第36卷 第15期 计 算 机 工 程 2010年8月
Vol.36 No.15 Computer Engineering August 2010
·软件技术与数据库· 文章编号：1000—3428(2010)15—0068—04 文献标识码：A 中图分类号：TP393

面向工作流和服务的电子邮件系统模型

刘昌余，王红军，陈佳鑫，邹湘军
(华南农业大学南方农业机械与装备关键技术重点实验室，广州 510642)

摘 要：针对不同业务系统之间互相发送邮件和请求服务过程中的工作流约束问题，提出一个面向工作流和服务的多线程电子邮件系统模型。该模型在面向服务的体系结构基础上引入单元流处理思想，通过采用信号量触发并进行线程池调度管理，实现邮件服务和工作流之间的有序粒化组合。

关键词：面向服务；工作流；线程池；邮件

Workflow and Service Oriented E-mail System Model

LIU Chang-yu, WANG Hong-jun, CHEN Jia-xin, ZOU Xiang-jun
(Key Laboratory of Key Technology on Agricultural Machine and Equipment, South China Agricultural University, Guangzhou 510642)

【Abstract】According to workflow constraint problem existing in the process of sending mail and requesting service among different business system, this paper presents a Workflow and Service oriented Multithread E-mail System model(WSMES). Introducing mail work flow unit into Service Oriented Architecture(SOA), the model uses semaphore and thread pool to realize ordered and granulated service composition between mail service and workflow.

【Key words】service oriented; workflow; thread pool; mail

1 概述

现有电子邮件系统在实现和使用上具有较好的灵活性，但应用到面向服务的体系结构(Service Oriented Architecture, SOA)中时存在如下不足：(1)在企业应用方面，难以满足不同系统之间互相发送业务邮件、请求服务、共享资源过程中的工作流约束问题；(2)在个人应用方面，现有多邮箱管理软件主要有GMAIL、Fuser、Techemail等，此类多邮箱管理软件一般不支持扩展的邮件服务。

为解决上述问题，本文提出面向工作流和服务的多线程电子邮件系统模型(Workflow and Service oriented Multithread E-mail System model, WSMES)，并给出具体的实现算法。

2 相关工作

目前国内外对电子邮件系统的研究主要集中在邮件分类、系统安全性、系统架构等方面。

邮件分类技术的应用多集中于垃圾邮件的自动过滤。过滤方法主要有2类：(1)根据垃圾邮件的网络传播特征，运用黑白名单、IP过滤、群发过滤以及对发送方进行认证等技术，将类似于Spam的邮件过滤掉；(2)基于文本分类的过滤方法，有贝叶斯分类算法、词集合算法、基于概念的文档分类算法、K-最近邻接参照分类算法、支持向量机、Boosting方法等。

在安全电子邮件方面，研究者已提出多种方案，如PGP、S/MIME、MOSS等，它们在特定领域中发挥作用。此类方案通常采用混合加密算法进行加密、解密和签名、验证操作。对邮件系统安全性的研究还表现在对邮件病毒的传播规律进行分析，文献[1]研究了相关网络中病毒的传播，提出一个节点的连接度与其邻居节点的连接度相关，但没有可信的证据能支持连接类型。

在系统架构方面，人们提出了SOA。SOA的基本框架由3个参与者和3个基本操作构成[2]。目前，对SOA的研究集中在服务组合、服务协同和服务管理方面。在SOA结构下，待研究的问题主要是在服务建立协同之后，如何保障服务的可靠性，以及随外界环境的变化，组合服务如何进化[2]。SOA作为一种较新的体系架构，已被应用在很多领域，但在邮件系统领域，该方面的研究还处于起步阶段。文献[3]采用WCF(Windows Communication Foundation)技术实现一种面向服务的邮件系统，并对部分工作流进行处理，但在该系统中，对邮件的建模局限在特定系统中，没有组合外在的底层邮件服务，且缺少对请求服务进行优化处理的工作。本文在文献[3]的基础上，通过引入线程池和工作流处理单元解决服务组合和服务优化问题。

3 面向工作流和服务的邮件系统模型

3.1 基本概念

在定义WSMES模型之前，先给出以下定义：

定义1(邮箱帐户信息实体 MUserInfo) 邮箱帐户信息是登录到各电子邮件服务器的帐户信息，其集合称为邮箱帐户实体集(MUserInfos)，记为 MUIS。可以用一个元组〈email, password, alias, tagLog〉表示，其中，email为电子邮箱帐户；password为对应的帐户密码；alias为邮箱帐户别名；tagLog取布尔型值，tagLog=true表示自动登录，否则需要手动登录。加密后的邮箱帐户信息实体集记为EMUIS。

定义2(服务 Service) 在SOA中，服务是指能够提供某种业务功能的逻辑单元，服务的集合记为S。

基金项目：国家自然科学基金资助项目(50775079)；广东省自然科学基金资助项目(91510642010000)

作者简介：刘昌余(1984－)，男，硕士研究生，主研方向：虚拟现实；王红军(通讯作者)，副教授；陈佳鑫，硕士研究生；邹湘军，教授、博士生导师

收稿日期：2010-03-06 **E-mail：**yezhichunac@tom.com

定义 3(RSA 密钥对产生函数) 设 S={0, 1, 2, 3, 4}，分别表示 128 位、256 位、512 位、768 位和 1 024 位 RSA 加密，对于□KeySize∈S，定义 RSA 密钥对产生函数为 GenerateKeyPair(KeySize)={(KU,KR)| KU ∈ KUS, KR ∈ KRS}，其中，KUS 为公钥集；KRS 私钥集。

定义 4(RSA 加密函数和解密函数) 设已产生的密钥对为(KU, KR)=GenerateKeyPair(KeySize)。定义 RSA 加密函数为 $C=RSAE_{KU}(M)$，其中，M 为明文数据；KU 为已产生的公钥；RSAE 为 RSA 加密算法；C 是输出数据。RSA 解密函数为 $M=RSAD_{KR}(C)=RSAD_{KR}[RSAE_{KU}(M)]$，其中，KR 为已产生的私钥，与 KU 是一对密钥。

定义 5(线程池 ThreadPool) 线程池是管理线程的工具，可以用一个元组〈RQ, WHA, SMT〉表示，其中，RQ(Request Queue)是用户请求队列；WTA(Thread Array)是线程池中的工作线程集合；SMT(Scheduling Management Thread)是调度管理线程。

定义 6(邮件工作流单元(Mail work Flow Unit, MFU)) 邮件工作流单元指在一个邮件工作流程中对被处理对象的一次处理过程，是一个基本执行单元，其集合为 MFUS。在 MFU 中，由主线程根据线程调度算法，从线程池中取出空闲线程，以执行读邮件、发送邮件等操作。

3.2 WSMES 模型结构

针对传统多邮箱管理模型在企业应用和个人应用中存在的问题，本文提出 WSMES 模型。在该模型中，工作流被封装在 MFU 中，流的执行对应 MFU 切换，客户对邮件服务的请求采用线程池技术来响应和管理。WSMES 模型结构见图 1。

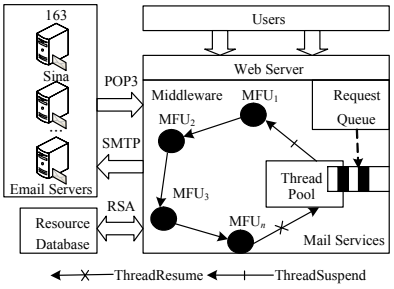

图 1 WSMES 模型结构

定义 7(WSMES 模型) WSMES=(U, WebServer, EmailServer, S, Middleware)，其中，U 是用户集；WebServer 是提供邮件服务的 Web 服务器；EMailServer 是各电子邮件服务器集；S 是服务集；Middleware 是邮件服务中间件。

图 1 描述了一个完整的 WSMES 邮件服务流程，具体描述如下：(1)WebServer 服务器接到用户的邮件服务请求，将请求者标识、请求内容传递给 Middleware 应用服务器。(2)Middleware 服务器把此请求放入 RequestQueue 队列。(3)ThreadPool 线程池根据调度算法从池中取出空闲的主工作线程，并在 ThreadResume 中等待内核中 Semaphore 对象，等到以后，主工作线程从工作队列中取出待执行的请求。(4)被执行的主工作线程进行 MFU 切换，在 MFU 单元中启动并同步其他工作线程，主要包括对资源库的 RSA 运算、用 POP3/STMP 协议与其他 EMailServer 通信以及进行其他流操作。(5)线程池用 ThreadSuspend 挂起并归并主工作线程。(6)Middleware 应用服务器将 MFU 单元的执行结果返回给 WebServer 服务器，并通过调用 WebService 将结果返回给用户。

3.3 WSMES 模型的特点

与传统的多邮箱模型相比，WSMES 模型具有以下优点：(1)提供可扩展的邮件服务。WSMES 模型建立在 SOA 架构上，可以对外提供组合形式的多邮箱邮件服务。各电子邮件服务器的服务被看作是 WSMES 中的即插模块，因此，WSMES 的实现不会局限于 POP3、SMTP 等协议，且具有动态扩展功能。(2)支持业务工作流。WSMES 模型将流的执行交给 MFU，并通过发送和释放信号量 Semaphores 来控制线程池，调度执行相应的工作线程。与传统多邮箱邮件模型相比，WSMES 模型能较好地适应不同企业应用环境。(3)支持参数化管理。在 WSMES 模型中，用户可以配置相应的邮箱参数，以满足多样化的服务需求。

4 基于线程池的模型实现

4.1 线程类与基函数

在 WSMES 模型中，除了应用程序主线程和主工作线程外，还使用了 5 种工作线程类：初始化线程 InitThread，邮箱登录线程 LogThread，邮件到达检测线程 HintThread，邮件收发线程 SendRecivThread，用户邮箱参数配置线程 ConfigThread。

定义 8(5 个线程类实例) 结合 WSMES 模型，设计上述 5 个工作线程类，以封装 Win32 API 函数中一些与线程操作相关的函数。用 new 函数生成相应的实例，即 InitThread* pInitT、LogThread* pLogT、HintThread* pCheck、SendRecivThread* pSendRecivT、ConfigThread* pConfigT。

函数 1(邮箱帐户信息解密函数 ReadAndDecry UserInfo) 假设用户 S(其私钥为 KR_S、公钥为 KU_S)，m_pUList[]为解密后的邮箱帐户列表数据，它是一个数组，数组的元素代表邮箱帐户信息实体。在 WSMES 模型中，当线程池调度到队列中某个用户时，对该用户的邮箱设置配置文件进行解密，即

ReadAndDecryUserInfo(m_pUList[]:MUIS) ◁
Set: $u_i \in MUIS$, $0 \leqslant i \leqslant n$
EMUIS= $RSAE_{KU_S}(u_1, u_2, \ldots u_n)$
return m_pUList[]=$RSAD_{KR_S}$ (EMUIS) ▷

函数 2(邮箱帐户信息分串函 DepartUserInfo) 定义 Depart UserInfo(ui, d, e, f, g)函数为 ui–>(d, e, f, g)上的关系，其中，ui 为输入；d、e、f、g 为输出。该函数主要将 MUIS 中的每个邮箱帐户信息实体字符串划分成 4 个子串，具体描述如下：

DepartUserInfo(ui, d, e ,f ,g) ◁
Set: ui∈MUIS , d∈email , e∈password,
f∈alias, g∈tagLog
pos [3] = {GetPos(ui, ",")}
d = ui.SubString(1, pos [0])
e= ui.SubString(pos [0], pos [1])
f = ui.SubString(pos [1], pos [2])
g= ui.SubString(pos [2], ui.Length())
return (d,e,f,g) ▷

函数 3(初始化 RSA 函数 InitRSAPara)，即

InitRSAPara() ◁
Set: KeySize∈{0, 1, 2, 3, 4} , KU∈KUS, KR∈KRS
(KU, KR) = GenerateKeyPair(KeySize)
RSA-> PublicKey = KU
RSA-> PrivateKey = KR ▷

4.2 线程的同步与控制

多线程[4]的使用使程序更灵活，但也带来了数据毁坏的访问冲突问题，因此，必须在适当的场合采取同步机制来确

保多线程应用程序的稳定运行。在 Windows 平台上，有多种同步机制，可以使用互斥器 Mutexes、信号量 Semaphores、事件 Events 以及临界区 Critical Sections 等。其中，临界区是用户模式的线程同步，事件、互斥器、信号量是内核对象的线程同步。

跨线程的控制流可以用 ICG(Inter-thread Call Graph)图来表示。ICG 即跨线程的调用图，它用节点表示方法，用节点之间的有向边表示线程间以及函数间的调用。结合 WSMES 模型，同一线程类的不同实体用相同的线程节点表示，用主线程 MainThread 对应 ICG 的入口，用虚线边表示一个跨线程的函数调用，用实线边表示线程内部的方法调用，图 2 是上述 WSMES 模型定义中的邮件服务流程对应的 ICG。

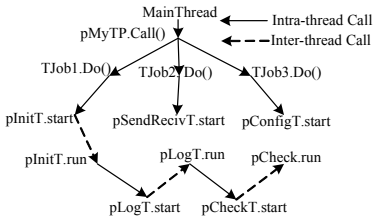

图 2　WSMES 模型的 ICG

4.3　算法实现

由 WSMES 模型的 ICG 可以看出 WSMES 邮件服务的实现过程主要包括 2 个过程，即服务请求分类过程和请求处理过程，均由线程池完成。

服务请求分类过程分为 2 个阶段(见算法 1)。线程池初始化阶段创建算法所需要的线程池环境包括若干个主工作线程、信号量和事件。假设请求的平均等待时间为 *AWT*，平均服务时间为 *AST*，那么对于一个具有 *N* 个处理器的系统，线程池中最佳的主工作线程的个数大约需要设置为 *N*×(1+*AWT*/*AST*)个。请求分类阶段根据请求字符串中的请求模式 iMode，得到 pMyTP.Call()调用需要的 TJob 处理类类型。

算法 1 服务请求分类算法

```
INPUT:   NUM, Req;//NUM=N×(1+AWT/AST);Req:请求字符串
OUTPUT: Ret;        //处理结果
S1. WTA* pMT[NUM];
S2. CreateThreadPool (pMyTP); CreateSemaphore();
S3. for (int i=0; i< NUM; ++i){
S4. pMyTP.CreateThread(pMT[i]);       pMyTP.AddThread
(pMT[i]); }
S5. int iMode = Req.GetRequireType();
S6. switch(iMode) { //请求类型
S7.     case 1:   Ret= UserLogin(Req); break; //登录
S8.     case 2:   pMyTP.Call(TJob1);   break; //启动 pInitT
S9.     case 3:   pMyTP.Call(TJob2);   break; //启动 pSendRecivT
S10.    case x:   pMyTP.Call(…);   break; } //启动其他 MFU 线程
```

服务请求处理过程根据服务请求分类算法得到的工作流处理类 $TJob_i$ 调用相应的 $TJob_i$.Do()，并返回处理结果。由 WSMES 模型的 ICG 可以看出服务请求大致能分为 3 类：$TJob_1$、$TJob_2$ 和 $TJob_3$。$TJob_1$ 主要完成多邮箱的自动登录、邮件自动检测、加密等操作，并通过调用线程 pInitT 来异步启动其他多个相关线程(见算法 2)。

算法 2 pInitT.run()线程算法

```
INPUT:   ui[]∈MUIS;
OUTPUT: d∈email, e∈password, f∈alias, g∈tagLog;
Ret;   //请求结果
S1. InitRSAPara();
S2. ReadAndDecryUserInfo(ui);
S3. for (第一个帐户 To 最后一个){
S4.   If (!结束信号){
S5.       DepartUserInfo(ui[i], d, e, f, g);
S6.       if (g) {   // true:自动登录
S7.          pLogT = new LogThread(d, e);
S8.          pLogT ->Resume(); //启动线程
S9.          pLogT ->WaitFor(); }
S10.      Ret = pLogT->GetLogResult(); } }
```

WSMES 模型的特点之一是支持业务工作流，流的执行促使 MFU 进行切换，并在切换时通过调用 ReleaseSemaphore()释放一个代表该流结束的信号量，然后在 $TJob_2$ 中等待并捕获该信号量，进行进一步的邮件处理(见算法 3)。比如在公司进货系统中，如果成功生成了订单，那么就会发送成功生成订单的邮件给厂商的联系人。

算法 3 MFU 工作流处理算法

```
S1. while(true){
S2.   WaitForMultipleObjects(信号量);
S3.   if (结束线程的信号) 退出该线程函数;
S4.   else if (工作流 1 信号) pSendRecivT.run(1); //发送流邮件
S5.   else if (工作流 i 信号) pSendRecivT.run(i); }
```

4.4　性能分析

为了简化测试，本实验设计一个自动发送服务请求的客户端调试器，其发送速度为 2 000 个请求/秒，并自动激活 WSMES 模型中工作流处理的信号量。实验的环境配置参数可参照表 1。本文设计了 2 组实验对 WSMES 模型系统的性能进行评估，测试内容包括：线程池的尺寸对服务器程序的性能影响和服务请求数对服务器程序的性能影响。测试方法来自 cnblogs 网站[5]。

表 1　实验环境配置

Type	Item	Configuration
Server	OperatingSystem	Microsoft Windows Server 2003
	Hardware	Intel Core 2 Duo E8400 CPU, 2 GB RAM
Client	OperatingSystem	Microsoft Windows XP
	Hardware	AMD Athlon(tm)64Dual CPU, 2 GB RAM
	Network	100 Mb/s
Other	Test Software	WSMES Debugger
	AnalysisSoftware	Excel

实验 1 线程池尺寸对服务器程序的性能影响

当服务请求数 *R_NUM*=4 000 时，测试 WSMES 模型系统的处理时间 *T* 与线程池尺寸 *P_NUM* 之间的关系。将 WSMES 系统线程池的尺寸配置为 $P\_NUM=2^i$，其中 $0 \leqslant i \leqslant 8$，并记录每种配置情况下的服务处理总时间 *T*。为了便于分析线程池的使用带来的影响，实验中记录了没有应用线程池时的服务处理总时间 *T*，结果数据见图 3。

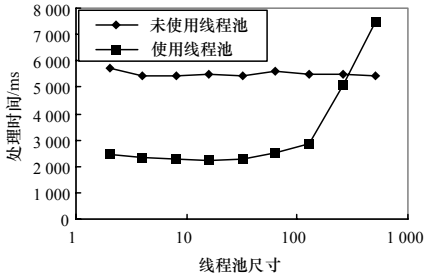

图 3　线程池的尺寸对服务器的性能影响

通过对图 3 的分析可以得到 2 个结论：

(1)线程池的使用可以有效提高服务器的处理性能，特别

是在大量并发请求存在的情况下。

(2)线程池尺寸变化对特定的服务程序影响较大，主要表现为：如果尺寸太小，将出现任务不能及时处理的问题，如果尺寸太大，将导致线程间同步、切换开销过大。因此，在设计时，要考虑线程池的优化问题。在 WSMES 模型算法中的解决方案如下：预先设置线程池的参考尺寸为 $NUM=N\times(1+AWT/AST)$，并在运行中动态改变工作线程数。

实验 2 服务请求数对服务器程序的性能影响

在线程池尺寸 *P_NUM*=32 时，测试 WSMES 模型系统的处理时间 *T* 与服务请求数 *R_NUM* 的关系。将 WSMES 系统服务请求数配置为 $R\_NUM=1\,000\times j$，其中，$1\leqslant j\leqslant 6$，并记录每种配置情况下的服务处理总时间 *T*。与实验 1 相同，在本组实验中记录了没有应用线程池时的服务处理总时间 *T*，结果数据见图 4。

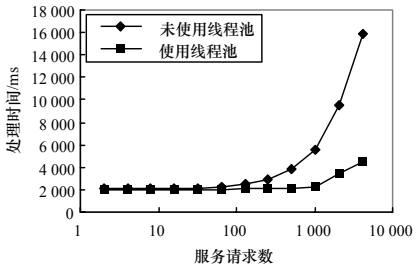

图 4　服务请求数对服务器程序的影响

通过对图 4 的分析可以看出，当 $R\_NUM\geqslant 258$ 时，服务请求数的进一步增加对 2 种技术的影响是不同的。线程池的优势表现在对于大量并发的任务的处理具有稳定性，适合处理复杂邮件工作流环境中的问题。

5　结束语

2 组相关实验结果显示，WSMES 模型在提高邮件服务处理的同时，能通过优化线程池的方式很好地适应复杂的工作流环境。

未来的工作包括：(1)以插件方式，添加其他类型协议的邮件服务，如 IMAP 邮件服务；(2)进一步提高 WSMES 系统的安全性，增加更多安全机制，如 MOSS 方案；(3)进一步完善 MFU 的处理能力，增加更多工作流，模拟更复杂的邮件服务处理。

参考文献

[1] Pastor-Satorras R, Vespignani A. Epidemics and Immunization in Scale-free Networks[J]. Physical Review Letters, 2001, 86(14): 3200-3203.

[2] 徐　罡, 黄　涛, 刘绍华, 等. 分布应用集成核心技术研究综述[J]. 计算机学报, 2005, 28(4): 433-444.

[3] 张　海. 基于 SOA 架构的邮件服务系统[D]. 四川: 西南交通大学, 2008.

[4] 王江鹏, 李先国. 基于多线程与缓冲池的 WebGIS 数据传输[J]. 计算机工程, 2010, 36(4): 79-81.

[5] Jonson 的技术空间. 线程池的介绍及简单实现[Z]. (2002-08-22). http://www.cnblogs.com/licheng/archive/2008/09/23/1296843.html.

编辑　陈　晖

（上接第 67 页）

3.4　结果分析

在 3.2 节和 3.3 节的 2 个实验中分别使用相同的数据源。表 4 给出了 *ABC* 分类和本文模型之间的比较分析。本文使用 2 种方法生成的分类预测 2009 年下一季度的情况。正确率是正确预测数量与实际数量的百分比。结果表明，本文方法更有效，在物料需求计划中具有更高准确率。

表 4　*ABC* 分类和本文方法的性能比较　(%)

方法	*A* 类比例	*B* 类比例	*C* 类比例	准确率
ABC 分类	12.32	22.41	65.27	81.12
本文方法	11.24	19.78	68.98	87.24

4　结束语

物料需求计划是一种现代生产控制模型、现代制造企业标准和管理技术。本文提出一种基于数据场和云模型的物料控制新方法，该方法能在供应链物料需求计划中自动找出重要材料。其模型包括 4 个部分：数据采集，数据预处理，分析与处理以及材料知识数据库建设。

在基本数据和物料需求计划的研究过程中，对一些因素和某些特殊情况考虑不周，下一步工作应深入研究提前决策因素、预先设置的准确性以及各种干扰的影响。

参考文献

[1] Hu Changchun. Study on the Re-designation of the Cost Management System of Small and Medium Manufacturing Enterprises Based on the ERP Theory[D]. Tianjin, China: Tianjin University, 2006.

[2] Shan Miyuan, Gu Hengping, Pu Lida. Research on Production Planning System of Mass Customization[J]. Packaging Engineering, 2006, 17(3): 183-186.

[3] Hamid M S, Shahram S, Fereydoon K. An Efficient Optimal Algorithm for the Quantity Discount Problem in Material Requirement Planning[J]. Computers and Operations Research, 2009, 36(6): 1780-1788.

[4] Mula J, Poler R, Garcia-Sabater J P. Material Requirement Planning with Fuzzy Constraints and Fuzzy Coefficients[J]. Fuzzy Sets and Systems, 2007, 158(7): 783-793.

[5] Miao Wenming, Chen Yong, Chen Guanlong, et al. Dynamic Adjustment of Material Requirement Planning for Postponement Manufacturing[J]. Acta Automatica Sinica, 2008, 34(8): 950-956.

[6] Li Deyi, Liu Changyu. Study on the Universality of the Normal Cloud Model[J]. Engineering Science, 2004, 6(8): 28-34.

[7] Qian Weining, Zhou Aoying. Analyzing Popular Clustering Algorithms from Different Viewpoints[J]. Journal of Software, 2002, 13(8): 1382-1394.

[8] Oyang Y J, Chen C Y, Yang T W. A Study on the Data Clustering Algorithm Based on Gravity Theory[C]//Proc. of PKDD'01. Berlin, Germany: Springer-Verlag, 2001: 350-361.

编辑　陈　晖