situações que podem ocorrer durante a transmissão de uma mensagem: uma transmissão normal, uma contenção ou um conflito. Uma transmissão normal ocorre quando uma mensagem é transmitida com sucesso do *buffer-origem* para o *buffer-destino*. Uma contenção ocorre sempre que uma mensagem é destinada a um *buffer* completamente cheio (sem espaço para

armazená-la). Neste caso a mensagem permanece no *buffer*-origem. Um conflito ocorre quando, em um mesmo ciclo, duas mensagens são destinadas a um mesmo *buffer*. Como apenas uma das mensagens pode ser transmitida (a outra permanece no *buffer*-origem), foram propostas três políticas para a resolução de conflitos: aleatória, rotativa e preferencial.

Na política aleatória, a cada conflito, é sorteado aleatoriamente o vencedor, independente dos vencedores dos conflitos anteriores. Na política rotativa, existe um registro que informa quem foi o vencedor do último conflito. A prioridade é dada, então, ao outro *buffer* que passa a ser o último vencedor no caso da transmissão ser realizada com êxito (sem conflito nem contenção).

Algoritmo para a política rotativa:

```
if Ultimo_Vencedor := Buff0 then
  Vencedor_Atual := Buff1
else
  Vencedor_Atual := Buff0;
Realiza_Acesso;
if Acesso_Realizado_Com_Exito then
  Ultimo_Vencedor := Vencedor_Atual;
```

Na política preferencial, a prioridade é dada ao *buffer* que estiver mais cheio. No caso de empate (ambos com a mesma ocupação), é sorteado aleatoriamente o vencedor.

Modelagem das Condições de Contorno e Geração de Acessos

Entende-se por condições de contorno quaisquer eventos relativos a interface da rede com o mundo exterior, que são suas portas de entrada e de saída.

Como condição de contorno na entrada, é gerado um vetor de mensagens (vetor de acessos) a cada ciclo que é submetido à rede. Os acessos podem ser divididos em 2 tipos: acessos comuns e acessos a um *hot-spot*.

Nos acessos comuns é definida uma **probabilidade de acesso** (P_{ac}) como sendo a probabilidade que cada entrada m tem de receber uma mensagem, em cada ciclo. A cada mensagem é atribuído um endereço de destino, escolhido aleatoriamente entre 0 e $M - 1$ e diferente de m , com distribuição uniforme.

Algoritmo:

```
for I := 0 to M - 1 do
begin
  RND := Random(100)1;
  if RND < Pac then
    Vetor_Entrada[I] := Sorteio(M,I)2
  else
```

¹Random(I) - Gera um número aleatório entre 0 e I - 1.

²Sorteio(I,J) - Gera um número aleatório entre 0 e I - 1, diferente de J.

```

    Vetor_Entrada[I] := Vazio;
end

```

A simulação de *hot-spots* é feita atribuindo-se um peso maior para uma determinada saída no sorteio do endereço de destino da mensagem. É definida uma **probabilidade de hot-spot** (P_{hs}) como sendo a probabilidade do acesso gerado ser dirigido a um *hot-spot*. Sem perda de generalidade, o endereço do *hot-spot* é fixo e igual a $M - 1$. Se o acesso não for dirigido ao *hot-spot*, o endereço de destino da mensagem é escolhido aleatoriamente entre 0 e $M - 1$ e diferente de m , com distribuição uniforme.

Algoritmo:

```

for I := 0 to M - 1 do
begin
    RND := Random(100);
    if RND < Pac then
begin
    RND := Random(100);
    if RND < Phs then
        Vetor_Entrada[I] := M - 1
    else
        Vetor_Entrada[I] := Sorteio(M, I)
    end
end
else
    Vetor_Entrada[I] := Vazio;
end;

```

Como condição de contorno na saída, a cada ciclo é gerado um vetor de contenções que reflete o estado (livre ou bloqueado) de cada saída da rede. É definida uma **probabilidade de contenção** (P_{cn}) como sendo a probabilidade que cada saída tem de estar bloqueada. Uma saída bloqueada não aceita mensagens vindas da rede.

Algoritmo:

```

for I := 0 to M - 1 do
begin
    RND := Random(100);
    if RND < Pcn then
        Vetor_Saida[I] := Bloqueado
    else
        Vetor_Saida[I] := Livre;
    end;
end;

```

Definição dos Parâmetros de Saída do Simulador

Nesta seção são descritos os parâmetros que podem ser obtidos com o simulador e como eles são calculados.

- Taxa de Acessos

Um acesso é definido como qualquer tentativa de se enviar um pacote de um estágio para o estágio seguinte (incluindo as portas de entrada e de saída). Existem 3 situações que podem ocorrer: o pacote ser enviado normalmente, ocorrer um conflito ou ocorrer uma contenção. O simulador fornece como saída a quantidade de acessos normais, de acessos com contenção, e de acessos com

conflito. Estes parâmetros são normalizados em relação ao número total de acessos.

O algoritmo para classificação de um acesso é o seguinte (supondo uma rede com m portas):

```

for Estágio := m downto 1 do
begin
  if Destino_Mensagem_1 = Destino_Mensagem_2 then
  begin
    Conflitos := Conflitos + 1;
    Escolhe_Vencedor;
    Realiza_Acesso;
    if Acesso_Realizado_Com_Exito then
      Normais := Normais + 1
    else
      Contenção := Contenção + 1;
    end
  else
    for I := 1 to 2 do
    begin
      Realiza_Acesso;
      if Acesso_Realizado_Com_Exito then
        Normais := Normais + 1
      else
        Contenção := Contenção + 1;
      end;
    end;
  end;
end;

```

- Atraso Médio

É definido como sendo o tempo médio, em ciclos, que as mensagens levam para atravessar a rede. Utiliza-se normalmente o **atraso médio normalizado (AMN)**, que fornece o atraso médio por estágio da rede. O cálculo do atraso médio é dado por:

$$AMN = \frac{AM}{n} = \frac{1}{n \cdot TM} \sum_{i=1}^{TM} T_i \quad (1)$$

onde,

AM - Atraso médio
 AMN - Atraso médio normalizado
 TM - Total de mensagens que saíram da rede
 T_i - Tempo gasto pela mensagem i para atravessar a rede
 n - Número de estágios da rede

- Ocupação Média

É definida como o valor médio de ocupação de todos os *buffers* da rede, desde o instante inicial de simulação (ciclo 0) até o instante atual. Este parâmetro é normalizado em relação ao tamanho do *buffer*. O cálculo é feito da seguinte maneira:

A cada ciclo,

$$\text{TOT_BUFFER} := \text{TOT_BUFFER} + \sum_{i=1}^{\text{NTF}} \text{OC}_i \quad (2)$$

$$\text{OM} = \frac{\text{TOT_BUFFER}}{\text{NTF.ciclo}} \quad (3)$$

onde,

OC_i - Ocupação do *buffer* i
 NTF - Número total de *buffers*
 OM - Ocupação média
ciclo - Ciclo atual de simulação

- Throughput Médio

É definido como sendo o número médio de pacotes que atravessaram a rede por unidade de tempo (ciclo). Este parâmetro dá uma noção da capacidade de transmissão (vazão) da rede. Normalmente é utilizado o **throughput médio normalizado** (ou simplesmente **throughput normalizado**), que é a razão entre o *throughput* médio e o *throughput* máximo. Para uma rede com m portas o *throughput* máximo é igual a m pacotes por ciclo. O *throughput* médio normalizado permite a comparação de redes com diferentes características, já que ele nos dá uma idéia de quão próximo do desempenho máximo a rede está operando. O cálculo é feito da seguinte forma:

$$\text{THN} = \frac{\text{THM}}{m} = \frac{\text{TM}}{m.\text{ciclo}} \quad (4)$$

onde,

THM - Throughput médio
 THN - Throughput médio normalizado
 TM - Total de mensagens que saíram da rede
 CICLO - Ciclo atual de simulação
 m - Número de portas de saída

4 - MODELAGEM ANALÍTICA DA REDE

O simulador analítico aqui apresentado é baseado em um conjunto de equações de estado proposto por Yoon et al. [Yoon90] que são resolvidas iterativamente até que se alcance uma situação de regime permanente.

Características Gerais do Modelo Analítico

O modelo analítico utilizado apresenta as seguintes características principais:

- A rede possui N entradas, N saídas e n estágios de N/a chaves ($a \times a$), onde $N = a^n$.
- As mensagens são geradas com igual probabilidade em cada entrada e a

- distribuição dos endereços de destino é uniforme: cada mensagem tem uma probabilidade de $1/N$ de ir para cada uma das N saídas.
- As mensagens tem tamanho fixo e igual a um pacote.
 - A política adotada para a resolução de conflitos é a aleatória.
 - Cada chave possui a *buffers* em suas entradas com espaço para m mensagens.

O fato do perfil de distribuição dos endereços de destino ser uniforme, faz com que, para cada estágio da rede, se tenha um padrão de distribuição idêntico para todas as chaves. Portanto, cada estágio pode ser caracterizado por uma única chave.

Modelo Analítico para Chave com Buffer na Entrada

Algumas modificações foram introduzidas no conjunto de equações de estado afim de adaptá-lo ao modelo funcional descrito na seção 3. Estas alterações dizem respeito principalmente as condições de contorno ($r(n,t)$ e $q(1,t)$), já que o modelo analítico assume que, se uma mensagem está disponível na saída da rede, ela é sempre removida ($P_{cn} = 0$). A probabilidade de contenção (P_{cn}) é embutida no modelo analítico com a introdução do fator $(1 - P_{cn})$ no cálculo de $r(n,t)$ (equação [12] de [Yoon90]). Na entrada da rede, a probabilidade de acesso (P_{ac}) é expressa fazendo $q(1,t) = P_{ac}$.

Modelo Analítico para Chave com Buffer na Saída

O conjunto de equações utilizado se refere a chaves com *buffer* na entrada. Porém, é possível adaptá-lo para o caso dos *buffers* estarem localizados na saída. As condições de contorno devem ser alteradas, pois a posição dos *buffers* foi invertida. A probabilidade de uma mensagem sair da rede, $r(n,t)$, é expressa por (equação [12] de [Yoon90]):

$$r(n,t) = 1 - P_{cn} \quad (5)$$

Como condição de contorno na entrada, a probabilidade de uma mensagem entrar na rede passa a ser:

$$q(1,t) = 1 - [1 - P_{ac}/a]^a \quad (6)$$

E a probabilidade de uma mensagem passar do estágio $k - 1$ para o estágio k , $q(k,t)$, passa a ser (equação [10] de [Yoon90]):

$$q(k,t) = 1 - [1 - \bar{p}_0(k-1,t)/a]^a \quad 2 \leq k \leq n \quad (7)$$

5 - ANÁLISE DOS RESULTADOS OBTIDOS

Parâmetros Utilizados para Avaliação do Desempenho

Os principais parâmetros utilizados para avaliar e comparar o desempenho de redes de interconexão são o *throughput* e o atraso médio. Em um sistema onde a rede de interconexão é o ponto crítico, o *throughput* pode fornecer uma indicação de seu desempenho. Já no caso onde a dependência da geração de mensagens é grande (as mensagens atuais dependem das respostas das mensagens anteriores), o atraso médio é mais significativo que o *throughput*, pois a taxa com que as mensagens são geradas depende fortemente do tempo gasto pela transmissão e resposta das mesmas.

Uma forma de se avaliar e comparar os limites superiores de desempenho das 3 arquiteturas propostas é submetê-las às condições de máximo desempenho (CMD)

[Dias81], que são:

- As mensagens são apresentadas às entradas na taxa máxima que elas podem aceitar. Isto quer dizer que, no início de cada ciclo, haverá sempre uma mensagem para cada entrada (probabilidade de acesso = 100%).
- As mensagens são removidas da rede no momento em que elas chegam nas portas de saída. Isto quer dizer que não existem obstáculos nas portas de saída (probabilidade de contenção = 0%).

Throughput x Tamanho do Buffer

Pode-se imaginar que um aumento no tamanho do *buffer* cause sempre um aumento no *throughput*, pois este aumento diminuiria a probabilidade de ocorrer uma contenção. A análise do gráfico da fig. 3 mostra que a partir de um determinado tamanho (chamado de **tamanho útil máximo**, que para o modelo utilizado corresponde a um *buffer* com aprox. 16 posições), o aumento do *buffer* não acarreta em um aumento significativo do *throughput*. Por outro lado, este mesmo aumento implica em um aumento significativo do atraso médio (fig. 4). Extendendo a análise feita por Dias e Jump [Dias81], pode-se identificar dois fatores limitantes do *throughput*: conflitos e contenção. Estes dois fatores atuam em conjunto até que o *buffer* atinja o tamanho útil máximo. A partir deste ponto o *buffer* se torna grande o suficiente para aceitar as mensagens sem contenção (fig. 5). Então o fator limitante passa a ser apenas o conflito interno (fig. 6), que é uma característica intrínseca da chave e independente do tamanho do *buffer*.

Ocupação Média x Tamanho do Buffer

A fig. 8 mostra como varia a ocupação média dos *buffers* (por estágio da rede) com o aumento do tamanho dos mesmos. Para a chave com *buffer* na entrada (fig. 8.a), a ocupação média do primeiro estágio é de 100%, independente do tamanho do *buffer*. Isto ocorre porque a taxa média de preenchimento dos *buffers* do primeiro estágio (100%) é maior que a taxa média de retirada (50%). Para os demais estágios, a ocupação média decai com o aumento do *buffer*, pois as taxas de preenchimento e retirada são iguais (50%).

Para a chave com *buffer* na saída (fig. 8.b), as taxas médias de preenchimento e de retirada são sempre iguais (50%), exceto para o último estágio, onde a taxa média de retirada é maior (100%). Isto pode explicar o fato da taxa de ocupação para esta estrutura de chave ser a menor entre as três (fig. 8.d).

Para a chave com *buffer* duplo na entrada e multiplex na saída (fig. 8.c), a taxa média de retirada é igual para todos os estágios (50%) enquanto que a taxa média de preenchimento é de 50% para o primeiro estágio (pois existem dois *buffers* ligados a cada entrada) e entre 50% e 100% para os demais estágios.

Atraso Médio x Tamanho do Buffer

O atraso médio verificado (fig. 4) pode ser explicado a partir da ocupação média dos *buffers*. Quanto maior for a ocupação média, maior será o atraso médio sofrido pelas mensagens, pois estas tenderão a ficar mais tempo no interior dos *buffers*.

Política de Resolução de Conflitos x Throughput

Com relação as 3 políticas propostas, não existe diferença significativa no desempenho apresentado (fig. 7). As duas primeiras (aleatória e rotativa) são, na prática, equivalentes pois atendem a cada uma das entradas com probabilidades iguais (50%). A política 3 (preferencial) poderia apresentar um resultado diferente das duas anteriores, devido a não-uniformidade do seu atendimento as entradas. Porém, como os acessos são uniformemente distribuídos entre todas as saídas, a ocupação dos *buffers* tende a ser uniforme também, fazendo com que esta política seja equivalente as duas anteriores. No caso de tráfego não uniforme, esta política pode apresentar diferenças no sentido de homogeneizar a ocupação dos *buffers*.

Comparação dos Resultados dos Modelos Analítico e Funcional

Os gráficos das figuras 9.a e 9.b apresentam uma comparação dos resultados obtidos com os modelos analítico e funcional. Pode-se observar que o modelo analítico apresenta praticamente o mesmo desempenho (*throughput* relativo) tanto para a chave com *buffer* na entrada quanto para a chave com *buffer* na saída (fig. 9.a). Isto ocorre porque, embora as estruturas das duas chaves sejam diferentes, o processo de competição é o mesmo para ambas: dois *buffers* disputando o acesso através do *crossbar*. Para a chave com *buffer* na entrada, estas três estruturas (dois *buffers* e um *crossbar*) estão dentro de uma mesma chave e para a chave com *buffer* na saída, em três chaves diferentes. É interessante observar que o atraso médio da chave com *buffer* na saída (que é bastante peculiar para esta arquitetura) é praticamente o mesmo para ambos os modelos (fig. 9.b). Para a chave com *buffer* na entrada, observa-se uma diferença significativa no atraso médio apresentado pelos dois modelos, sendo que o modelo apresenta um resultado mais otimista (fig. 9.b). Acredita-se que isto ocorra devido ao fato da hipótese de distribuição uniforme para o modelo analítico ser mais real que para o modelo funcional, pois sabe-se que as redes de interconexão apresentam um melhor desempenho quando submetidas a um tráfego uniforme [Yoon90].

6 - CONCLUSÕES

Neste trabalho são apresentados resultados de simulação de três estruturas de elementos chaveadores (chave): chave com *buffer* na entrada, chave com *buffer* na saída e chave com *buffer* duplo na entrada e multiplex na saída. O objetivo destas simulações é definir a estrutura de chave que será utilizada na construção da rede de interconexão do MULTIPLUS.

Dois fatores são importantes no momento desta definição: o custo associado a implementação e o desempenho da chave. O custo de implementação está diretamente ligado ao tamanho do *buffer* e o desempenho da pode ser avaliado pelo *throughput* ou pelo atraso médio.

Das três estruturas de chave simuladas, a chave com *buffer* na saída é a que apresenta o menor atraso médio. Esta característica é muito importante em sistemas onde os processadores precisam aguardar a transmissão e resposta das mensagens para poder continuar o processamento (caso do MULTIPLUS). Com relação ao *throughput*, a diferença apresentada pelas três estruturas não é tão significativa quanto a diferença verificada no atraso médio.

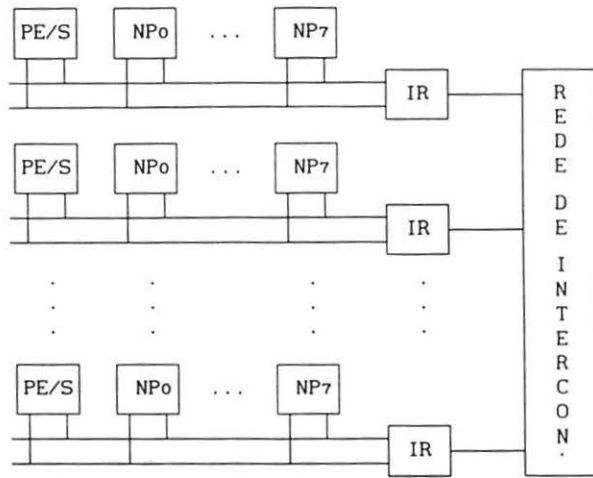
A chave com *buffer* na saída também possui o menor custo de implementação, pois apresenta a menor taxa de ocupação dos *buffers*. Isto implica em *buffers* menores e conseqüentemente mais baratos.

7 - AGRADECIMENTOS

Os autores agradecem ao CNPq e à FINEP o apoio ao desenvolvimento deste projeto.

8 - REFERÊNCIAS

- [Alma89] Almasi, G. S. e Gottlieb, A., *Highly Parallel Computing*, Benjamin/Cummings Publishing, 1989.
- [Aude90] Aude, J. S. et al., "MULTIPLUS: Um multimicroprocessador de alto desempenho", *Anais do X Congresso da SBC*, Vitória, Jul. 1990.
- [Cruz90] Cruz, A. J. O. e Bronstein, G., "Um estudo sobre redes de interconexão e a sua utilização no projeto MULTIPLUS", Relatório Técnico NCE-12/90, NCE/UFRJ, Abr. 1990.
- [Dias81] Dias, D. M. e Jump, J. R., "Packet switching interconnection networks for modular systems", *IEEE Computer*, vol. 14, no. 12, Dez. 1981.
- [Fall86] Faller, N. e Salenbauch, P., "Plurix: A multiprocessing Unix-like operating system", *Proceedings of the 2nd Workshop on Workstations Operating Systems*, Pacific Grove, CA, Set. 1986.
- [Feng81] Feng, T. Y., "A survey of interconnection networks", *IEEE Computer*, vol. 14, no. 12, Dez. 1981.
- [Hwan85] Hwang, K. e Briggs, F. A., *Computer Architecture and Parallel Processing*, McGraw-Hill, 1985.
- [Mena] Menascé, D. A. e Barroso, L. A., "A methodology for performance evaluation of parallel applications on multiprocessors", -.
- [Tami88] Tamir, Y. e Frazier, G. L., "High-performance multi-queue buffers for VLSI communication switches", *ACM Computer Architecture News*, vol. 16, no. 2, Mai. 1988.
- [Sieg79] Siegel, H. J., "Interconnection networks for SIMD machines", *IEEE Computer*, vol. 12, no. 6, Jun. 1979.
- [Sieg81] Siegel, H. J. e McMillen, R. J., "The multistage cube: a versatile interconnection network", *IEEE Computer*, vol. 14, no. 12, Dez. 1981.
- [Sieg89] Siegel, H. J. et al., "Using the multistage cube network topology in parallel supercomputers", *Proceedings of the IEEE*, vol. 77, no. 12, Dez. 1989.
- [Sun87] -, *The SPARC Architecture Manual*, Sun Microsystems Inc., Out. 1987.
- [Yoon90] Yoon, H. et al., "Performance analysis of multibuffered packet-switching networks in multiprocessor systems", *IEEE Trans. Comput.*, vol. 39, no. 3, Mar. 1990.



PE/S - Processador de Entrada e Saída
 NP_i - Nó de Processamento i
 IR - Interface com a Rede de Interconexão

Fig. 1 - Arquitetura do MULTIPLUS

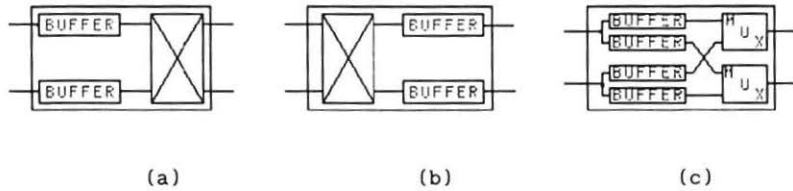


Fig. 2 - Estruturas simuladas. (a) Buffer na entrada. (b) Buffer na saída. (c) Buffer duplo na entrada e multiplex na saída.

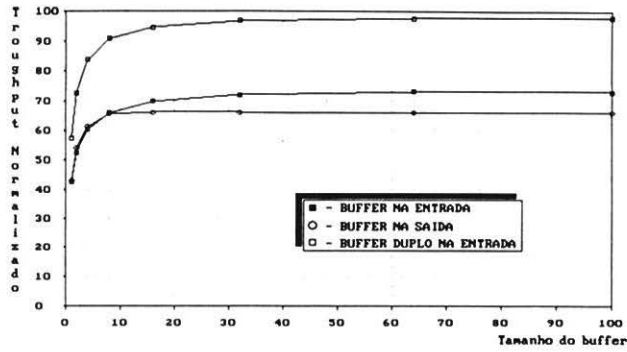


fig. 3 - Throughput relativo em função do tamanho do *buffer* ($P_{ac} = 100\%$, $P_{cn} = 0\%$, $n = 5$ estágios).

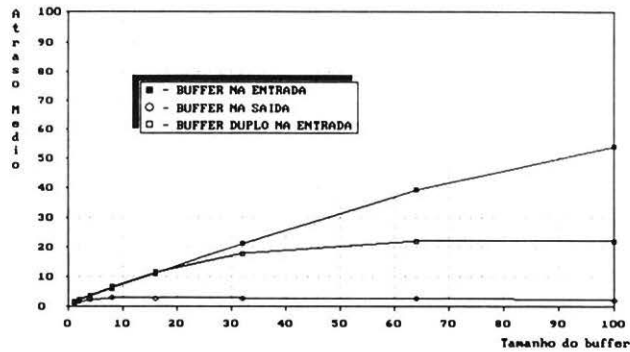


fig. 4 - Atraso médio em função do tamanho do *buffer* ($P_{ac} = 100\%$, $P_{cn} = 0\%$, $n = 5$ estágios).

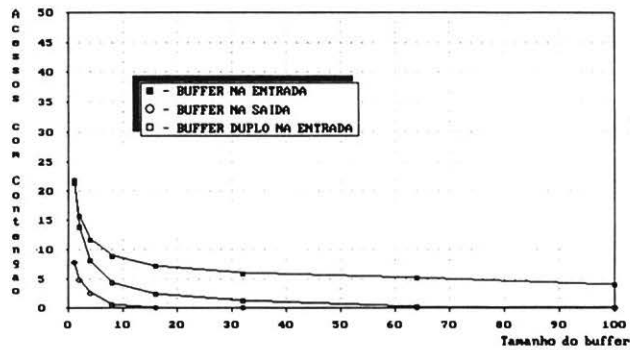


fig. 5 - Porcentagem de acessos com contenção em função do tamanho do *buffer* ($P_{ac} = 100\%$, $P_{cn} = 0\%$, $n = 5$ estágios).

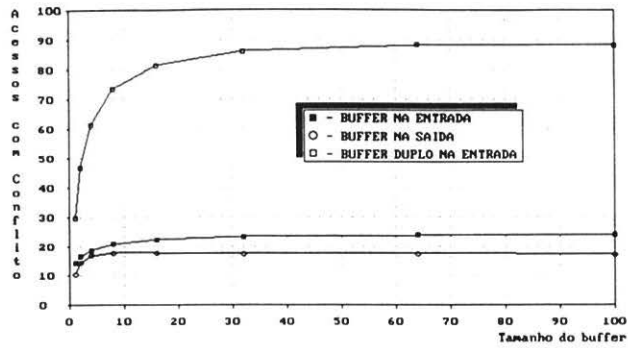


fig. 6 - Porcentagem de acessos com conflito em função do tamanho do *buffer* ($P_{ac} = 100\%$, $P_{cn} = 0\%$, $n = 5$ estágios).

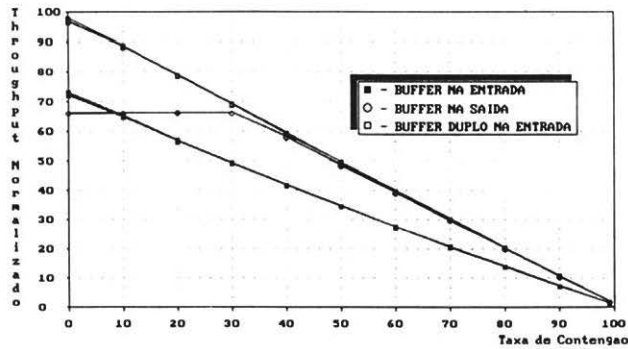
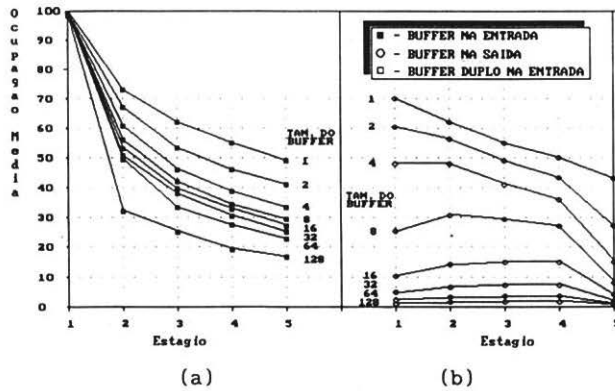


fig. 7 - Comparação das diversas políticas de resolução de conflitos ($P_{ac} = 100\%$, $P_{cn} = 0\%$, $n = 5$ estágios, tam. do *buffer* = 32).



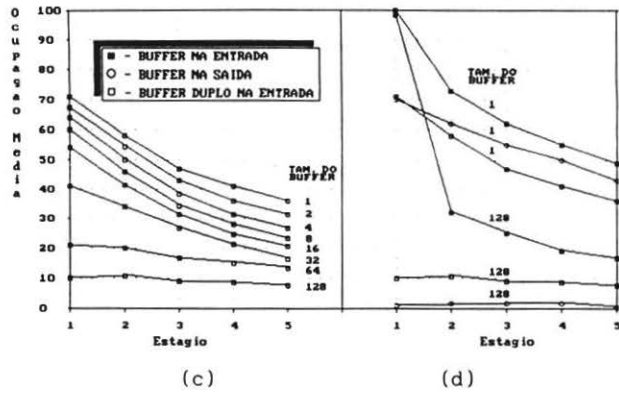
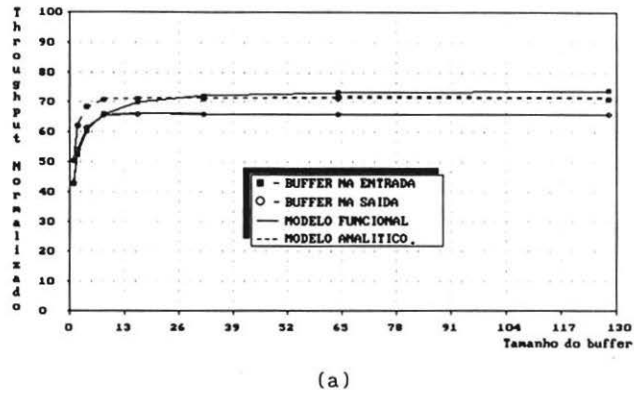
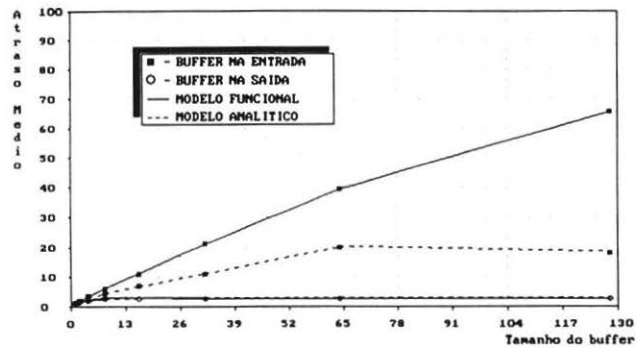


fig. 8 - Ocupação média por estágio ($P_{ac} = 100\%$, $P_{cn} = 0\%$, $n = 5$ estágios). (a) chave com *buffer* na entrada. (b) chave com *buffer* na saída. (c) chave com *buffer* duplo na entrada e multiplex na saída. (d) comparação das 3 estruturas



(a)



(b)

fig. 9 - Comparação entre os modelos funcional e analítico ($P_{ac} = 100\%$, $P_{cn} = 0\%$, $n = 5$ estágios). (a) *Throughput* normalizado. (b) Atraso médio.