KFKI-1980-63

T.L. TÖRÖK
G. MESSING

SYSTEM-BUS LOAD INVESTIGATIONS

Hungarian Academy of Sciences

CENTRAL RESEARCH INSTITUTE FOR PHYSICS

BUDAPEST

KFKI-1980-63

SYSTEM-BUS LOAD INVESTIGATIONS

T.L. Török and G. Messing

Central Research Institute for Physics
H-1525 Budapest 114, P.O.B. 49, Hungary

HU ISSN 0368 5330
ISBN 963 371 699 3

ABSTRACT

The loadability of a tightly-coupled multiprocessor system with a common System-Bus is investigated by means of a population process. As against the classical network models the time parameter is discrete because the bus cycle time is of unit length. Since the state space turned out to be very large several approximations are given. Some states are lumped thus a process is defined and discussed using a less detailed state space. This procedure seems to be relevant to models other than just the present one.

АННОТАЦИЯ

В статье описывается исследование нагрузочных параметров многопроцессорной структуры, использующей системную магистраль. Описание структуры осуществлено с помощью "популяционного процесса". В отличие от классических моделей, описывающих сети, параметр времени имеет дискретный характер, что обьясняется дискретным значением времени цикла магистрали - единицы времени. Поскольку полученное пространство состояний огромно, дается несколько приближенных решений. Они заключаются в обьединении некоторых состояний, и на основе менее подробного пространства состояний исследуется определенный процесс. Данный метод имеет больше возможностей чем те, о которых упоминается в статье.

KIVONAT

Az osztott rendszer-buszt használó több processzoros struktura terhelhetőségét vizsgáljuk. A leirás egy populációs folyamattal történik. A hálózatokat leiró klasszikus modellekkel szemben az időparaméter diszkrét, amit a busz ciklusidő egységnyi volta indokol. Minthogy a kapott állapottér igen nagy, több közelitést adunk. Ez ugy történik, hogy bizonyos állapotokat összevonunk, és egy kevésbé részletes állapottéren definiált folyamatot vizsgálunk. Az eljárás tulmutat a cikkben kimeritett lehetőségeken.

INTRODUCTION

Increasing system throughput by using parallel processing techniques instead or besides endeavouring to increase the working speed of electronic components has become a noteworthy tendency in the computer design of recent years. The idea itself is not new but from the practical point of view it is only the achievements of the last decade's semiconductor technology that have given actual possibility, notwithstanding some earlier special implementations.

Parallel processing seems to be particularly suitable in real-time applications where tasks are, in general, sufficiently independent so that their separate treatment does not imply large organizational problems.

Systems forming a subset and capable of parallel processing form the tightly-coupled distributed systems. In such systems active system parts /processors/ have direct access not only to their local resources but to common resources as well. Access to common resources is maintained mostly via a commonly used, shared bus, the system-bus. As the access time of the most commonly used, resources /e.g. memory/ is of the same high order as the cycle-time of the bus, the system-bus may become the bottleneck of the whole system. In view of this the organization and the load of the system-bus are both of key importance. The point is to find the balance between the accessibility of system resources /i.e. the flexibility of the system/ and the load of the system-bus or, in other words, to establish a well balanced system based on local and common resources.

A well-proved method for bus load investigations is the simulation of the traffic on the bus. If appropriate codes are given and the particular loads of the processors along the bus are known; simulation can be performed successfully.

A principal problem arises when the necessary tools for the simulation are not given. It is no exaggeration to say that this problem is very much a real one since integrated processors and peripheral controllers are easily accessible and the implementation of self-made and problem-oriented system by customers has become possible.

As the investigation of the system-bus load is in no case neglectable an alternative method had to be found which is relatively easily applicable without creating too many other difficulties. If the number of active units on the bus, the distribution of their requests, and the duration of their bus--occupation are known, mathematical methods originally used for handling queuing problems, e.g. in computer networks, can be used.

In the following an attempt will be made both to calculate the load of a multiprocessor system-bus model in order to achieve to optimal balance of local and common resources and to develop a tool for handling Markov chains of large state space.

The first section contains the description of the computer system. The second one establishes the mathematical model in detail. Section 3 is the theoretical part of the investigation but it is pointed out that the reader need not become immersed in it if interested solely in the system optimization. A knowledge of this section is not essential to understanding the further details. Section 4 involves the performance evaluation of the model and answers the questions arising in the first section. Finally, Section 5 considers utilization.

I. SYSTEM REPRESENTATION

In connection with the development of a multi-microprocessor system a model is analysed. The model describes the system as follows. Processors of the system are placed along a commonly used bus, the System-Bus /*Fig. 1*/, which provides communication and data exchange between the processors, and between the processors and passive system parts /common resources: CR/ on the System-Bus. Some of the processors can be equipped with local bus facility where, if any, the local resources /*LR*/ of the processor are placed.

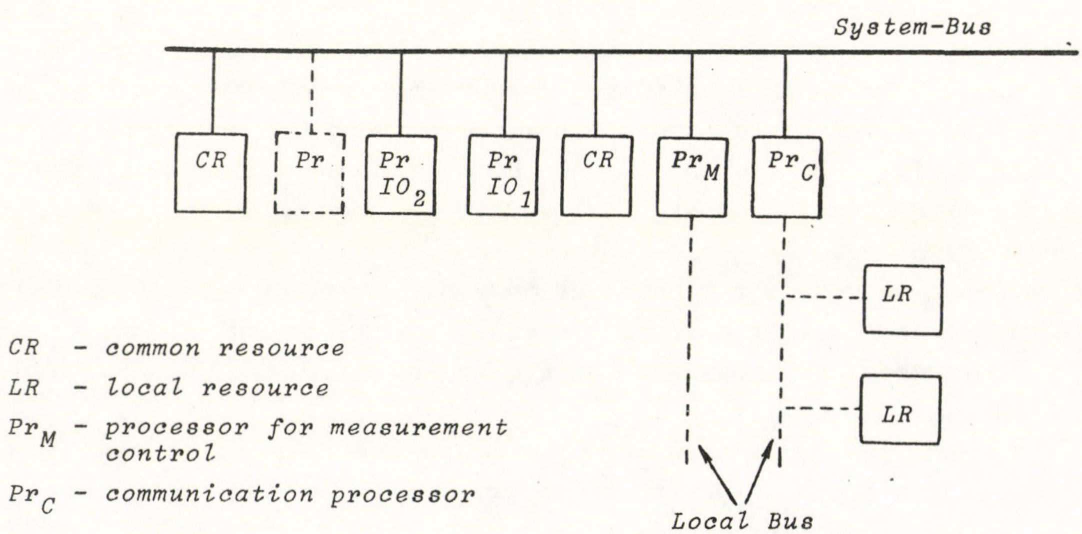


Figure 1

System-Bus structure

So far as the bus bargaining is concerned, the system is represented by a closed loop /*Fig. 2*/.

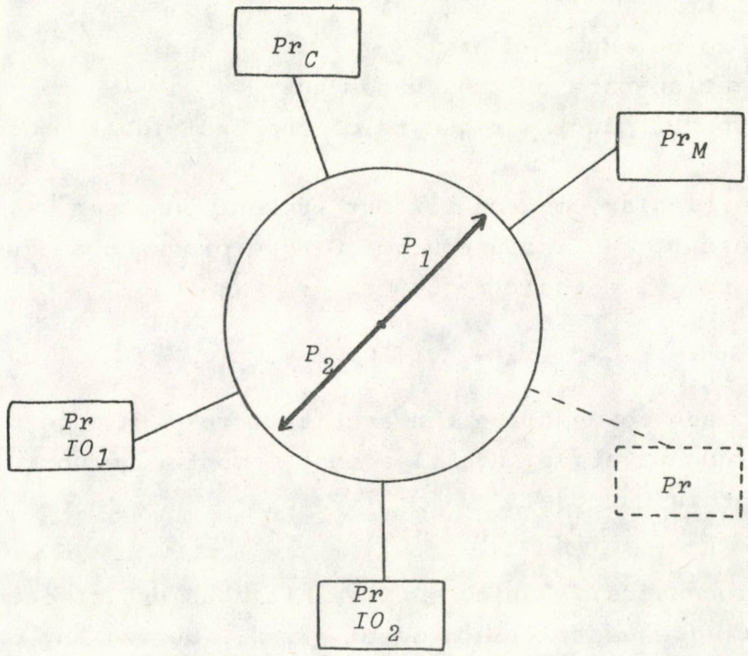


Figure 2

Bus bargaining structure

Bus cycles and bus bargaining are overlapped. Two asynchronously rotating pointers P_1 and P_2 point to the processors along the loop. P_1 enables bus requests and P_2 grants the request. Granted request means that the rotation stops and the processor waits until the bus becomes free from the bus-cycle currently in progress. Once the unit has had the request granted, it occupies the bus and the rotation of the pointers starts again and during the bus-cycle the next bus-master can be encoutered.

Parameters of the model are the number of processors on the bus and the rate of their bus occupation. The cycle-time on the bus has been taken as unity which does not differ much from reality.

Concerning bus requests the system implies two priority levels. As the rate of the higher level requests is more than two orders less than that of the lower level requests and as one granted request yields always only one bus-cycle, in the model only one priority level has been introduced.

To get closer to real circumstances some additional parameters are introduced: Two types of processors are distinguished, viz. the "R" type and the "Q" type. R-type processors may queue their bus requests /as is the case, for example, in some background storage processors/; Q-type processors are "halted" while their bus requests are pending. Further, we distinguish between models, where the time T between the termination of the bus-cycle of a

Q-type processor and the generation of the next cycle of the same processor is

$$\text{a/} \quad 0 \leq T \quad \text{and} \quad \text{b/} \quad 1 \leq T \;. \qquad /1.1/$$

The questions to be answered are

/i/ - the utilization rate of the system-bus;

/ii/ - the duration of pending requests of the individual processor for the system-bus;

and, of particular concern for our present purposes,

/iii/ - the decrease of the throughput of Q-type processors caused by their inability to queue their requests.

II. THE MODELS

An attempt is made to describe the architecture with a terminology which is appropriate for quantitative investigation. Computer network models seem to be suitable for this purpose.

Model 1 /M1/

We have N nodes /processors/. During a time unit /bus cycle/ each processor generates a customer /request/ with probability p_i. During any one time unit one simple request may be served. The service order is cyclic.
We are interested in the number of requests accumulated in the nodes in consequence of the occupied system-bus.

The described one is very similar to a Markov population model often used when evaluating a computer network. The more essential deviations are the following. We constructed a discrete model, i.e. there exists a time unit and each occurrence takes place during a multiple of it. Due to this the model fails to be ordinary bacause during a single time unit several events may occur. Service is not realized at the nodes but on another level common bus thus the service of one processor is not independent of the others.

Taking these facts into account let us define the following stochastic process of discrete time

$$\underline{\upsilon}^{(1)}(n) = (\upsilon_1^{(1)}(n), \upsilon_2^{(1)}(n), \ldots, \upsilon_N^{(1)}(n); k(n)) \qquad /2.1/$$

where $\upsilon_i^{(1)}(n)$ is the number of customers accumulated at moment n at the i-th; k(n) takes the values 0,1,2,...,N according to which node is served at moment n. It is easy to see that $\underline{\upsilon}^{(1)}(n)$ is a discrete time homogeneous Markov chain /MC/ of $\mathcal{N}$ state space. Since the accumulation of a certain number of requests in a single node may be fatal it is worthy investigating finite modifications where either the total population or the number in nodes is limited. This yields a finite model of rather large state space. The facultative reduction of certain states fails essentially to improve the situation.

Let therefore

$$\mu_n^{(1)} = \sum_{i=1}^{N} \upsilon_i^{(1)}(n) \qquad /2.2/$$

be the number of all requests. It is easy to see that $\mu_n^{(1)}$ is a Markov chain with transition probability matrix /*TPM*/

$$P\{\mu_n = k+\ell \mid \mu_{n-1} = k\} = \begin{cases} P\{\ell+1 \text{ arrivals}\} & \text{if } k>0 \text{ and } \ell \le N-1 \\ P\{\ell \text{ arrivals}\} & \text{if } k=0 \text{ and } \ell \le N \\ 0 & \text{if } \ell > N \end{cases} , \qquad /2.3/$$

where

$$P\{\ell \text{ arrivals}\} = \sum_{1 \le k_1 \le \ldots \le k_\ell \le N} \prod_{i=1}^{\ell} p_{k_i} \quad . \qquad /2.4/$$

This is a matrix of easily calculable elements. Disregarding the first row it is a Toeplitz type matrix in which case the determination of the stationary distribution /*SD*/ of $\mu_n^{(1)}$ meets hardly any difficulties.

Model 2 /M2/

For concrete purpose a more specific structure is discussed. Two types of processors are distinguished $\{Q_1, Q_2, \ldots, Q_N\}$ and $\{R_1, R_2, \ldots, R_K\}$. Since the Q-type processors fail to queue their requests the restriction

$$P\{\upsilon_i^{(2)}(n) \le 1\} = 1 \text{ if } i=1,\ldots,N \qquad /2.5/$$

is assumed. The distinction concerning whether the Q-type requests may be generated immediately after each other or not influences only the TPM. The investigation concentrates on the case T = 0. Further we will point out that the model with T > 0 considerably less than 1 does not cause a significant deviation from T = 0.
Thus the process

$$\underline{\upsilon}^{(2)}(n) = (\upsilon_1^{(2)}(n), \upsilon_2^{(2)}(n), \ldots, \upsilon_N^{(2)}(n), \ldots, \upsilon_{N+K}^{(2)}(n); k(n)) \qquad /2.6/$$

is defined. It is easy to see that $\underline{\upsilon}^{(2)}(n)$ is a Markov chain with $\mathfrak{n}$ state space. Calculation of the elements of its TPM is extraordinary tedious. Its SD is denoted by $\underline{\pi}^{(2)}$.

Let

$$\mu_n^{(2)} = \sum_{i=1}^{N+K} \upsilon_i^{(2)}(n) \quad .$$

In general this fails to be Markovian. The state space of $\mu_n^{(2)}$, however, arises from concentrating the states of $\upsilon^{(2)}(n)$ thus its SD is not without meaning. It can be determined from $\underline{\pi}^{(2)}$ by summing up the stationary probabilities of the concentrated states. Without calculating $\underline{\pi}^{(2)}$ the SD of $\mu_n^{(2)}$ is not evaluable thus approximations will be given in the following.

Let us formulate the questions in the first section using the terminology of the queueing theory. We are interested in

1. the utilization rate of the server /system-bus/;
2. the waiting time of the individual processors
3. the intensity of the real arrival process from the Q-type processors taking into account /2.5/.

III. MATHEMATICAL TOOLS

In the following a capital letter denotes a matrix and the same lower case letter with two indices refers to its elements.

At first an ordering relation on discrete probability distributions is introduced. The vector $\underline{p} = (p_o, p_1, \dots, p_n, \dots)$ is said to be greater than $\underline{q} = (q_o, q_1, \dots, q_n, \dots)$ $(\underline{p} \geq \underline{q})$ if

$$\sum_{i=0}^{n} p_i \leq \sum_{i=0}^{n} q_i \qquad /3.1/$$

for all $n = 1,2,\dots$ /cf. [2]/ loosely speaking it means that the random variable with $\underline{p}$ takes the larger values with larger probability than $\underline{q}$.

For finite distributions $\underline{p} = (p_o, p_1, \dots, p_n)$ $\underline{p_i} = 0$ $i>n$ makes the definition complete.

The Markov chain ξ_n is said to be greater than η_n $/\xi_n \geq \eta_n/$ if its SD is greater than that of η_n. This definition will be used for processes whose SD is interpreted.

The Markov chain is said to be monotonic if either $\xi_n \leq \xi_{n+m}$ or $\xi_n \leq \xi_{n+m}$ for all n and m. The monotonicity is not independent of the initial probability vector.

Theorem 1 /[2]/

If the ergodic MCs ξ_n and η_n have TPMs P and Q, respectively and one of them is monotonic then from

$$\sum_{j=0}^{k} p_{ij} \geq \sum_{j=0}^{k} q_{ij} \quad \text{for all } k = 1,2,\dots \qquad /3.2/$$

follows $\xi_n \leq \eta_n$.

In the following, condition /3.2/ will be referred to shortly as $P \leq Q$.

Theorem 2

Let ξ_n be an MC with state space S, with TPM P and SD $\underline{\pi}$. Let $A_1 \cup A_2 \cup \dots \cup A_n = S$ be a disjoint partition and let us define the quantities

$$q_{ij} = \sum_{\ell \in A_i} \pi_i \Big(\sum_{s \in A_i} \pi_s \Big)^{-1} \cdot \sum_{r \in A_j} p_{\ell r} \qquad /3.3/$$

In this case the solution $\underline{x}$ of the system $\underline{x} = \underline{x}Q$ satisfies

$$x_i = \sum_{\ell \in A_i} \pi_\ell \quad . \qquad /3.4/$$

Proof: The case $S = \{0,1,\dots,n\}$; $A_1 = \{0,1\}$, $A_i = \{i\}$ will be discussed.

Let B be given as follows:

$$b_{i1} = p_{i0}+p_{i1} \qquad \text{if } i \geq 2$$

$$b_{ij} = p_{ij} \qquad \text{if } i \geq 2 \text{ and } j \geq 2$$

and the values b_{1j} will be determined in the way that

$$\underline{x} = \underline{x}B \; . \qquad /3.5/$$

In the latest system b_{1j}-s are the variables and $x_1 = p_0+p_1$; $x_i = p_i$ are assumed to be known. If /3.5/ is supplemented with the condition $\sum_{j=1}^{n} b_{1j} = 1$ it is a correct problem.

Let us specify the system $\underline{\pi} = \underline{\pi}P$ and add the first two equations

$$\begin{array}{llll} \pi_0(p_{00}+p_{01}) & +\pi_1(p_{10}+p_{11}) & +\dots+\pi_n(p_{n0}+p_{n1}) & = \pi_0+\pi_1 \\ \pi_0 \cdot p_{02} & +\pi_1 \cdot p_{12} & +\dots+\pi_n \cdot p_{n2} & = \pi_2 \\ \vdots & \vdots & \vdots & \vdots \\ \pi_0 \cdot p_{0n} & +\pi_1 \cdot p_{1n} & +\dots+\pi_n \cdot p_{nn} & = \pi_n \end{array} \qquad /3.6/$$

Subtracting /3.5/ from /3.6/ we have

$$\begin{array}{lll} \pi_0(p_{00}+p_{01}) & +\pi_1(p_{10}+p_{11}) & =(\pi_0+\pi_1)b_{11} \\ \pi_0 \cdot p_{02} & +\pi_1 \cdot p_{12} & =(\pi_0+\pi_1)b_{12} \\ \vdots & & \\ \pi_0 \cdot p_{0n} & +\pi_1 \cdot p_{1n} & =(\pi_0+\pi_1)b_{1n} \end{array}$$

From this we get the values b_{1i}. The coincidence of Q and B is evident.

The sense of the proof is completely similar for an arbitrary partition.

REMARK 1 /3.3/ is a convex linear combination of the rows of matrix P corresponding to the set A_i where the weight of the rows is proportional to the stationary probabilities of the corresponding state.

REMARK 2 The above theorem is a generalization of a classical result [1] namely if

$$\sum_{r \in A_j} p_{\ell r} = c_j^{(i)} \quad \text{for } \ell \in A_i \qquad /3.7/$$

does not depend on ℓ then $q_{ij} = c_j^{(i)}$. /We have an arbitrary combination of identical elements./

Theorem 3

Let $A_o = \{0,1,\ldots,m\}$; $A_i = \{m+i\}$ if $1 \leq i \leq n-m$ be a partition and $\underline{b}^{(i)} = \{p_{io}, p_{i1}, \ldots, p_{in}\}$ $0 \leq i \leq m$. Let $\underline{u} = \max_i\{\underline{b}^{(i)}\}$; $\underline{v} = \min_i\{\underline{b}^{(i)}\}$* and the notation of Th.2 supplemented with

$$u_{ij} = p_{i+m\ j+m} \text{ if } i,j > 0; \quad u_{oo} = \sum_{i=0}^{m} u_i, \quad u_{oj} = u_j, \quad u_{io} = \sum_{j=0}^{m} p_{ij}$$

$$v_{ij} = p_{i+m\ j+m} \text{ if } i,j > 0; \quad v_{oo} = \sum_{i=0}^{m} v_i, \quad v_{oj} = v_j, \quad v_{io} = \sum_{j=0}^{m} p_{ij}$$

we have $\underline{v} \leq \underline{x} \leq \underline{z}$ if it is assumed that U and V are monotonic** where

$$\underline{y} = \underline{y}V \quad \text{and} \quad \underline{z} = \underline{z}U \quad . \qquad /3.8/$$

Proof: The first row of Q is a convex linear combination of vectors $\underline{b}^{(i)}$ with unknown weights. It is easy to see that an arbitrary combination of this kind is between the maximum and minimum of the basis vectors. Thus $V \leq Q \leq U$ and from Th.1 the statement follows.

Finally some guiding principles are given for solving fixed point problems of large - occasionally ∞ - matrices. This tedious procedure consists of the following steps.

1. Generating the possible states and ordering them into a vector /S = = {0,1,...,N,...}/. If we have too many of them a limitation is needed as detailed below.

2. Enumerating the TPM. A sparse one is organized into a list form.

3. The SD is approximated by iteration.

The limitation has two possibilities

1. Some states are neglected /the state space will be S' = {0,1,...,N} instead of S/ and iteration is executed by a truncated /substochastic/ matrix.

2. Some states are concentrated /S' = {0,1,...,N-1}∪{N} instead of S/. Transition probabilities to N are determined to obtain a stochastic matrix. Transition from N are more difficult. Two extreme cases are considered

$$P\{N \to N\} = 1-\varepsilon \qquad \text{- N is nearly an absorbing state if } \varepsilon \text{ is small} \qquad /3.9/$$

$$P\{N \to 0\} = 1 \qquad /3.10/$$

The real case is somewhere between /3.9/ and /3.10/ if ε is small enough.

---

* max and min are to be understood in the sense of /3.1/.

** This means that the corresponding MCs are monotonic.

IV. MODEL DISCUSSION DEMONSTRATED BY EXAMPLES

4.1 Some bounds on the stationary distributions

The numerical evaluation of some specific models is executed. Architectures of four processors are considered with Q_1, Q_2 and R_1, R_2. The corresponding values of p_i are

	p_1	p_2	p_3	p_4
Example 1 /E1/	.35	.35	.1	.05
E2	.4	.4	.1	.05
E3	.45	.45	.1	.05
E4	.5	.5	.1	.05
E5	.6	.6	.1	.05

/4.1/

The stationary distribution of $\mu_n^{(1)}$ in the model M1 is

$P\{\mu = k\}$	0	1	2	3	4	5	6
E1	.15	.254	.219	.139	.086	.053	.033
E2	.065	.145	.162	.139	.117	.098	.081
E3	DOES						
E4			NOT				
E5					EXIST		

/4.2/

It is obvious that the SD $\underline{x}$ of $\mu_n^{(2)}$ is less than that of $\mu_n^{(1)}$ in the sense of /3.1/. This approximation coincides with $\underline{z}$ in /3.8/. V in /3.8/ is easily determined by $p_4 = \min p_i$ and from Th.3.

	y_o	y_1	y_2	y_3	y_4
E1	2607	.4605	.2463	.0319	.0006
E2	2037	.4577	.2975	.0403	.0008
E3	1550	.4441	.3506	.0495	.0011
E4	1138	.4184	.4067	.0597	.0014
E5	0557	.3512	.5116	.0788	.0021

/4.3/

Before further investigation the solution of $\underline{\upsilon}_n^{(2)}$ will be determined and thus the exact evaluation of $\mu_n^{(2)}$ can be obtained. This lengthy procedure is sketched at the end of Section 3.

The state space of $\upsilon_n^{(2)}$ is $\mathcal{N}$, thus the mentioned limitation is needed. If the total population

$$/\sum_{i=1}^{K+N} \upsilon_i^{(2)}(n) = M/$$

is supposed as being $M \leq 5$ the deviation of /3.9/ and /3.10/ is 10^{-4} if $\varepsilon=.001$. After summation we get

	$P\{\mu_2 = 0\}$	=1	=2	=3	=4	>4	
E1	.2473	.4369	.2538	.0554	.0059	.0006	
E2	.1899	.4267	.3011	.0726	.0085	.0010	
E3	.1412	.4061	.3474	.0915	.0115	.0015	/4.4/
E4	.1026	.3769	.3918	.1115	.0149	.0024	
E5	.0479	.3017	.4703	.1533	.0226	.0037	

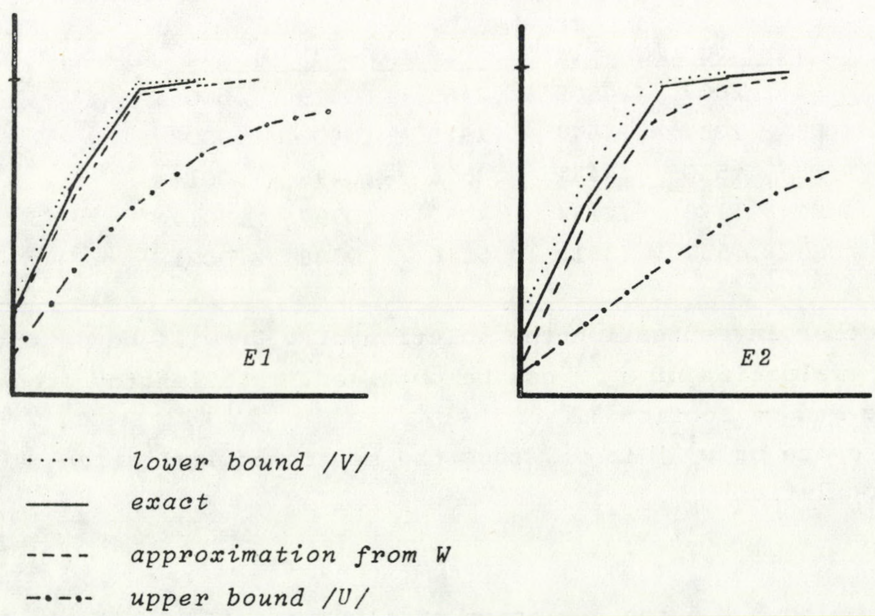
The above results are from a very tedious calculation. This is illustrated by the table

	M= 2	3	4	5
number of states	20	46	84	134
number of probab.	400	2116	7506	17956
positive probab.	148	386	693	1153
rate of saturation	37%	18,2%	9,8%	6,4%

If the exact values /4.4/ are compared with the upper /4.2/ and lower bounds /4.3/ the latter turns out to have a better fitting. This is not surprising since the requests for Q-type processors arise much more frequently thus they are busy more frequently than the R-type ones. This supports heuristically that $W = \frac{1}{2}(U+V)$ is an upper bound too:

$$\underline{x} \leq \underline{w} \qquad \text{where} \qquad \underline{w} = \underline{w}W \ .$$

The bounds and correct values are compared in the figures below.



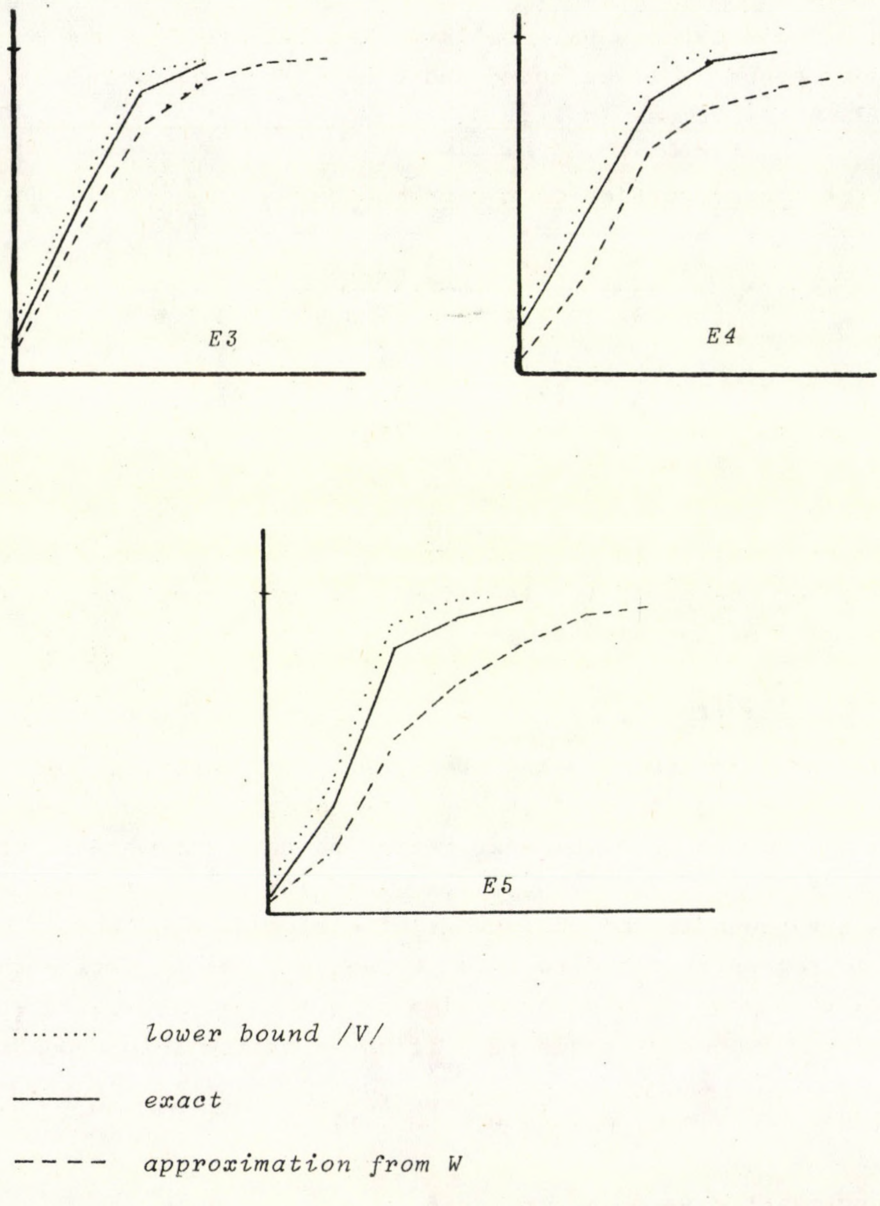


Figure 3

The cumulative distribution for the examples E1-E5

4.2 Answers to questions

Based on the above facts let us try to answer the questions raised in Section 2.

The following notations will be used: The letters U, V and W always refer to an upper bound, a lower bound and a heuristic approximation, respectively, based on the values in *Fig. 3*.

1. Utilization proportion of the common bus

	U	W	Exact	V
E1	85 %	78,4 %	75,3 %	73,9 %
E2	95	83,1	81	75,6
E3	100	88,2	85,8	84,5
E4	100	92,8	89,7	88,6
E5	100	97,3	95,2	94,4

/4.5/

The real utilization is slightly greater than we obtained because of the asynchronicity but the deviation is not significant. More precise discussion needs further data on the working.

2. Mean waiting time

Correct discussion is not easy. We have to discriminate the waiting time of different processors for the bus /W_i/. The exact procedure based on the SD of $\upsilon^{(2)}$ does not result in exact information on this parameter. For example, if the system is in the state /1,0,0,2;4/ then the second request at R_2 may wait 2,3,4 units depending on the number of arrivals to Q_2 and R_1 during the service of the requests preceding it at Q_1 and R_2. Let us define the stochastic variables T_i and Q_i /the sojourn time of a request generated at the i-th processor and the number of requests - if any - at the i-th processor/. This means

$$P\{T_i = k\} = P\{W_i = k-1\}$$

since the service time is equal to 1 and

$$P\{Q_i = k\} = \frac{P\{\upsilon_i^{(2)} = k\}}{P\{\upsilon_i^{(2)} > 0\}}$$

It is obvious from the above example that $Q_i \leq S_i$ in the sense of /3.1/. On the other side $\upsilon_i^{(2)}$ is majorized by $\mu^{(2)}$. The deviation between Q_i and T_i seems to be less than that of $\upsilon_i^{(2)}$ and $\mu^{(2)}$. /The former concerns only some states mostly of small probability./ These are summarized in the following table for E1.

i	$E(Q_i)$	$E(T_i)$	$E(Q)$	B_1	B_2
1	1,43	1,449			
2	1,43	1,455	1,51	1,675	1,779
3	1,69	1,739			
4	1,68	1,73			

/4.6/

where Q is the total number of requests, if any

$$P\{Q = k\} = \frac{P\{\mu^{(2)} = k\}}{P\{\mu^{(2)} > 0\}}$$

is derived from the SD of $\upsilon_n^{(2)}$; B_1 and B_2 are bounds for E(Q) derived from the approximations W, U and W, V.

$$E(W_i) \leq \frac{1}{2} \quad /i = 1,2/ \quad \text{and} \quad E(W_i) \leq 1 \quad /i = 3,4/$$

seem to be persuasive from /4.6/ and they are not unfavourable.

An other approximation will be given by means of the throughput.

3. The throughput

The utilization of the single processors of Q-type decreases because of the restriction /2.5/. This is investigated for E1 based on the M2 $\upsilon_n^{(2)}$. The probability that the i-th processor is busy is as follows

i=	1	2	3	4
p_{B_i}	.4377	.4387	.1723	.0857

/4.7/

In the same way the table

i=	1	2	3	4
p_{S_i}	.3010	.3015	.1	.05

/4.8/

gives the probability that the i-th processor is being served. This characterizes the output rate /throughput/. Since these procedures are based on the exact model of rough calculation, approaching possibilities are needed. The model M1 is evaluated with different values of $p_1 = p_2$. If these distributions are compared with those from the approximations W

	E_1	.30	E_2	.33	E_3	.35	E_4	.37
P{.=0}	.3100	.3780	.3447	.4220	.4404	.4780	.5112	.5360
=1	.4332	.3794	.4265	.3769	.4060	.3657	.3749	.3461
=2	.1868	.1683	.1562	.1477	.1263	.1219	.0970	.0967
=3	.0513	.0523	.0339	.0395	.0227	.0273	.0142	.0177
=4	.0153	.0154	.0065	.0100	.0035	.0056	.0018	.0029
=5	.0034	.0045	.0014	.0025	.0006	.0011	.0003	.0004
=6	.0007	.0013	.0004	.0006	.0001	.0002	.00008	.00008

/4.9/

a rather good fit is found - especially for the larger values of the variable. Thus the estimated decrease of the throughput is compared with the correct values /obtained from $\upsilon_n^{(2)}$ /.

	E_1	E_2	E_3	E_4
correct	0.301 /85,7%/	0.331 /82,7%/	0.354 /78,7%/	0.374 /74,8%/
estimated	0.30	0.33	0.35	0.37

The real throughput of the two Q-type processors /4.10/

The real throughputs of the two Q-type processors have a deviation of approximately 10^{-3} /Q_1 is .3010 and Q_2 is .3015 if $p_1 = p_2 = .35$/. This is not by chance. The worst circumstance of Q_2 is because of its position after Q_1. If an R-type node reserves the bus both of the Q-types can generate a request. If both of them do it Q_1 is always the first to be served. Therefore, it is in a better position. The deviation is extremely small because of the relatively small values of p_3 and p_4.

It seems to be worth mentioning that the decreasing of the throughput is connected with the occurrence of waiting. Loosely speaking decreasing from /4.1/ to /4.7/ reflects the blocking of processors Q_1, Q_2. Decreasing from /4.7/ to /4.8/ is because of the waiting for the bus reserved by others. Therefore it is easy to see that

$$E(T_i):1 = p_{B_i} : p_{S_i} \qquad /4.11/$$

This gives for E1

i	1	2	3	4
$E(T_i)$ from /4.6/	1.449	1.455	1.739	1.73
$E(T_i)$ from /4.11/	1.454	1.455	1.729	1.714

/4.12/

Another possibility is obtained for estimating $E(T_i)$. It is obvious that $p_3 = p_{S_3}$; $p_4 = p_{S_4}$.
Thus the values p_{S_i} are approximated from /4.9/ and /4.12/. It is obvious that

$$p_i \geq p_{B_i} \geq p_{S_i} \quad \text{if} \quad i = 1,2 \quad \text{and} \quad p_3 \leq p_{B_3}, \quad p_4 \leq p_{B_4} \quad .$$

From these

$$E(T_1) \leq \frac{p_1}{p_{S_1}} = 1,45; \qquad E(T_2) \leq \frac{p_2}{p_{S_2}} = 1,452 \quad .$$

The bounds for $E(T_3)$, $E(T_4)$ derived in a similar way need several inequalities and are worse than in /4.6/.

Finally the deviations caused by the different values of T are summarized only for the exact E1 model

	T=1	T=.1	T=0
$P\{\mu_2=0\}$	.3663	.2480	.2473
=1	.4513	.4383	.4369
=2	.1589	.2529	.2538
=3	.0220	.0542	.0554
=4	.0014	.0058	.0060
4	.0006	.0006	.0006
p_{S_i}	.2414 /68,8%/	.301 /86%/	.302 /86,14%/
p_{B_i}	.3052	.439	.438

/4.13/

V. UTILIZATION OF RESULTS

When implemented the system will contain with all its extensions four processors, one for measurement control with $p_1 = p_M = 0.35$, one communications processor with $p_2 = p_C = 0.35$ /both Q-type processors/ and two input--output processors of R-type, with $p_3 = p_{IO_1} = 0.1$ and $p_4 = p_{IO_2} = 0.05$.

In utilizing the results two effects have to be considered, viz. the influence of the minimum distance /T_{min}/ between consecutive bus cycles of the same processor /cf. /1.1//; the increase in throughput as a consequence of applying local resources.

The first effect can be influenced by the appropriate choice of the bus arbitration system. Table /4.12/ shows alteration of the throughput as the consequence of T_{min} between two consecutive pulses. One can see that if $T_{min} = 1$, the throughput of the Q-type processor decreases to 68,8% against the 86,14% for $T_{min} = 0$. The difference is considerable. The bus arbitration used by the system comes very near to the model of $T_{min} = 0$, as small divergences of the limit T = 0 do not affect significantly the throughput /see table /4.12/ $p_{S_1} = p_{S_2} = 86\%$ instead of 86,14% by $T_{min} = 0$/.

So far as the local memories are concerned the following should be noted:

Without using local memories, i.e. if the system-bus is loaded by the whole traffic of the four processors, the throughput of a Q-type processor decreases according to table /4.10/ to 86% of the optimum value /which corresponds to the sojurn time S = 1/.

As communication jobs are relatively independent of other system activities local memory can be used as a communication program store and p_2 takes the approximate value $p_C' = 0.05$. If the calculations described in Section 4 are performed the throughput for the "M" processor becomes 0.3313 /94,6%/ and for processor "C" 0.0491 /98,2%/. It is obvious that the figures encountered for the throughput are normed to one processor.

As the most affected processor M represents only a part of the whole system activity the system throughput will become approximately 90% of the ideal case. The difference in speed is not large, the increase in performance, however, in advantageous.

VI. CONCLUSIONS

The method described is used for the investigation of system-bus load of tightly-coupled multiprocessor systems. It does not replace the simulation of the system completely as the accuracy of input parameters obviously influences the results, but it does give very useful preliminary information on system--bus load at the design stage before simulation is possible. Though results have not as yet been verified in practice they seem to provide a good fit to real circumstances. Utilization of the method enables an optimum proportion of local and common resources to be established.

REFERENCES

[1] Burke, C.M.- M. Rosenblatt: A Markovian function of a Markov chain, Ann. Math. Stat., 29, 1112-1122 /1972/

[2] Stoyan, D.: Über einige Eigenschaften monotoner stoshastischer Prozesse, Math. Nachr., 52, 21-34 /1972/

[3] Messing, Gy.: CAMAC-based microcomputer system for data acquisition, Conf. on Real-Time Data, Berlin, 1979. ed. H. Meyer, North-Holland, 1980.

Kiadja a Központi Fizikai Kutató Intézet
Felelős kiadó: Sándory Mihály
Szakmai lektor: Csákány Antal
Nyelvi lektor: Harvey Shenker
Példányszám: 290 Törzsszám: 80-531
Készült a KFKI sokszorosító üzemében
Budapest, 1980. szeptember hó