

# 基于 Domino 邮件服务器的性能建模

王磊 梁毅

(中国科学院计算技术研究所国家智能计算机研究开发中心 北京 100080)

**摘要** 现有的邮件服务器的评测方式大多是定量的,缺乏对系统的定性描述和有效建模。本文中,我们利用排队论理论,对基于 Domino 的邮件服务器进行性能建模,并对模型进行了验证和分析。实验表明,该模型较好地描述了邮件服务器的系统行为,是对邮件服务器的性能进行定性分析的一种较为理想的工具。

**关键词** Domino, 邮件服务器, 排队论, 性能模型

## A Performance Model for Domino Mail Server

WANG Lei LIANG Yi

(National Research Center for Intelligent Computing Systems, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080)

**Abstract** Nowadays, people usually measure the performance of E-mail server in quantitative manners, and the research of setting the qualitative performance model for E-mail server remains in the immature state. In this paper, with consideration of the Queue Theory, we propose a performance model for Domino mail server. The verification and analysis of the model are also presented. The good experimental results show that this model simulates the behavior of Domino mail server well and prove that it is useful for the qualitative performance measurement of Domino mail server.

**Keywords** Domino, Mail server, Queue theory, Performance model

### 1 引言

E-mail 是 Internet 上历史最悠久的服务之一。随着网络技术的发展, E-mail 服务的需求日益增加, E-mail 服务系统的性能已经成为了人们关注的一个问题。对邮件系统进行性能评测可以发现已有系统的性能缺陷和瓶颈从而为进行改进和提高提供了参考, 同时还可以为用户在一定的价格范围内选择性能最好的系统, 使其达到较好的性能/价格比。

现有的对于邮件系统的评测方式大多是定量的, 即采用测量法进行性能评测, 对实际的邮件系统本身进行观测, 收集各种事件的统计资料, 再加以分析, 来评价邮件系统性能。这种方式缺乏对系统的定性描述和有效建模。很大程度上依靠测试人员的经验且和测量设备以及测量程序密切相关, 在系统规划阶段难以对系统性能进行准确的预测。

如果采用模型方法, 根据系统的工作原理, 对系统的功能进行建模, 则可以对系统性能进行定性分析, 从而解决上述定量评测无法解决的问题。排队论作为一种数学工具, 在系统性能的定性分析中一直发挥着重要的作用<sup>[5]</sup>。

Lotus Domino 服务器是为企事业单位设计的集通讯、协作及 Web 应用功能于一体的软件平台。基于 Domino 的邮件服务系统已具有广泛的应用场景。本文中我们采用排队论理论对基于 Domino 的邮件服务器进行性能建模和分析, 通过建立该模型我们实现了两个主要功能: 对系统支持的在线用户数和响应时间等指标进行预测; 对 Domino 邮件服务器的资源瓶颈进行考察。最后我们对模型进行了验证和分析。

### 2 Domino 邮件系统工作原理

电子邮件是 Domino 最基本和核心的服务功能。Domino R5 是一个标准的 Internet 邮件服务器, 支持一切 Internet 标准的邮件工作方式, 邮件在服务器之间以标准的 Internet 寻址方式和路由方式传递。

Domino 邮件系统由三个基本组件组成: Domino 邮件服务器、Domino 邮件文件和邮件客户机, 支持 Notes 邮件功能以及多种 Internet 邮件标准。Domino 邮件服务器作为 Domino 邮件系统的核心组件和客户机一起提供了收信、发信等主要服务。

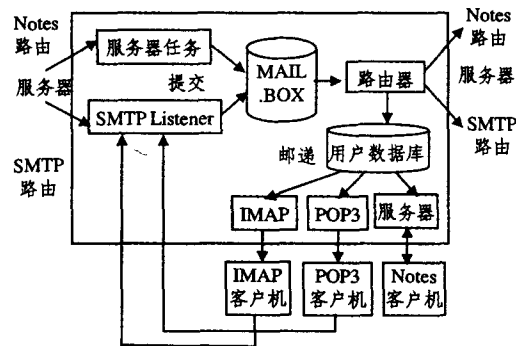


图 1 Domino 邮件系统的工作原理图

图 1 是 Domino 邮件系统的工作原理图, 黑框内为 Domino 邮件服务器的工作原理。Domino 邮件服务器主要实现了响应用户请求、分发邮件、投递邮件和传送邮件的功能, 实现这些功能的进程/线程如表 1 所示。

表 1 主要进程/线程功能表

功能描述	具体功能	实现的进程/线程
响应用户请求	接收和处理用户的收信、发信等请求。	dbServer 线程
分发邮件	从 Mail.Box 中读取信件, 并对本地和外地信件进行分类。	Router 进程
传送邮件	发送外地信件	Transfer 线程
投递邮件	处理本地邮件, 存放到用户数据库中	Deliver 线程

### 3 排队论理论及其在性能评测中的应用

排队论是运筹学的一个分支,它又叫做随机服务系统理论。它的研究目的是如何改进服务机构或组织满足服务的对象使得某种指标达到最优的问题<sup>[1,4]</sup>。一个典型排队系统如图 2 所示,其特性主要由顾客到达方式、排队方式和服务方式决定。

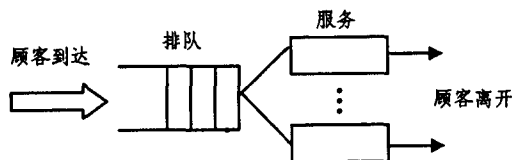


图 2 排队论原理图

排队系统的表达方式为: A/B/n/S/Z, 其中各符号的含义为: A 顾客到达的分布, B 服务时间的分布, n 服务个数, S 服务队列的容量, Z 服务规则。常用的服务系统, 顾客到达服从泊松分布, 服务队列容量无限, 服务时间服从负指数分布, 服务规则为先来先服务, 因此可表示为 M/M/N/∞/FCFS, 简化为 M/M/N, 如果服务台为 1 个, 则简化为 M/M/1<sup>[3]</sup>。

衡量单个服务系统的主要指标有:

1. 系统的利用率  $\rho$ : 它是排队系统的顾客平均到达率(到达强度)与系统的平均服务率(服务强度)的比值。

2. 顾客在系统内的平均排队时间  $W_q$ : 它是系统中顾客等待时间的期望值。

3. 顾客在系统内的平均逗留时间  $W_s$ : 它是系统中顾客逗留时间的期望值。如果设服务时间为 S, 则  $W_s = W_q + S$ 。

下面我们给出 M/M/1 和 M/M/N 系统中  $\rho$ 、 $W_s$  和  $W_q$  的计算方法。

M/M/1 服务系统: 如果顾客的到达强度为  $\lambda$ , 服务台的服务强度为  $\mu$ 。

$$\text{系统的利用率: } \rho = \frac{\lambda}{\mu}$$

$$\text{顾客在系统内的平均排队时间: } W_q = \frac{\rho}{\mu - \lambda}$$

$$\text{顾客在系统内的平均逗留时间: } W_s = \frac{1}{\mu - \lambda}$$

M/M/N 服务系统: 如果顾客的到达强度为  $\lambda$ , 服务台的服务强度为  $\mu$ , 服务台的个数为  $n$ 。

$$\text{系统的利用率: } \rho = \frac{\lambda}{n * \mu}$$

设  $\rho_1 = \frac{\lambda}{\mu}$ ; 设  $P_0$  为系统中顾客数为 0 个的概率, 顾客在

$$\text{系统内的平均排队时间: } W_q = \frac{\rho_1^n * P_0}{\mu * n * n! * (1 - \rho)^2}$$

$$\text{顾客在系统内的平均逗留时间: } W_s = W_q + \frac{1}{\mu}$$

### 4 Domino 邮件服务器性能模型

在本节中, 我们利用排队论知识, 对 Domino 邮件服务器进行建模。该模型分析系统中用户数一定的情况下, 系统对用户请求的响应时间, 因此不考虑用户登录的负载。同时在模型中, 我们假设网络带宽足够, 在网络上未设队列。

#### 4.1 邮件服务器工作流程分析

下面我们给出收信和发信的细化流程图。

收信的流程

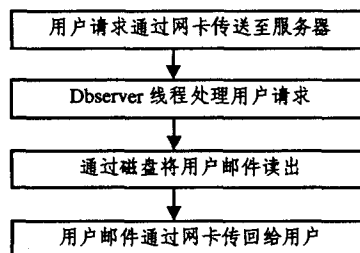


图 3 Domino 邮件服务器收信功能细化流程图

在收信功能上, 我们主要考虑磁盘和网卡的性能瓶颈, 因此, 在流程图中, 除了在 Dbserver 处的处理外, 我们只着重画出了磁盘和网卡这两个处理步骤。

发信的流程

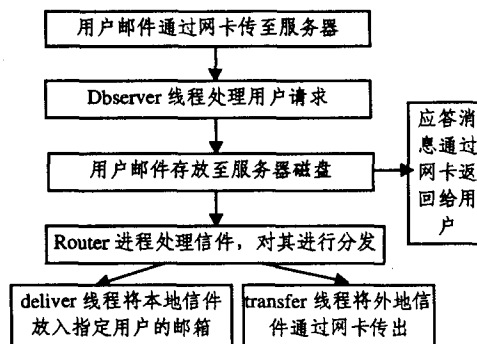


图 4 Domino 邮件服务器发信功能细化流程图

由图 4 可知, 对于用户而言, 当邮件存放至服务器磁盘, 即收到应答消息。因此在计算用户响应时间时, 只需要计算至用户邮件存盘, 返回应答信息即可。此外, 值得注意的是, 一封邮件可能发送给多个接收者。因此, 一封邮件经过 Router 服务台的分发处理后, 将产生多封邮件输出。

4.2.1 模型假设条件 对于模型我们做如下假设:

(1) 对小消息(如发信返回信息、收信请求信息), 由于他们在各个队列中的服务时间非常短, 因此在模型中, 我们假设小消息的服务时间为 0。

(2) 由于邮件分发中, 存在一份邮件分发为多封邮件的情况, 因此在 transfer 和 deliver 处应设置  $M_k/M/N$  队列<sup>[4]</sup>, 在计算中, 为了方便计算我们设 transfer 和 deliver 处队列的输入流符合泊松分布<sup>[4]</sup>。

4.2.2 模型中的定义 以下给出模型中的一些定义:

$\lambda_1$ : 用户发信的强度; 单位为 个/秒。  $\lambda_2$ : 用户收信的强度; 单位为 个/秒。  $k_1$ : 用户的一封信发往的其他域个数的期望。  $k_2$ : 用户的一封信发往本域用户个数的期望。  $F_{mail}$ : 邮件文件的平均文件大小; 单位为 kb。  $N_{io}$ : 服务器端网卡带宽; 单位为 kb/s。  $D_{io}$ : 磁盘 I/O 带宽; 单位为 kb/s。

#### 4.2 邮件服务器建模

综合第 2 节的 Domino 邮件服务器工作流程图, 我们利用排队论对 Domino 邮件服务器进行建模, 模型图如图 5 所示。

4.2.3 模型中各排队系统分析 我们对各个节点处的排队队列进行分析, 计算各个队列的到达强度、服务强度, 从而得到  $\rho$ 、 $W_q$ 、 $W_s$  等性能指标。

节点  $M_{n,i}$  代表网卡处的排队队列, 为 M/M/1 排队系统。在  $M_{n,i}$  处的到达强度:  $\lambda_{n,i} = \lambda_1 + \lambda_2 + k_1 * \lambda_1$ ; 到达请求为: 用户发信的请求, 用户发信的回应, 用户收信的请求, 用户收信

的请求,用户收信的反应,TRANSFER 线程发信的请求。其

中用户发信的请求回应和用户收信请求的消息大小被忽略。

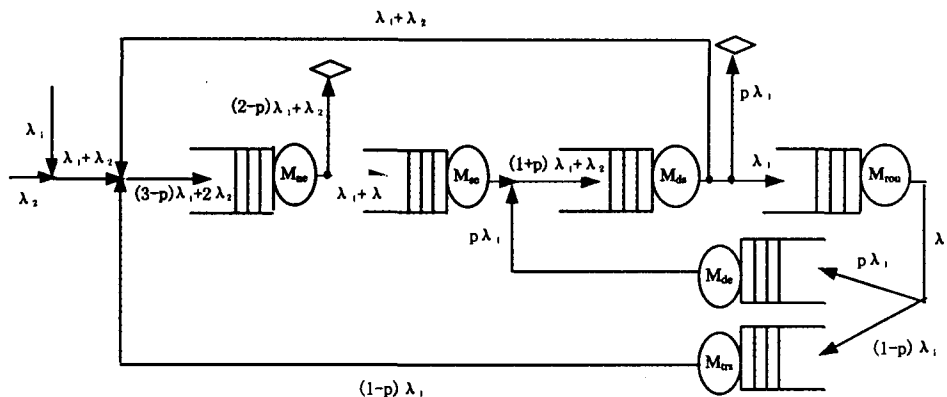


图 5 Domino 邮件服务器性能模型

在  $M_{ser}$  处的服务强度:  $\mu_{ser} = \frac{Nio}{F_{mail}}$

在  $M_{ser}$  处的系统利用率:  $\rho = \frac{\lambda_{ser}}{\mu_{ser}}$

在  $M_{ser}$  处的平均排队时间:  $W_{q_{ser}} = \frac{\lambda_{ser}}{\mu_{ser} * (\mu_{ser} - \lambda_{ser})}$

在  $M_{ser}$  处的平均停留时间:  $W_{s_{ser}} = W_{q_{ser}} + (1/\mu_{ser})$

节点  $M_{ser}$  代表 Dbsolver 线程处的排队队列,为  $M/M/N$  排队系统,其中  $N$  是由系统管理员设定的 Dbsolver 的线程个数。

在  $M_{ser}$  处的到达强度:  $\lambda_{ser} = \lambda_1 + \lambda_2$ ; 到达的请求为: 用户发信的请求, 用户收信的请求。

在  $M_{ser}$  处的服务强度:  $\mu_{ser} = 1/Z$ ; 其中,  $Z$  表示对于一个请求的平均服务时间, 它是平均邮件大小和用户到达强度的函数。

在  $M_{ser}$  处的系统的利用率:  $\rho = \lambda_{ser} / (N * \mu_{ser})$

在  $M_{ser}$  处的平均等待时间:

$$W_{q_{ser}} = \frac{\rho^n * P_0}{\mu_{ser} * n * n! * (1-\rho)^2}$$

其中:  $\rho_1 = \lambda_{ser} / \mu_{ser}$

$$P_0 = \left[ \sum_{k=0}^{n-1} \frac{\rho_1^k}{k!} + \left( \frac{\rho_1^n}{n!} \right) * \left( \frac{1}{1-\rho} \right) \right]^{-1}$$

在  $M_{ser}$  处的平均停留时间:  $W_{s_{ser}} = W_{q_{ser}} + (1/\mu_{ser})$

节点  $M_{dis}$  代表磁盘处的排队队列,为  $M/M/1$  排队系统。

在  $M_{dis}$  处的到达强度:  $\lambda_{dis} = \lambda_1 + \lambda_2$ ; 到达的请求为: 用户发信请求, 用户收信请求。

在  $M_{dis}$  处的服务强度:  $\mu_{dis} = Nio / F_{mail}$

在  $M_{dis}$  处的系统的利用率:  $\rho = \frac{\lambda_{dis}}{\mu_{dis}}$

在  $M_{dis}$  处的平均排队时间:  $W_{q_{dis}} = \frac{\lambda_{dis}}{\mu_{dis} * (\mu_{dis} - \lambda_{dis})}$

在  $M_{dis}$  处的平均停留时间:  $W_{s_{dis}} = W_{q_{dis}} + (1/\mu_{dis})$

节点  $M_{del}$  代表 Deliver 线程处的排队队列,为  $M/M/N$  排队系统,其中  $N$  是由系统管理员设定的 Deliver 的线程个数。

在  $M_{del}$  处的到达强度:  $\lambda_{del} = k_2 * \lambda_1$ ; 到达请求为: 用户发给本地的信。

在  $M_{del}$  处的服务强度:  $\mu_{del} = 1/Z$ ;  $Z$  表示对于一个请求的平均服务时间, 它是平均邮件大小和用户到达强度的函数。

在  $M_{del}$  处的系统的利用率:  $\rho = \lambda_{del} / (N * \mu_{del})$

在  $M_{del}$  处的平均排队时间:

$$W_{q_{del}} = \frac{\rho^n * P_0}{n * \mu_{del} * n! * (1-\rho)^2}$$

其中:  $\rho_1 = \lambda_{del} / \mu_{del}$

$$P_0 = \left[ \sum_{k=0}^{n-1} \frac{\rho_1^k}{k!} + \left( \frac{\rho_1^n}{n!} \right) * \left( \frac{1}{1-\rho} \right) \right]^{-1}$$

在  $M_{del}$  处的平均停留时间:  $W_{s_{del}} = W_{q_{del}} + (1/\mu_{del})$

节点  $M_{rou}$  代表 Router 进程处的排队队列,为  $M/M/1$  排队系统。

在  $M_{rou}$  处的到达强度:  $\lambda_{rou} = \lambda_1$ ; 到达请求为: 用户所发的信件。

在  $M_{rou}$  处的服务强度:  $\mu_{rou} = 1/Z$ ; 其中,  $Z$  表示对于一个请求的平均服务时间, 它是平均邮件大小和用户到达强度的函数。

在  $M_{rou}$  处的系统利用率:  $\rho = \frac{\lambda_{rou}}{\mu_{rou}}$

在  $M_{rou}$  处的平均排队时间:  $W_{q_{rou}} = \frac{\lambda_{rou}}{\mu_{rou} * (\mu_{rou} - \lambda_{rou})}$

在  $M_{rou}$  处的平均停留时间:  $W_{s_{rou}} = W_{q_{rou}} + (1/\mu_{rou})$

节点  $M_{trs}$  代表 Transfer 线程处的排队队列,为  $M/M/N$  排队系统,其中  $N$  是由系统管理员设定的 Transfer 的线程个数。

在  $M_{trs}$  处的到达强度:  $\lambda_{trs} = k_1 * \lambda_1$ ; 到达请求为: 用户发给外地的信。

在  $M_{trs}$  处的服务强度:  $\mu_{trs} = 1/z$ ; 其中,  $z$  表示对于一个请求的平均服务时间, 它是平均邮件大小和用户到达强度的函数。

在  $M_{trs}$  处的系统的利用率:  $\rho = \lambda_{trs} / (N * \mu_{trs})$

在  $M_{trs}$  处的平均排队时间:

$$W_{q_{trs}} = \frac{\rho^n * P_0}{n * \mu_{trs} * n! * (1-\rho)^2}$$

其中:  $\rho_1 = \lambda_{trs} / \mu_{trs}$

$$P_0 = \left[ \sum_{k=0}^{n-1} \frac{\rho_1^k}{k!} + \left( \frac{\rho_1^n}{n!} \right) * \left( \frac{1}{1-\rho} \right) \right]^{-1}$$

在  $M_{trs}$  处的平均停留时间:  $W_{s_{trs}} = W_{q_{trs}} + (1/\mu_{trs})$

### 4.3 邮件服务器模型结论

根据 4.1 节对于发信、收信流程的分析以及 4.2 节对于各个排队队列的分析我们可以得到以下的模型结论。

发信的用户平均响应时间为:

$$W_1 = W_{s_{ser}} + W_{s_{ser}} + W_{s_{dis}} + W_{q_{ser}}$$

收信的用户平均响应时间为:

$$W_2 = W_{q_{ser}} + W_{q_{ser}} + W_{s_{dis}} + W_{s_{ser}}$$

对于本域信件的系统平均响应时间为:

$$W_3 = W_{s_{ser}} + W_{s_{ser}} + W_{s_{dis}} + W_{s_{rou}} + k_2 * (W_{s_{del}} + W_{s_{dis}})$$

对于外域信件的系统平均响应时间为:

$$W_4 = W_{S_{net}} + W_{S_{er}} + W_{S_{disk}} + W_{S_{net}} + k_1 * (W_{S_{er}} + W_{S_{net}})$$

### 5 模型验证

我们选取用户发信的响应时间来进行模型正确性的验证。

#### 5.1 实验环境及负载模型

我们选取曙光超级服务器作为实验的环境平台,其中作为邮件服务器的节点配置为2个PⅢ 1GHZ的CPU,3G的RAM,17G的磁盘,100M以太网卡,操作系统为Linux;客户端的配置为2个PⅢ 1GHZ的CPU,1G的RAM,17G磁盘,100M以太网卡,操作系统为Windows2000,服务端的Dbserver线程池的线程数设置为系统缺省值40。

由于我们的目标是验证模型正确性,因此这里选取客户端的发信行为作为验证。对于负载模型,我们的设置为:客户

端使用发送强度和信件大小可以指定的负载发生器,发送信件的间隔符合负指数分布。发信的规则为:每封信分别发给本地和外地各一个用户,并且在客户端记录每次发信的实际响应时间。这里我们所选取的可变参数(发送强度、信件大小)来源于乐捷邮件系统提供的数据<sup>[6]</sup>。

#### 5.2 实验验证与结果分析

我们首先给出理论响应时间的计算方法。计算理论响应时间需要知道各处到达强度和服务强度的值。由上节可知,发信流程会经过网卡、Dbserver和磁盘这3处服务台。下面我们分析各个服务台服务强度的求解:

1. 网卡处的服务强度  $\mu_{net} = \frac{N_{io}}{F_{mail}}$ ,  $N_{io}$  的值为 8192kb/s

2. 对于 Dbserver 处的服务强度。由上一节我们知道 Dbserver 对一个请求的服务时间是用户平均邮件大小和用户到达强度的函数。表 2 为我们的测试数据。

表 2 Dbserver 实际响应时间

实验序列	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
邮件大小(kb)	0.5	0.5	0.5	0.5	5.7	5.7	5.7	5.7	15.6	15.6	15.6	15.6	43.5	43.5	43.5	43.5
到达强度	5	10	15	20	5	10	15	20	5	10	15	20	5	10	15	20
响应时间(ms)	22	28	28	36	23	25	30	37	23	27	31	37	27	30	34	39

我们根据测试用例拟合出服务时间公式为:

$$z = 16.9163 + 0.1033 * x + 0.8859 * y;$$

其中:z:服务时间(ms),x:邮件文件的平均文件大小(kB),y:用户到达强度(个/秒);图 6 为拟合值与实际测量值的对比图。

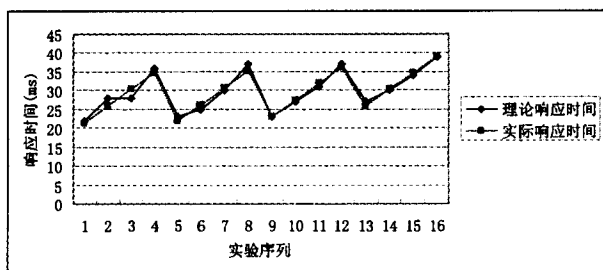


图 6 Dbserver 的实际响应时间和理论响应时间对比图

所以 Dbserver 处的服务强度  $\mu_{ser} = 1/z$

3. 磁盘处的服务强度  $\mu_{disk} = N_{io}/F_{mail}$ ,  $D_{io}$  的值为 30720kb/s

根据用户发信的理论响应时间的计算公式:  $W_1 = W_{S_{net}} + W_{S_{er}} + W_{S_{disk}} + W_{S_{net}}$ , 我们计算出用户发信的理论响应时间,同时我们测量出实际的发信响应时间。表 3 为测试和计算的数据。

表 3 发信流程的理论响应时间和实际测试响应时间数据表

实验序列	1	2	3	4	5	6	7
信件大小(kb)	0.5	5.7	5.7	15.6	15.6	43.5	43.5
到达强度(个/秒)	20	15	20	10	15	10	15
实际响应时间(ms)	39	33	42	33	37	39	46
理论响应时间(ms)	36.12	31.32	38.33	30.69	34.77	41.17	45.0

图 7 为理论响应时间和实际测试响应时间的对比图。

通过图 7 可见理论时间和实际测试的时间基本吻合。但是理论时间小于实际测试的时间。这是因为我们计算理论时

间的时候,没有考虑客户端的处理时间。这说明我们所建模型的队列设置是正确的,而且对于 Dbserver 的分析也是正确的。

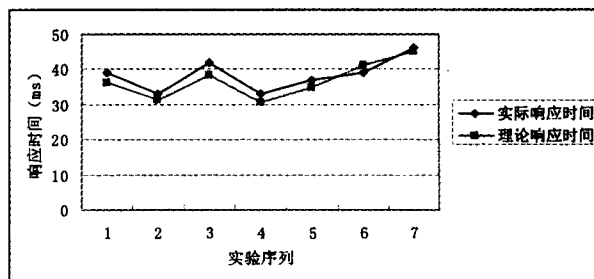


图 7 发信流程理论响应时间值和实际测试响应时间对比图

### 6 模型效用

通过建立该模型我们实现了:

- 对系统支持的在线用户数和响应时间等指标进行预测。

- 对 Domino 邮件服务器的资源瓶颈进行考察。

#### 6.1 对系统支持的在线用户数和响应时间进行预测

通过该模型,我们可以实现对于给定的用户数要求,预测 Domino 邮件服务器对用户请求的平均响应时间;对于给定的平均响应时间要求,预测 Domino 邮件服务器能支持的相应用户数。具体做法为:将用户数  $n$  给定( $n$  要满足使所有队列的资源使用率  $\rho$  小于 1)算出平均响应时间  $T$  ( $T$  就是在用户数为  $n$  情况下用户请求的平均响应时间)。根据一组  $n$  和  $T$  的结果,拟合  $n$  和  $T$  的函数  $T = f(n)$ ;这样在一定范围内,任意给出  $n$  值或是  $T$  值就能得到对应的  $T$  值或是  $n$  值。

#### 6.2 对 Domino 邮件服务器的资源瓶颈进行考察

通过该模型,我们可以通过计算 Domino 邮件服务器的资源瓶颈,考察 Domino 邮件服务器的资源容量和资源处理能力。目前,我们主要考察的 Domino 邮件服务器资源有网卡资源、磁盘资源、CPU 资源和内存资源。具体方法如下:

• 网卡资源:主要考察网卡带宽。在网卡队列处,网卡资源使用率  $\rho_{net} = \lambda/\mu$ ,如果发现在指定的条件下  $\rho_{net}$  的值接近 1 时,可知 Domino 邮件服务器在网卡处存在瓶颈。可以考虑更换速度更快的网卡。

• 磁盘资源:主要考察磁盘带宽。计算方法与网卡相同。在磁盘队列处,磁盘资源使用率  $\rho_{disk} = \lambda/\mu$ ,如果发现在指定的条件下  $\rho_{disk}$  的值接近 1 时,可知 Domino 邮件服务器在磁盘处存在瓶颈。可以考虑更换读写速度更快的磁盘。同时还需考察磁盘容量。根据实验可知,每个用户邮箱初始化时占用磁盘空间为 5.5M,如果为每个用户分配的邮箱最大容量为 50M,在计算所需的磁盘空间时,我们取均值 25M。给定用户数  $n$ ,则除了 Domino 应用程序所占空间外,磁盘还需留出至少  $(25 * n)M$  空间。

• CPU 资源:主要考察 router 进程,dbserver 线程,transfer 线程,deliver 线程的占用 CPU 时间的情况。根据到达率和服务率,可得进(线)程的使用率  $\rho$ ,若各使用率均接近 1,则说明 CPU 的处理能力是瓶颈,需要更换处理速度更快的 CPU,若只有个别使用率接近 1,则可以通过调整该进(线)程的优先级来解决。

• 内存资源:主要考察内存的使用量。我们在模型中没有将内存带宽作为瓶颈考虑,因此没有建立内存资源的队列,而且在现实中内存容量的需求可以很容易地被满足。所以我们只给出 Domino 邮件服务器内存需求的经验公式。设给定并发用户数位  $n$ ,内存使用的公式如下:

$$\text{所需的内存容量的最小值} = (n * 300 * 3)k$$

小结 本文中我们利用排队论理论对基于 Domino 的邮件服务器进行性能建模和分析,通过建立该模型我们实现了对系统支持的在线用户数和响应时间等指标进行预测,对 Domino 邮件服务器的资源瓶颈进行考察。

我们的模型只是基于一个 Domino 域建立的模型,在下一步的工作中我们将进一步考虑多个 Domino 域的情况。

### 参考文献

- 1 随机过程. 中国统计出版社,1997
- 2 管理 DOMINO 系统(第一卷). LOTUS
- 3 运筹学. 清华大学出版社,1982
- 4 排队论. 北京邮电学院出版社,1994
- 5 Slothouber L P. A Model of Web Server Performance. Starine Com, 1996
- 6 乐捷 DOMINO 邮件系统用户手册. 乐捷公司,2001

(上接第 26 页)

2 和图 3 所示。这里 PB 帧编码没有采用 intra 模式。

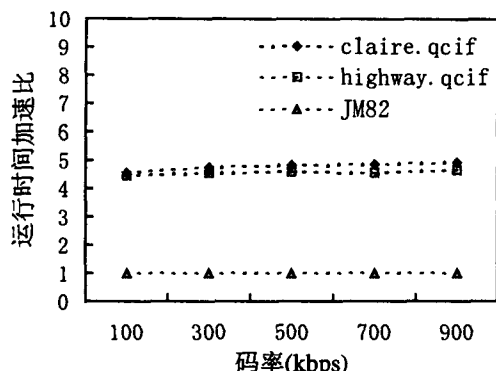
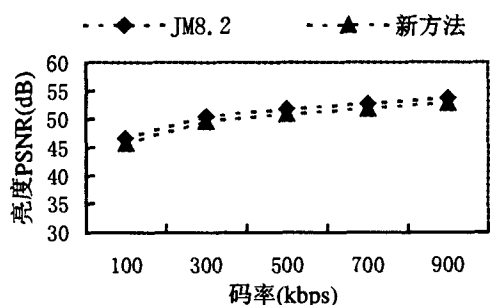
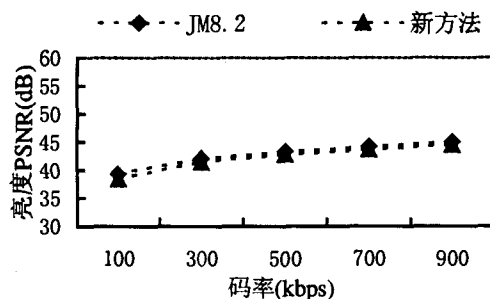


图 2 几种典型序列运行时间加速比



(a) claire 序列亮度率失真比较



(b) highway 序列亮度率失真比较

图 3

图 2、图 3 表明对于几种代表序列该算法相对 JM8.2 的运行时间加速比可以达到 4.5 以上,PSNR 略有下降,但实际当中不会对视频质量造成很大影响。而且在该算法步骤 2 中 intra 模式的粒度可以放宽,如可以只采用 intra16×16 模式,测试表明这样 PSNR 下降较少,只是加速比要稍有降低,实际中可以根据需要来调整 intra 的粒度。

结束语 H.264 视频编码标准具有编码效率高、图像质量好、适合低码率传输等优点,并能很好地处理信息包丢失问题,必将促进视频压缩编码技术和多媒体通信技术的进一步发展。H.264 主要面向的是低比特率、低延时的应用。因此,减少运算时间、降低码率是其中一个关键因素。本文在分析目前 H.264 中采用的关键技术的基础上,针对其中较为耗时的块分割模式选择模块进行优化,提出了一种新的块分割模式选择算法,通过对比实验测试,验证了该算法和 JM8.2 相比,在 PSNR 稍有下降的情况下运行时间加速比达到了 4.5 以上,大大提高了编码的实时效果。

### 参考文献

- 1 ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC, Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification, document JVT-G050. March 2003
- 2 Wiegand T, Sullivan G J, Bjøntegaard G, Luthra A. Overview of the H.264/AVC Video Coding Standard. IEEE Transactions on Circuits and Systems for Video Technology, 2003,13(7)
- 3 李宾,高平. H.264 编码系统的特点及其应用前景. 数字电视与数字视频,2003 (6)
- 4 王嵩,薛全,张颖,陈建乐. H.264 视频编码新标准及性能分析. 数字电视与数字视频,2003 (6)
- 5 Richardson I E G. H.264 / MPEG-4 Part 10 White Paper: Prediction of Inter Macroblocks in P-slices
- 6 Richardson I E G. H.264 and MPEG-4 Video Compression. UK: Wiley & Sons, 2003
- 7 朱冬冬,戴琼海. H.264 快速帧间编码模式选择算法. 有线电视技术,2004(9)
- 8 Evaluation of the H.264 codec Internal Report. Carlos Miguel Tavares Calafate, Manuel Pérez Malumbres,2003
- 9 Joint Video Team (JVT). Test Model JM8.2. at <http://bs.hhi.de/~suehring/tml/download/>, July 2004