計算機アーキテクチャ　68－5
(1987・11・12)

複数の同一のメモリを持つマルチ·プロセッサシステムに対する性能評価
A Performance Analysis of Multi-processor System with Copied Memory Modules

栗 熹文　白鳥 則郎　野口 正一
LI SHI WEN, NORIO SHIRATORI, AND SHOICHI NOGUCHI
東北大学電気通信研究所
Research Institute of Electrical Communication, Tohoku University

あらまし　コンピュータ·システムの性能をあげるためにマルチプロセッサがよく用いられる。共用バスはデータ伝送時間が速いため、マルチプロセッサ·システムのインタコンネックション·ネットワークとしてよく用いられる。複数の共用バスと共用メモリを使用することによって、共用バスと共用メモリに対する競争を減少することができる。しかし、複数のプロセッサが同時に共用データをアクセスしようとすると、この共用データに対する競争が起こり、システムの性能は下がる。このような競争を避けるため、われわれは本論文で複数の同一メモリを持つマルチプロセッサ·システムを提案した。そのシステムの性能を定量的に評価し、単一のメモリを持つシステム(S-M)と分散されたメモリを持つシステム(D-M)と比較し、その有効性を示した。

Abstract　Multiprocessor systems have been used to improve the performance of computer systems such as to increase execution times. As the interconnection network of computer systems, global bus is specially effective for its high speed transfer.To reduce the contention for global bus and common memory, multiple buses and multiple common memories are used. In the application in which different processors frequently request to access a common data at the same time, the contention for the common data decreases system performance. To reduce such contention thus obtaining high performance, in this paper, we have proposed multiple bus multiprocessor systems with copied memories. We have evaluated the performance of the systems and have compared the results with previously proposed S-M(single-memory module) and D-M(distributed-memory module) systems.

1. Introduction

Multiple bus multiprocessor systems are characterized by the presence of several processors, one or more common memories(used by the processors for the exchange of information) and common buses(to connect the processors and common memories).

Multiple bus multiprocessor systems with single memory modules(S-M systems) and multiple bus multiprocessor systems with distributed memory modules(D-M system) aimed at reducing bus and memory module conflicts were proposed. In previous work[2][3], their performance has been analyzed such that processors were assumed to execute programs stored in their own private memories. The execution of a program was assumed to be interleaved with accesses to common memories.

In some applications in which many processors may want to access the same data in a memory at the same time, the conflicts for common memory become intolerant and the system performance decreases. To reduce the conflicts(i.e.,obtaining high performance), we have proposed a multiprocessor system with copied memory modules(denoted as C-M system). The proposed C-M system is constructed by inserting multiple memories within each memory module such that all the memories in each memory module have even contents. Therefore same data in every memory module can be accessed by different processors *simultaneously*.

In this paper, we evaluate the performance of the proposed C-M systems under the same assumptions of [2] and [3]. We have compared our evaluation results with those of S-M and D-M systems obtained in [2] and [3]. The comparison results show an obvious improvement of our C-M systems compared with S-M and D-M systems.

In the next section we explain the analytical model of the proposed C-M system. In section 3 we introduce the exact performance evaluation of the C-M system. In section 4 we analyze the system approximately which gives closed form solution, and then conform the approximate results. We give numerical examples of the evaluation results and compare them with S-M and D-M systems in Section 5. Conclusions appear in the final section.

2. the Model and Assumptions

The system we propose in this paper consists of N processors, M memory modules, B buses and L copies in each memory module as shown in Fig. 1. We denote it by N*(M-L)*B. We use here dualport memory which permits simultaneously one read and one write accesses if the read access and the write access are not for the same memory location. The data written into common memories will firstly be multiplied-copied at the replicator. Then multiplied-

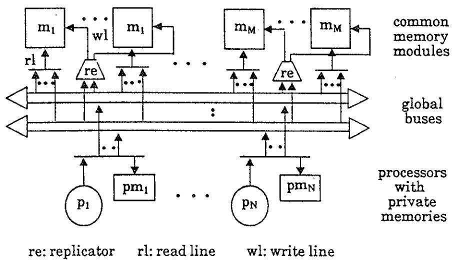

Fig. 1. The model of C-M systems.

copied data are written into the L memories of that memory module. Therefore, all memories of each memory module maintain the same contents all the time. Data will be read through the L read lines directly. Consequently each memory module permits L read accesses and one write accesses simultaneously if there is no memory module conflict.

The memory location conflict occurs if a processor wants to write(read) into the memory location from(into) which another processor is reading(writing).

The memory module conflict occurs if a processor wants to read from a memory module which is being read by L other processors or a processor wants to write into a memory module which is being written by another processor.

The replicators make L copies of the data from its input and send the L copies into the L memories in that memory module.

[Definition 2.1]

Any processor can be in one of the following five states.

1) active state represents the execution state using the private memory.

2) accessing state represents the state accessing a common memory.

3) waiting-for-memory state represents the state waiting to access a common memory because of the common memory module conflicts.

4) waiting-for-bus state represents the state waiting to access a common memory because of the global bus conflicts.

5) rejected state represents the state that the processor's request is rejected because of the memory location conflicts and the processor is waiting to request again after some time. ■

[Definition 2.2]

A memory module being read(written) by n processors(m processors) is called n-read(m-write) memory module, where $n=0, 1, 2$ $m=0, 1$ ■

According to the above definitions, we can explain our system's operaton as follows:

Each processor executes by accessing the private memory for instructions and local data if the data needed are stored in the private memory(in active state). When the data needed are in common memories or the processor wants to write data in common memories, the processor stops executing for the present and requests to access the common memory in which the data needed are stored or are to be written(in accessing state). If there is memory location conflict, the processor's request will be rejected, and the processor has to request again after some time(in rejected state). During this period the processor does nothing.

Even though there is no memory location conflict, but if there is memory module conflict, the processor has to wait until the conflict is dissolved(waiting-for-memory state; Definition 2.1).

If there is neither memory location conflict nor memory module conflict, but no bus is available, the processor has to wait until some bus is released(waiting-for-bus state; Definition 2.1). Then it can access for read or write operation.

After the accessing, the processor returns to active state i.e., executing by using private memory until the next access to common memories.

We make the following assumptions regarding the operation of the system for performance analysis:

1.1) The duration of accesses to common memories is an independent, exponentially distributed random variable with average $1/\mu$ for all the common memory modules.

1.2) Upon memory access completion,memory and bus are immediately released (with zero delay) and the processor resumes its background activity. The interval between subsequent access requests is an independent, exponentially distributed random variable with average $1/\lambda$.

1.3) An access request from any processor is directed to all the memory modules with the same probability $1/M$.

1.4) When a processor requests access to common memories, it requests for reading and writing in probabilities β and $(1-\beta)$ respectively.

1.5) When memory location conflicts occur, the conflicting request is rejected and the processor has to request again in an exponentially distributed random time, with average $1/\lambda$.

1.6) The location conflict occurs with probability $(1-\alpha)$.

[Notations]

N:	number of processors
M:	number of memory modules
B:	number of global buses
L:	number of copied memories in each memory module
S-M:	single memory module multiprocessor system
D-M:	distributed memory module multiprocessor system
C-M:	copied memory module multiprocessor system
N*(M-L)*B:	Multiple bus multiprocessor system consisting of N processors M memory modules L copies in each memory module and B global buses
$1/\mu$:	average access time
λ:	average request rate
$(1-\alpha)$:	probability that location conflicts occur
β:	probability that an access requests reading operations.
ρ:	the ratio: λ/μ
λ_i:	transition rate from state i to state (i-1) in the approximate Markov chain.
$\mu_i=\mu_i * \mu$:	transition rate from state i to state (i+1) in the approximate Markov chain.
S:	system state of the lumped Markov chain
S_a:	system state of the approximate Markov chain
P_n:	probability that system is in state n in the approximate Markov chain
P:	processing power

Because of the complexity of the performance evaluation of L-copied memory modules system, we analyze the performance of 2-copied memory module system denoted by N*(M-2)*B and compare the results with those of S-M and D-M systems(as shown in Fig. 2 and Fig. 3 respectively) obtained in [2] and [3]

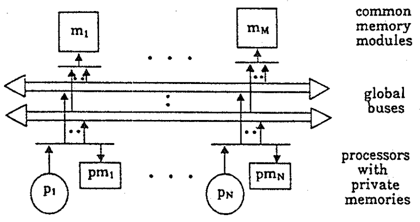

Fig. 2 The model of S-M systems.

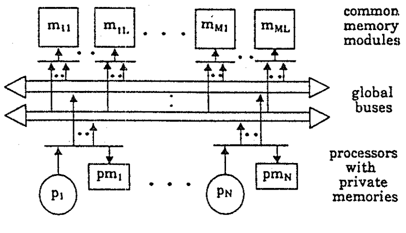

Fig. 3 The model of D-M systems.

3.Performance Evaluation----Exact Analysis

Processing power is the performance measure which also has been used in [1] and [2]. Processing power is defined as follows:

[Definition 3.1]

Processing power=[the average number of active processors] ■

[Definition 3.2] To obtain a Markov chain and to reduce the number of system states as small as possible, we define lumped system state as :

$$S=(S_1, S_2, S_3)$$

where

$S_1=(S_{11}, S_{12}, S_{13}, S_{14}, S_{15})$

$S_{11}=n_{r_1}$ $S_{12}=n_{r_2}$ $S_{13}=n_{w_1}$

$S_{14}=n_{r_1w_1}$ $S_{15}=n_{r_2w_1}$

$S_2=(S_{21}, S_{22}, \cdots, S_{25})$

$S_{21}=n^{(1)}_{w_1|q_m|w}, \cdots, n^{(i1)}_{w_1|q_m|w}$

$S_{22}=n^{(1)}_{r_2|q_m|r}, \cdots, n^{(i2)}_{r_2|q_m|r}$

$S_{23}=n^{(1)}_{r_1w_1|q_m|w}, \cdots, n^{(i3)}_{r_1w_1|q_m|w}$

$S_{24}=n^{(1)}_{r_2w_1|q_m|w}, \cdots, n^{(i4)}_{r_2w_1|q_m|w}$

$S_{25}=n^{(1)}_{r_2w_1|q_m|r}, \cdots, n^{(i5)}_{r_2w_1|q_m|r}$

$S_3=(S_{31}, S_{32}, \cdots, S_{37})$

$S_{31}=n^{(1)}_{r_1|q_b|r}, \cdots, n^{(j1)}_{r_1|q_b|r}$

$S_{32}=n^{(1)}_{|q_b|r}, \cdots, n^{(j2)}_{|q_b|r}$

$S_{33}=n^{(1)}_{|q_b|w}, \cdots, n^{(j3)}_{|q_b|w}$

$S_{34}=n^{(1)}_{r_1|q_b|w}, \cdots, n^{(j4)}_{r_1|q_b|w}$

$S_{35}=n^{(1)}_{r_2|q_b|w}, \cdots, n^{(j5)}_{r_2|q_b|w}$

$S_{36}=n^{(1)}_{w_1|q_b|r}, \cdots, n^{(j6)}_{w_1|q_b|r}$

$S_{37}=n^{(1)}_{r_1w_1|q_b|r}, \cdots, n^{(j7)}_{w_1r_1|q_b|r}$

where $i1 \sim i5 \leqq B$

$j1 \sim j7 \leqq M$

$n_{r_jw_k}$: number of processors accessing the j-read k-write memory module.

$n^{(i)}_{q_m|r_jw_1|w}$: number of processors waiting to write into the ith j-read 1-write memory modules because of memory module conflicts

$(j=0, 1, 2)$

$n^{(i)}_{q_m|r_2w_k|r}$: number of processors waiting to read from the ith 2-read k-write memory modules because of memory module conflicts.

$(j=0, 1, 2 \quad k=0, 1)$

$n^{(i)}_{q_b|r_j|w}$: number of processors waiting to write into the ith j-read memory module because no bus is available.

$(j=0, 1, 2)$

$n^{(i)}_{q_b|r_jw_k|r}$: number of processors waiting t read from the ith j-read k-write memory module because no bus is available.

$(j=0, 1 \ ; \ k=0, 1)$

$(n_{r_0w_1}=n_{w_1}, n_{r_1w_0}=n_{r_1}$ etc.) ■

For example, $S_{14}=n_{r_1w_1}=4$ means that there are two memory modules both of which are being read by a processor and being written by another processor. $S_{23}=n^{(1)}_{r_1w_1|q_m|w}=2$ means that there is a 1-read 1-write memory module and two processors are waiting for this memory module because of the memory module conflict. $S_{37}=n^{(1)}_{r_1w_1|q_b|r}$, $n^{(2)}_{w_1r_1|q_b|r}=1, 1$ means that there are two 1-read 1-write memory modules and a processor is waiting for one of them and another processor is waiting for the other one.

4.Performance Evaluation----Approximate Analysis

4.1 Approximate Solution with Closed Form

Even in the lumped markov chains, the number of system states increases rapidly as system size becomes larger. For example, a 3*(2-2)*2 system has 26 states, but in 4*(2-2)*2 system the number of system states increases to 59 only by adding one processor. The explosive growth is due to the detailed information that the state must includes about the queues inside the system.

In order to reduce the number of system states and to obtain a closed form solution, we construct an approximate Markov chain in which the system states include very little information about system queues. This approximate markov chain has smallest number of states and is suitable especially for calculating the processing power.

[Definition 4.1]

System state for the approximate chain is defined as

$$S_a=(n)$$

where n is the number of active processors. ■

The state transition rate is given by

$$\lambda_n=n\,\lambda$$

$$\mu_n=\mu\mu^*_n=\mu\Big(\sum_{i+j=N-n}\beta^i(1-\beta)^j\sum_{i1+i2+i3=i}a\,b\Big)$$

where

$$a=\begin{cases}(i2+i1+j1) & ; \ i2+i1+j1\leqq B\\ B & ; \ i2+i1+j1>B\end{cases}$$

$$b=p_{i2}(i3)\,p_{j1}(j2)/p_M(i)\,p_M(j)$$

where

i: the number of processors which are reading or are waiting to read

j: the number of processors which are writing or are waiting to write

i1: the number of processors reading from 1-read memory modules

i2: the number of processors reading from 2-read memory modules

i3: the number of processors waiting to read from two-read memory modules

j1: the number of processors which are writing

j2: the number of processors waiting to write into one-write memory modules

$p_k'(n)$: the number of unordered partitions of n into k parts,with k and n integers.

μ_n is the transition rate from state n to state (n+1) and is obtained by taking the average of the transition rates for all the possible states of the (N-n) non-active processors.

For simplicity, in calculating μ_n, the memory location conflict is ignored. In fact, the probability that memory location conflict occurs is generally very small.

We have reduced the system description to a birth and death Markov chain, whose solution is easily obtained. Denote by P_n the steady state probability of state n, then

$$P_n = P_N \prod_{i=n}^{N-1} \lambda_{(i+1)}/\mu_i = P_N \prod_{i=n}^{N-1} (i+1)\lambda/\mu_i$$

where

$$P_N = 1/(1 + \sum_{n=0}^{N-1} \prod_{i=n}^{N-1} (i+1)\lambda/\mu_i)$$

The expression of the processing power P is then as follows:

$$P = \sum_{n=0}^{N} nP_n = \sum_{n=0}^{N} \{(\lambda/\mu)^{N-n}(N!/n!) \prod_{i=n}^{N-1} 1/\mu_i^*\}/$$

$$\{1 + \sum_{n=0}^{N-1} (\lambda/\mu)^{N-n}(N!/n!) \prod_{i=n}^{N-1} 1/\mu_i^*\}$$

The approximate approach has greatly reduced the number of system states. For example, a 3*(2-2)*2 system has 26 states in the case of lumped Markov chain but only 4 states in approximate approach. A 4*(2-2)*2 system has 59 states in the case of lumped Markov chain but only 5 states in the approximate approach.

4.2 Computer Simulation

To conform our approximate solution, we compare the approximate results with those of exact solution for small number of system components. To test the performance of the approximate techniques on large systems, a simulation program was written for general N*(M-2)*B systems.

Exact and approximate analytic results were compared by considering 5*(2-2)*2 system as shown in Fig. 4. We can see that the difference between the approximate and the exact results is very small.

Fig. 5.1, Fig. 5.2 and Fig. 5.3 are the approximate results and simulation results of large system size for different number of processors, memory modules, and global buses respectively.

It can be seen that the results of the approximate solution and the results of the simulation approach are very close in any case. This conforms our approximate solution.

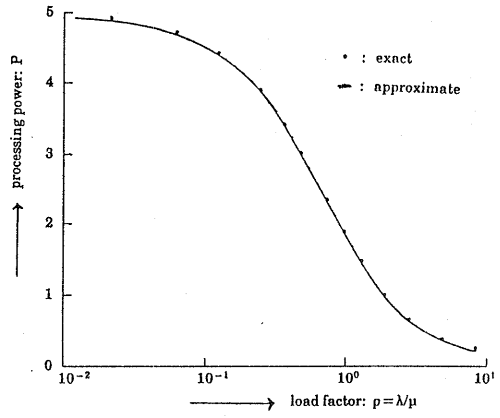

Fig. 4. The comparison of approximate results and Exact results in the case of 5*(2-2)*2 C-M systems

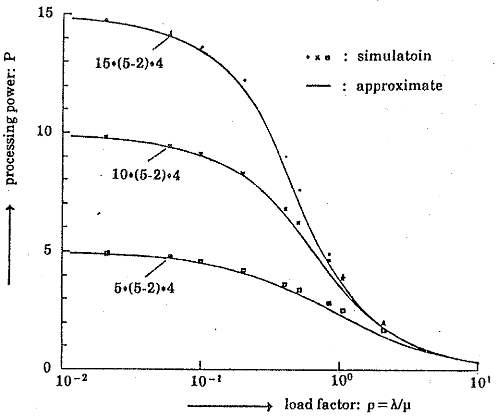

Fig. 5.1. The comparison of approximate results and simulation results in the case of 15*(5-2)*4, 10*(5-2)*4, and 5*(5-2)*4 C-M systems

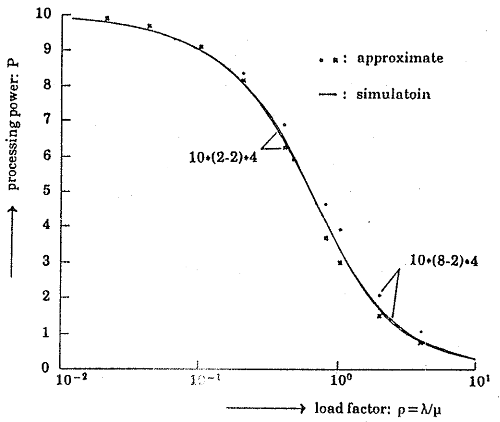

Fig. 5.2. The comparison of approximate results and simulation results in the case of 10*(2-2)*4 and 10*(8-2)*4 C-M systems

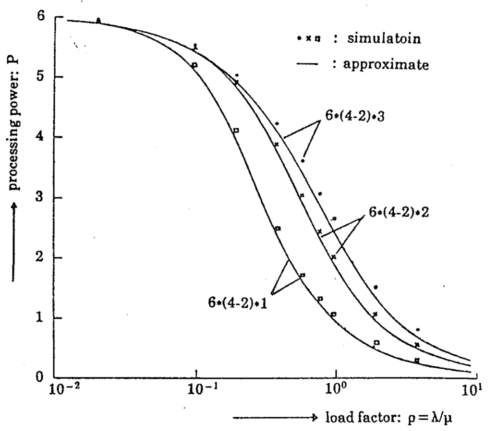

Fig. 5.3. The comparison of approximate results and simulation results in the case of 6•(4-2)•3, 6•(4-2)•2, and 6•(4-2)•1 C-M systems

5 Numerical Examples and Comparisons of the Three Systems: S-M, D-M and C-M

Fig. 6 shows the results of C-M system of different read probabilities β's. We find that $\beta=2/3$ is the optimum value at which the system obtains the best performance. This is because a memory module in our C-M systems permits two reads and one write accesses at the same time. If the read access and write access occur in the ratio 2:1, the memory module conflicts are the fewest. So we obtain the best performance at the point of $\beta=2/3$.

We compare the performance of the three systems: S-M, D-M and C-M (shown in Fig. 2, Fig. 3 and Fig. 1 respectively), by using our approximate results. Some examples are shown in Fig. 7.1(Table 1.1) and Fig. 7.2(Table 1.2).

Fig. 7.1(Table 1.1) and Fig. 7.2(Table 1.2) show the results of the three systems with the global bus conflict and the results of the three systems without the global bus conflic respectively. We observe that the maximum difference of processing power between D-M system and S-M system is 50%. But in our C-M system, we can obtain about 124% maximum improvement of processing power over S-M system when ρ is about 2. It can also be seen that even in the case with bus conflict, our C-M systems achieves improvement over S-M systems more than D-M systems can achieve.

6 Conclusions

In this paper, we have proposed a multiple bus multiprocessor system with
L-copied memory modules. We have evaluated the performance of the system for L = 2 and have compared the results with S-M systems and D-M system.

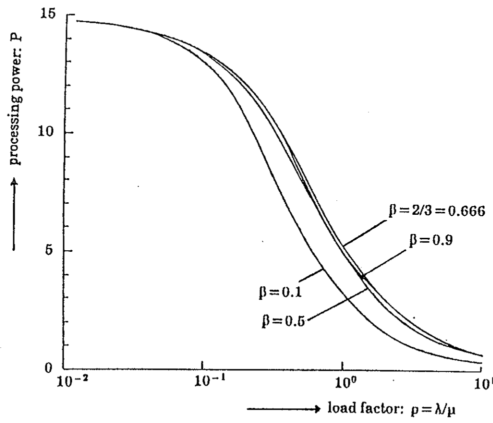

Fig. 6. Results of 15•(4-2)•12 C-M systems for different values of β's.

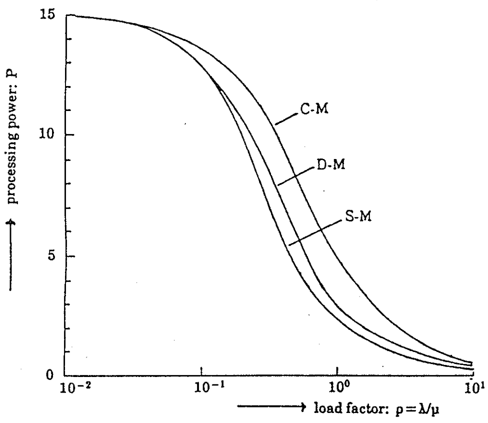

Fig. 7.1. The comparison of the three systems: S-M, D-M, and C-M in thr case of 15•(4-2)•12, without the global bus conflict.

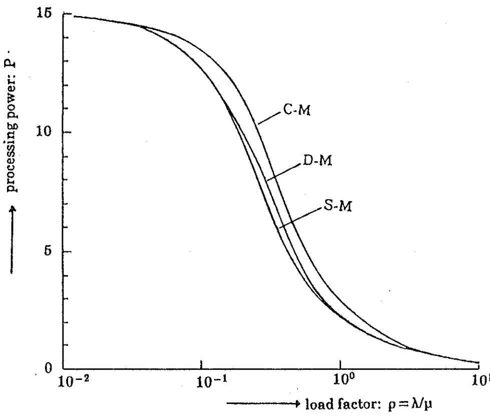

Fig. 7.2. The comparison of the three systems: S-M, D-M, and C-M in the case of 15•(4-2)•3, with the global bus conflict.

$\rho=\lambda/\mu$	$P_{S\text{-}M}$	$P_{D\text{-}M}$	$P_{C\text{-}M}$	$P_{D\text{-}M}-P_{S\text{-}M}$ (%)	$P_{C\text{-}M}-P_{S\text{-}M}$ (%)
0.1	12.75	12.75	13.50	0.00	5.88
0.2	9.79	10.60	12.05	8.27	27.68
0.4	5.80	7.20	9.58	24.13	41.03
0.6	3.75	5.15	7.38	37.33	96.80
0.8	2.80	3.60	5.75	28.57	105.35
1	2.25	2.90	4.85	28.88	115.55
2	1.25	1.57	2.80	25.60	124.00
4	0.70	1.00	1.50	42.85	114.28
6	0.50	0.75	1.02	50.00	104.00
8	0.45	0.60	0.80	33.33	77.77
10	0.40	0.5	0.60	25.00	50.00

Table 1.1 The comparison of the three systems: S-M, D-M and C-M in the case of 15•(4-2)•12, without the global bus conflict

$\rho=\lambda/\mu$	$P_{S\text{-}M}$	$P_{D\text{-}M}$	$P_{C\text{-}M}$	$P_{D\text{-}M}-P_{S\text{-}M}$ (%)	$P_{C\text{-}M}-P_{S\text{-}M}$ (%)
0.1	12.30	12.30	13.50	0..00	9.75
0.2	9.60	10.10	11.50	5.20	19.79
0.4	5.30	6.00	7.10	13.20	33.96
0.6	3.50	3.80	4.70	8.50	34.28
0.8	2.70	2.80	3.70	3.70	37.03
1	2.07	2.10	2.90	1.44	40.09
2	1.09	1.10	1.50	0.91	37.61
4	0.68	0.68	0.69	0.00	1.40
6	0.50	0.50	0.50	0.00	0.00

Table 1.2 The comparison of the three systems: S-M, D-M, and C-M in the case of 15•(4-2)•3, with the global bus conflict.

For small number of system components, we have obtained the exact solution and compared with that of S-M systems. For large number of system components, seeking for exact solution become very tedious because of the explosion of the system states. As a result, we have obtained an approximate solution which has not only very small number of system states but also can give closed form which is very desirable in analytical approach. Then we have compared the approximate results with those of exact solution for small number of system components, and with those of computer simulation for large number of system components. We have found that the difference between the approximate results and the exact results or simulation results is very small thus conforming our approximate solution.

We also have found that the read probability $\beta=2/3$ obtains the best performance for the systems with 2-copied memory modules.

By comparing the three systems: S-M, D-M, and C-M, we have also found that their performances can be ordered as: $P_{C\text{-}M} > P_{D\text{-}M} > P_{S\text{-}M}$.

We have evaluated multiprocessor systems consisting of memories with 1 read port and 1 write port, memory modules with 2-copied memories, and arbitrary number of processors, memory modules and global buses which we denote by s=(1,1,2,N,M,B). The problem of performance evaluation of multiprocessor systems consisting of memories with arbitrary number of read and write ports and memory modules with arbitrary number of copies(ie. s=(R,W,C,N,M,B)) remains as the future work.

REFERENCES

[1] Marco Ajmone Marsan and Mario Gerla, "Markov Models for Multiple Bus Multiprocessor Systems,"IEEE Trans. Comput., Vol.C-31, No.3, pp.239-248, (Mar. 1982).

[2] Marco Ajmone Marsan, Gianfranco Balbo, and Gianni Conte, "Comparative Performance Analysis of Single Bus Multiprocessor Architectures," IEEE Trans. Comput., Vol.C-31, No.12, pp.1179-1191, (Dec. 1982).

[3] M. A. Holliday and M. K. Vernon, "Exact Performance Estimates for Multiprocessor memory and Bus Interference," IEEE Trans. Comput., Vol.C-36, No.1, pp.175-189, (Jan. 1987).

[4] Leonard S. Haynes, Richard L. Lau, and Daniel P. Siewiorek, "A Survey of Highly Parallel Computing," IEEE Computer, pp.9-24, (Jan. 1982).

[5] Daniel D. Gajski and Jih-Kwon Peir, "Essential Issues in Multiprocessor Systems," IEEE Computer, pp.9-27, (June 1985).

[6] Peter C. Patton, "Multiprocessors:Architecture and Applications," IEEE Computer, pp.29-40, (June 1985).

[7] Keki B. Irani and Ibrahim H.Onyuksel, " A Closed-Form Solution for the Performance Analysis of multiple-Bus Multiprocessor Systems," IEEE Trans. Comput., Vol.C-33, No.11, pp.1004-1013, (Nov.1984).

[8] Joseph A. Fisher, "Trace Scheduling: A Technique for Global Microcode Compaction," IEEE Trans. Comput., Vol.C-30, No.7, pp.478-491, (July 1981).
[9] Tomas Lang, Mateo Valero, and Ignacio Alegre, "Bandwidth of Crossbar and Multiple-Bus Connections for Multiprocessors," IEEE Trans. Comput., Vol.C-31, No.12, pp.1227-1235, (Dec. 1982).
[10] Tse-Yun Feng, "A Survey of Interconnection Networks," IEEE Computer, pp.12-27,(Dec. 1981).
[11] Eli T. Fathi and Moshe Krieger, "Multiple Microprocessoer System: What, Why, and When," IEEE Computer, pp.23-32, (Mar.1 983).
[12] Sheldon S. and L. Chang, "Multiple-Read Single-Write Memory and Its Appliications," IEEE Trans. Comput., Vol.C-29, No.10, pp.689-694, (Oct. 1985).
[13] Chitar Das and Laxmi N. Bhuyan, "Bandwidth Availability of Multiple Bus Multiprocessors," IEEE Trans. Comput., Vol.C-34, No.10, pp.918-926, (Oct. 1985).
[14] Leonard Kleinrock, "Distributed Systems," Communications of the ACM, Vol.28, No.11, pp.1200-1213, (Nov. 1985).
[15] David A. Padua, David J. Kuck, and Duncan H. Lawrie, "High-Speed Multiprocessors and Compilation Techniques," IEEE Trans. Comput., Vol.C-29, No.9, pp.763-776, (Sep. 1980).
[16] M. J. Flynn and J. L.Hennessy, "Parallelism and Representation Problems in Distributed Systems," IEEE Trans. Comput., Vol.C-29, No.12, pp.175-189, (Dec. 1980).
[17] A. Gottlieb and others, "The NYU Ultracomputer--Designing an MIMD Shared Memory Parallel Computer," IEEE Trans. Comput., Vol.C-32, No.2, pp.175-189, (Feb. 1983).
[18] B. W. Wah, "A Comparative Study of Distributed Resourse Sharing on Multiprocessors," IEEE Trans. Comput., Vol.C-33, No.8, pp.700-711, (Aug. 1984).
[19] T. J. Leblanc and S. A. Friedberg, "HPC: A Model of Structure and Change in Distributed Systems," IEEE Trans. Comput., Vol.C-34, No.12, pp.1114-1129, (Dec. 1985).
[20] P. heidelberger and K. S Trivedi, "Queueing Network Models for Parallel Processing with Asynchronous Tasks," IEEE Trans. Comput., Vol.C-31, No.11 pp.1099-1109, (Nov. 1982).
[21] A. Nicolau and J. A. Fisher, "Measuring the Parallelism Available for Very Long Instruction Word Architectures," Trans. comput., Vol.C-33, No.11, pp.968-976, (Dec.1984).
[22] D. B. Gannon and J. V. Rosendale, "On the Impact of Communication Complexity on the Design of Parallel Numerical Algorithms," IEEE Trans. Comput., Vol.C-33, No.12, pp.1180-1194, (Dec. 1984).
[23] H. Kasahara and S. Narita, "Practical Multiprocessor Scheduling Algorithms for Efficient Parallel Processing," IEEE Trans. comput., Vol.C-33, No.11, pp.1023-1029, (Nov. 1984).
[24] K. G. Shin and J. C. Liu, "A Cost-Effective Multistage Interconnection Network with Network Overlapping and Memory Interleaving," IEEE Trans. Comput., Vol.C-34, No.12, pp.1088-1100, (Dec. 1985).
[25] C. P. Kruskal and M. Snir, "The Performance of Multistage Interconnection Networks for Multiprocessors," IEEE Trans. Comput., Vol.C-32, No.12, pp.1091-1098, (Dec. 1983).
[26] J. R. Goodman and C. H. Sequin, "Hypertre: A multiprocessor Interconnection Topology," IEEE Trans. Comput., Vol.C-30, No.12, pp.923-933, (Dec. 1981).
[27] D. R. Cox, "Renewal Theory," London: Methuen & Co Ltd, New York: John Wiley & Sons Inc., 1962.
[28] Charles H. Sauer and K. Mani Chandy, "Computer Systems Performance Modeling," Prentice-Nall, Inc.
[29] J. Laurie Snell and John G. Kemeny, "Finite Markov Chains," D. Van Nostrand Company, Inc., 1960.
[30] Leonard Kleinrock, "Queueing Systems," John Wiley & Sons, Inc., 1976.
[31] Y. Paker, "Multiprocessor systems," Academic Press 1983.