

TÉCNICAS DE MULTIPLEXAÇÃO POR DIVISÃO NO TEMPO PARA INTERCONEXÃO PROCESSADOR MEMÓRIA EM SISTEMAS DE MULTIPROCESSADORES: UM ESTUDO COMPARATIVO

MARCOS A.G. BRASILEIRO* E J. ANTÃO B. MOURA*

SUMÁRIO

Quatro técnicas de multiplexação por divisão no tempo são investigadas para determinar a mais adequada para a comunicação processador-memória em sistemas com multiprocessadores. O estudo é realizado através de simulações.

ABSTRACT

Four TDM scanning disciplines are simulated in order to determine the technique best suited for the interconnection mechanism of a multiprocessor system.

* DEE/CCT/UFPb

Grupo de Redes de Computadores

Fone: (083)322.1948

58.100 — Campina Grande - Pb.

A proliferação de micro e mini computadores tem instigado o projeto e implementação de sistemas com multiprocessadores para explorar o paralelismo inerente das tarefas submetidas. O objetivo destas implementações é geralmente, melhorar a vazão de tarefas processadas. Para tanto, vários processadores operam simultaneamente em várias sub-tarefas ou em um número de tarefas. É importante notar que os projetos desses sistemas a multiprocessadas são realizáveis com o nível de tecnologia de hardware já atingido; o desempenho dos sistemas porém, ainda é distante do caso ideal. Em geral, sabe-se que qualquer arquitetura de computador sofre uma degradação em desempenho devido às limitações impostas pelo mecanismo de interconexão, responsável por interligar as suas várias unidades funcionais, i.e., processadores, unidade de entrada e saída (E/S) e blocos de memória.

O sistema com multiprocessador de interesse aqui é a máquina de Instruções Múltiplas e fluxos de Dados Múltiplos (IMDM) [FLYN 72, HAND 77]. O mecanismo de interconexão empregado num sistema IMDM oferece vários graus de "acoplamento" entre as unidades funcionais. Assim, as redes de computadores ou processadores distribuídos podem ser classificadas como sistemas IMDM acoplados frouxamente ("loosely coupled"). Nosso interesse, entretanto, focaliza as organizações acopladas "rigidamente" ("tightly coupled") onde os processadores e memórias são conectados via alguma malha de interconexão, constituindo-se em sistemas fisicamente "fechados" ou restritos a unidades isoladas, com identidade própria. Como esperado, a malha de interconexão pode ser um fator limitante com respeito à vazão.

O problema de interconectar sub-unidades tem recebido atenção considerável, destacando-se as redes telefônicas reconfiguráveis [BATC 68], os arranjos celulares de Kautz [KAUT 68], redes de permutação serial [JOEL 68], redes de Lawrie [LAWR 73], interconexão "shuffle" [STON 75], memória multiporta [ENSL 77]. Sugestões de interconexão via um único barramento com acesso aleatório (como em redes locais) foram também feitas [MEHR 79]. Entre estes, os mecanismos de interconexão mais freqüentemente adotados em sistemas IMDM são o arranjo "cross point", o barramento compartilhado no tempo, memórias multiportas e redes de chaves binárias. Apesar de que o escopo dos esquemas de interconexão é bem mais amplo e complexo, estas categorias contudo, formam uma base útil para a discussão da organização de máquinas IMDM.

Enslow [ENSL 77] e Pearce [PEAR 77] investigaram e compararam o desempenho dos mecanismos mais freqüentes e concluíram que o sistema de interconexão mais simples, com menor custo para um ambiente de multiprocessadores, é um barramento compartilhado no tempo, o qual conecta todas as unidades funcionais. Uma maneira de se efetuar este compartilhamento, conhecida por Multiplexação por Divisão no Tempo (MDT), é alocar um intervalo de tempo (chamado de quadro ou slot) a cada unidade funcional, a qual acessa o barramento apenas durante o quadro a ela alocado. Um controlador (unidade de controle de multiplexação) é responsável por alocar os quadros às unidades funcionais. Uma alternativa a esta maneira é fazer com que cada unidade que deseje utilizar o barramento determine a disponibilidade do mesmo e o acesso num intervalo conveniente, minimizando a chance de possí

veis conflitos. O controle de acesso assim especificado é dito ser distribuído. A vantagem do controle centralizado é o seu baixo custo.

O objetivo deste artigo é investigar a adequação de algumas técnicas de MDT sob controle centralizado para acesso a uma configuração de barramento comum, no contexto de sistemas IMDM.

2. ARQUITETURA IMDM DE INTERESSE

Para efeito do presente estudo, considera-se um cenário para a comunicação entre as unidades funcionais (processadores e bancos de memória) de um sistema IMDM que corresponde à comunicação unidirecional oriunda de N Processadores para M Bancos de Memória. Este cenário é de interesse pois, enquanto simples, permite considerações preliminares sobre as características da arquitetura IMDM com interconexão via métodos MDT. Ele será também utilizado numa avaliação comparativa destes métodos a fim de selecionar o método mais adequado para uso em sistemas IMDM.

2.1. Interconexão das Unidades Funcionais

O cenário em consideração é baseado na arquitetura de um sistema IMDM, ilustrado na figura 1. Como mostra a figura, as unidades funcionais são "rigidamente" acopladas ("tightly coupled") através de um barramento. Os N Processadores compartilham o barramento através de um multiprocessador com N linhas de entrada (uma para cada processador) e com o barramento na sua saída. O Decodificador é utilizado para fazer o encaminhamento das mensagens no barramento a um dos M Bancos de Memória. Para tanto, cada mensagem no barramento contém o endereço do Banco de Memória destinatário.

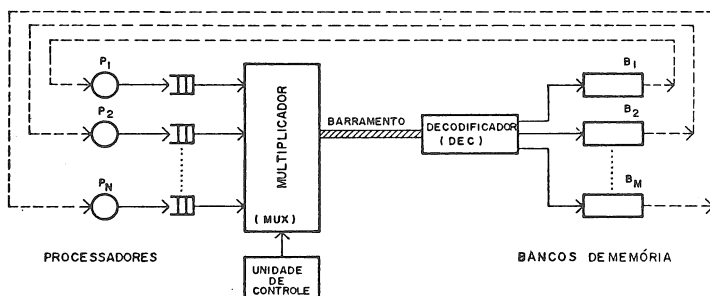


Figura 1: Conexão Processador-Memória

Convém observar que o barramento na figura 1 é compartilhado (por MDT) entre as unidades que a ele se ligam. MDT é um método de multiplexação no qual o tempo de um canal multiplexado é dividido em quadros (de duração fixa ou variável). Os quadros são então alocados, em instantes diferentes, a diferentes usuários do canal. A alocação pode ser repetida regularmente (MDT Síncrona, MDTs) ou pode ser feita sob demanda (MDT Assíncrona ou MDTA) [DAVI 73].

Com métodos MDT, apenas uma unidade pode ter acesso ao barramento por vez. Assim, para que não se percam mensagens geradas por uma unidade enquanto ela não tem acesso ao barramento e/ou para liberar mais rapidamente os processadores de tarefas de comunicação (ex.: evitar que um processador fique à espera do uso do barramento), fornecem-se buffers a cada linha de entrada dos multiplexadores (vide figura 1). As mensagens são então armazenadas nos buffers enquanto aguardam transmissão no barramento.

2.2. Disciplina de Atendimento

Quando várias unidades funcionais compartilham um barramento por meio de métodos MDT, deve-se definir uma "disciplina de atendimento" (implementada pelo controlador do multiplexador) para que cada unidade possa requisitar e obter controle do barramento (a fim de efetuar suas transmissões). A dificuldade maior quanto à disciplina de atendimento diz respeito à resolução de conflitos nos pedidos das unidades para acesso ao barramento. Como vimos, apenas uma unidade deve ter acesso em qualquer instante. O presente artigo trata de quatro disciplinas de atendimento, as quais são descritas abaixo.

2.2.1. Disciplina de Atendimento Cíclico (MDTS)

A disciplina de atendimento MDTs é cíclica no sentido de que o controlador do multiplexador reserva um quadro fixo para cada linha de entrada independentemente de sua necessidade de transmissão. Este comportamento implica em que as linhas sejam atendidas em sequência, numa frequência fixa. Com a disciplina cíclica, a sequência de quadros no tempo é como mostra a figura 2.

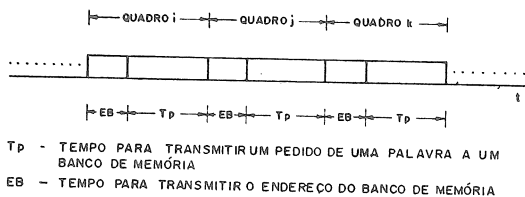


Figura 2: Sequência de Quadros em MDTs

2.2.2. Disciplina com Lista de Chamada (LIC)

Um controlador implementando LIC verifica o estado do buffer de cada linha. Se o buffer sendo verificado tem algo a transmitir, o controlador aloca um quadro ao primeiro item no buffer. Caso contrário, o controlador passa para a próxima linha segundo uma "ordem de chamada" fixa. Observe que a chamada segue um procedimento síncrono já que um intervalo de tempo fixo é gasto para verificar cada buffer, independentemente do seu estado. Esta disciplina apresenta uma sequência de quadros como ilustra a figura 3.

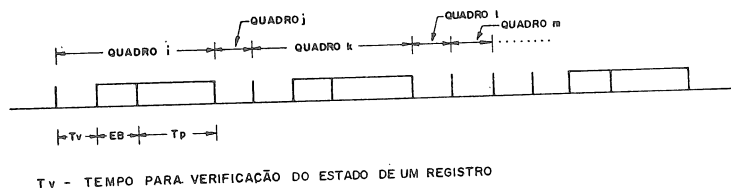


Figura 3: Possível Seqüência de Quadros com LIC

2.2.3. Disciplina com Verificação Múltiplas (VEM)

VEM é semelhante a LIC, mas com a diferença de que, se o buffer se encontrar vazio inicialmente, o controlador insiste na verificação por um certo número de vezes, n , antes de passar à próxima linha. Por exemplo, se $n=3$, o controlador desperdiçará, no máximo $3T_t$ segundos antes de desistir e passar para a linha seguinte. A figura 4 ilustra uma possível seqüência de quadros com a disciplina VEM para o caso de $n=3$.

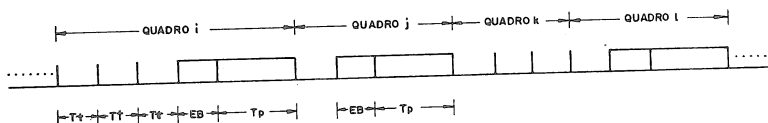


Figura 4: Exemplo de uma Seqüência de Quadros com VEM

2.2.4. Disciplina com Verificação Aleatória (VEA)

VEA faz com que a verificação de um dado buffer de linha se dê em um certo período de tempo, s , de acordo com uma probabilidade $p(s)$. Uma possibilidade é fazer $p(s)$ uma função do número de linhas de entrada do multiplexador, como por exemplo (vide figura 1),

$$p(s) = \frac{1}{N}$$

A duração da verificação, como no caso da disciplina VEM, é T_t seg. A seqüência de quadros com VEA é semelhante a obtida com LIC (figura 3). Na próxima seção, investigamos as características de desempenho das quatro disciplinas de atendimento apresentadas acima, sob as condições do cenário na figura 1.

3. AVALIAÇÃO DO DESEMPENHO DAS DISCIPLINAS DE ATENDIMENTO

3.1. Descrição do Modelo

A definição do modelo para o sistema IMDM da figura 1 requer suposições e considerações sobre os aspectos operacionais do sistema. Primeiro, assumimos que durante um intervalo de tempo T_s , cada processador, independentemente dos outros, gera um pedido para um dos Bancos de Memória apenas uma vez, com probabilidade λ . Segundo, os tempos de interchega

das de pedidos tem distribuição uniforme com média T_s . Assim, quando um processador faz um pedido, apenas após um intervalo médio de tempo de T_s segundos é que ele fará um outro pedido com probabilidade λ . Esta suposição leva então a uma distribuição binomial para o número de pedidos durante o intervalo T_s [RUDI 71]. Deve-se ressaltar que a distribuição de Poisson para o número de pedidos, freqüentemente adotada a fim de facilitar a solução do modelo [BIRD 62, DOR 67, CHU 69], é aqui descartada devido às características reais do sistema - tais como número finito de processadores, buffers de linha com capacidade finita, etc. Quando um processador pede uma palavra (uma instrução ou um dado) à memória, o processador gera o endereço da palavra e coloca o pedido no seu buffer de linha. Os pedidos de palavra são armazenados no buffer de linha por ordem de chegada. Assume-se que o comprimento dos pedidos de palavra é fixo e composto apenas do endereço da palavra, "EP", e do endereço do Banco de Memória, "EB".

Cada pedido de palavra permanece no buffer de linha até que o multiplexador, controlado pela disciplina de atendimento, remova o pedido e o enderece ao banco de memória via o barramento. Como o sistema IMDM sendo discutido é "rigidamente acoplado", o barramento tem uma alta capacidade de transmissão. Assumimos que os atrasos de transmissão dos pedidos de palavra no barramento são desprezíveis. Uma última suposição é a de que a decodificação dos endereços da palavra leva T_d segundos. O decodificador tem condições de detectar se o banco de memória endereçado está ocupado ou não. Caso o esteja, o pedido é bloqueado, permanecendo na cabeça do buffer de linha aguardando o próximo acesso ao barramento. Caso contrário, o acesso é garantido depois de um atraso de T_a segundos, o equivalente ao tempo de acesso à memória.

3.2. Medidas de Desempenho

As medidas de desempenho de interesse são:

L - Comprimento Médio das filas de pedidos de palavra nos buffers de linha.

E - Tempo Médio de Espera dos pedidos de palavra nos buffers de linha. "E" é a média dos intervalos de tempo definidos entre a chegada de um pedido de palavra ao buffer e o início de sua transmissão através do multiplexador.

P - Probabilidade de Bloqueio de pedidos de palavra nos bancos de memória.

V - Vazão Média do sistema IMDM. "V" fornece o número de pedidos (por unidade de tempo) que passam pelo multiplexador e obtêm sucesso no acesso ao banco de memória endereçado.

A solução do modelo descrito na seção 3.1 mostrará o comportamento das medidas acima em função de variações nos parâmetros N, M e do sistema IMDM com cada uma das quatro disciplinas de atendimento de interesse.

3.3. Solução do Modelo

As suposições feitas na seção 3.1 e a dependência estatística entre as filas que se for

mam nos buffers de linha tornam intratável a solução analítica exata do modelo. Assim, decidimos por uma simulação do sistema.

Um simulador gatilhado por eventos foi implementado na linguagem C e executado num PDP 11 com sistema operacional UNIX*. O simulador inclui todas as características julgadas importantes e inerentes a cada disciplina de atendimento em conjunção com a arquitetura e ambiente da figura 1.

O programa simulador foi dividido em seções modelando os processadores, os buffers de linha, multiplexador, decodificador e bancos de memória. Foram implementadas ainda as seguintes rotinas de suporte: "escalador de eventos" e "coletor de estatística". As semáforas permitidas de Dijkstra (P e V) foram também implementadas. Cada parte do programa foi escrita como co-rotina ou procedimento. Maiores detalhes acerca do programa simulador são fornecidos em [BRAS 85].

3.4. Resultados

O simulador da seção anterior é utilizado para fornecer as características de desempenho das quatro disciplinas de atendimento da seção 2.2, a saber: MDTs, LIC, VEM e VEA.

A tabela 1 fornece os valores dos parâmetros utilizados nas execuções do simulador. Estes valores foram escolhidos para aproximar atrasos típicos [TTL 76] das unidades funcionais. Observe-se, que o simulador é genérico e qualquer outro conjunto de valores poderia ter sido utilizado. Os resultados abaixo têm intervalos de confiança variando de $\pm 0,12\%$ a $\pm 5,23\%$ dos valores verdadeiros com um nível de confiança de 95%.

Parâmetro	Valor
uts - unidade de tempo simulado	25 ns
Ts - média dos tempos de interchegadas	64 uts
Tm - tempo de transporte de um pedido através dos multiplexadores	3 uts
Tv - tempo de verificação de um buffer de linha	1 uts
Td - tempo de decodificação do endereço do banco de memória	3 uts
Ta - tempo de acesso à memória	13 uts
N - número de processadores	2, 4, 8 e 16
M - número de bancos de memória	2, 4, 8 e 16

Tabela 1: Valores dos Parâmetros utilizados no modelo de Simulação do Sistema IMDM da figura 1.

* UNIX é marca registrada da AT&T.

As figuras 5, 6 e 7 apresentam, respectivamente, resultados para L, E e P em função de λ , a probabilidade de um processador gerar um pedido de palavra em Ts segundos, quando N, o número de processadores, é fixado em 8 e $M = 2, 4, 8$ (M , é o número de bancos de memória).

É importante observar que nas figuras 5, 6 e 7, os pedidos bloqueados nos bancos de memória são colocados de volta na cabeça dos respectivos buffers de linha. O desempenho de cada disciplina de atendimento apenas como função da atividade de geração pura de pedidos, sem re-inserção de pedidos bloqueados nos buffers de linha, é ilustrado nas figuras 8, 9, 10 e 11. Nestas figuras, adotou-se $\lambda=1.0$ para se observar o comportamento das disciplinas sob condições de alta intensidade de tráfego.

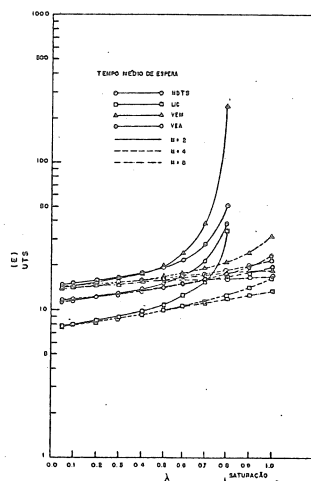
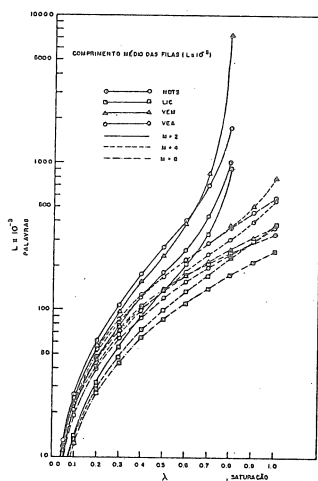


Figura 5: L vs λ para as disciplinas MDTs, LIC, VEM e VEA no caso de Comunicação Processador-Memória.

Figura 6: E vs λ para as disciplinas MDTs, LIC, VEM e VEA no caso de Comunicação Processador-Memória.

As figuras 5 e 6 mostram que a disciplina LIC oferece o melhor desempenho para L e E. Com relação a P, VEM e VEA apresentam menor probabilidade de bloqueio, mas à medida que λ e M crescem, as diferenças se tornam insignificantes.

No caso das figuras 8-11, LIC e MDTs são em geral, superiores. Pode-se observar que LIC é melhor que MDTs com relação a L, E e V quando $M \leq 11$. A partir daí, MDTs se torna mais vantajosa. Isto porque o tráfego, sendo mais intenso, não justifica a verificação de cada buffer de linha (feito por LIC) já que praticamente todos os buffers terão pedidos para transmitir. Nesta situação, apenas poucos quadros da disciplina MDTs serão desperdiçados. Estas considerações são novamente aplicáveis às curvas da vazão na figura 10.

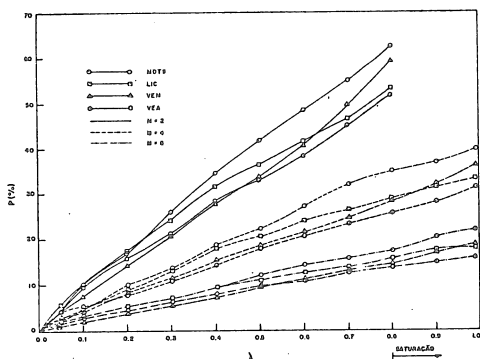


Figura 7: P vs λ para as disciplinas MDTs, LIC, VEM e VEA no caso de Comunicação Processador-Memória.

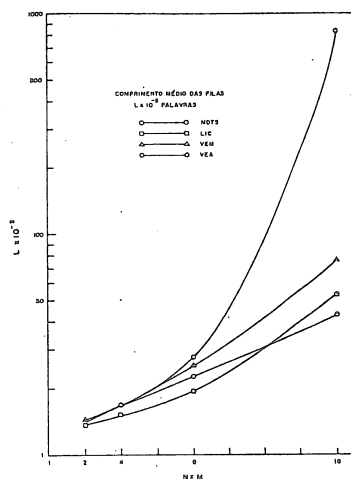


Figura 8: L vs $(N=M)$ para as disciplinas MDTs, LIC, VEM e VEA no caso de Comunicação Processador-Memória.

A disciplina VEA apresenta a pior vazão. Isto se deve à aleatoriedade com que VEA serve cada uma das linhas do multiplexador. De fato, como ela serve um determinado buffer de linha com uma probabilidade igual a $1/N$, pode acontecer que algum buffer nunca seja servido (o que não ocorre com as outras três disciplinas). Quando N cresce, as chances de serviço diminuem fazendo com que VEA apresente uma redução acentuada na vazão.

Finalmente, com relação a P vs $(N=M)$ e $\lambda=1$, o melhor desempenho é alcançado com a disciplina VEA. Como o acesso dos processadores aos bancos de memória é aleatório, é possível que um mesmo buffer de linha seja atendido mais de uma vez consecutivamente, reduzindo assim a possibilidade de bloqueio para esse buffer.

A probabilidade de K verificações sucessivas de um mesmo buffer de linha, com a disciplina VEA, é $(1/N)^K$. Para valores de N grande, essa probabilidade é pequena. Assim, comparando-se as curvas para P das disciplinas VEA e LIC na figura 11 vemos que as diferenças são, no máximo, 5% e que as curvas tendem a se confundir quando N cresce. VEA perde pois sua vantagem sobre LIC. Convém notar também que a complementação de LIC, comparada à implementação de VEA, é mais simples. No cômputo global-figuras 5-11 - LIC é a disciplina com melhor nível de desempenho.

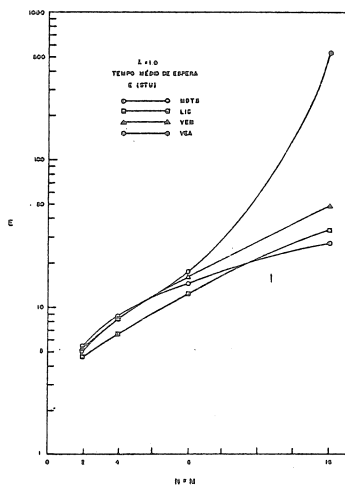


Figura 9: E vs (N=M) para as disciplinas MDTs, LIC, VEM e VEA no caso de Comunicação Processador-Memória

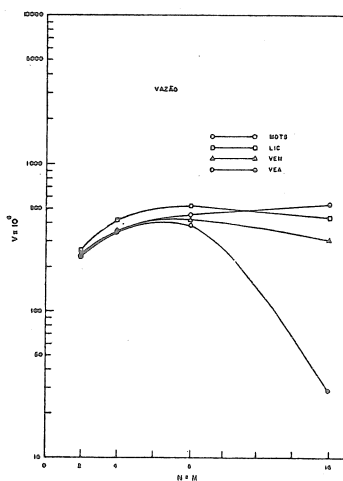


Figura 10: V vs (N=M) para as disciplinas MDTs, LIC, VEM e VEA no caso de Comunicação Processador-Memória

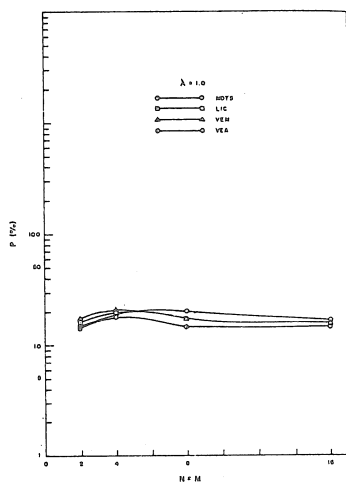


Figura 11: P vs (N=M) para as disciplinas MDTs, VEM e VEA no caso de Comunicação Processador-Memória.

No projeto de um sistema com multiprocessadores (IMDM), uma questão crítica é prover um método de comunicação eficiente e de baixo custo entre os processadores e os bancos de memória. Técnicas de Multiplexação por Divisão no Tempo (TDM) propiciam esta comunicação com as qualidades desejadas. A vazão do sistema porém, depende da disciplina de atendimento. Este artigo apresentou um estudo comparativo de quatro disciplinas: Atendimento Cíclico (MDTS), Lista de Chamadas (LIC), Verificação Múltipla (VEM) e Verificação Aleatória (VEA). Inicialmente, apenas quando a comunicação dos processadores para as memórias foi considerada, chegou-se às seguintes observações:

- 1) As quatro disciplinas apresentam desempenho comparável quando a atividade dos processadores é baixa.
- 2) LIC aparenta ser a melhor disciplina para pequenos sistemas. Para um grande número de processadores, porém, MDTs é superior.
- 3) Os experimentos de simulação mostram que o tempo de acesso à memória, T_a , é um fator importante. Reduzindo-se T_a , minimiza-se a probabilidade de bloqueio, P , e aumenta-se a vazão.
- 4) Observa-se também que P diminui quando um maior número de memórias, M , é empregado. O custo de tal sistema é maior, obviamente.

Limitando a investigação a sistemas multiprocessadores pequenos, LIC deve ser estudada em maiores detalhes sob um cenário de comunicação em ambas direções, i.e., processador-memória-processador. Este cenário é mais realista (e por isto mais complexo), exigindo um modelo mais refinado. O seu estudo não foi apresentado aqui por questões de brevidade, mas, o leitor interessado pode dirigir-se a [BRAS 85] para mais informações.

Como sugestões para continuação deste trabalho, destacam-se um estudo mais detalhado dos sistemas IMDM-LIC com memória locais juntamente com o mecanismo de Busca Prêvia ("Prefetching") a fim de estabelecer a sua utilidade e limitações; e uma análise matemática, possivelmente através da Teoria das Filas, do referido sistema que proporcione maior eficiência de solução, haja visto que o custo de execução do simulador aqui utilizado é apreciável. Referências para essas sugestões seriam [BUZE 73, PEAR 77, WONG 78]. Finalmente, os autores agradecem ao CNPq pelo suporte a este trabalho.

REFERÊNCIAS

- [BATC 68] Batcher, K.E. - "Sorting Networks and their Applications" - Proc. do AFIPS, SJCC, 1968, pp. 307-314.
- [BIRD 62] Birdsall, T.G., Ristenbatt, M.P. e Weinstein, S.B. - "Analysis of Asynchronous Time Multiplexing of Speech Sources" - IRE Trans. Comun. Syst., Vol. CS-10, dezembro 1962, pp. 390-397.

- [CHU 69] Chu, W.W. - "A Study of Asynchronous Time Division Multiplexing for a Time Sharing Computer System" - 2nd Annual Hawaii Conference, 1969.
- [BRA 85] Brasileiro, M.A.G. e Moura, J.A.B. - "Interconexão Processador-Memória via Técnicas de Multiplexação por Divisão no Tempo em Sistemas e Multiprocessadores" - Grupo de Redes de Computadores da UFPb, RT-GRC/UFPb, abril 1985.
- [BUZE 73] Buzen, J.P. - "Computational Algorithmus for Closed Queuing Networks with Exponential Servers" - ACM - setembro 1973 - pp. 527-531.
- [DAVI 73] Davis, D.W. e Barber, D.L.A. - "Communication Networks for Computers" - John Willy & Sons, 1973.
- [DOR 67] Dor, N.M. - "Guide to the Length of Buffer Storage Required for Random Input and Constant Rates" -, IEEE Trans. on Electronic Computers, outubro 1967, pp. 683-684.
- [ENSL 77] Enslow, P.H. Jr. - "Multiprocessor Organization - A Survey", Computing Surveys, Vol. 9, Nº 1, março 1977 - pp. 103-129.
- [FLYN 72] Flynn, M.J. - "Some Computer Organizations and their Effectiveness" - IEEE Trans. Comm. C-21, 9-1972 - pp. 948-960.
- [JOEL 68] Joel, A.E. - "On Permutation Switching Networks" - BSTV maio-junho 1968, pp. 813-822.
- [HAND 77] Handler, W. - "The Impact of Classification Schemes on Computer Architecture" - Proc. 1977 Intl. Conf. on Parallel Processing - agosto 1977 - pp. 7-15.
- [LAWR 73] Lawrie, D.H. - "Memory-Processor Interconnection Networks" - Relatório da Universidade de Illinois UIUCDCS-R - fevereiro 1973, pp. 73-557.
- [MEHR 79] Mehra, S.K. e Majithia, J.C. - "A Modified Ethernet for Multiprocessor Interconnection" - Proc. Comp. Networking Symposium, dezembro 1979 - IEEE - CH 1467 - pp. 132-136.
- [PEAR 77] Pearce, R.C. - "Performance Studies of Parallel Computer Architecture" - Tese de Ph.D., University of Waterloo, Waterloo, Ontario, Canada - 1977.
- [RUDI 71] Rudin, H.J. - "Performance of Simple Multiplexer-Concentrators for Data Communication" -, IEEE Trans. Comm., Vol. COM-19, Nº 2, abril 1971.
- [STON 75] Stone, H.S. - "Introduction to Computer Architecture" - Science Research Associates, Inc. - 1975.
- [TTL 76] TTL Databook 1976.
- [WONG 75] Wong, J.W. - "Queueing Network Models for Computer Systems" - UCLA-ENG-7579 - School of Engineering and Applied Science, University of California at Los Angeles - 1975.