第 34 卷 第 8 期 计 算 机 工 程 2008 年 4 月
Vol.34 No.8 Computer Engineering April 2008
·软件技术与数据库· 文章编号：1000—3428(2008)08—0062—04 文献标识码：A 中图分类号：F49

基于多队列和多线程的短信实时并发控制算法

沈 斌，李兴国，钟金宏，沈丽娜
(合肥工业大学管理学院，合肥 230009)

摘 要：针对 Modem 控制的短信应用在实时性和并发性等性能上的不足，以实际项目为背景，提出基于多队列和多线程的短信实时并发控制算法。以排队论模型为依据，采用多队列、多线程、池等技术手段保证短信控制的实时性、并发性以及数据的共享性。通过排队论模型和实验对算法进行验证，结果表明，该算法大幅提高了短信发送和接收的速度与效率，满足了短信应用的实时性和并发性需要，优化了短信应用系统的整体性能。

关键词：短信；多队列；多线程；实时；并发

Real-time Concurrent Algorithm for Short Message Control Based on Multi-queue and Multithread

SHEN Bin, LI Xing-guo, ZHONG Jin-hong, SHEN Li-na
(School of Management, Hefei University of Technology, Hefei 230009)

【Abstract】 Dedicated to the deficiencies in real-time characteristic, concurrency and other performance of short message applications controlled by the Modem, a real-time concurrent algorithm for short message control based on multi-queue structure and multithread technology is introduced, with a practical project as the background. The algorithm is built on the basis of queuing theory model and adopts multi-queue, multithreading, pool and some other technical means to ensure the real-time characteristic, concurrency and data sharing of short message control. This paper also validates the algorithm via queuing theory model and experiments, so that the results indicate that the algorithm increases the speed and the efficiency of the short message sending and receiving to a great extent, consequently satisfies the requirement for real-time characteristic and concurrency, and improves the overall performance of the short message applications.

【Key words】 short message; multi-queue; multithread; real-time; concurrent

1 概述

国内外许多企事业单位将短信应用于日常的信息收集、处理和发送[1]。短信的应用促进了及时、准确的信息交流，提高了信息交流的速度和效率。

短信应用系统通常通过 3 种方式接入无线网络：

(1)专线接入。通过专线或互联网接入移动运营商短信网关。该方式发送和接收速度快，能够承受大数据量；但是该方式依赖 Internet，费用高，并且运营商对用户的设备和业务有要求。该方法适用于业务量大、业务稳定、场所固定的大型企业。

(2)虚拟运营商接入。通过开发接口或者中间件服务的方式联入互联网，实现短信对接。该方式资金和设备投入少，但是依赖 Internet 及虚拟运营商，业务内容和服务质量受其限制。该方法适用于业务量中等、业务相对稳定、场所相对固定的中小型企业。

(3)Modem 接入。通过 Modem 接入移动通信网，实现点对点收发。该方式不受运营商限制，不依赖 Internet，费用低廉，灵活性大，但是发送和接收速度受设备限制。该方法适用于业务量不大、业务不稳定、场所不固定的中小型企业。

本文的应用背景是使短信服务方便快捷地集成到中小型企业的信息系统中，选择第(3)种方式作为接入方式。实验设备选择的是WAVECOM的GSM Modem，支持AT指令[2]。

随着短信应用范围和规模的扩大，短信应用的实时性和并发性备受关注。短信控制算法是保障短信实时性和并发性的关键，对减少系统消耗和提高系统效率非常重要。Modem方式的短信应用一般采用轮询和线程的方式对短信进行控制，由于Modem的收发速率限制，未能实现实时并发控制[3]。本文研究在Modem方式的短信应用系统中使用多Modem、Modem池、Modem集群(以下简述为多Modem)控制的短信实时并发控制算法，即基于多队列和多线程的短信实时并发控制算法(以下简称为算法)。

2 算法的整体方案

对于算法的解释如下：

(1)多队列：具有多种功能的队列，且每种功能的队列只有一个；

(2)多线程：具有多种功能的线程，且每种功能的线程有多个；

(3)实时：以最小的时间延迟处理产生的短信；

(4)并发控制：对短信和多 Modem 进行并发控制。

该算法能够实现短信的实时并发控制和数据的安全共享，以最小的系统消耗达到最佳的应用效果，对于其他方式的短信应用有很好的借鉴意义。

基金项目：安徽省自然科学基金资助重点项目(2006KJ025A)
作者简介：沈 斌(1985－)，男，本科生，主研方向：信息系统，信息化与管理创新，项目管理；李兴国，教授；钟金宏，副教授、博士后；沈丽娜，本科生
收稿日期：2007-12-09 **E-mail：**nduziyou@163.com

2.1 算法项目背景

在短信应用中，实时性、并发性和数据的共享性是衡量短信服务质量的重要指标。为了解决实时并发问题，根据多处理器并行计算的理论公式,类似地得到多 Modem 并行处理的理论公式：

$$Speedup(n\ \text{Modems}) = \frac{Performance(n\ \text{Modems})}{Performance(1\ \text{Modem})} = n$$

$$Speedup_{\text{fixed problem}}(n\ \text{Modems}) = \frac{Time(1\ \text{Modem})}{Time(n\ \text{Modems})} = \frac{1}{n}$$

即：n 个 Modem 的处理速度是 1 个 Modem 处理速度的 n 倍；对于一个确定的问题，n 个 Modem 的处理时间是 1 个 Modem 处理时间的 $1/n$。因此，算法采用多 Modem 的并发控制来提高收发速率，从而提高实时性和并发性。

本文以 SMSEngine 短信系统(以下简称 SMSEngine)为项目背景介绍算法。SMSEngine 是基于多 Modem 的短信应用系统，它将移动通信中的短信服务(Short Messaging Service, SMS)引入信息系统，能够根据组织的需求方便快捷地集成到信息系统中，使 SMS 与信息系统应用层的各应用成分之间实现跨网络协同工作，从而方便快捷地进行信息交换。通过这种集成，充分发挥 SMS 业务的优势，提高信息系统的整体性能和效率，降低系统的成本，实现系统的优化配置。其应用结构如图 1 所示。

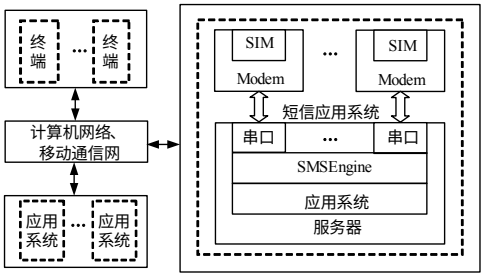

图 1　基于多 Modem 的短信应用结构

2.2 算法理论模型

排队系统是由一个或多个服务台和一些等待服务的顾客所组成的离散事件系统。排队系统有 3 个基本组成部分：输入过程，排队规则和服务机构。排队系统模型用于研究排队系统运行的效率，确定系统参数的最优值，以决定系统结构是否合理，研究改进措施等。

SMSEngine是一个排队系统，每种队列的输入过程是短信的产生，服务机构是Modem，排队规则采用单队列、并列的多服务台的标准M/M/c模型(M/M/c/∞/∞)[4]。用以判断短信排队系统性能的指标主要有：

(1)队长，指在系统中的短信数，它的期望值记作L_s；队列长，指在系统中排队等待服务的短信数，它的期望值记作L_q。则有：L_s=L_q+正被服务的短信数。

一般情形，L_s(或L_q)越大，服务率越低。

(2)逗留时间，指一条短信在系统中的停留时间，它的期望值记作W_s；等待时间，指一条短信在系统中排队等待的时间，它的期望值记作W_q。则有：W_s=W_q+服务时间。

此外，忙期指从短信到达空闲 Modem 起到 Modem 再次空闲为止这段时间长度，即 Modem 连续繁忙的时间长度，它关系到 Modem 的工作强度。忙期和一个忙期中平均服务的短信数都是衡量 Modem 效率的指标。

2.3 算法功能结构

要实现 SMSEngine 需要解决以下关键性问题：多 Modem 控制，计算机与 Modem 数据通信，数据共享。这些关键问题都可以通过算法来解决。算法的结构如图 2 所示。

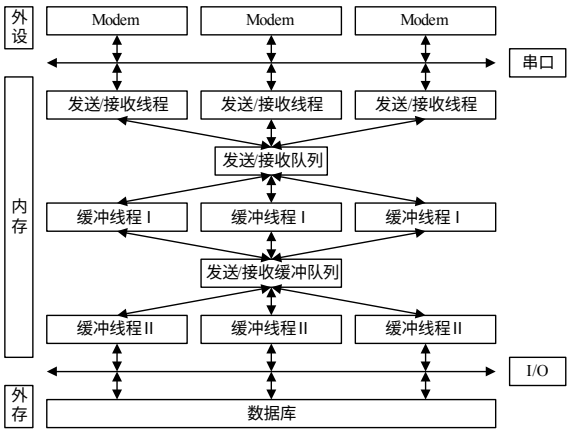

图 2　算法结构

算法采用的方法以及实现的功能如下：

(1)采用支持多线程互斥的两级缓冲多队列数据结构，避免低速的数据缓冲方式，提高短信收发的速度和效率，方便短信的收发、格式化和流量控制。

(2)采用多线程技术对多 Modem 进行控制，全双工、异步并发地处理大流量的短信，利用线程同步有效地解决数据共享冲突问题。

(3)加入线程池技术、数据库连接池技术，使系统在低消耗情况下高效率运作，提高系统的异步并发性和数据共享性。

3　算法多队列数据结构

队列是一种运算受限的线形表，它只允许在表的一端进行插入，而在另一端进行删除，对队列的操作满足先进先出原则。

为了保证短信的实时性，短信的发送和接收顺序应按照其产生的先后顺序进行，所以短信的发送和接收控制遵循先进先出(First In First Out, FIFO)原则。短信的收发特性和队列的特性相符合，因此算法采用队列结构对短信进行存储，让先产生的短信先发送或接收，后产生的短信后发送或接收。

3.1 短信队列设计

在 SMSEngine 中，为满足短信控制的需要，在内存中设置 2 个短信队列：发送队列，接收队列。发送队列用于存放需要发送的短信；接收队列用于存放接收到的短信。短信队列在短信的控制中起缓冲作用。

为了对短信的发送和接收进行流量控制和速度控制，发送队列和接收队列是支持多线程互斥读写的循环队列，这 2 个队列在短信的收发控制中起滑动窗口的作用。发送队列和接收队列的结构如图 3 所示。

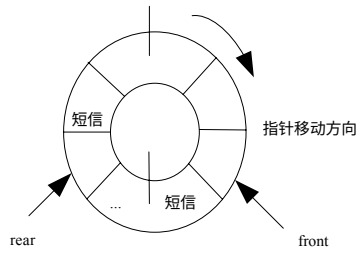

图 3　短信队列结构

短信队列中的单元是协议数据单元(Protocol Data Unit, PDU)编码的短信[5]。

短信发送过程中，发送队列未满时，需发送的短信直接入队；发送队列已满时，超过流量限制的短信先在缓冲队列中排队缓冲，等待发送队列有空单元时再入队。短信接收过程中：接收队列未满时，接收短信并入队；接收队列已满时，暂停接收短信，等待接收队列有空单元时再接收短信并入队。通过循环队列结构可以方便地对短信的发送和接收进行流量控制和速度控制。

3.2 缓冲队列设计

在 SMSEngine 中，短信的收发要频繁地与数据库交换数据，并进行格式转换。为了提高数据库的访问速度和格式转换速度，算法采用两级缓冲技术，在内存中给每种队列再开辟相应的缓冲区作为缓冲队列，即再设置发送缓冲队列、接收缓冲队列。缓冲队列起数据缓冲和格式转换作用。

缓冲队列是支持多线程互斥读写的带头结点的 FIFO 链表，队列的长度可根据短信的流量动态改变。缓冲队列的结构如图 4 所示。

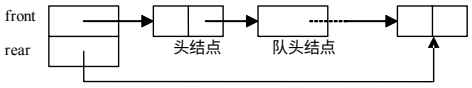

图 4 缓冲队列结构

缓冲队列的单元是一个结构体，发送缓冲队列的结构体中存放将要进行 PDU 编码的各项数据元素，接收缓冲队列的结构体中存放 PDU 解码后的各项数据元素。两级缓冲能够有效地提高多 Modem 收发短信的速度和效率。

3.3 数学分析

设$N(t)$表示在时间$[0,t)$内到达的短信数($n>0$)，令$P_n(t_1,t_2)$表示在时间区间$[t_1,t_2)$($t_2>t_1$)内有$n(\geq 0)$条短信到达的概率，即$P_n(t_1,t_2)= P\{N(t_2)- N(t_1)=n\}$($t_2>t_1,n\geq 0$)。

当$P_n(t_1,t_2)$符合下列 3 个条件时，短信的到达形成泊松流：

(1)在不相重叠的时间区间内短信到达数相互独立；

(2)对充分小的Δt，在时间区间$[t,t+\Delta t)$内有一条短信到达的概率与t无关，而与区间长Δt接近正比关系，即$P_n(t,t+\Delta t)=\lambda\Delta t+o(\Delta t)$。其中当$\Delta t\to 0$ 时，$o(\Delta t)$是关于Δt的高阶无穷小。$\lambda>0$是常数，表示单位时间有一条短信到达的概率，称为概率强度。

(3)对充分小的 Δt，在时间区间$[t,t+\Delta t)$内有 2 条或 2 条以上短信到达的概率极小，以至于可以忽略，即

$$\sum_{n=2}^{\infty} P_n(t,t+\Delta t) = o(\Delta t)$$

在$[0,t+\Delta t)$到达n条短信的概率$P_n(t+\Delta t)$为$P_n(t+\Delta t)=P_n(t)(1-\lambda\Delta t)+P_{n-1}(t)\lambda\Delta+o(\Delta t)$。

在表达式两边同时除以 Δt，令 $\Delta t\to 0$，则

$$\begin{cases} \mathrm{d}P_n(t)/\mathrm{d}t = -\lambda P_n(t) + \lambda P_{n-1}(t),\ n \geq 1 \\ P_n(o) = 0 \end{cases}$$

当 $n=0$ 时，有

$$\begin{cases} \mathrm{d}P_0(t)/\mathrm{d}t = -\lambda P_n(t) \\ P_0(o) = 1 \end{cases}$$

解上述 2 式，得

$P_n(t)=(\lambda t)^n \mathrm{e}^{-\lambda t}/n!$, $t>0$, $n=0,1,2,\cdots$

其中，$P_n(t)$表示长为t的时间区间内到达n条短信的概率，它的数学期望和方差分别是$E[N(t)]=\lambda t$, $Var[N(t)]=\lambda t$。

4 算法的多线程技术

短信的实时并发要求计算机与 Modem 通信的安全和高效。在 SMSEngine 中，计算机与 Modem 的通信是耗时的工作，容易堵塞其他任务的执行。

在算法中采用多线程技术可以使系统在实现并发通信的同时完成其他的任务，而不会因为一项任务的执行堵塞其他任务的执行，从而解决多 Modem 的控制问题，提高了短信控制的实时性和并发性。

在算法中利用多线程和重叠 IO 机制实现多线程下 Modem 的全双工、异步通信，以解决短信发送、接收、编解码、数据存储等过程中的 Modem 通信和多队列控制问题。在数据共享的过程中，对多线程进行线程同步，防止数据共享冲突。同时通过线程池技术提高系统并发处理性能，并通过数据库连接池技术提高数据库的访问效率。

4.1 线程设计

(1)发送与接收线程

在算法中，发送与接收线程有多个，主要任务是控制 Modem 对短信的发送和接收。发送线程从发送队列中取出数据包发送；接收线程将接收到的数据包放入接收队列。发送线程和接收线程将数据包的操作过程生成 SQL 语句写入日志文件，以方便对系统进行监控。

发送线程的工作过程：该线程平时处于睡眠状态，当发送队列中有短信需要发送时，发送线程被主管线程唤醒；发送线程从发送队列中提取短信，然后驱动 Modem 发送短信；如果短信发送成功，发送线程进入睡眠状态直到被主管线程唤醒，如果短信发送失败，则继续发送直到超过次数限制，然后进入睡眠状态直到被主管线程唤醒，发送操作完成后将发送操作写入日志文件。

接收线程的工作过程：该线程平时处于睡眠状态，当 Modem 中有短信到来时，接收线程被主管线程唤醒；接收线程驱动 Modem 接收短信，将接收到的短信放入接收队列，并将接收操作写入日志文件，然后进入睡眠状态直到被主管线程唤醒。

对于发送队列、接收队列的数据共享冲突问题，可以用临界区来解决。通过临界区可以保证同一时刻只有一个线程对这块数据缓冲区进行操作，其他线程如果要访问这些共享数据就会被阻塞，直到这个线程离开临界区域，这样就解决了数据共享冲突问题。

(2)缓冲线程

算法采用两级缓冲技术，缓冲线程有多个，主要任务是提高数据访问速度，方便数据的格式化。缓冲线程分为 2 种：第 1 种是在缓冲队列和短信队列间进行数据操作的线程，称为缓冲线程 I (如图 2)，该线程完成将发送缓冲队列中的数据编码并放入发送队列，或者将接收队列中的数据解码并放入接收缓冲队列的数据格式化操作；第 2 种是在数据库和缓冲队列间进行数据操作的线程，称为缓冲线程 II (如图 2)，该线程完成从数据库中提取数据到发送缓冲队列，或者从接收缓冲队列提取数据到数据库的数据交换操作。缓冲线程平时处于睡眠状态，只有当主管线程需要时才会被唤醒，缓冲线程完成数据交换或数据格式化任务后进入睡眠状态。

4.2 池技术

SMSEngine 需要迅速处理大批突发任务，每个任务的耗时短。如果频繁地创建、销毁线程或数据库连接，将耗费大量的系统资源。算法的池技术，就是预先建立一批线程或连接放在池中排队，需要的时候从池中取出线程或连接，操作完成后放回池中而不销毁。池技术使系统避免频繁创建、销毁线程或连接，从而提高并发处理性能。

线程池是用一个双向链表实现的，初始化时产生一个主管线程负责线程的调度，池中的元素是处于睡眠状态的发送线程、接线程或缓冲线程。当有任务到达时，主管线程从链表头摘取一个线程，将其唤醒转入工作状态。若线程池为空，则创建一个新的线程。线程完成任务后返回池中睡眠。主管线程定期杀死超过睡眠时限的线程，以实现线程池大小的自动调整和系统性能优化。

数据库连接池将数据库连接的信息集(包括编号、数据库连接描述符、繁忙标识等信息)定义为一个节点，空闲节点存储在一个用双向循环链表实现的 FIFO 队列中，繁忙节点存储在可随机访问的哈希表(繁忙节点表)中。要使用数据库连接时，从空闲节点队列中获取一个节点，将它放入繁忙节点表。若无空闲节点，则创建一个数据库连接，形成新节点放入繁忙节点表中。连接用完后，用哈希算法从繁忙节点表中迅速找出相应节点，再将它转移到空闲节点队列中。数据库连接池初始化时产生一个主管线程来定时检查空闲节点是否超过空闲时限，超时则关闭连接、删除节点，以使连接池的大小随系统的繁忙程度自动调整。

4.3 数学分析

短信相继到达的间隔时间 T 的概率密度若是

$$f_T = \begin{cases} \lambda e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases}$$

则称 T 服从负指数分布。它的分布函数是

$$F_T = \begin{cases} 1 - e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases}$$

数学期望 $E[T]=1/\lambda$；方差 $Var[T]=\lambda t$；标准差 $\sigma[T]=1/\lambda$。

则短信相继到达的间隔时间有下列性质：

(1)$P\{T>t+s|T>s\}=P\{T>t\}$，即无记忆性。说明短信到达是纯随机的。

(2)当输入过程是泊松流时，T必须服从负指数分布。因为对于泊松流，在$[0,t)$区间内至少有一条短信到达的概率是$1-P_0(t)=1-e^{-\lambda t}$, $t>0$，又可表示为$P\{T\leq t\}=F_T(t)$。

因此，T是独立且同负指数分布(密度函数为$\lambda e^{-\lambda t}$, $t\geq 0$)，与输入过程为泊松流(参数为λ)是等价的。

对于泊松流，λ 表示单位时间平均到达的短信数，所以 $1/\lambda$ 就表示短信相继到达平均间隔时间，而这正和 $E[T]$的意义相符。

短信的服务时间是在忙期相继离开系统的 2 条短信的间隔时间，有时也服从负指数分布。这时设它的分布函数和密度分别是$F_v(t)=1-e^{-\mu t}$。其中，μ表示单位时间能被服务完成的短信数，称为平均服务率；而 $1/\mu=E(v)$表示一条短信的平均服务时间，这里的平均就是期望值。

5 系统性能分析

由上述分析可知，短信的产生服从泊松过程，服务时间服从负指数分布，短信的各项服务符合排队论中单队列、并列的多服务台的标准 M/M/c 模型(M/M/c/∞/∞)。

将由算法实现的SMSEngine应用到某上市公司的OA(Office Automation)系统中，该OA系统每月有 10^7条的短信发送量，则短信平均到达率λ=3.86 条/s。该OA系统采用 8 口Modem池进行短信收发，8 口Modem池的平均服务率μ=1.6 条/s，服务台数量c=8，则λ/μ=2.41，服务强度$\rho=\lambda/c\mu$=2.41/8(<1)，代入M/M/c/∞/∞，得

Modem 池的空闲概率：

$$P_0 = [\sum_{k=0}^{c-1} \frac{1}{k!}(\frac{\lambda}{\mu})^k + \frac{1}{c!}\cdot\frac{1}{1-\rho}\cdot(\frac{\lambda}{\mu})^c]^{-1} = 8.9572\%$$

队列中等待发送的短信数：

$$L_q = \sum_{n=c+1}^{\infty}(n-c)P_n = \frac{(c\rho)^c\rho}{c!(1-\rho)^2}P_0 = 0.0016 \text{ 条}$$

系统中等待发送的平均短信数：

$$L_s = L_q + \frac{\lambda}{\mu} = 2.4141 \text{ 条}$$

短信在队列中等待时间的期望值：

$$W_q = \frac{L_q}{\lambda} = 0.0004 \text{ s}$$

短信在系统中逗留时间的期望值：

$$W_s = \frac{L_s}{\lambda} = 0.6254 \text{ s}$$

短信到达后必须等待的概率：

$$P_n(n>c) = \frac{1}{c!c^{n-c}}(\frac{\lambda}{\mu})^n P_0 = 0.3650\%$$

从实验结果可以看出：

(1)Modem 池空闲概率小，即 Modem 池长期处于繁忙状态，提高了利用率。

(2)短信在系统中逗留时间的期望值小，即一条短信从产生到发送成功的时间短，实时性高。

(3)短信到达后必须等待的概率小，即一条短信产生后几乎不需要等待就能被 Modem 处理，最大程度地利用了Modem。

6 结束语

本文提出了基于多队列和多线程的短信实时并发控制算法，该算法建立在多队列数据结构基础上，设计了短信队列和缓冲队列；采用多线程技术实现短信的实时并发控制，设计了多线程和线程池。该算法利用排队论模型和相关数学理论分析了性能。

SMSEngine 使用 Visual C++实现，可以与常见的数据库接口兼容。经实验，SMSEngine 可以方便快捷地被集成到 C/S 和 B/S 结构的信息系统中，目前已投入使用。

理论和实践证明，基于多队列和多线程的短信实时并发控制算法能够合理地控制短信的收发，最大限度地利用设备，从而提高设备的利用率和短信收发的速度与效率。该算法能够保证短信的实时性、并发性和数据共享性，满足了短信应用的需要。

参考文献

[1] 汤敬华, 曹　健. 基于短信的信息服务平台研究[J]. 计算机工程, 2004, 30(12): 238-240.

[2] ESTI. Digital Cellular Telecommunications System (Phase 2+): AT Command Set for GSM Mobile Equipment(ME)(GSM07.07) [EB/OL]. (1996-01-11). http://www.etsi.org.

[3] 房永龙, 周书民. 基于线程的短信实时并发算法[J]. 计算机应用与软件, 2006, 23(4): 87-89.

[4] 《运筹学》教材编写组. 运筹学[M]. 3 版. 北京: 清华大学出版社, 2005: 324-337.

[5] ESTI. Digital Cellular Telecommunications System(Phase 2+): Technical Realization of the Short Message Service(SMS) Point-to-Point(PP) (GSM03.40)[EB/OL]. (1996-03-28). http://www.etsi.org.