http://www.zju.edu.cn/jzus; E-mail: jzus@zju.edu.cn
ISSN 1009-3095 *Journal of Zhejiang University SCIENCE* V.5, No.1, P.31-39, Jan., 2004

Using multi-class queuing network to solve performance models of e-business sites*

ZHENG Xiao-ying (郑小盈), CHEN De-ren (陈德人)

(*College of Computer Science, Zhejiang University, Hangzhou 310027, China*)

E-mail: zhengbetty@hotmail.com; drchen@zju.edu.cn

Received Sept.5, 2002; revision accepted Jan.30, 2003

Abstract: Due to e-business's variety of customers with different navigational patterns and demands, multi-class queuing network is a natural performance model for it. The open multi-class queuing network (QN) models are based on the assumption that no service center is saturated as a result of the combined loads of all the classes. Several formulas are used to calculate performance measures, including throughput, residence time, queue length, response time and the average number of requests. The solution technique of closed multi-class QN models is an approximate mean value analysis algorithm (MVA) based on three key equations, because the exact algorithm needs huge time and space requirement. As mixed multi-class QN models, include some open and some closed classes, the open classes should be eliminated to create a closed multi-class QN so that the closed model algorithm can be applied. Some corresponding examples are given to show how to apply the algorithms mentioned in this article. These examples indicate that multi-class QN is a reasonably accurate model of e-business and can be solved efficiently.

Key words: Queuing network (QN), Multi-class, Performance, E-business
Document code: A **CLC number:** TP393.0

INTRODUCTION

E-Business has become an important means of transacting business spanning a wide range of interactions, including business-to-business, business-to-consumer, and individual-to-individual (Zhong *et al.*, 2002). While issues related to creating Web Sites strongly drove e-commerce's first phase, the inclusion of performance analysis will drive the next phase. The goal of this article is to provide techniques regarding how performance models can be constructed, solved, and used in the context of electronic business environments. There are many types of performance models and various analytic solution techniques (Kleinrock, 1975; Menasce and Almeida, 1998). This article will introduce one potential model, which is based on a variant of queuing theory (Kleinrock, 1975) called "operational analysis" (Menasce and Almeida, 1998; Buzen, 1978; Denning and Buzen, 1978), and focus on using multi-class queuing network (QN), including open multi-class models, closed models and mixed models, to solve performance models of e-business sites.

This article is organized as follows. Section 2 will introduce background of using multi-class QN to solve performance models of e-business sites. Section 3 discusses open multi-class QN solution. Section 4 describes closed multi-class QN model in detail. Section 5 will combine open and closed model to set up a mixed one. Section 6 is summaries and hints at our future work.

BACKGROUND OF USING MULTI-CLASS QN TO SOLVE PERFORMANCE MODELS OF E-BUSINESS SITES

Performance is a key issue in electronic business. Performance models help us understand the quantitative behavior of complex electronic commerce applications. After a suitable model has been developed and its parameters have been

* Project (No.2002AA414070) supported by the High-Tech. Research & Development Program (863) of China

determined, analytic techniques can be used to solve the model to get results, i.e., the performance metrics, and even to predict future system performance. When choosing a solution technique, simplicity, accuracy, computational cost and availability of information about the system under analysis should be considered. Because queuing network models achieve a favorable balance between accuracy and efficiency, they become suitable models of e-business sites. And since customers of e-business sites exhibit different navigational patterns with different demands on the site resources, the transactions associated with each e-business function which are the basic components of the e-business workload differs from each other. So it is impractical to represent all transactions in a single class. Thus multi-class QN model becomes the natural choice. Open queues and closed queues differ in that the former do not place any limits on the maximum number of request present in the system. There exist situations in which systems should be modeled with a fixed and finite number of requests. Therefore we need to study both open and closed multi-class QN models.

Some fundamental concepts must be introduced before we discuss in detail modeling e-business sites with multi-class QN.

The term queue stands for a resource (e.g., processor, I/O and network) and the requests waiting to use the resource. The resources can be categorized into two groups:

1. Load-dependent resources are used to represent resources where queuing and the average service time depend on the load, i.e., the current queue length.

2. Load-independent resources represent resources whose average service time is independent of the current load. These resources include queuing resources and delay resources. Requests at a queuing resource are queuing for the use of the resource. But the average service time does not depend on the load, and $S(n) = S$ for all values of n, where $S(n)$ is the service time of request n spent at a resource. Requests at a delay resource are each allocated their own server, so these is no queuing. And the average service time does not depend on the number of requests present at the resource, $S(n) = S$ for all values of n.

APPLYING OPEN MULTI-CLASS QN MODEL TO E-BUSINESS SITES PERFORMANCE ANALYSIS

Open multi-class QN model (Lazowska *et al.*, 1984)

Let C be the number of classes and K be the number of service centers in the model. Each class c is an open class with arrival rate λ_c. Denote the vector of arrival rates by $\boldsymbol{\lambda} \equiv (\lambda_1, \lambda_2, \cdots, \lambda_C)$. The inequality (1) should satisfactorily guarantee no service center is saturated as a result of the combined loads of all the classes.

$$\max_k \{\sum_{c=1}^{C} \lambda_c D_{c,k}\} < 1 \qquad (1)$$

where $D_{c,k}$ is the service demand of class c request at center k.

The following are the formulas used to calculate performance measures.

1. By the forced flow law the throughput of class c at center k as a function of $\boldsymbol{\lambda}$ is:

$$X_{c,k} = \lambda_c V_{c,k} \qquad (2)$$

where $V_{c,k}$ is the visit count of class c request to service center k. Therefore, the throughput of class c in the network is:

$$X_c(\boldsymbol{\lambda}) = \sum_{k=1}^{K} \lambda_c V_{c,k} = \lambda_c \qquad (3)$$

2. Utilization:

$$U_{c,k}(\boldsymbol{\lambda}) = X_{c,k}(\boldsymbol{\lambda}) S_{c,k} = \lambda_c D_{c,k} \qquad (4)$$

where $S_{c,k}$ is average service time of class c request at service center k per visit to the center.

3. Residence time

$$R_{c,k}(\boldsymbol{\lambda}) = \begin{Bmatrix} D_{c,k} & \text{(delay)} \\ D_{c,k}[1 + A_{c,k}(\boldsymbol{\lambda})] & \text{(queuing)} \end{Bmatrix}$$

where $A_{c,k}(\boldsymbol{\lambda})$ is the average number of requests seen at queue k by an arriving class c request. According to the Arrival Theorem, for queuing centers $R_{c,k}(\boldsymbol{\lambda}) = D_{c,k}[1 + Q_k(\boldsymbol{\lambda})]$ where $Q_k(\boldsymbol{\lambda})$ is the time averaged queue length at center k. Applying Little's law (Little, 1961), $R_{c,k}(\boldsymbol{\lambda}) = D_{c,k}[1 + \sum_{j=1}^{C} \lambda_j R_{j,k}(\boldsymbol{\lambda})]$. After transforming the equation, we get

$$R_{c,k}(\boldsymbol{\lambda}) = \frac{D_{c,k}}{1 - \sum_{j=1}^{C} \lambda_j D_{j,k}} = \frac{D_{c,k}}{1 - \sum_{j=1}^{C} U_{j,k}(\boldsymbol{\lambda})}.$$

So we have:

$$R_{c,k}(\boldsymbol{\lambda}) = \begin{cases} D_{c,k} & \text{(delay)} \\ \dfrac{D_{c,k}}{1 - \sum_{j=1}^{C} U_{j,k}(\boldsymbol{\lambda})} & \text{(queuing)} \end{cases} \quad (5)$$

4. Applying Little's law to the residence time Eq. (5), the queue length of class c at center k, $Q_{c,k}(\boldsymbol{\lambda})$, is:

$$Q_{c,k}(\boldsymbol{\lambda}) = \lambda_c R_{c,k}(\boldsymbol{\lambda}) = \begin{cases} U_{c,k} & \text{(delay)} \\ \dfrac{U_{c,k}(\boldsymbol{\lambda})}{1 - \sum_{j=1}^{C} U_{j,k}(\boldsymbol{\lambda})} & \text{(queuing)} \end{cases} \quad (6)$$

5. The response time for a class c request, $R_c(\boldsymbol{\lambda})$, is the sum of its residence times at all service centers:

$$R_c(\boldsymbol{\lambda}) = \sum_{k=1}^{K} R_{c,k}(\boldsymbol{\lambda}) \quad (7)$$

6. The average number of class c requests in system, $Q_c(\boldsymbol{\lambda})$, is the sum of the class c queue lengths at all centers:

$$Q_c(\boldsymbol{\lambda}) = \lambda_c R_c(\boldsymbol{\lambda}) = \sum_{k=1}^{K} Q_{c,k}(\boldsymbol{\lambda}) \quad (8)$$

A simple example

Let us consider an online bookstore site composed of one Web server, one application server and one database server. The IT infrastructure of the site is shown in Fig. 1. The Web server provides static HTML pages to customers, collects data, and transfers them to the application server. The application server gets requests from the Web server, parses them, and activates the processes that carry out the requested e-business function (e.g., Book Selection, Personal Data Entry, Deliver Infor, Hold Book, View Status, Cancel Orders and Extended Services).

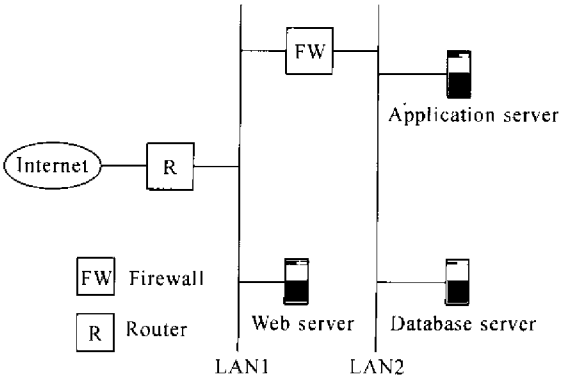

Fig. 1 IT infrastructure of the online bookstore

Because these e-business functions have different service demand and average arrival rate, different classes should represent them. And this site is accessible to a very large population, which is referred to as the infinite population case, so the servers can be viewed as receiving requests of these e-business functions with corresponding average arrival rates. Therefore, it is suitable to set up an open multi-class QN model to represent the site's behaviour and caculate performance measures. The QN model is shown in Fig. 2.

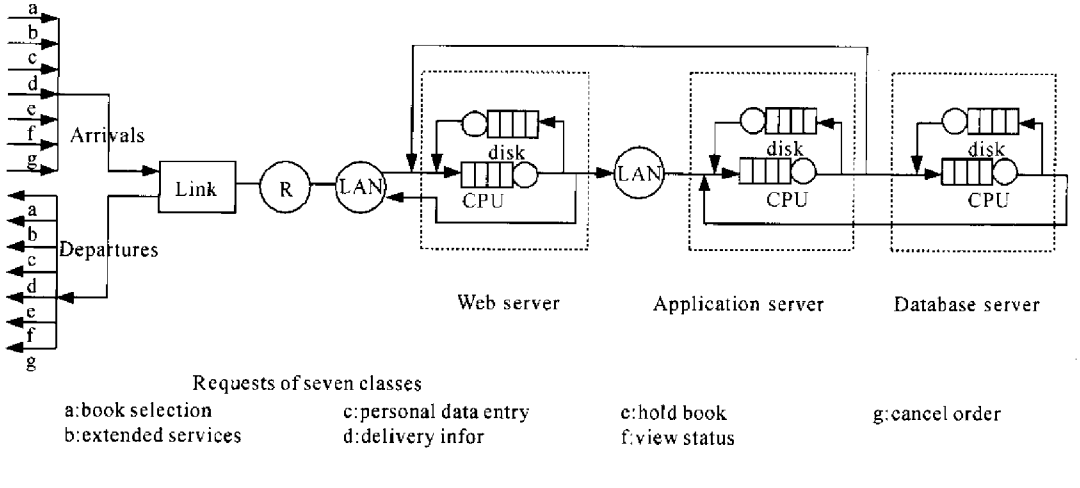

Fig. 2 Open multi-class QN of the online bookstore

The input parameters of the QN model, request arrival rates and service demands are obtained through running a series of these e-business functions, shown in Table 1. Eqs.(1) – (8) are used to calculate the performance measures of these e-business functions and the results are shown in Table 2 – Table 4.

This model can also be used to predict the performance of the site. Let us continue the example. The management of the site conducted an advertisement campaign in TV, newspapers and magazines, and promised to provide a big discount. Due to the campaign, the traffic volumes of the site will grow largely. Then what will the response time and utilization be if the current arrival rate grew by a factor of three? Applying Eqs.(1) – (8) again, we get utilization and response time in Table 3 – Table 6.

Table 1 Request arrival rate and service demand (msec) of the online bookstore (Menasce and Almeida, 2000)

E-business function	Request arrival rate (req/s)	WS CPU	WS IO	AS CPU	AS IO	DB CPU	DB IO	LAN 1	LAN 2	Link to ISP
Book selection	5.556	5.2	9.5	25.0	15.0	20.0	40.0	0.492	0.532	16.4
Extended services	2.228	5.1	8.4	12.0	10.0	13.0	20.0	0.295	0.492	11.5
Personal data entry	0.248	32.0	15.0	16.0	30.0	0.0	0.0	0.655	0.000	32.8
Delivery info	0.186	32.0	14.0	18.0	24.0	0.0	0.0	0.410	0.000	19.1
Hold book	0.162	31.0	15.0	35.0	90.0	30.0	80.0	0.819	0.901	43.7
View status	0.025	4.8	6.7	12.0	0.0	11.0	15.0	0.205	0.229	27.3
Cancel order	0.009	5.2	7.1	13.0	0.0	18.0	30.0	0.246	0.262	30.0

Table 2 Utilization of the online bookstore

E-business function	WS CPU	WS IO	AS CPU	AS IO	DB CPU	DB IO	LAN 1	LAN 2	Link to ISP
Book selection	0.028891	0.052782	0.138900	0.083340	0.111120	0.222240	0.002731	0.002958	0.091030
Extended services	0.011363	0.018715	0.026736	0.022280	0.028964	0.044560	0.000657	0.001095	0.025552
Personal data entry	0.007936	0.003720	0.003968	0.007440	0.000000	0.000000	0.000163	0.000000	0.008126
Delivery info	0.005952	0.002604	0.003348	0.004464	0.000000	0.000000	0.000076	0.000000	0.003555
Hold book	0.005022	0.002430	0.005670	0.014580	0.004860	0.012960	0.000133	0.000146	0.007078
View status	0.000120	0.000168	0.000300	0.000000	0.000275	0.000375	0.000005	0.000006	0.000683
Cancel order	0.000047	0.000064	0.000117	0.000000	0.000162	0.000270	0.000002	0.000002	0.000270
Utilization	0.059331	0.080483	0.179039	0.132104	0.145381	0.280405	0.003767	0.004208	0.136295

Table 3 Residence time (msec) of the online bookstore

E-business function	WS CPU	WS IO	AS CPU	AS IO	DB CPU	DB IO	LAN 1	LAN 2	Link to ISP	Response Time
Book selection	5.35	10.03	29.03	16.36	22.5	51.43	0.49	0.53	18.02	153.76
Extended services	5.16	8.56	12.33	10.23	13.39	20.93	0.30	0.49	11.77	83.15
Personal data entry	32.26	15.06	16.06	30.22	0.00	0.00	0.66	0.00	33.04	127.29
Delivery info	32.19	14.04	18.06	24.11	0.00	0.00	0.41	0.00	19.18	107.99
Hold book	31.16	15.04	35.2	91.33	30.15	81.05	0.82	0.90	44.00	329.64
View status	4.80	6.70	12.00	0.00	11.00	15.01	0.20	0.23	27.33	77.27
Cancel order	5.20	7.10	13.00	0.00	18.00	30.01	0.25	0.26	30.05	103.87

Table 4 Queue length (req.[a]) of the online bookstore

E-business function	WS CPU	WS IO	AS CPU	AS IO	DB CPU	DB IO	LAN 1	LAN 2	Link to ISP
Book selection	0.029751	0.055723	0.161305	0.090917	0.125011	0.285744	0.002738	0.002967	0.100146
Extended services	0.011493	0.019072	0.027470	0.022788	0.029828	0.046638	0.000657	0.001096	0.026223
Personal data entry	0.007999	0.003734	0.003984	0.007496	0.000000	0.000000	0.000163	0.000000	0.008193
Delivery info	0.005988	0.002611	0.003359	0.004484	0.000000	0.000000	0.000076	0.000000	0.003568
Hold book	0.005047	0.002436	0.005702	0.014796	0.004884	0.013130	0.000133	0.000146	0.007128
View status	0.000120	0.000168	0.000300	0.000000	0.000275	0.000375	0.000005	0.000006	0.000683
Cancel order	0.000047	0.000064	0.000117	0.000000	0.000162	0.000270	0.000002	0.000002	0.000270
Queue Length	0.060445	0.083807	0.202238	0.140480	0.160160	0.346157	0.003775	0.004218	0.146211

[a]. the average requests in system

Table 5 Utilization of the online bookstore

E-business function	WS CPU	WS IO	AS CPU	AS IO	DB CPU	DB IO	LAN 1	LAN 2	Link to ISP
Book selection	0.101119	0.184737	0.486150	0.291690	0.388920	0.777840	0.009558	0.010355	0.318603
Extended services	0.039770	0.065503	0.093576	0.077980	0.101374	0.155960	0.002300	0.003833	0.089434
Personal data entry	0.027776	0.013020	0.013888	0.026040	0.000000	0.000000	0.000569	0.000000	0.028443
Delivery info	0.020832	0.009114	0.011718	0.015624	0.000000	0.000000	0.000267	0.000000	0.012444
Hold book	0.017577	0.008505	0.019845	0.051030	0.017010	0.045360	0.000464	0.000511	0.024773
View status	0.000420	0.000586	0.001050	0.000000	0.000963	0.001313	0.000018	0.000020	0.002389
Cancel order	0.000164	0.000224	0.000410	0.000000	0.000567	0.000945	0.000008	0.000008	0.000946
Utilization	0.207658	0.281689	0.626637	0.462364	0.508834	0.981418	0.013183	0.014727	0.477031

Table 6 Residence time (msec) of the online bookstore

E-business function	WS CPU	WS IO	AS CPU	AS IO	DB CPU	DB IO	LAN 1	LAN 2	Link to ISP	Response Time
Book selection	5.78	11.65	48.65	21.18	32.73	180.05	0.50	0.54	24.04	325.13
Extended services	5.31	8.99	13.24	10.85	14.47	23.70	0.30	0.49	12.60	89.93
Personal data entry	32.91	15.20	16.23	30.80	0.00	0.00	0.66	0.00	33.73	129.52
Delivery info	32.68	14.13	18.21	24.38	0.00	0.00	0.41	0.00	19.36	109.17
Hold book	31.55	15.13	35.71	94.84	30.52	83.80	0.82	0.90	44.80	338.07
View status	4.80	6.70	12.01	0.00	11.01	15.02	0.20	0.23	27.37	77.36
Cancel order	5.20	7.10	13.01	0.00	18.01	30.03	0.25	0.26	30.07	103.92

The utilization of DB I/O reaches 98%, the response time of book selection increases by 211% and DB I/O has become the bottleneck of the site. If the arrival rate grows continually, the DB I/O will be saturated, and the performance will be damaged. Therefore, the management should take action to improve the performance due to the ad campaign.

APPLYING CLOSED MULTI-CLASS QN MODEL TO E-BUSINESS SITES PERFORMANCE ANALYSIS

Closed multi-class QN model (Lazowska *et al.*, 1984)

Let C be the number of classes and K be the number of service centers in the closed model. Each class c is a closed class with a fixed popu-

lation size N_c. Denote the workload intensity by $\boldsymbol{N} \equiv (N_1, \cdots, N_C)$. The solution technique discussed is a mean value analysis (MVA) algorithm (Reiser and Lavenberg, 1980), which relies on three key equations:

1. Applying Little's law to the whole QN

$$X_c(\boldsymbol{N}) = \frac{N_c}{Z_c + \sum_{k=1}^{K} R_{c,k}(\boldsymbol{N})} \quad (9)$$

where Z_c is the think time of class c request.

2. Applying Little's law to the service centers

$$Q_{c,k}(\boldsymbol{N}) = X_c(\boldsymbol{N}) R_{c,k}(\boldsymbol{N}) \quad (10)$$

thus the total queue length at center c:

$$Q_k(\boldsymbol{N}) = \sum_{c=1}^{C} Q_{c,k}(\boldsymbol{N}) \quad (11)$$

3. The residence time at service center is

$$R_{c,k}(\boldsymbol{N}) = \begin{cases} D_{c,k} & (\text{delay}) \\ D_{c,k}[1 + A_{c,k}(\boldsymbol{N})] & (\text{queuing}) \end{cases} \quad (12)$$

where $A_{c,k}(\boldsymbol{N})$ is the arrival instant queue length at center k seen by an arriving class c request. There are two methods to compute $A_{c,k}(\boldsymbol{N})$, exact and approximate. However, the exact algorithm needs huge time and space requirement, which makes the exact one impractical. Therefore, we study the approximate solution to solve closed multi-class QN model.

The approximate algorithm estimates the arrival instant queue lengths based on the assumption that the removal of a request from the network does not affect the placement of requests in other classes, and only reduces queue lengths in its own class in proportion to their original size.

$$A_{c,k}(\boldsymbol{N}) = Q_k(\boldsymbol{N} - 1_c) \approx h_c[Q_{1,k}(\boldsymbol{N}), \cdots, Q_{C,k}(\boldsymbol{N})] \equiv \left[\frac{N_c - 1}{N_c} Q_{c,k}(\boldsymbol{N})\right] + \sum_{j=1}^{C} Q_{j,k}(\boldsymbol{N}) \quad (13)$$

Using Eq. (13), we get the approximate algorithm, shown as follows.

Algorithm 1 Approximate MVA solution for closed multi-class QN

1. Set $Q_{c,k}(\boldsymbol{N}) \leftarrow \frac{N_c}{K}$ for all c, k
2. Approximate $A_{c,k}(\boldsymbol{N})$ by Eq. (13), for all c, k
3. Apply Eqs. (9) – (11) to compute a new set of $Q_{c,k}(\boldsymbol{N})$ for all c, k
4. If the $Q_{c,k}(\boldsymbol{N})$ resulting from Step 3 do not agree to within some tolerance (e.g., 0.1%) with those as inputs in Step 2, return to Step 2 using the new $Q_{c,k}(\boldsymbol{N})$.

In this process, we initialize a supposed value for time averaged queue lengths, iteratively apply Eq. (13) to the approximate arrival instant queue lengths until the difference between the successive estimates of time averaged queue lengths and the previous values are within the tolerance. This means estimates of time averaged queue lengths are sufficiently closed. The storage requirement is proportional to CK. The time requirement is based on the number of iterations. The number of operations required per iteration is proportional to CK. And typically, the convergence to less than a 0.1% change in queue lengths requires less than two dozens of iterations. The accuracy is typically within a few percent of the exact solution for throughputs and utilization, and within 10% for queue lengths and residence times (Lazowska *et al.*, 1984).

A simple example

Let us continue the online bookstore example. We have mentioned DB I/O can be the bottleneck of the site, and now we analyze the DB server in detail. Every request to the database server requires the use of a database connection. As traffic to an e-business site grows, a common bottleneck is database connectivity, which may not scale up to handle thousands of requests simultaneously. Therefore there is a maximum number of request limits in DB server. It is more suitable to use a closed QN to model the DB server. And we will use the approximate technique to solve the model.

Assume the DB server is composed of one processor and one disk. The service demand for basic database transactions, Search, Insert, Update, and Delete is shown in Table 7. The maximum numbers of simultaneous connections of these four transactions are set to 10, 2, 2, and 1 respectively. Here, we ignore the think time of class c request, i.e., $Z_c = 0$.

Using the Algorithm 1, we get the performance measures of the DB Server, shown in Table 8 – Table 10.

Table 7 Service demand of DB Server (msec) (Menasce and Almeida, 2000)

	Search	Insert	Update	Delete
DB server CPU	20.00	30.00	25.00	18.00
DB server I/O	40.00	80.00	55.00	30.00

Table 8 Queue length of DB Server (req.)

	Search	Insert	Update	Delete	Queue Length
DB server CPU	0.60	0.09	0.11	0.07	0.87
DB server I/O	9.40	1.91	1.89	0.93	14.13

Table 9 Residence time of DB Server (sec.)

	Search	Insert	Update	Delete
DB server CPU	0.03630	0.05487	0.04550	0.03248
DB server I/O	0.56740	1.13369	0.77990	0.42587
Response Time	0.60370	1.18856	0.82540	0.45835

Table 10 Throughput of DB Server (req./sec.)

	Search	Insert	Update	Delete
Throughput per class	16.56	1.68	2.42	2.18

Using Utilization formula $U_{c,k}(\boldsymbol{\lambda}) = X_{c,k}(\boldsymbol{\lambda})D_{c,k}$, utilization is caculated, shown in Table 11.

Table 11 Utilization of DB Server

	Search	Insert	Update	Delete	Utilization
DB server CPU	0.33129	0.05048	0.06058	0.03927	0.48162
DB server I/O	0.66258	0.13462	0.13327	0.06545	0.99592

It is obvious that the I/O of the DB server is the bottleneck. Search transaction consumes 66% of I/O time and becomes the major obstacle of the simultaneous level improvement. We will resolve the problem in the next section.

APPLYING MIXED MULTI-CLASS QN MODEL TO E-BUSINESS SITES PERFORMANCE ANALYSIS

Mixed multi-class QN models (Lazowska *et al.*, 1984)

Mixed queuing network models are those in which some classes are open and some are closed. Denote the workload intensity vector of the entire model by $\boldsymbol{I} \equiv (N_1 \text{ or } \lambda_1, N_2 \text{ or } \lambda_2, \cdots, N_C \text{ or } \lambda_C)$. Mixed model solution is described in Algorithm 2.

Algorithm 2 MVA solution for mixed multi-class QN

Let $\{O\}$ be the set of open classes and $\{C\}$ the set of closed classes.

1. For each center k, apply the forced flow law and the utilization law to obtain its utilization by each open class:

$$U_{c,k}(\boldsymbol{I}) = \lambda_c D_{c,k} \quad c \in \{O\} \tag{14}$$

and its total utilization by all open classes:

$$U_{\{O\},k}(\boldsymbol{I}) = \sum_{c \in \{O\}} \lambda_c D_{c,k} \tag{15}$$

2. Remove the open classes and solve the closed model consisting of the K centers and the remaining closed classes. Compute the new service demand $D^*_{c,k}$ of each class $c \in \{C\}$ at each center k in the closed model using equation:

$$D^*_{c,k} = \frac{D_{c,k}}{1 - U_{\{O\},k}(\boldsymbol{I})}, \quad c \in \{C\} \tag{16}$$

where $D_{c,k}$ is the service demand of class c at center k in the original mixed model. Apply the approximate algorithm to $D^*_{c,k}$ and get the throughputs, queue length, and residence times, which are the performance measures for the corresponding closed classes in the mixed models. Utilization can be computed by applying the utilization law to the original set of service demands $D_{c,k}$.

3. Compute residence time and queue lengths for the open classes using equations:

$$R_{c,k}(\boldsymbol{I}) = \frac{D_{c,k}[1 + Q_{\{C\},k}(\boldsymbol{I})]}{1 - U_{\{O\},k}(\boldsymbol{I})} \quad c \in \{O\} \tag{17}$$

$$Q_{c,k}(\boldsymbol{I}) = \lambda_c R_{c,k}(\boldsymbol{I}) \quad c \in \{O\} \tag{18}$$

where $Q_{\{C\},k}(\boldsymbol{I})$ is the total queue length of all closed classes at center k obtained in Step 2.

In Step 2 of Algorithm 2, we eliminate the open classes and create a closed multi-class QN model. Thus we have to use Eq. (18) to compute the new service demand of closed classes by eliminating the effect of the open classes. The factor $1 - U_{\{O\},k}$ is the percentage of time that the processor is not used by the open classes.

A simple example

Let us continue the online bookstore example. We have noticed in the closed queuing network example that the I/O of the database server is the bottleneck, and Search transcation consumes 66% of I/O. Because most of the Search transaction involves that the search for book information, and the search for customers and their orders comprises only a minor part. The database is separated into two parts, one for book information and the other for customers and their orders. Now a second disk is added to the database server. The first disk keeps the database of information on books, and the second is the database of customers and their orders. Because book information can be constant in a period, and maintance of this database can be executed in a special period, we can assume the transaction executed in the first disk is only Search transaction. And the mechanism makes the Search transaction in the first disk be executed at a very high simultaneous level, so the first disk can be represented by an open queue. However, Select, Insert, Update and Delete transactions are executed in the second disk, which has a limit on the number of the transactions executed simultaneously, and can be modeled by a closed queue. So a mixed model is set up to calculate the performance measures. The service Demand is shown in Table 12.

Table 12 Service demand of DB Server (msec) (Menasce and Almeida, 2000)

	Search(disk1)	Search(disk2)	Insert(disk2)	Update(disk2)	Delete(disk2)
DB CPU	22.00	20.00	35.00	30.50	16.30
DB Disk1 I/O	17.50	0.00	0.00	0.00	0.00
DB Disk2 I/O	0.00	40.00	80.00	55.00	30.00

The request arrival rate of Search transaction in disk1 is 12.8 req/s. The maximum number of simultaneous connections of Search, Insert, Update and Delete transaction in disk2 are set to 10, 3, 5, and 2 respectively. Algorithm 2 is used to evaluate the performance measures.

1. Compute the total utilization of the devices by the open classes:

$$U_{\{0\},\mathrm{CPU}}(I) = \lambda_{\mathrm{Search1}} D_{\mathrm{Search1,CPU}} = 0.2816$$
$$U_{\{0\},\mathrm{Disk1}}(I) = \lambda_{\mathrm{Search1}} D_{\mathrm{Search1,Disk1}} = 0.224$$
$$U_{\{0\},\mathrm{Disk2}}(I) = \lambda_{\mathrm{Search1}} D_{\mathrm{Search1,Disk2}} = 0$$

2. Eliminate the open classes, compute the new service demand of closed classes, and get the closed model. The new service demand of the closed classes is shown in Table 13.

Table 13 Service demand of closed classes (msec)

	Search (disk2)	Insert (disk2)	Update (disk2)	Delete (disk2)
DB CPU	27.84	48.72	42.46	22.69
DB Disk1 I/O	0.00	0.00	0.00	0.00
DB Disk2 I/O	40.00	80.00	55.00	30.00

Using the new Service Demand, the closed model can be used to calculate the performance measures, shown as follows. Here, we still ignore the think time of class c request, i.e., $Z_c = 0$.

Table 14 Queue length of closed classes in DB Server (req.)

	Search (disk2)	Insert (disk2)	Update (disk2)	Delete (disk2)	Queue Length
DB CPU	1.05	0.28	0.57	0.22	2.12
DB Disk1 I/O	0.00	0.00	0.00	0.00	0.00
DB Disk2 I/O	8.95	2.72	4.43	1.78	17.88

Table 15 Residence time of closed classes in DB Server (sec.)

	Search (disk2)	Insert (disk2)	Update (disk2)	Delete (disk2)
DB CPU	0.08401	0.14757	0.12771	0.06829
DB Disk1 I/O	0.00000	0.00000	0.00000	0.00000
DB Disk2 I/O	0.71930	1.43768	0.98957	0.53971
Response Time	0.80331	1.58525	1.11728	0.60800

Table 16 Throughput of closed classes in DB Server (req./s)

	Search (disk2)	Insert (disk2)	Update (disk2)	Delete (disk2)
Throughput per Class	12.45	1.89	4.48	3.29

Applying the utilization law $U_{c,k}(\vec{\lambda}) = X_{c,k}(\vec{\lambda})D_{c,k}$ to the original set of service demands, utilization is computed as shown in Table 17.

Table 17 Utilization of DB Server

	Search(disk1)	Search(disk2)	Insert(disk2)	Update(disk2)	Delete(disk2)	Utilization
DB CPU	0.2816	0.2490	0.0662	0.1365	0.0536	0.7869
DB Disk1 I/O	0.2240	0.0000	0.0000	0.0000	0.0000	0.2240
DB Disk2 I/O	0.0000	0.4979	0.1514	0.2461	0.0987	0.9942

3. Using the queue lengths of the closed classes, compute the performance measures of the open classes. The result is shown in Table 18 and Table 19.

Table 18 Residence time of open class in DB Server (msec)

	DB CPU	DB disk1 I/O	DB disk2 I/O
Search(disk1)	95.55	76.00	0.00

Table 19 Queue length of open class in DB Server (req.)

	DB CPU	DB disk1 I/O	DB disk2 I/O
Search(disk1)	1.22	0.97	0.00

From the result, the utilization of the database CPU has improved greatly. Now the utilization of the second disk's I/O is still near 100%, which however has no effect on the Search transaction in the first disk. So the system can permit the request arrival rate of the Search transaction in the first disk to grow in peak hour to some degree, and it will not damage the performance of the site.

SUMMARY AND FUTURE WORK

In this article we have focused on using multi-class QN to solve performance model for e-business sites. Multi-class QN is a reasonably accurate model of e-business and can be solved efficiently. Although multi-class models are more useful and natural for describing workloads of real e-business sites, they present some difficulties to the modelers. By identifying distinct workload components, input parameter values are required for each individual class. This typically requires considerable additional effort over that for a single class model, as monitoring tools often do not provide measurements on a per-class basis. So it needs future study.

Currently, we are using QN models to analysis an e-business site in our campus network. Our future work is to set up more powerful measurement tools to monitor the site. Thus we will get more sufficient and accurate information about resource consumption by transaction classes and push our study of performance modeling of e-business sites into further phase.

References

Buzen, J. P., 1978. "Operational Analysis: An Alternative to Stochastic Modeling" in Performance of Computer Installations. North Holland, p.175 – 194.

Denning, P. and Buzen, J. P., 1978. The operational analysis of queuing network models. *Computing Surveys*, **10**(3):225 – 261.

Kleinrock, L., 1975. Queuing Systems. Vol. I: Theory, Wiley, New York, p. 69 – 97.

Lazowska, E. D., Zahorjan, J., Graham, G.. S. and Sevcik, K. C., 1984. Quantitative System Performance: Computer System Analysis Using Queuing Network Models. Prentice-Hall, Englewood Cliffs, New Jersey 07632, p. 127 – 151.

Little, J., 1961. A proof of the Queuing Formula L = λW. *Operations Research*, **9**:383 – 387.

Menasce, D. A. and Almeida, V. A. F., 1998. Capacity Planning for Web Performance: Metrics, Models and Methods. Prentice Hall, Upper Saddle River, NJ, p. 197 – 220.

Menasce, D. A. and Almeida, V. A. F., 2000. Scaling for E-Business Technologies, Models, Performance, and Capacity Planning. Prentice Hall, Upper Saddle River, NJ, p. 223 – 374.

Reiser, M. and Lavenberg, S., 1980. Mean-value analysis of closed multi-chain queuing networks. *J. ACM*, **27**(2):313 – 323.

Zhong, Y.S., Chen, D.R. and Shi, M.H., 2002. Estimation of financial loss ratio for E-insurance: a quantitative model. *Journal of Zhejiang University SCIENCE*, **3**(2):140 – 147.