Computers Math. Applic. Vol. 24, No. 1/2, pp. 41–50, 1992
Printed in Great Britain.
0097-4943/92 \$5.00 + 0.00

QUEUEING MODELS OF CONCURRENCY CONTROL IN DATABASE WITH POISSON ARRIVALS

SHOICHI NISHIMURA AND YONG JIANG
Institute of Socio-Economic Planning, University of Tsukuba
Tsukuba, Ibaraki 305, Japan

Abstract—We study the mean performance of concurrency control in database with Poisson arrival. The computer system is formulated as open queueing systems with two cases, no waiting room case and unlimited waiting room case. To avoid the complexity of the state space, we propose two aggregate models, birth-death process and quasi-birth-death process, to predict system performance. The death rates of two models are defined by the throughput of closed systems. Instead of DC-thrashing appeared in closed system, we obtain the monotonicity of response time and loss probability in birth-death models. From these properties we can obtain the optimal of multiprogramming level. In quasi-birth-death models we give the algorithms to calculate the equilibrium probability. We obtain numerical results from both analytic model and computer simulation. Comparisons show that the results of birth-death models and quasi-birth-death models are very similar, and the predictions of our models fit well with the simulation results.

1. INTRODUCTION

We consider a computer system with multiprogramming level n. That means n transactions can be processed simultaneously in the system. In such a computer system concurrency control is indispensable for consistent processing of transactions. Many concurrency control methods have been proposed. The most popular one is locking. Under the concurrency control with locking, the system operates in the following basic way.[1] Transaction which requires to a granule of the database has to lock the granule before accessing it. Until the transaction completes its job and releases the lock, other transactions cannot access the granule. One of the most important property of the multiprogramming system is DC-thrashing of throughput $t(n)$, i.e., $t(n)$ is a unimodal function of n. The reason of DC-thrashing is intuitive. There are two opposite forces affecting the throughput. Increasing n tends to increase the throughput; on the other hand, increasing n also increases conflicts among transactions, which finally reduces the throughput. Hence, there is one thrashing point n^* at which throughput achieves the maximum.

In [1,2] the analytic models are proposed and the DC-thrashing phenomenon is shown by the simple deterministic model in equilibrium. It is assumed that the system is so crowded that n transactions are always processed and no idle processor exists in the system. Under the same assumptions, the closed queueing model is considered in [3]. For direct modeling the computational complexity is caused by the complexity of state space. To avoid the difficulty, states of the system are aggregated to the number of granules locked by transactions. Other analytic models of locking performance can be found, for example, in [4–6]. In this paper we assume that transactions arrive to the computer system according to Poisson process with parameter λ. Instead of the DC-thrashing, we discuss the mean performance of an open system multiprogramming level n in two cases. Let v and m be the number of transactions of the system and locked granules, respectively. The first is no waiting room case. If $v < n$, an arrival transaction receives service immediately. If not, the arrival transaction will be rejected (loss transaction). If the transaction completes processing all its required granules without encountering any conflict, it terminates and then departs from the system. If not, the transaction restarts with new requests.

[1] To simplify description, we consider writelock only here. A detailed description about both writelock and readlock can be found in next section.

Typeset by $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX

The second is the unlimited waiting room case. If $v < n$, an arrival transaction enters the system and then receives service immediately. If not, the arrival transaction stays in a queue waiting for entering the system. Once there is an idle processor, waiting transactions (if any) enter the system according to the FIFS discipline. In this paper, we propose two aggregations for these open cases. In the first, we formulate the system as a birth-death process with the state v to be the number of transactions in the system. In the second, we model the system as a quasi-birth-death process with the state of (v, m).

A description of the model is given in Section 2. In Section 3 we study a closed queueing model, and obtain its throughput, which will be used in the open system as death rate. In Section 4 no waiting room case is considered. Using a birth-death model, we prove the monotonisities of the response time and the loss probability. In a quasi-birth-death model we propose a computational algorithm to calculate equilibrium probability, which is an application of the recursive solution method [7]. In Section 5 the unlimited waiting room case is considered. We prove the optimality of thrashing point n^* with respect to the response time. In a quasi-birth-death model, we obtain the equilibrium probability using the Neuts's method. In Section 6, numerical results about our analytic models and computer simulation are obtained. Comparisons show that predictions of these models fit well with results of simulation.

2. DESCRIPTIONS OF THE MODEL

We consider that a database is a set of D granules and denote it by $\mathcal{DB}$. A granule is the smallest entity in the database that may be locked. According to the access frequency, $\mathcal{DB}$ is partitioned into ν kinds of granules. Let $\mathcal{DB}_u$ be a partition of $\mathcal{DB}$, i.e., $\mathcal{DB}_u \bigcap \mathcal{DB}_{u'} = \emptyset$ for $u \neq u'$, $u, u' = 1, 2, \ldots, \nu$ and $\mathcal{DB} = \bigcup_{u=1}^{\nu} \mathcal{DB}_u$. Let $c_u D$ be the number of granules in $\mathcal{DB}_u$ such that $c_u > 0$, $\sum_{u=1}^{\nu} c_u = 1$.

A transaction has a sequence of requests for locking granules. The number of granules required by a transaction is called the *length* of the transaction and we classify transactions by their lengths. Let π_k be the probability of the transaction in class k, $k = 1, 2, \ldots, C$, and $\pi_0 = 0$.[2] Let p_k, $k = 1, 2, \ldots, C$, be the probability that the k^{th} request is for termination, $p_0 = 0$, then we have that $p_k = \pi_k / \sum_{j=k}^{C} \pi_j$.

Assume that every transaction is independent of its class in making requests. Transaction requires a granule of $\mathcal{DB}_u$ with the probability of b_u and in $\mathcal{DB}_u$ every granule is required equally likely. To ensure consistent processing, transaction has to lock the required granules before it accesses them. Two kinds of locks, writelock and readlock, are considered. Let b_u^w and b_u^r be the probabilities that a transaction requires writelock and readlock on $\mathcal{DB}_u$, respectively, such that $b_u = b_u^w + b_u^r$ and $\sum_{u=1}^{\nu} b_u = 1$. Suppose that the request of a processing transaction in class k is for writelocking (or readlocking) a granule. If the granule has already been locked (writelocked) by any other transaction, then the transaction *conflicts* with other transactions. If not, the transaction writelocks (readlocks) granule, and then makes the next request. In this way, if the transaction locks and then completes processing k^{th} required granules without encountering any conflict, it releases all its locks and then *terminates*. If the transaction encounters a conflict at some step of its requests, it releases all its locks and *restarts*. We assume that transaction restarts with new requests. To simplify analysis, we assume that the time for processing requested granules is independent and identically distributed according to exponential distribution with rate μ.

Consider that a transaction holds I_u writelocks and J_u readlocks on $\mathcal{DB}_u$, $u = 1, 2, \ldots, \nu$, and let K be the total number of locks which the transaction holds. Then $K = \sum_{u=1}^{\nu}(I_u + J_u)$. Let i_u and j_u be the realizations of I_u and J_u, respectively. Then

$$\mathrm{S} = \{\mathbf{s} \mid \mathbf{s} = (i_1, j_1, \ldots, i_\nu, j_\nu), \qquad 0 \leq |\mathbf{s}| \leq C\} \tag{2.1}$$

[2] Since the granules required by a transaction are distinct, transaction must be terminated before the D^{th} request. Hence, there is some $C \leq D$, $\pi_k = 0$ for all $k > C$ and $\sum_{k=1}^{C} \pi_k = 1$.

is the state space of a transaction, where $|\mathbf{s}| = \sum_{u=1}^{\nu} i_u + j_u = k$. Let $N(\mathbf{s})$ be the number of transactions in state $\mathbf{s}$ for $\mathbf{s} \in \mathbf{S}$, N_k the number of transactions with k locks and N_L the number of locks held by all transactions. Then we have

$$V = \sum_{\mathbf{s}\in\mathbf{S}} N(\mathbf{s}), \qquad N_k = \sum_{|\mathbf{s}|=k} N(\mathbf{s}), \qquad N_L = \sum_{\mathbf{s}\in\mathbf{S}} |\mathbf{s}|\, N(\mathbf{s}), \qquad 0 \le k \le C. \tag{2.2}$$

Let M be the total number of locked granules of $\mathcal{DB}$, $M \le D$. We know that the existence of readlock makes $M \le N_L$.

Denote M_u as the number of locked granules of $\mathcal{DB}_u$ such that $M = \sum_{u=1}^{\nu} M_u$. Let M_u^w and M_u^r be the number of writelocked and readlocked granules of $\mathcal{DB}_u$, respectively, $M_u = M_u^w + M_u^r$.

3. THE CASE OF NO IDLE PROCESSOR

First we consider the case where the system is so busy that there is no idle processor in system all the time. Hence the number of transactions v always equals to the multiprogramming level n. From modeling point of view, the system is formulated as a closed queueing network. Let $n(\mathbf{s})$ be the realization of $N(\mathbf{s})$ for $\mathbf{s} \in \mathbf{S}$. Then $n(\mathbf{s})$ denotes the number of transactions in state $\mathbf{s}$. In general we can define the state of the system by $\mathbf{n} = (n(0), \dots, n(\mathbf{s}), \dots)^\top$. Hence the state space is

$$\Phi_v = \{\mathbf{n} \mid \mathbf{n} = (n(0), \dots, n(\mathbf{s}), \dots)^\top,\ n(\mathbf{s}) \ge 0,\ \mathbf{s} \in \mathbf{S},\ \sum_{\mathbf{s}\in\mathbf{S}} n(\mathbf{s}) = v\}. \tag{3.1}$$

With the increment of v and C, the number of states of Φ_v are geometrically increased, which makes the computation become very hard. To avoid the computational complexity, we proposed an aggregate model to estimate the mean value of throughput $t(v)$ and restart rate $a(v)$ in [3].

We considered that M, the total number of locked granules in $\mathcal{DB}$, is the most important factor associated with conflict behaviors. Since the direct estimation of m is difficult, we use N_L, the total number of locks held by all transactions, to estimate M indirectly. Observing the difference between N_L and M, we know that $N_L - M$ is the number of granules which share the same readlock with other transactions. In general the difference is not large so that we may put $N_L \approx M$. Let m be a realization of N_L, then the state space of N_L is

$$\Psi_v = \{m \mid 0 \le m \le vC\}. \tag{3.2}$$

We define a mapping from Φ_v to Ψ_v such that

$$\left\{\mathbf{n} \mid \mathbf{n} = (n(0), \dots, n(\mathbf{s}), \dots)^\top \in \Phi_v,\ \sum_{\mathbf{s}\in\mathbf{S}} |\mathbf{s}|\, n(\mathbf{s}) = m\right\} \overset{mapping}{\longmapsto} m \in \Psi_v. \tag{3.3}$$

In this way the multidimensional state space Φ_v is aggregated as a one-dimensional state space Ψ_v.

Let $p_{c,k}(v,m)$ be the conflict probability at the $k+1^{\text{st}}$ request of a transaction and $\bar{N}_k(v,m)$ be the conditional expectation of N_k when $N_L = m$. We estimate $p_{c,k}(v,m)$ and $\bar{N}_k(v,m)$ under the following assumptions. (1) The number of u^{th} kind of granules which are locked by a transaction is small compared to the number of granules of $\mathcal{DB}_u$, i.e., for any $u = 1, 2, \dots, \nu$, $i_u + j_u \ll c_u D$. (2) Under the average environment of the database, each transaction transfers from $\mathbf{s}$ to $\mathbf{s}'$, $\mathbf{s}, \mathbf{s}' \in \mathbf{S}$, according to an independent Markovian process as a whole. The results about $p_{c,k}(v,m)$ and $\bar{N}_k(v,m)$ can be found in [3]. Denote the mean transition rate from state m to m' by $q(v)_{m,m'}$, $m,\ m' \in \Psi_v$. Then we have

$$\begin{cases} q(v)_{m,m} = -n\mu + \mu p_{c,0}(v,m)\, \bar{N}_0(v,m), & m = 1, \dots, vC-1 \\ q(v)_{m,m+1} = \mu \sum_{k=0}^{C-1} (1-p_k)(1-p_{c,k}(v,m))\, \bar{N}_k(v,m), & m = 0, 1, \dots, vC-1 \\ q(v)_{m,m-C} = \mu \bar{N}_C(v,m), & m = C, C+1, \dots, vC \\ q(v)_{m,m-k} = \mu((1-p_k)\, p_{c,k}(v,m) + p_k)\, \bar{N}_k(v,m), & m = 1, \dots, vC-1; \\ & k = 1, \dots, C-1 \\ q(v)_{m,m'} = 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

Let $P_v(m)$ be the equilibrium probability of state, $m \in \Psi_v$, $\mathbf{p}_v = [P_v(0), P_v(1), \ldots, P_v(vC)]^\top$ and $\mathbf{Q}_v$ be the transition matrix given by (3.4). Then we can obtain $\mathbf{p}_v$ by solving the following equations.

$$\mathbf{p}_v^\top \mathbf{Q}_v = 0 \qquad \text{and } \mathbf{p}_v^\top \mathbf{e} = 1, \tag{3.5}$$

where $\mathbf{e}^\top = (1, \ldots, 1)$. From (3.4) we know that $\mathbf{Q}_v$ is the skip-free-to-right transition rate matrix, i.e., the transition from state i to state j is not possible if $j > i+1$. By using this property, $P_v(m)$ is iteratively calculated from $P_v(vC)$ to $P_v(1)$ and then normalized. In this way we can analysis a large scale model because of memory saving. Let $t(v)$ and $a(v)$ be the mean values of throughput and restart rate of the system, respectively. Then $t(v)$ and $a(v)$ can be given approximately as follows:

$$\begin{aligned} t(v) &= \sum_{k=1}^{C} \sum_{m=k}^{vC} \mu p_k \alpha_k \bar{N}_k(v,m)\, P_v(m), \\ a(v) &= \sum_{k=0}^{C-1} \sum_{m=k}^{vC} \mu(1-p_k)\, p_{c,k}(m)\, \alpha_k \bar{N}_k(v,m)\, P_v(v,m). \end{aligned} \tag{3.6}$$

We verified the model through a variety of locking schemes such as uniform and non-uniform, write and read lock, fix and indeterminate length of transactions. Comparison shows that our models fit well with the simulation results [3]. From both analytic and simulation results, two properties of $t(v)$ can be found.

Property (i): $t(v)/v$ is monotone decreasing in v, i.e., the individual efficiency per processing transaction is monotone decreasing.

Property (ii): $t(v)$ is DC-thrashing (or unimodal), which means that the efficiency of v processing transactions is monotone increasing in $[0, n^*]$ and monotone decreasing in $[n^*, \infty)$, where n^* is called as the thrashing point of the closed queueing model.

4. THE CASE OF NO WAITING ROOM

Suppose that transactions arrive to a database system according to Poisson process of λ. Let n be the multiprogramming level and V_n be the number of processing transactions in the system. We study the performance evaluation of the open systems with respect to n in two cases. The first is no waiting room case. If $V_n < n$, an arrival transaction receives service immediately. If not, the arrival transaction is rejected (lost transaction). If the transaction completes processing all its required granules without encountering any conflict, it terminates and then departs from the system. If not, the transaction restarts with new requests. The second is the unlimited waiting room case. If $V_n < n$, the arrival transaction enters the system and then receives the process immediately. If not, the arrival transaction stays in a queue waiting for entering the system. Once the idle processor has arisen, a waiting transaction (if any) enters the system according to the FIFS discipline.

Like the closed queueing system mentioned in the Section 3, the open system also has multidimensional state space. For a large scale model, the computation of equilibrium distribution becomes very hard. In this paper we propose two aggregate methods for these open cases. In the first, we formulate the system as a birth-death process with the state v to be the number of transactions in the system. In the second, we model the system as a quasi-birth-death process with the state of (v, m), the number of transactions and the number of locked granules. For no waiting room and unlimited waiting room cases, we study birth-death and quasi-birth-death methods in following sections.

4.1. Birth-Death Model of the No Waiting Room Case

Regarding the number of the transactions as the state of the system, we can define the system approximately as a birth-death process with the birth rate of $\lambda_v = \lambda$, $0 \le v \le n-1$, $\lambda_n = 0$ and the death rate of $t(v)$, $0 \le v \le n$, where $t(v)$ is the result of a closed system with the multiprogramming level of v given by (3.6). Hence, v is the state of the system with the state space of $\mathcal{SP}_1 = \{v \mid 0 \le v \le n\}$. Let $p_n(v)$ be the equilibrium probability of $\mathcal{SP}_1$ with the multiprogramming level of n, then we have

$$\begin{cases} p_n(v) = \frac{\lambda^v}{\prod_{i=1}^{v} t(i)} p_n(0) & 1 \le v \le n, \\ p_n(0) = \left\{ \sum_{v=0}^{n} \frac{\lambda^v}{\prod_{i=1}^{v} t(i)} \right\}^{-1}. \end{cases} \tag{4.1}$$

Let $\bar{V}_n$ be the expectation of V_n in equilibrium, $\bar{V}_n = \sum_{v=1}^{n} v p_n(v)$. Since transactions cannot enter to a crowded system, $p_n(n)$ is the loss probability of transactions, $\tilde{\lambda}_n = (1 - p_n(n))\lambda$ is the accept rate of transactions. Hence, the mean value of throughput t_n, restart rate a_n and response time r_n of the no-waiting-room system can be given as follows:

$$t_n = \sum_{v=1}^{n} t(v) p_n(v), \qquad a_n = \sum_{v=2}^{n} a(v) p_n(v), \qquad r_n = \frac{\bar{V}_n}{\tilde{\lambda}_n}, \tag{4.2}$$

where the last equality comes from the Little's formula.

Theorem 1. *If $\lambda < t(n+1)$ then the loss probability satisfies $p_n(n) > p_{n+1}(n+1)$.*

Proof. From (4.1) we know that $1/p_{n+1}(0) > 1/p_n(0)$. Because of $\lambda < t(n+1)$, we have

$$p_n(n) - p_{n+1}(n+1) = p_n(n) \left(1 - \frac{\lambda p_{n+1}(0)}{t(n+1) p_n(0)}\right) > 0. \qquad \blacksquare$$

Theorem 2. *If $t(n)/n$ is monotone decreasing in n, then the response time r_n is monotone increasing in n.*

Proof. From (4.2) we have

$$r_{n+1} - r_n = \frac{(1 - p_n(n)) \bar{V}_{n+1} - (1 - p_{n+1}(n+1)) \bar{V}_n}{\lambda(1 - p_n(n))(1 - p_{n+1}(n+1))}. \tag{4.3}$$

Through elementary calculations, the numerator of (4.3) can be given by

$$\sum_{u=0}^{n-1} \sum_{v=0}^{n} \{v p_n(u) p_{n+1}(v) - v p_n(v) p_{n+1}(u)\} + (n+1) p_{n+1}(n+1) \sum_{u=0}^{n-1} p_n(u) - p_{n+1}(n) \sum_{v=0}^{n} v p_n(v). \tag{4.4}$$

Substituting (4.1), the first term of (4.4) is vanished. We know that $p_{n+1}(n+1) = \frac{\lambda}{t(n+1)} p_{n+1}(n)$, then remaining terms can be rewritten as follows:

$$\begin{aligned} &(n+1) p_{n+1}(n+1) \sum_{u=0}^{n-1} p_n(u) - p_{n+1}(n) \sum_{v=0}^{n} v p_n(v) \\ &\quad = \left\{ (n+1) \frac{\lambda}{t(n+1)} \sum_{v=0}^{n-1} p_n(v) - \sum_{v=0}^{n} v p_n(v) \right\} p_{n+1}(n) \\ &\quad = \left\{ \sum_{v=1}^{n} \left(\frac{t(v)}{t(n+1)} (n+1) - v \right) p_n(v) \right\} p_{n+1}(n) > 0. \end{aligned} \tag{4.5}$$

Hence the numerator of (4.3) is positive. This completes the proof of the theorem. $\blacksquare$

In a practical problem, the assumptions of $\lambda < t(n+1)$ and $t(n)/n$ are natural (see Property (i)). Under these assumptions, we have proved that there is a trade-off relation between the loss probability and response time for making the decision of multiprogramming level n. The same phenomena can be observed both in simulation results and quasi-birth-death results in Section 6.

4.2. Quasi-Birth-Death Model of the No Waiting Room Case

We now aggregate the open system through v, the number of transactions processing in the system and m, the number of the locks held by all processing transactions. Hence (v, m) is the state of the system with the state space of $\mathcal{SP}_2 = \{(v,m) \mid 0 \le v \le n,\ 0 \le m \le Cv\}$.

Let $q(v,m;v',m')$ be the transition rate from state (v,m) to (v',m'). There are four kinds of transitions due to four kinds of locking performance in the system. The first is $q(v-1,m;v,m)$, which results from an arrival at the state of $(v-1,m)$. The second is $q(v,m-1;v,m)$, which is due to a locking at the state of $(v,m-1)$. $q(v,m+k;v,m)$, $0 \le k \le C-1$, is the third, which is the result of a conflict that occurred at the $k+1^{\text{st}}$ request of the transaction. The final is $q(v+1,m+k;v,m)$, which is due to a throughput of the transaction of class k, $1 \le k \le C$. These transition rates can be derived easily from (3.4).

$$\begin{cases} q(v-1,m;v,m) = \lambda, & \text{arrival,} \\ q(v,m-1;v,m) = l(v,m-1) = \mu \sum_{k=0}^{C-1} (1-p_k)(1-p_{c,k}(v,m))\, \bar{N}_k(v,m), & \text{locking,} \\ q(v,m+k;v,m) = a_k(v,m+k) = \mu(1-p_k)\, p_{c,k}(v,m)\bar{N}_k(v,m), & \text{restart,} \\ q(v+1,m+k;v,m) = t_k(v+1,m+k) = \mu p_k \bar{N}_k(v,m), & \text{throughput.} \end{cases} \tag{4.6}$$

Let $p_n(v,m)$ be the equilibrium probability of the system with the multiprogramming level n, $(v,m) \in \mathcal{SP}_2$. Then $p_n(v,m)$ satisfies following equilibrium equations

$$\begin{cases} \lambda p_n(v-1,m) + l(v,m-1)\, p_n(v,m-1) + \left[-\mu n - \lambda + a_0(v,m)\right] p_n(v,m) \\ \quad + \sum_{k=1}^{C-1} a_k(v,m+k)\, p_n(v,m+k) + \sum_{k=1}^{C} t_k(v+1,m+k)\, p_n(v+1,m+k) = 0, \\ \sum_{(v,m)\in SP_2} p_n(v,m) = 1. \end{cases} \tag{4.7}$$

Let $\mathbf{p}_n = \{p_n(0,0), p_n(1,0), \ldots, p_n(1,C), p_n(2,0), \ldots, p_n(n,Cn)\}$ be the equilibrium probability vector and $\mathbf{Q}$ be the square rate matrix of order $(n+1)(1+nC/2)$, then (4.7) can be rewritten in following matrix form.

$$\mathbf{p}_n\mathbf{Q} = 0, \qquad \mathbf{p}_n\mathbf{e} = 1, \tag{4.8}$$

where $\mathbf{Q}$ has the submatrix structure of

$$\mathbf{Q} = \begin{bmatrix} \mathbf{A}_{0,1} & \mathbf{A}_{0,2} & & & & \\ \mathbf{A}_{1,0} & \mathbf{A}_{1,1} & \mathbf{A}_{1,2} & & & \\ & \mathbf{A}_{2,0} & \mathbf{A}_{2,1} & \mathbf{A}_{2,2} & & \\ & & & \ddots & & \\ & & & & \ddots & \mathbf{A}_{n-1,2} \\ & & & & \mathbf{A}_{n,0} & \mathbf{A}_{n,1} \end{bmatrix}. \tag{4.9}$$

In (4.9), $\mathbf{A}_{v,0}$, $1 \le v \le n$, is a throughput matrix of $[1+vC] \times [1+(v-1)C]$ with elements of

$$(\mathbf{A}_{v,0})_{i,j} = \begin{cases} t_{i-j}(v,i), & 1 \le i \le Cv, \quad 1 \le i-j \le C, \\ 0, & \text{otherwise.} \end{cases} \tag{4.10}$$

$\mathbf{A}_{v,1}$, $0 \le v \le n$, is a square matrix of order $[1+vC]$ with the elements of

$$(\mathbf{A}_{v,1})_{i,j} = \begin{cases} l(v,i), & 0 \le i \le Cv-1,\ j = i+1, \\ -\mu v - \lambda + a_0(v,i), & 0 \le i \le Cv,\ j = i, \\ a_{i-j}(v,i), & 0 \le i \le Cv-1,\ 1 \le i-j \le C-1, \\ 0, & \text{otherwise.} \end{cases} \tag{4.11}$$

$\mathbf{A}_{v,2}$, $0 \le v \le n$, is an arrival matrix of $[1+vC] \times [1+(v+1)C]$ with the elements of

$$(\mathbf{A}_{v,2})_{i,j} = \begin{cases} \lambda, & j = i \\ 0, & \text{otherwise.} \end{cases} \tag{4.12}$$

By solving (4.8) we obtain $p_n(v,m)$ and then calculate the throughput t_n, restart rate a_n and response time r_n just like we did in (4.2). The remaining work is to solve the equations of (4.8). Since $\mathbf{Q}$ is the square rate matrix of order $(n+1)(1+nC/2)$, some techniques are needed to solve the large scale problem. As the closed queueing model, diagonal submatrixes of $\mathbf{Q}$, $\mathbf{A}_{v,1}$, $v = 1,2,\dots,n$, are skip-free-to-right transition rate matrixes. Let $p_n(v,Cv) = x_v$, where x_v is an undetermined coefficient, $0 \le x_v \le 1$, $v = 1,2,\dots,n$. Then we can obtain all other $p_n(v,m)$ through recurrence relations. Finally, we have $n-1$ equilibrium equations unused, which are those of (4.7) when $m = 0$. Adding the normalizing condition, we get n linear equations to determine x_v. As a result, we can solve (4.8) only through solving n linear equations. This recursive solution method can be found in [7], which is effective in saving memory.

5. THE CASE OF UNLIMITED WAITING ROOM

In this section we study the birth-death model and the quasi-birth-death model in the unlimited waiting room case.

5.1. Birth-Death Model of the Unlimited Waiting Room Case

Consider an open system with the multiprogramming level n. Let V_n be the number of transactions in the system, i.e., V_n includes processing and waiting transactions. Let $p_n(v)$ be the equilibrium probability of the system. Suppose that $\lambda < t(n)$, then we have

$$\begin{cases} p_n(v) = \dfrac{\lambda^v}{\prod_{i=1}^{v} t(i)}\, p_n(0), & 1 \le v \le n, \\ p_n(v) = \left(\dfrac{\lambda}{t(n)}\right)^{n-v} \dfrac{\lambda^n}{\prod_{i=1}^{n} t(i)}\, p_n(0), & v \ge n, \end{cases} \tag{5.1}$$

where

$$p_n(0) = \left\{ \sum_{v=0}^{n-1} \frac{\lambda^v}{\prod_{i=1}^{v} t(i)} + \frac{t(n)}{t(n)-\lambda}\, \frac{\lambda^n}{\prod_{i=1}^{n} t(i)} \right\}^{-1}. \tag{5.2}$$

Let $\bar{V}_n$ be the expectation of V_n, which is the mean number of the transactions in the system. Then we have

$$\bar{V}_n = \sum_{v=1}^{n-1} v p_n(v) + \left(1 - \sum_{v=1}^{n-1} p_n(v)\right)\left(n + \frac{\lambda}{t(n)-\lambda}\right). \tag{5.3}$$

Hence, the mean value of throughput t_n, restart rate a_n and response time r_n of the unlimited waiting room system are

$$t_n = \sum_{v=1}^{n} t(v)\, p_n(v) + \frac{\lambda t(n)}{t(n)-\lambda}\, p_n(n), \quad a_n = \sum_{v=2}^{n} a(v)\, p_n(v) + \frac{\lambda a(n)}{t(n)-\lambda}\, p_n(n), \quad r_n = \frac{\bar{V}_n}{\lambda}. \tag{5.4}$$

LEMMA 1. *In unlimited waiting room case, if $t(n) < t(n+1)$, then $\mathbf{V}_n$ is stochastically larger than $\mathbf{V}_{n+1}$, i.e., $\mathbf{V}_n \ge_{st} \mathbf{V}_{n+1}$. If $t(n) > t(n+1)$, then $\mathbf{V}_n$ is stochastically smaller than $\mathbf{V}_{n+1}$, i.e., $\mathbf{V}_n \le_{st} \mathbf{V}_{n+1}$.*

PROOF. Suppose $t(n) < t(n+1)$. From (5.2), we know that

$$p_{n+1}^{-1}(0) - p_n^{-1}(0) = \frac{\lambda^n}{\prod_{i=1}^{n} t(i)} \left\{ \frac{\lambda \left(\frac{1}{t(n+1)} - \frac{1}{t(n)}\right)}{\left(1 - \frac{\lambda}{t(n+1)}\right)\left(1 - \frac{\lambda}{t(n)}\right)} \right\} \le 0. \tag{5.5}$$

Define the distribution function as $F_n(v) = \sum_{i=0}^{v} p_n(i)$. When $0 \le v \le n$, it follows from (5.1) and (5.5) that

$$F_{n+1}(v) - F_n(v) = \sum_{i=0}^{v} \frac{\lambda^i}{\prod_{j=1}^{i} t(j)}\, (p_{n+1}(0) - p_n(0)) \ge 0. \tag{5.6}$$

When $v > n$, we have

$$\begin{aligned} F_{n+1}(v) - F_n(v) &= \sum_{i=v+1}^{\infty} (p_n(i) - p_{n+1}(i)) \\ &= \frac{\lambda^n}{\prod_{j=1}^{n} t(j)} \sum_{i=v+1-n}^{\infty} \left\{ \left(\frac{1}{t(n)}\right)^i - \left(\frac{1}{t(n+1)}\right)^i \right\} \lambda^i \geq 0. \end{aligned} \tag{5.7}$$

Hence, for any v, $F_{n+1}(v) \geq F_n(v)$, which means $\mathbf{V}_n \geq_{st} \mathbf{V}_{n+1}$. For $t(n) > t(n+1)$, we can prove $\mathbf{V}_n \leq_{st} \mathbf{V}_{n+1}$ in the same way. ∎

Let n^* be the thrashing point of the closed queueing model. From Property (ii), $t(n)$ is monotone increasing in $[0, n^*]$ and monotone decreasing in $[n^*, \infty)$. For the same n^* we have

THEOREM 3. *In the case of unlimited waiting room, r_n is monotone decreasing in $[0, n^*]$ and monotone increasing in $[n^*, \infty)$.*

PROOF. If $t(n)$ is monotone increasing in $[0, n^*]$, then from Lemma 1, we know that for $n+1 < n^*$, $\mathbf{V}_n \geq_{st} \mathbf{V}_{n+1}$. Hence, $\bar{V}_n \geq \bar{V}_{n+1}$. Using the Little's formula, we obtain $r_n = \bar{V}_n/\lambda \geq \bar{V}_{n+1}/\lambda = r_{n+1}$, i.e., r_n is monotone decreasing in $[0, n^*]$. By the same way we can prove that r_n is monotone increasing in $[n^*, \infty)$. ∎

In the unlimited waiting room case, we can determine the thrashing point n^* with the respect to response time, that is, n^* attains the minimum of the mean number of transactions in the system.

5.2. Quasi-Birth-Death Model of the Unlimited Waiting Room Case

We consider (v, m) to be the state of the system, then in the case of infinite waiting room, the state space of the system is given by $\mathcal{SP}_4 = \{(v, m) \mid v \leq 0,\ 0 \leq m \leq Cv\}$. Transition rate matrix of the system $\mathbf{Q}_1$ has the structure of

$$\mathbf{Q}_1 = \begin{bmatrix} \mathbf{A}_{0,1} & \mathbf{A}_{0,2} & & & & & & \\ \mathbf{A}_{1,0} & \mathbf{A}_{1,1} & \mathbf{A}_{1,2} & & & & & \\ & & \ddots & & & & & \\ & & \mathbf{A}_{n-1,0} & \mathbf{A}_{n-1,1} & \mathbf{A}_{n-1,2} & & & \\ & & & \mathbf{A}_{n,0} & \mathbf{A}_{n,1} & \mathbf{A}_0 & & \\ & & & & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & \\ & & & & & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 \\ & & & & & & & \ddots \end{bmatrix}, \tag{5.8}$$

where the submatrix $\mathbf{A}_{v,i}$, $0 \leq v \leq n$, $0 \leq i \leq 2$, are given in Equations (4.10), (4.11) and (4.12). $\mathbf{A}_0$ is the arrival submatrix such that $\mathbf{A}_0 = \lambda\mathbf{I}$, where $\mathbf{I}$ is the unit matrix of order $[nC+1]$. $\mathbf{A}_1 = \mathbf{A}_{n,1}$ and $\mathbf{A}_2 = [\mathbf{A}_{n,0}, 0]$, the square submatrix of order $[nC+1]$.

Let $\mathbf{p}_n = (y_0, \mathbf{y}_1, \ldots, \mathbf{y}_{n-1}, \mathbf{x}_0, \mathbf{x}_1, \ldots)$ be the equilibrium probability of the system such that $\mathbf{p}_n \mathbf{Q}_1 = 0$. Then from the Neuts's method we know that $\mathbf{x}_k = \mathbf{x}_0 \mathbf{R}^k$, $k \geq 0$, where $\mathbf{R}$ is the minimal non-negative solution to the equation $\mathbf{R}^2\mathbf{A}_2 + \mathbf{R}\mathbf{A}_1 + \mathbf{A}_0 = 0$. Let $\mathbf{R}(i)$ be the successive substituted sequence such that $\mathbf{R}(i+1) = \mathbf{A}_0(\mathbf{A}_1 + \mathbf{R}(i)\mathbf{A}_2)^{-1}$, and $\mathbf{R}(0) = 0$. Then $\mathbf{R}(i)$ converges increasingly to $\mathbf{R}$ (see [8]). By using the Neuts's method, we solve the equilibrium probability of the system from following equations.

$$\begin{cases} y_0 A_{0,1} + \mathbf{y}_1 \mathbf{A}_{1,0} = 0, & \\ \mathbf{y}_{r-1}\mathbf{A}_{r-1,2} + \mathbf{y}_r \mathbf{A}_{r,1} + \mathbf{y}_{r+1}\mathbf{A}_{r+1,0} = 0, & 2 \leq r \leq n-1 \\ \mathbf{y}_{n-1}\mathbf{A}_{n-1,2} + \mathbf{x}_0(\mathbf{A}_1 + \mathbf{R}\mathbf{A}_2) = 0, & \\ y_0 + \mathbf{y}_1\mathbf{e} + \cdots + \mathbf{y}_{n-1}\mathbf{e} + \mathbf{x}_0(\mathbf{I} - \mathbf{R})^{-1}\mathbf{e} = 1. & \end{cases} \tag{5.9}$$

Replacing $\mathbf{A}_{n,1}$ by $\mathbf{A}_1 + \mathbf{R}\mathbf{A}_2$, we can obtain the matrix form of (5.9) from (4.9). Let $\mathbf{x}_0$ be an undetermined coefficient, we can obtain $\mathbf{y}_{n-1}, \ldots, y_0$ through recurrence relations. Since

$\mathbf{A}_1 + \mathbf{R}\mathbf{A}_2$ is no longer a skip-free-to-right matrix, we must solve $(n-1)C+1$ linear equations, from which $\mathbf{x}_0$ can be determined. By solving (5.9), we obtain the equilibrium distribution of the system and the mean number of the transactions $\bar{V}_n$.

$$\bar{V}_n = \sum_{i=1}^{n-1} i\mathbf{y}_i\mathbf{e} + n\mathbf{x}_0(\mathbf{I}-\mathbf{R})^{-1}\mathbf{e} + \mathbf{x}_0\mathbf{R}(\mathbf{I}-\mathbf{R})^{-2}\mathbf{e} \tag{5.10}$$

Hence, we can get the mean value of throughput t_n, restart rate a_n and response time r_n by

$$t_n = \sum_{v=1}^{n} t(v)\,\mathbf{y}_v\mathbf{e}, \qquad a_n = \sum_{v=2}^{n} a(v)\,\mathbf{y}_v\mathbf{e}, \qquad r_n = \frac{\bar{V}_n}{\lambda}, \tag{5.11}$$

where $\mathbf{y}_n = \mathbf{x}_0(\mathbf{I}-\mathbf{R})^{-1}$.

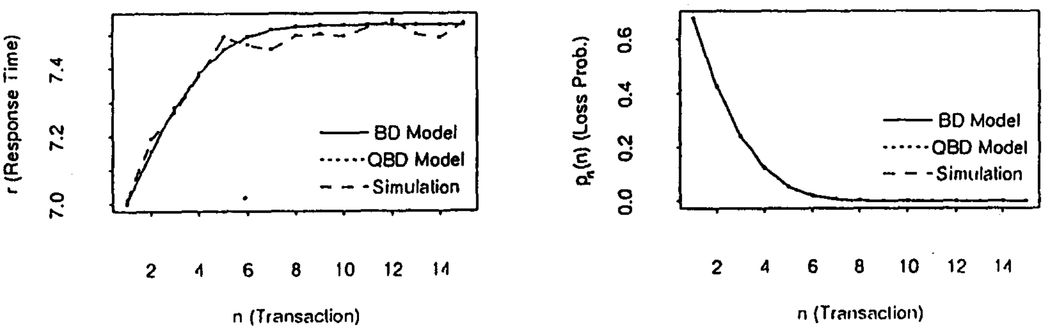

Figure 1. The no waiting room case, with $D = 500$, $C = 6$, $\mu = 1.0$, $\lambda = 0.3$, $p_w = 1.0$, $b = 0.7$, $c = 0.3$ and in fixed length transactions.

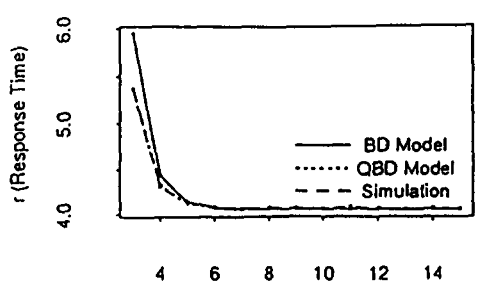

Figure 2. The unlimited waiting room case, with $D = 300$, $C = 5$, $\mu = 1.0$, $\lambda = 0.5$, $p_w = 1.0$, $b = c = 0.5$ and uniform distribution in length of transactions.

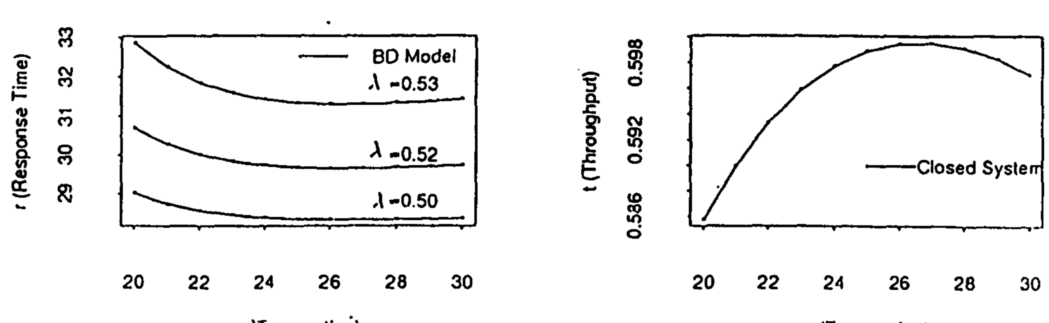

Figure 3. The unlimited waiting room case, with $D = 1950$, $C = 12$, $\mu = 1.0$, $p_w = 1.0$, $b = 0.8$, $c = 0.2$ and fixed length transactions.

6. NUMERICAL RESULTS

In this section we investigate numerical results of analytic model and simulation. Simulation results are the expectations of 10 runs of 20,000 locking requests. In graphs we cannot find the difference between the results of birth-death models and quasi-birth-death models. No waiting room cases are given in Figure 1. Our analytic models fit well with the simulation ones. As proved in Theorem 1 and Theorem 2, the trade off between the loss probability and response time are given. The unlimited waiting case of our analytic models is given in Figure 2 and Figure 3, in which the fitness of our analytic models and the property of Theorem 3 are shown.

REFERENCES

1. Y.C. Tay, N. Goodman and R. Suri, Locking performance in centralized database, *ACM Transactions on Database Systems* **10** (4), 415–462 (1985).
2. Y.C. Tay, R. Suri and N. Goodman, A mean value performance model for locking in database: The no-waiting case, *Jour. of the Association for Computing Machinery* **32** (3), 618–651 (1985).
3. S. Nishimura and Y. Jiang, An analytic model for locking performance in database, *Discussion Paper Series in Institute of Socio-Economic Planning, University of Tsukuba*, p. 443, (1990).
4. D. Mitra, Probabilistic models and asymptotic results for concurrent processing with exclusive and non-exclusive locks, *SIAM Journal of the Computers* **14** (4), 1030–1051 (1985).
5. D. Mitra and P.J. Weinberger, Probabilistic models of database locking: Solutions, computational algorithms, and asymptotics, *Jour. of the Association for Computing Machinery* **31** (4), 855–878 (1984).
6. I.K. Ryu and A. Thomasian, Analysis of database performance with dynamic locking, *Jour. of the Association for Computing Machinery* **37** (3), 491–523 (1990).
7. C.H. Sauer and K.M. Chandy, *Computer Systems Performance Modeling*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, (1981).
8. M.F. Neuts, *Matrix-Geometric Solutions in Stochastic Models*, The Johns Hopkins University Press, Baltimore, (1981).