Queuing Theoretic Model for a Multiprocessor With Private Caches and Shared Memory

Angel V. Nikolov
Department of Mathematics and Computer Science, National University of Lesotho, angel_nikolov10@yahoo.com

Abstract

We develop an analytical model of multiprocessor with private caches and shared memory and obtain the steady-state probabilities of the system. Behaviour in equilibrium can be studied and analyzed. We show that results can be applied to determine the output parameters for both blocking and non-blocking caches.

Keywords: *Invalidate cache-coherence protocol, queuing system, discrete transform*

Introduction

Shared memory multiprocessors are widely used as platforms for technical and commercial computing [2]. Performance evaluation is a key technology for design in computer architecture. The continuous growth in complexity of systems is making this task increasingly complex [7]. In general, the problem of developing effective performance evaluation techniques can be stated as finding the best trade-off between accuracy and speed.

The most common approach to estimate the performance of a superscalar multiprocessor is through building a software model and simulating the execution of a set of benchmarks. Since processors are synchronous machines, however, simulators usually work at cycle-level and this leads to enormous slowdown [9]. It might take hours even days to simulate.

For memory structures relatively accurate analytical models are developed [3, 7, 8, 9] through extensive use of various queuing systems. Open queue system with Poisson arrivals and exponential service times is considered quite good for description of memory hierarchies [7]. Our focus is on the impact of the cache-coherence protocols on the overall system performance. The most commonly used technique for this purpose is the Mean Value Analysis (MVA) [3, 5, 7, 8, 9]. It allows the total number of the customers to be fixed (closed queue system), and this seems to be more adequate representation of the processes of self-blocking requestors [5]. Calculations of output parameters such as residency times, waiting times and utilization are shown in [3, 8, 9]. MVA suggests exponential service times but in fact both bus cycle times and memory access times are close to constants. It will be seen later in this paper that state probabilities depend on the server's time density function.

We assume general distribution of the service times and introduce the supplementary variable x, elapsed service time, to describe the behaviour of the multiprocessor implementing cache-coherence protocols. A system of differential equations is set and solved and the steady-state probabilities are obtained.

2. Definition and Analysis of the Model

A multiprocessor consists of several processors connected together to a shared main memory by a common complete transaction bus. Each processor has a private cache. When a processor issues a request to its cache, the cache controller examines the state of the cache and takes suitable action, which may include generating bus transaction to access main memory. Coherence is maintained by having all cache controllers "snoop" on the bus and monitor the transaction. Snoopy cache-coherence protocols fall in two major categories: Invalidate and Update [2, 3, 9]. Invalidating protocols are studied here but the concepts can be applied with some modifications to updating

protocols too. Transactions may or may not include the memory block and the shared bus. Typical transaction that does not include memory block is Invalidate Cache Copy which occurs when a processor requests writing in the cache. All other processors simply change the status bit(s) of their on copies to Invalid. If the memory block is uncached or not clean it can be uploaded from the main memory, but in today's multiprocessors it is rather uploaded from another cache designated as Owner (O) (cache-to cache transfer). Memory-to-cache transfer occurs when the only clean copy is in the main memory. A cache block is written back (WB) in the main memory (bus is used) when a dirty copy is evicted [6]. The evicted block is maintained in the write-back buffer until the block is written back. The responsibility of handling the WB transaction rests solely with the processor's cache controller and thus the processor can resume processing immediately after completion of its blocking request. Apparently the bus can be considered as the bottleneck of the system.

We shall refer to the processors as customers and to the bus as server.

Inter-arrival times are exponentially distributed with parameter λ. This assumption is adequate for most applications [7]. The number of the processors is N. Requests are served on First Come First Served (FCFS) basis. Immediately after issuing a request for cache-to-cache transfer or synchronization procedure the customer blocks itself. Service time for blocking request has a density function $f_b(x)$. When service is completed the processor resumes processing with probability p or resumes processing and generates a new WB request with probability q $(p+q=1)$. The new request joins the queue at its tail or is taken immediately into service if there is no queue at the server. Details on how to obtain the input parameters are given in [2, 3, 8, 9]. This new request has a different density function $f_w(x)$ and corresponds to WB transaction. It does not block the customer but the server is held until completion of WB transaction therefore adding to the queue. System's states can be described by two components:1) number of customers doing internal processing, and 2) ordering z_r of blocking(b) and WB(w) requests (waiting and in service) at the server. Transitions between these states are illustrated in Fig. 1.

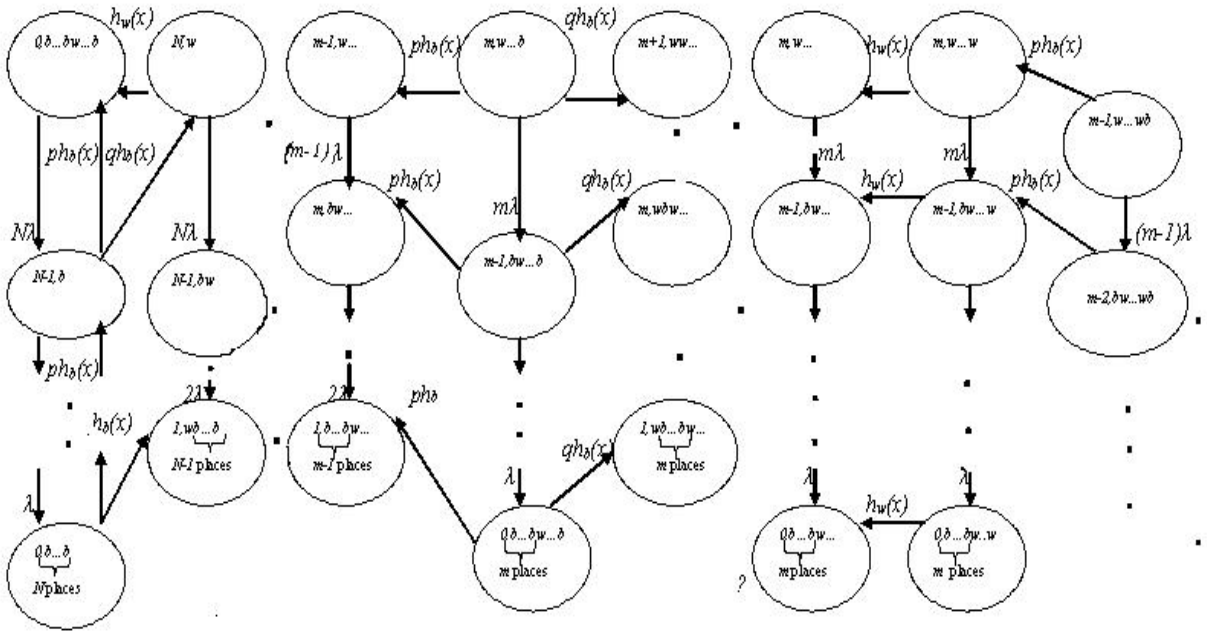

FIG.1

Each processor at any moment can have one blocking and one write-back request at the server, so that the maximum length of z_r is 2N.

Throughout this paper we use the following notations

b blocking request

w write-back request

z_r ordering of *b*'s and *w*'s

$\boldsymbol{Z}$ $\{z_r\}$ set of all orderings at the server

$LM(z_r)$ leftmost character of the ordering z_r

$RM(z_r)$ rightmost character of the ordering z_r

y_k ordering in which the $LM(y_k)=w$; parent state (node)

$\pm char + z_r$ $(char=b,w)$ ordering originating from z_r by adding/removing the $RM(z_r)$; example: $z_r=wbbbw$, $-w+z_r=bbbw$, $w+z_r=wwbbbw$

$z_r \pm char$ $(char=b,w)$ ordering originating from z_r by adding/removing the $LM(z_r)$; example: $z_r=wbbbw$, $z_r-w_r=wbbb$, $z_{r+}w=wbbbww$

$\boldsymbol{Y}$ $\{y_k\}$, $\boldsymbol{Y}\subset\boldsymbol{Z}$ subset of the parent states; Although the leftmost character of the state *N-1,b* is not *w* we refer to it as a parent state

j,z_r system's state (node), where *j* is the number of customers doing internal processing

P_N *P*[in equilibrium all *N* customers are doing internal processing]

$P_{j,z_r}(x)$ *P*[in the equilibrium state *j* customers are doing internal processing, *N-j* are in the queue and/or in the server, the ordering of *b* and *w* requests is z_r, and the elapsed service time lies between *x* and *x+dx*].

P_N $\lim_{t\to\infty} P_N(t)$

P_{j,z_r} $\int_0^\infty P_{j,z_r}(x)dx$ steady-state probabilities

β_j $j\lambda;\ j=1\leq j\leq N$

$F_{srv}(x)$ c.d.f. of the service time of type *srv* ; $srv=b,w$

$f_{svr}(x)$ density function of the service time of type

$\frac{1}{\mu_{srv}}$ $\int_0^\infty x f_{srv}(x)dx$

$h_{srv}(x)$ $\frac{f_{srv}(x)}{1-F_{srv}(x)}$ service rate for type *srv*

$\underline{f_{srv}(s)}$ Laplace transform of $f_{srv}(x)$

*multiplication sign

The algorithm below generates the states of the system:

Number_nonblocked_customers*(first_parent)=N;*
Seq*(first_parent)=Ø;*
Add *first-Parent* to New_Parent_Nodes;
Do while New_Parent _Nodes=*Ø*{
Parent_Nodes=New_Parent_Nodes;
New_Parent_Nodes=*Ø*;
$\forall Parent\_Node \in Parent\_Nodes$
{**Generate_all_children**(*parent_node*}
$\forall parent\_node \in Parent\_Nodes$ and $\forall$ its children
{**Generate_Parent** (*parent_node*)}
}
Generate_Child(*node,i*){
Number_nonblocked_customers(*child*)=Number_nonblocked_customers(*node*)-*i*;
Seq(*child*)=(Number_nonblocked_customers*(node)-i)*b*+seq(*node*);
Add *child* to *Nodes*}

Generate_all_children(*node*){
For *i=Number_nonblocked_customers* ,0
Generate_child(*node,i*);
Endfor}

Generate_parent(*node*){
If RM(seq(*node)*)=*b* **then**
Number_nonblocked_customers(*new_parent*)=
Number_nonblocked_customers(*node*)+1;
Seq(*new_parent*)=*w*+seq(*node*)-LM(seq(*node*));
Add *new_parent* to New_Parent_Nodes;
Endif}

In each step a subset of parent nodes is created according to the transition $m-1,-w+y_k+b \xrightarrow{qh_b(x)} m, y_k$, then the child nodes of each parent nodes are added to **Z.** Nodes with *w* as a rightmost character in the ordering do not generate parent nodes. The number of rightmost *b'*s in the generating ordering is decremented by one, so that the node $m,... \underbrace{b...b}_{l\_places}$ will be exhausted in *l* steps. Since the node $0, \underbrace{b...b}_{N\_places}$ of the first subset has the largest number of such *b'*s, *N* step will be needed to exhaust it. So the algorithm produces all states (nodes) in *N+1* steps.

We will prove that the algorithm produces all possible system's states. First we use induction to show that all $ph_1(x)$ transactions in a given subset occur between states in this subset. Let m,z_z and $m-1,z_r+b$ be two states in the i^{th} subset ($1 \le i \le N-1$). Obviously $m-1, z_r+b \xrightarrow{ph_b(x)} m, z_r$. If RM($z_r$)=*b* both states generate parent nodes in the $(i+1)^{st}$ subset and there is a $ph_1(x)$ transaction between them: $m-1, w+z_r \xrightarrow{ph_b(x)} m, w+z_r-b$. A $ph_1(x)$ transaction also exists between their child states $j,(m-1)*b+w+z_r-b$ and $j,(m-1)*b+w+z_r$.

Since in the last subset RM(z_r)=*w* for all states no $ph_1(x)$ transitions exist.

Let's denote two arbitrary states in the i^{th} subset $\mathbf{Z}_i, z_a$ and $j+l, z_c$ ($0 \le i \le N+1$, $0 \le j \le N$, $0 \le l \le N$, $z_a \in \mathbf{Z}_i$ and $z_c \in \mathbf{Z}_i$) and an arbitrary state in the $(i+1)^{st}$ subset $\mathbf{Z}_{i+1}$ by j, z_d ($z_d \in \mathbf{Z}_{i+1}$). The following relations can be proven by induction on *i*

$$Length(z_a)\text{-}length(z_c)=l \quad (1)$$

and

$$Length(z_d)\text{-}length(z_a)=1. \quad (2)$$

Proof: Transitions $j, z_a+b \xrightarrow{j\lambda} j-1, b+z_a \xrightarrow{qh_b(x)} j, w+b+z_a$ and $j+l, z_c+b \xrightarrow{(j+l)\lambda} j+l-1, b+z_c \xrightarrow{qh_b(x)} j+l, w+b+z_c$ generate two parent states for which apparently (1) and (2) hold. Proof for the child states is straightforward.

We can conclude now that transitions of type $h_w(x)$ occur from nodes in the $(i+1)^{st}$ subset to nodes in the i^{th} subset.

Viewing the nature of the system, we obtain the following set of differential equations

$$\beta_N P_N = p \int_0^\infty P_{N-1,b}(x) h_b(x) dx + \int_0^\infty P_{N,w}(x) h_w(x) dx \quad (3)$$

$$\left[\frac{d}{dx}+\beta_m+h_{srv}(x)\right]P_{m,y_k}(x)=0 \tag{4}$$

$$\left[\frac{d}{dx}+\beta_j+h_{srv}(x)\right]P_{j,(m-j)*b+y_k}(x)=\beta_{j+1}P_{j+1,(m-j-1)*b+y_k}(x) \tag{5}$$

$$\left[\frac{d}{dx}+h_{srv}(x)\right]P_{0,m*b+y_k}(x)=\beta_1 P_{1,(m-1)*b+y_k}(x) \tag{6}$$

having the following boundary and normalizing conditions

$$P_{N\text{-}1,b}(0)=p\int_0^\infty P_{N-2,bb}(x)h_b(x)dx+\int_0^\infty P_{N-1,bw}(x)h_w(x)dx+\beta_N P_N \tag{7}$$

$$P_{m,y_k}(0)=q\int_0^\infty P_{m-1,-w+y_k+b}(x)h_b(x)dx+\int_0^\infty P_{m,y_k+w}(x)h_w(x)dx \tag{8}$$

for the i^{th} subset ($2\le i\le N$),$1\le m\le N$, and no $ph_b(x)$ transition to m,y_k

$$P_{m,y_k}(0)=q\int_0^\infty P_{m-1,-w+y_k+b}(x)h_b(x)dx+\int_0^\infty P_{m,y_k+w}(x)h_w(x)dx \tag{9}$$

$$+p\int_0^\infty P_{m-1,y_k+b}(x)h_b(x)dx$$

For $2\le i\le N$,$1\le m\le N$, and $ph_b(x)$ transition to m,y_k

$$P_{m,y_k}(0)=q\int_0^\infty P_{m-1,-w+y_k+b}(x)h_b(x)dx \quad \text{for the last } (N+1)^{st} \text{ subset} \quad . \tag{10}$$

$$P_{j,(m-j)*b+y_k}(0)=\int_0^\infty P_{j,(m-j)*b+y_k+w}(x)h_w(x)dx \quad \text{for } 0\le j\le m \tag{11}$$

for the i^{th} subset ($2\le i\le N$),$1\le m\le N$, and no $ph_b(x)$ transition to $j,(m\text{-}j)*b+y_k$.

$$P_{j,(m-j)*b+y_k}(0)=p\int_0^\infty P_{j-1,(m-j)*b+y_k+b}(x)h_b(x)dx+\int_0^\infty P_{j,(m-j)*b+y_k+w}(x)h_w(x)dx \tag{12}$$

for the i^{th} subset ($2\le i\le N$),$1\le m\le N$, and $ph_b(x)$ transition to $j,(m\text{-}j)*b+y_k$.

$$P_{0,m*b+y_k}(0)=\int_0^\infty P_{0,m*b+y_k+w}(x)h_w(x)dx \qquad \text{for } 1\le i\le N \tag{13}$$

$$P_{j,(m-j)*b+y_k}(0)=0 \quad \text{for } 0\le j<m \quad \text{for the last } (N+1)^{st} \text{ subset} \tag{14}$$

$$P_N+\sum_{z_r\in\mathbf{Z}}P_{j,z_r}=1 \tag{15}$$

By using discrete transform [4] the equations (4-5) are transformed as follows

$$\left[\frac{d}{dx}+\beta_j+h_{srv}(x)\right]u_{j,(m-j)*b+y_k}(x)=0 \qquad \text{for } 1\le j\le m \tag{16}$$

where

$$u_{j,(m-j)*b+y_k}(x)=\sum_{n=j}^{m}(-1)^{n-j}\binom{n}{j}P_{n,(m-n)*b+y_k}(x) \text{ , and}$$

$$P_{j,(m-j)*b+y_k}(x)=\sum_{n=j}^{m}(-1)^{n-j}\binom{n}{j}u_{n,(m-n)*b+y_k}(x)$$

Let $v_{j,(m-j)*b+y_k}(x) = \dfrac{u_{j,(m-j)*b+y_k}(x)}{1-F_{srv}(x)}$ and $P'_{0,m*b+y_k}(x) = \dfrac{P_{0,m*b+y_k}(x)}{1-F_{srv}(x)}$

Then from (16) and (6) we have after some manipulations

$$\left[\frac{d}{dx}+\beta_j\right] v_{j,(m-j)*b+y_k}(x) = 0 \tag{17}$$

$$\left[\frac{d}{dx}\right] P'_{0,m*b+y_k}(x) = \beta_1 P'_{1,(m-1)*b+y_k}(x). \tag{18}$$

Hence solutions of (17-18) are

$$u_{j,(m-j)*b+y_k}(x) = [1-F_{srv}(x)]u_{j,(m-j)*b+y_k}(0)e^{-\beta_j x} \qquad \text{for } 1 \le j \le m \tag{19}$$

$$P_{0,m*b+y_k}(x) = [1-F_{srv}(x)]\beta_1\left[\sum_{n=1}^{m}(-1)^{n-1} n \frac{1-e^{-\beta_n x}}{\beta_n} u_{n,(m-n)*b+y_k}(0)\right] + [1-F_{srv}(x)]P_{0,m*b+y_k}(0) \tag{20}$$

By integrating (19-20), and from (3) we obtain the steady-state probabilities

$$P_{j,m*b+y_k} = \sum_{n=1}^{m}(-1)^{n-1}\binom{n}{j}\frac{1-\underline{f_{srv}(\beta_n)}}{\beta_n} u_{n,(m-n)*b+y_k}(0) \tag{21}$$

$$P_{0,m*b+y_k} = \sum_{n=1}^{m}(-1)^{n-1} n \frac{\beta_1}{\beta_n}\left[\frac{1}{\mu_{srv}} + \frac{1-\underline{f_{srv}(\beta_n)}}{\beta_n}\right] u_{n,(m-n)*b+y_k}(0) + \frac{P'_{0,m*b+y_k}(0)}{\mu_{srv}} \tag{22}$$

$$P_N = \frac{p u_{N-1,b}(0)\underline{f_b(\beta_{N-1})} + u_{N,w}(0)\underline{f_w(\beta_N)}}{\beta_N} \tag{23}$$

From (7-13) we get after some algebra the following linear equations

$$\begin{aligned} u_{N-1,b}(0) = p\sum_{n=N-2}^{N-1}(-1)^{n-N+2}\binom{n}{N-2}u_{n,(N-n)*b}(0)\underline{f_b(\beta_n)} + \\ \sum_{n=N-1}^{N}(-1)^{n-N+1}\binom{n}{N-1}u_{n,(N-n)*b+w}(0)\underline{f_w(\beta_n)} + \beta_N P_N \end{aligned} \tag{24}$$

where $y_r = -w+y_k+b$.

$$\begin{aligned} u_{m,y_k}(0) = q\sum_{n=m-1}^{l}(-1)^{n-m+1}\binom{n}{m-1}u_{n,(l-n)*b+y_r}(0)\underline{f_b(\beta_n)} + u_{m,y_k+w}(0)\underline{f_w(\beta_m)} \\ + p\sum_{n=m-1}^{m}(-1)^{n-m+1}\binom{n}{m-1}u_{n,(m-n)*b+y_k+b}(0)\underline{f_b(\beta_b)} \end{aligned} \tag{25}$$

where $y_r = -w+y_k+b$, for the i^{th} subset ($2 \le i \le N$), $2 \le m \le N$, and $ph_b(x)$ transition to m,y_k

$$u_{m,y_k}(0) = q\sum_{n=m-1}^{l}(-1)^{n-m+1}\binom{n}{m-1}u_{n,(l-n)*b+y_r}(0)\underline{f_b(\beta_n)} + u_{m,y_k+w}(0)\underline{f_w(\beta_m)} \tag{26}$$

for the i^{th} subset ($2 \le i \le N$), $2 \le m \le N$, and no $ph_b(x)$ transition to m,y_k .

$$u_{1,y_k}(0) = q\beta_1 \sum_{n=1}^{l} (-1)^{n-1} n \frac{1-\underline{f_b(\beta_n)}}{\beta_n} u_{n,(l-n)*b+y_r}(0) + qP'_{0,l*b+y_r}(0) + u_{1,y_k+w}(0)\underline{f_w(\beta_1)} \qquad (27)$$

for $2 \le i \le N$.

For the last subset we have

$$u_{m,y_k}(0) = q \sum_{n=m-1}^{l} (-1)^{n-j+1} \binom{n}{j-1} u_{n,(l-n)*b+y_r}(0)\underline{f_b(\beta_n)} \qquad \text{for } 2 \le m \le N \qquad (28)$$

$$u_{1,y_k}(0) = q\beta_1 \sum_{n=1}^{l} (-1)^{n-1} n \frac{1-\underline{f_b(\beta_n)}}{\beta_n} u_{n,(l-n)*b+y_r}(0) + qP'_{0,l*b+y_r}(0) \qquad (29)$$

$$\sum_{n=j}^{m} (-1)^{n-j} \binom{n}{j} u_{n,(m-n)*b+y_k}(0) = p \sum_{n=j-1}^{m} (-1)^{n-j+1} \binom{n}{j-1} u_{n,(m-n)*b+y_k+b}(0)\underline{f_b(\beta_n)} + \sum_{n=j}^{m} (-1)^{n-1} \binom{n}{j} u_{n,(m-n)*b+y_k+w}(0)\underline{f_w(\beta_n)} \qquad (30)$$

for the i^{th} subset ($1 \le i \le N$), $2 \le j \le m$, and $ph_b(x)$ transition to $j,(m-j)*b+y_k$.

$$\sum_{n=j}^{m} (-1)^{n-j} \binom{n}{j} u_{n,(m-n)*b+y_k}(0) = \sum_{n=j}^{m} (-1)^{n-1} \binom{n}{j} u_{n,(m-n)*b+y_k+w}(0)\underline{f_w(\beta_n)} \qquad (31)$$

for the i^{th} subset ($1 \le i \le N$), $2 \le j \le m$, and no $ph_b(x)$ transition to $j,(m-j)*b+y_k$.

$$\sum_{n=1}^{m} (-1)^{n-1} n u_{n,(m-n)*b+y_k}(0) = p\beta_1 \sum_{n=1}^{m} (-1)^{n-1} n \frac{1-\underline{f_1(\beta_n)}}{\beta_n} u_{n,(m-n)*b+y_k+b}(0) + pP'_{(m-1)*b+y_k+b}(0) + \sum_{n=1}^{m} (-1)^{n-1} n u_{n,(m-n)*b+y_k+w}(0)\underline{f_w(\beta_n)} \qquad (32)$$

for the i^{th} subset ($1 \le i \le N$), and $ph_b(x)$ transition to $1,(m-1)*b+y_k$.

$$\sum_{n=1}^{m} (-1)^{n-1} n u_{n,(m-n)*b+y_k}(0) = \sum_{n=1}^{m} (-1)^{n-1} n u_{n,(m-n)*b+y_k+w}(0)\underline{f_w(\beta_n)} \qquad (33)$$

for the i^{th} subset ($1 \le i \le N$), and no $ph_b(x)$ transition to $1,(m-1)*b+y_k$.

$$P'_{0,m*b+y_k}(0) = \beta_1 \sum_{n=1}^{m} (-1)^{n-1} n \frac{1-\underline{f_w(\beta_n)}}{\beta_n} u_{n,(m-n)*b+y_k}(0) + P'_{0,m*b+y_k+w}(0) \quad \text{for } 1 \le i \le N \qquad (34)$$

From (`4) by induction and using the relation $\sum_{n=0}^{j} (-1)^n \binom{j}{n} = 0$ we obtain

$$u_{j,(m-j)*b+y_k} = \binom{m}{j} u_{m,y_k} \qquad \text{for } i=N+1, 1 \le j < m \qquad (35)$$

Coefficients $u_{j,z_r}(0)$ can be determined from (15) and (24-35).

Various performance characteristics can be computed using the state probabilities. For example, the average number of waiting (blocked) customers (*ANBC*) in the case of blocking caches will be given by

$$ANBC = \sum_{z_r \in \mathbf{Z}} (N - j) P_{j,z_r}$$

In the case of non-blocking caches *ANBC* will be

$$ANBC = \sum_{\substack{z_r \in \mathbf{Z} \\ RM(z_r)=b}} (N - j - 1 + k) P_{j,z_r} + \sum_{\substack{z_r \in \mathbf{Z} \\ RM(z_r)\neq b}} (N - j) P_{j,z_t}$$

where *k* is the ratio of average memory stall time [2] . *k* depends strongly on the application. (*1-k)* actually refers to the fraction time the processor is consuming data while cache-to-cache or memory-to cache transfer is in progress.

Conclusion

This paper presented a model for a shared memory, shared bus multiprocessor maintaining Invalidate type cache coherence protocol. We obtained the steady-state probabilities of the system so that the behaviour in equilibrium can be studied and analyzed.

We showed that results can be applied to determine the output parameters for both blocking and non-blocking caches.

References

1. S. K. Bose, Introduction to Queuing Systems, Kluwer/Plenum Publishers, 2001
2. J. L. Hennessy, D. A. Patterson; Computer Architecture: A Quantitative Approach, Pearson Publishers, 2003
3. M. C. Chiang, Memory System Design For Bus Based Multiprocessor, *PhD Thesis,* University of Wisconsin, 1991
4. T. Itoi, T. Nishida, M. Kodama and E. Ohi, N-unit parallel redundant system with correlated failures and single repair facility, *Microelectronics and Reliability*, vol. 17, pp. 279-285, 1978
5. E. Lazowska, J. Zahorjan, G. Graham, and K. Sevcik, Quantitative System Performance, Computer System Analysis Using Queuing Network Models, Prentice-Hall, Englewood Cli_s, NJ, May 1984.
6. A. Louri, A.K. Kodi, An optical interconnection network and a modifying snooping protocol for the design of large-scale symmetric multiprocessors (SMPs), *IEEE Transactions on Parallel and Distributing Systems*, vol. 15, No. 12, Dec. 2004, pp. 1093-1104
7. R. E. Matick, Comparison of analytic performance models using closed mean-value analysis versus open-queuing theory for estimating cycles per instruction of memory hierarchies, *IBM Journal of Research and Development*, Jul 2003
8. D. J. Sorin et. al., A customized MVA model for ILP multiprocessors, *Technical report #1369,* University of Wisconsin-Madison, 1998
9. D. J. Sorin et. al., Evaluation of shared-memory parallel system with ILP processors , *Proc. 25th Int'l Symp. On Computer Architecture*, June 1998, pp. 180-191
10. J. Sustersic, A. Hurson, Coherence protocol for bus-based and scalable multiprocessors, Internet and wireless distributed computing environments: a survey , *Advances in Computers*, vol.59,2003 pp. 211-278.

Article received: 2008-06-27