◎研发、设计、测试◎

面向应用的通用服务平台设计与实现

章剑林[1,3],李 班[2],徐从富[1]
ZHANG Jian-lin[1,3],LI Ban[2],XU Cong-fu[1]

1.浙江大学 计算机科学与技术学院,杭州 310018
2.浙江理工大学 管理学院,杭州 310018
3.浙江经贸职业技术学院 信息技术系,杭州 310018
1.College of Computer Science,Zhejiang University,Hangzhou 310018,China
2.College of Management,Zhejiang University of Science & Engineering,Hangzhou 310018,China
3.Dept. of Information Technology,Zhejiang Institute of Economic & Technology,Hangzhou 310018,China
E-mail:zhangjohn@vip.sina.com

ZHANG Jian-lin,LI Ban,XU Cong-fu.Design and implementation of universal service platform for application.Computer Engineering and Applications,2008,44(18):66-69.

Abstract: From the mode of operation and the thought of design of the developed background software of the network server,a universal service platform oriented toward the application is designed,which can be applied in several industries,such as bank,securities business and telecommunication.Being expanding the design characteristics of the platform,such as high performance,high reliability,strong expansibility,good integration capabilities and the ability of Cross-Platform,the development technology based on this platform is introduced.Through testing,the developed universal service platform has reached the expected the requirements of design.Eventually,the information of the application of the universal service platform in securities business is introduced.
Key words: the universal service platform;application;design;implementation

摘 要:借鉴网络服务器成熟后台软件的工作模式和设计思路,设计了面向应用的通用服务平台,可广泛应用于银行、证券和电信等行业。在阐述平台具有高性能、高可靠性、强扩展性、较好集成能力及跨平台运行等设计特点同时,还介绍了基于平台的应用开发技术。经测试,开发的通用服务平台系统已达到预期设计要求。最后简要介绍了通用服务平台在证券业务中的开发应用情况。
关键词:通用服务平台;应用;设计;实施
DOI:10.3778/j.issn.1002-8331.2008.18.021 **文章编号:**1002-8331(2008)18-0066-04 **文献标识码:**A **中图分类号:**TP311

1 概述

在日益激烈市场竞争中,许多服务业、商务企业为了提高自身竞争力,不断地推出新业务新服务,并更新原有系统;而在推出一项新业务时,应用系统开发速度和稳定性测试,对企业来说十分重要。因此,需要一个能直接在应用环境基础之上进行业务开发的应用服务器平台。企业可以借此节省大量的时间和金钱来推出新的业务,从而提供自身的竞争力。

为此,本文提出了低成本、高性能、高可靠的企业级应用服务解决方案——通用服务平台。用户可以在通用服务平台上进行企业应用的二次开发。通用服务器平台不仅可应用于证券、银行、电信等业务性很强的行业,还可应用于解决网络瓶颈等技术难题。

2 服务平台原理及性能设计

2.1 服务平台的工作原理

通用服务平台是在借鉴大型数据库、Apache、Linux 等网络服务器成熟后台软件的工作模式和设计思路上进行的整体设计,无论是性能、可靠性还是可管理性,都比基于 Windows 窗口体系的原有通信平台有质的提高。

系统分核心应用部分和管理终端部分,核心应用由通用网关精灵进程和扩展模块两部分组成,用服务方式运行于服务器群上,管理终端可以远程运行,对核心应用的各网关进行统一管理。通用服务平台工作原理图如图 1 所示。

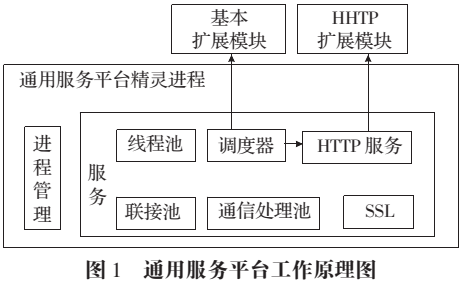
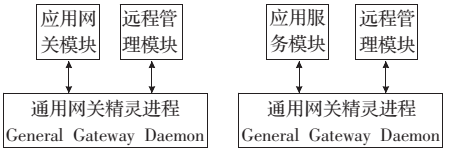


图 1 通用服务平台工作原理图

基金项目:浙江省科技计划项目(No.2007C33071)。
作者简介:章剑林(1966-),男,副教授,主要研究兴趣为电子商务与电子政务、商务智能和网络安全等方向;李班,MBA,副教授;徐从富,博士,副教授。
收稿日期:2007-10-29 **修回日期:**2008-01-17

整个核心应用由通用网关精灵进程和扩展模块两部分组成,通用网关精灵进程负责进程管理、线程管理、网络通信管理、日志等通用功能,而将与应用相关的通信协议分析、业务处理等功能封装到不同的扩展模块中。通用网关精灵进程配上不同的扩展模块就构成了不同的功能网关。如图 2 为网关结构和应用服务器结构。



(a)网关结构图　(b)应用服务器结构图

图 2　扩展模块结构图

2.2　服务平台的性能设计

2.2.1　功能可扩展性

根据通用服务平台的基本工作原理,服务平台主体是一个健壮的服务器平台和通信平台,所有商务应用功能均采用扩展模块的方式开发,既有适应不同应用领域的开发灵活性,又能让所有扩展功能享受基础平台提供的性能、可靠性和管理一致性的保证。

通用服务平台内置一个 HTTP 服务模块。Apache、MS IIS 等经典 Web 服务器,多按 HTTP 短网络联接而设计的,其内部处理线程/进程数和并发联接数在几百个的数量级。当出现类似聊天室聊天和 Dos 攻击等情况时,由于客户端并发数增多原因而易出现拥堵现象。而 HTTP/1.1 协议下的长网络联接方式访问网站时表现出较强的联接能力。通用服务平台内置的 Web 服务模块采用 HTTP/1.1 长联接方式,充分利用其并发联接能力强的特点,并取得了优于经典 Web 服务器的性能。

2.2.2　高性能设计

(1)分组多线程服务器结构(Grouped Multi-Thread Server)。常用服务器软件工作模式有两种:一种是专用进程/线程(Dedicated Process)模式,以分配一个专用的进程/线程来处理某个网络连接上的所有请求,直到网络连接终止;另一种是共享多线程服务器(Multi-Thread Server,MTS)模式,由一组固定数量的线程来为数量众多的网络连接服务,从线程组中分配一个线程为网络请求服务,该请求服务完毕线程即归还到线程组中。专用进程/线程模式处理效率高,但系统开销大,不适合于 Windows 等系统资源较紧张的操作系统;共享多线程服务器模式可以用较少的线程服务于大量的网络连接,比较适合做 CG-I,AG-X 等采用网络长连接的应用网关。但共享多线程服务器模式容易出现一种设计瓶颈,表现为系统进入一种饥饿状态:当系统的吞吐量(单位时间处理的总请求数)小于客户端加的总工作压力(单位时间客户端发出的总请求数)时,系统就忙不过来了,表现为客户请求出现拥堵和超时。但值得注意的一个现象是此时 CPU 利用率未必很高,而系统吞吐量却再也上不去了。这是因为网关作为一个中间件,很多工作是转发到后台去完成的,处理线程只是忙于“等待”后台返回结果,“等待”状态并不消耗本机 CPU,而此时却没有线程有空去处理新请求了。解决这个问题的一个方式是调节线程数,线程数增大时吞吐量随之增加。有很多服务器软件(Oracle/Apache)也利用了这种配置工作进程线程数的调优手段。

通用服务器平台系统采用线程分组的方式来改进标准的共享多线程服务器模式。将若干个网络连接归入一组,每组指定几个线程来专门为其服务,那么在一个组中,以共享多线程服务器的方式工作,而在不同组之间是独立的,又可以收到专用线程模型的好处。组内的网络连接数有限,比较容易处理。当客户端的网络连接数大量增加时,系统会自动创建新的组来满足要求。分组多线程服务器的策略,总工作线程数会随网络连接数同比增加,从而实现了自动调优,这也就是通用服务器平台籍以实现高性能的一个关键算法。

(2)并发能力和处理速度。设计并发处理能力为每服务器 5 000~10 000 联接,服务平台对每笔请求的平均处理延时在数百毫秒数量级。并保证系统处理速度平稳而高效,以满足不同的应用需求,提高客户满意度。

2.2.3　高可靠性设计

(1)采用可靠性较高的编程模型。引入开源项目中的成熟代码;实行核心应用与图形界面分离,采用标准服务器工作模式,核心应用设计成纯后台应用程序,运行不受窗口系统影响;将图形界面移入远程管理终端中,与核心应用跨机运行;核心应用抛弃了传统 Windows 程序消息机制和窗口系统,改用可靠性更高的 UNIX 编程风格。

(2)可靠传输。支持断点续传和数据一致性检查。这种传输方式具有极强的灵活性,即使信息接收端不工作,信息发送端仍可正常工作并保证接收端启动后收到这些信息。

(3)软件负载均衡器。专门设计软件负载均衡模块,采用 Round-Robin 算法、最短响应时间优先算法、最小联接数优先算法、IP 保持算法、HTTP SESSION 保持算法等多项算法。

2.2.4　较好的集成能力

平台内置 Web 服务功能,并兼容.NET、J2EE、Web Service、CORBA、消息队列和 LDAP 等多项技术的集成。

2.2.5　跨平台运行能力

包括 Windows 系列和 Linux/Unix 系列等诸多平台,系统更健壮。

另外在安全性和可管理性方面也必须考虑通用服务平台开发、运行的需要。

3　服务平台应用开发关键技术

通用服务平台作为一个开放的通信服务运行平台,可以通过自行开发扩展模块来定制各种业务功能。总的开发流程为:定义应用通信协议→开发通用服务平台扩展模块→配置运行→测试。

通用服务平台采用 Apache 的池式内存管理技术,以及相关 apache group,openssl.org 等开源代码,一部分函数和数据结构会带一个 apr_pool_t* 类型的参数,指向一个 Apache 内存池,并使用 Apache 提供的一组函数进行内存分配操作。内存生命周期与池相关,即每块内存使用完毕后不单独释放,而是整个池一起清空。内存池可以分配子池,子池生命周期短于父池,以适应不同的生命周期需要。内存池包括模块内存池、连接内存池和请求内存池等 3 个层次。其生命周期可如图 3 所示。

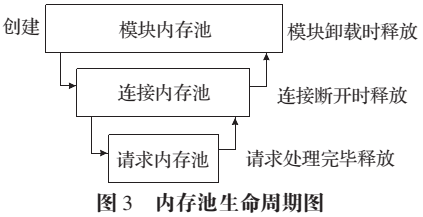


图 3　内存池生命周期图

3.1 应用通信协议设计

应用通信协议可以按业务需要自行设计,目前平台支持两种类型,一种是在基于 SOCKET 通信的,另一种是基于 HTTP 协议的。

例 1 自定义的 echo 服务协议

功能:客户端连接到 echo 服务器,向服务器发送字符串,服务器收到这些字符串后把相同的内容发还给客户端。

协议:每个请求和应答以回车换行为逻辑分隔,接收端从输入数据流中寻找下一个回车换行符,如果找到就把这行内容截取出来作为一个逻辑数据块进行处理。

例 2 HTTP 版的 echo 服务协议

请求: POST /echo HTTP/1.1

Host:www.myadd.com

Content-Type:text/plain

Content-Length:12

Hello World!

应答: HTTP/1.1 200 OK

Content-Type:text/plain

Content-Length:12

Hello World!

3.2 通用服务平台功能扩展模块开发(C 语言)

(1)自定义协议(SOCKET 通信)扩展模块。通过扩展模块来动态地扩充功能,在 Windows 平台上的通用服务平台扩展模块是个 DLL 文件,该 DLL 可以使用 C/C++开发。扩展模块 DLL 必须 export 一个 module 类型的数据结构,以便定义回调函数,供通用服务平台来调用。同时,通过登记对应几个系统事件的回调钩子函数来处理包括模块初始化、网络连接建立、从输入数据流中截取请求数据块、处理请求、网络连接断开、模块卸载等相关事件。

(2)HTTP/HTTPS 协议扩展模块。接口同基本扩展模块大致相同,但扩展模块必须实现请求处理例程,不必实现从输入流截取请求例程。

(3)公用函数。包括访问数据库、定时任务定制、访问运行环境、日志、统计、报警等。

3.3 配置

每个模块的配置信息可定义成 XML 格式, 放在 ggd.conf 中与模块有关的小节中,模块初始化时平台会把这部分内容的 XML 子树传给模块初始化事件处理例程。

4 服务平台性能测试

4.1 测试方案设计

(1)测试环境

测试中使用了 4 台 PC 便携机,各机配置如表 1。

表 1 PC 机配量

	PC1	PC2	PC3	PC4
CPU	PIII 933	PIII 933	PIII 700	PII 266
内存	256 M	192 M	128 M	128 M
网卡	100 M	100 M	100 M	100 M
操作系统	Win2000 Server	Win2000 Server	Win2000 Server	Win2000 Server

(2)测试分组

在服务器端模拟了申万 AGX 的登录、更密、查询余额、下单业务,进行了 4 组测试:

第 1 组:1 个客户端, 客户端与服务端运行在不同 PC 上。第 2 组:1 个客户端, 但客户端与服务端运行在同 1 台 PC 上。第 3 组:有 2 个客户端,客户端与服务端运行在不同 PC 上。第 4 组:有 3 个客户端,客户端与服务端运行在不同 PC 上。

测试时每个客户端启动后首先创建 3 000 个连接,然后每种业务各向服务器端发送 3 000 个请求,服务器端根据客户端请求的并发量自动创建服务线程,测试服务器端平均响应情况与每笔请求响应情况。所有测试时服务端都运行在 PC1 上,PC2、PC3、PC4 作客户端用。

4.2 测试过程及测试结果

(1)第 1 组。在只有 1 个客户端的测试中,测试 3 次,分别将客户端放在 PC2、PC3、PC4 上, 每次发送 12 000 个请求,系统未出现不稳定现象,服务端线程数最多达到 96 个,服务器内存使用 77 M,服务端 CPU 利用率不到 30%,网络带宽峰值不超过 1 089 K,服务端平均响应时间如表 2。

表 2 服务端平均响应时间(第 1 组)

功能	第 1 次	第 2 次	第 3 次
	PC4 性能	PC3 性能	PC2 性能
花费时间/s	31	24	16
每笔响应时间/ms	2.6	2.0	1.3
服务端 CPU 利用率/%	<30	<30	30
网络带宽峰值/K	<1 089	<1 089	1 089

(2)第 2 组。在只有 1 个客户端的测试中,测试 1 次,客户端与服务端都运行在 PC1 上,发送 12 000 个请求,系统未出现不稳定现象, 服务端线程数最多达到 96 个, 服务器内存使用 77 M,服务端平均响应时间如表 3。

表 3 服务端平均响应时间(第 2 组)

功能	PC1 性能
花费时间/s	21
服务端处理请求数/笔	12 000
每笔响应时间/ms	1.8
服务端 CPU 利用率/%	100

(3)第 3 组。在有 2 个客户端的测试中,测试 1 次,客户端运行在 PC2 与 PC3 上。每客户端发送 12 000 笔请求。系统未出现不稳定现象,服务端线程最多达至 174 个,服务器内存使用 143 M。客户端响应时间如表 4。

表 4 客户端响应时间(第 3 组)

功能	PC3 性能	PC2 性能
工作时间/s	29	18
每笔平均响应时间/ms	2.4	1.5

由于客户端速度不一样,将所有客户端同时工作时间作为有效时间,因此有效时间为 min(18,29)=18 s。在有效时间内 PC2 发送请求数 12 000 笔,PC3 为 12 000*18 s/29 s=7 448 笔,因此服务端处理请求 12 000+7 448=19 448 笔, 服务端平均响应时间如表 5。

(4)第 4 组。在有 3 个客户端的测试中,测试 1 次,客户端运行在 PC2、PC3 与 PC4 上。每客户端发送 12 000 笔请求。系

表 5　服务端平均响应时间(第 3 组)

功能	性能
服务端有效工作时间/s	18
服务端有效请求/笔	19 448
服务端 CPU 利用率/%	60
网络带宽峰值/K	5 102
服务端平均响应时间/ms	0.9

统未出现不稳定现象,服务端线程最多达至 284 个,服务器内存使用 230 M。但已出现瓶颈,网络带宽占用最大达 936 K,服务器端 CPU 平均利用率 90%,一度达 100%,其中系统内核占 70%,硬盘操作频繁。客户端响应时间如表 6。

表 6　客户端平均响应时间(第 4 组)

功能	PC4 性能	PC3 性能	PC2 性能
工作时间/s	57	51	41
每笔平均响应时间/ms	4.8	4.3	3.4

有效时间是指所有客户端同时工作时间,为 min(57,51,41)= 41 s。在有效时间内 PC2 发送请求数 12 000 笔,PC3 为 12 000× 41 s/51 s=9 647 笔,PC4 为 12 000×41 s/57 s=8 632 笔,因此服务端处理请求 12 000+9 647+8 632=30 279 笔。平均响应时间如表 7。

表 7　平均响应时间

功能	性能
服务端有效工作时间/s	41
服务端有效请求/笔	30 279
服务端 CPU 利用率/%	90(一度达 100)
服务端内核 CPU 利用率/%	50
网络带宽峰值/K	936
服务端平均响应时间/ms	1.4

4.3　测试结果分析

从第 1 组与第 3 组的测试结果来看,服务器在没有产生瓶颈时,服务端响应性能比较平稳。不管是因客户端提高处理能力而加大压力,还是增加客户端而增加压力,服务端的综合性能反而有所提高,如每笔平均响应时间从 1.3 ms/笔提高到 0.9 ms/笔。当服务端综合压力大幅增加时,每笔请求响应时间并没有同比增加。如第 1 组与第 3 组的 PC2,当服务端从 769 笔/秒增加到 1 111 笔/秒时,压力增加 44%,但 PC2 的单笔请求性能并没有多大变化,从 1.3 ms/笔增加至 1.5 ms/笔,只增加 15%。表 8 是第 1 组的 PC2 性能与第 3 组的 PC2 性能比较。

表 8　两组 PC2 性能比较

	第 1 组	第 3 组
服务端综合性能/(ms/笔)	1.3	0.9
服务端综合压力/(笔/s)	769	1 111
PC2 单笔响应速度/ms	1.3	1.5

从第 2 组与第 4 组的结果来看,当服务端产生 CPU 和内存瓶颈时,性能显著下降,如第 4 组服务端系统内核 CPU 占用 50%,经分析部分原因是内存不足产生的虚拟内存交换消耗,另一部分原因是服务端线程较多引起的线程切换消耗。由于服务器是 PIII 933 CPU,内存为 256 M,如增加 CPU 主频和加大内存,应可以提高系统响应性能,延缓出现瓶颈的压力点。当服务端出现瓶颈时,系统并没有崩溃,还能可靠的运行,只能响应变慢。

4.4　测试结论

(1)当服务端产生瓶颈时,系统不会崩溃,但性能有所降低。实际应用中,服务端达到上千笔/秒的可能性并不大,系统基本达到设计目标。

(2)软件本身没有设计瓶颈,影响响应性能主要因素为:一是 CPU/网络处理速度和内存大小,二是服务端业务复杂程度。可以通过提高服务端的 CPU 与内存来提高系统性能。

5　服务平台应用实例

本文以通用服务器平台上开发证券行业应用为实例,可以构建网上交易服务器、专家在线聊天室、开发应用服务器、开发报盘服务器、开发行情服务器、开发网络通讯服务器、银证转帐前置机等多种应用模式。

5.1　证券业务通讯平台

借助于服务平台整合本地交易和远程交易的实现,对分布或集中的交易服务进行灵活整合,满足各种不同要求,实现证券信息服务与交易服务通过通讯平台的集成(如图 4 所示)。

图 4　多种业务资源的整合

5.2　网上交易系统

图 5 是网上交易的模块结构图,其 WebServer 是用通用服务器开发平台开发的,通讯协议采用 Https,提供专用的 Https 服务,省去无关服务,大大提高系统并发性与稳定性。

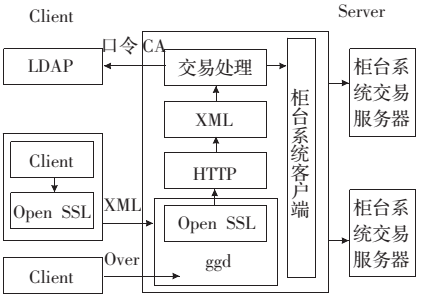


图 5　网上交易模块结构图

5.3　大容量在线交流平台

内置 Web 服务,类似于专家在线、网络聊天室,可以承受几千个人同时在线交流(在线人数看网络带宽而定),此时服务器反应时间不会出现明显变慢。

6　小结

经过系统测试和在相关领域的实际应用证明,本文所设计实现的通用服务平台具有低成本、高性能、高可靠的特点,且已经成功应用于证券行业的二次开发,有效地提高了网络通信效率。同样,根据系统设计的适应性和高性能特点,本通用服务器

(下转 75 页)