CACF:一种面向P2P客户端的应用程序框架*)

宋 杰 卢显良 韩 宏
(电子科技大学计算机学院 成都 610054)

摘 要 P2P客户端程序对网络通信的要求正变得越来越复杂。在开发过程中使用应用程序框架可以比较好地处理这种复杂性。本论文提出了框架CACF。它支持水平,垂直和协作过程等多种并行模式,能通过消息优先级和多种排队模型实现流控和端到端QoS,引入多路复用器实现对网络连接的多路复用,复合消息则消除了费时的内存拷贝。作为P2P系统Virtual Helpdesk的客户端,JCVIEWER的开发实践证明了CACF的有效性。

关键词 P2P,应用程序框架,网络通信,多路复用,QoS

CACF:A Communication-Oriented Framework for Client Applications

SONG Jie LU Xian-Liang HANG Hong
(Computer Science Department of UESTC,Chengdu 610054)

Abstract Complexity for communication requirements of the modern client applications is increasing dramatically. Framework techniques provide principles,methods,and tools that significantly reduce the complexity and cost of developing communication software. This paper presents CACF,a communication-oriented framework for client applications. It supports vertical,horizontal and co-routine parallelism,introduces layer-to-layer flow control with message priority and queue,multiplex the network connection with multiplexer and demultiplexer,and avoids the memory copy with composable message. A thin-client application,JCVIEWER,proves that CACF is flexible and efficient.

Keywords P2P,Framework techniques,Communication-oriented,Multiplex,QoS

1 引言

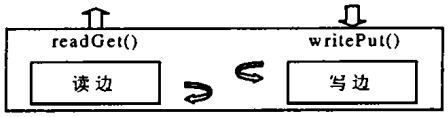
目前,P2P客户端对网络通信的要求正变得越来越复杂。从连接管理的角度来看,客户程序需要同时管理多条网络连接,每条连接往往又有自己不同的持久特性。从数据处理的角度来看,客户程序需要对数据做诸如加密压缩等更多更复杂的变换。从信道复用的角度来看,客户程序往往需要在单一信道上传送多种不同类型的数据。某些网络会议系统的客户端就需要在一条连接上同时传送文本,语音,视频和屏幕图像等四种不同的数据。从QoS的角度来看,传送延迟和吞吐量不再是决定服务质量的唯一指标,视频音频等应用对网络传送提出了稳定的新要求。

为了简化应用程序的编写,一些应用程序框架被陆续开发出来。MFC和OWL是Windows程序员最常用到的两个应用程序框架,但它们主要着眼于用户界面的开发,没有专门涉及网络通信。JAVA的IO包和NIO包以及UNIX的套接字实现了基本的通信机制,但它们没有向程序员提供高层次的通信抽象。STREAMS和CLICK是两个专门为通信而开发的框架,但前者主要用于开发操作系统内核的网络协议栈,后者则主要用于开发软件路由器中的IP处理单元,并且缺少对多线程的支持。JAVA RMI,DCOM和CORBA实现了高层次的通信抽象,但它们缺少诸如异步IO,QoS等特性,更重要的是,作为一种通用的中间层,它们不可能针对某个具体的程序进行优化。SEDA是一个为开发高度并行的网络服务器而设计的框架,但由于线程调度引入的不确定因素,使它不适合用来开发对时间比较敏感的通信程序。

本论文提出了一个面向P2P客户通信的框架CACF。流是CACF最重要的概念,代表一条全双工的信道。一条流由流头和多个链接在一起的模块组成。程序的通信处理功能就封装在这些模块中。其中,最底层的模块往往担负直接管理网络连接的任务,而中间模块则负责对数据做加密压缩等变换。此外,还有的模块实现了多种排队模型,通过和消息优先级相配合,程序能实现多种流控和QoS算法。模块之间的数据交换通过消息来实现,CACF的消息结构能有效地避免对性能影响很大的内存拷贝操作。

2 框架结构

流是一个由流头和若干模块链接而成的全双工信道。模块(图1)封装了程序对通信各个环节的需求。它通常由一个读边和一个写边构成,读边接收从下侧模块来的数据,处理后再送到上面的模块,写边接收从上侧模块来的数据,处理后再送到下面的模块。图2的流由三个模块组成,连接模块负责管理和服务器之间的TCP连接,SSL模块负责加密去往服务器的数据并解密从服务器来的数据,Zlib模块则负责数据的压缩与解压。一个程序可以同时打开多条不同配置的流,并且能动态地向流中压入新的模块。



一个模块由一个读边和一个写边构成。模块通过调用下侧模块的readGet()方法从下侧模块的读边获取数据;通过调用下侧模块的writePut()方法向下侧模块的写边派送数据。一模块可以有两个独立的线程。一个负责处理读变的异步事件,另一个负责处理写边的异步事件。

图1

*)本文受国家95重点攻关项目支持。宋 杰 博士研究生,主要研究方向:计算机网络、网络存储、分布式操作系统等。卢显良 教授,博士生导师,主要研究方向:计算机网络、操作系统。韩 宏 博士,主要研究方向:计算机网络、软件工程。

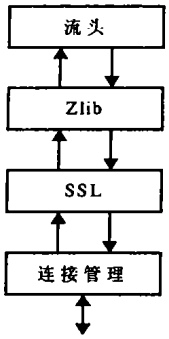


图 2　一个简单的流

下面，论文根据文[2]提出的分类法详细描述 CACF 各个方面的特性。

框架的扩展机制—框架的扩展主要是通过开发更多的具有特定功能的模块来实现的。一方面，开发人员可以直接继承基类 Module，然后通过实现 Module 中相应的方法来获得新模块，这种方式具有最大的灵活性；另一方面，CACF 提供了一个能满足大多数模块需求的 C++模版 ModuleT，开发人员只需要把读边写边的功能封装在基类 Task 的派生类中，然后用它们作为模版的参数，就可以直接得到新的模块。

模块之间的数据传递机制—消息是在相邻模块之间传递数据和控制信息的唯一方式。模块通过调用下侧模块的 writePut 方法向下传送消息，以及 readGet 方法获得来自下侧模块的消息。消息由消息头部和连接消息数据各个片段的链表组成。头部保存了消息类型和优先级等信息；链表的存在避免了在数据两端或者中部添加新数据带来的内存拷贝的开销，而内存拷贝往往是通信类程序的性能瓶颈。

模块间流控和 QoS 支持—除了加密解密这种有具体功能的模块以外，开发人员还可以用模块来实现各种排队模型。通过对各种排队模块与消息优先级的结合使用，流就能实现不同的流控和端到端 QoS 算法。图 4 通过一个 PQ 排队模型保证了图像流和鼠表键盘流的带宽要求。

多路复用—有时候多个流需要复用同一条网络连接，CACF 提供了多路复用模块来实现这个功能。与普通模块不同，一个多路复用模块可以有多个相邻的上侧模块，每个上侧模块接受分属于不同数据流的消息。CACF 允许程序向/从多路复用模块中动态地加入/删除流。

并行模型—CACF 支持三种并行模型。一种是垂直并行，模块本身没有属于自己的线程，而是借用程序的线程，通过 writePut 和 readGet 以过程调用的方式驱动流中所有的模块。一种是水平并行，有些模块需要处理诸如定时时钟、网络事件这样的异步事件，通过定义 readService 和 writeService 方法，CACF 会在创建模块时，自动创建两个独立的线程来驱动模块的读边和写边。另一种是协作过程(coroutine)，这种并行模型通常在多路复用器中实现，共享同一网络连接的多个流通过协作过程来实现并发可以避免线程间同步带来的系统开销。这三种并行模式也可以同时存在于一个程序中，下面的 JCVIEWER 就是这样一个例子。

3　实例分析

JCVIEWER 是我们正在开发的 P2P 系统 Virtual Helpdesk 中的一个客户端程序。通过 JCVIEWER，通信的双方可以共享对方的屏幕，远程操作对方的电脑，传送数据文件，以及发送即时消息。如图 3 所示，JCVIEWER 的通信需求是比较复杂的。JCVIEWER 需要周期性地和控制服务器交换控制信息，从通信服务器接收屏幕图像、数据文件和即时消息等信息，同时又要向通信服务器发送即时消息、数据文件和鼠标键盘事件等信息。在这些数据当中，屏幕图像和鼠标键盘事件有比较高的实时性要求；数据文件、控制信息和即时消息需要加密解密；屏幕图像需要解压，数据文件需要压缩。

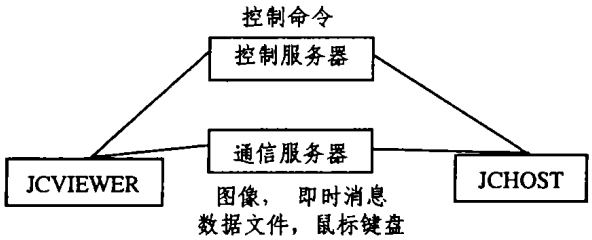


图 3　JCVIEWER 的工作环境

图 4 给出了 JCVIEWER 网络通信部分流的结构。最顶端的流负责和控制服务器交换控制消息，其中 SSL 模块负责消息的加密解密，HTTP 模块负责把消息封装在 HTTP 请求中，连接模块负责维护和控制服务器之间的 TCP 连接。

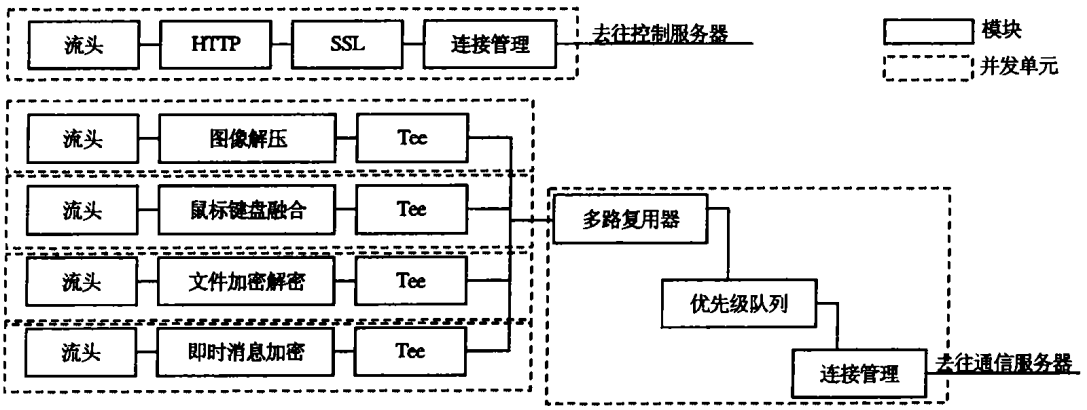


图 4　JCVIEWER 的流结构

即时消息、屏幕图像、鼠标键盘事件和数据文件复用一条和通信服务器之间的 TCP 连接。在多路复用器左侧，各个流借用用户线程实现垂直并发模型，复用器本身实现了协作过程并发模型。连接管理模块左侧的优先级队列模块为屏幕图像和鼠标键盘事件保留了足够的带宽。

4　相关研究

套接字以及 JAVA 的 IO 包和 NIO 包为开发网络程序提

供了基本的通信原语，但它们没有实现更高层次的抽象。如果直接以它们为基础进行开发，结果往往是工作量大，程序容易出错，程序模块的封装程度低，可复用性差。

JAVA RMI、基于 WINDOWS 的 DCOM 以及跨平台的 CORBA 是目前应用比较广泛的中间件平台。开发人员可以按其指定的规则把应用程序的功能封装在一系列可复用的模块中，然后通过它们提供的基础平台实现模块间的通信交互。这大大提高了开发效率和系统的模块化程度。但它们往往对底层平台有较高的要求，也缺少 QoS、异步通信等特性。同时，作为一个通用平台，它们不可能针对某个应用程序作专门的优化。

STREAMS 是 UNIX 内核实现的一个框架。它被广泛应用于网络协议栈、设备驱动程序和终端服务的开发。CACF 借鉴了 STREAMS 的流的概念，运行时模块的压入与弹出和避免内存拷贝的复合消息结构。CACF 与 STREAMS 的不同之处在于，STREAM 中的所有模块都必须使用唯一的由系统提供的队列模型及基于其上的流控算法，而 CACF 允许应用程序实现自己的队列模型和流控算法。

CLICK 是为开发软件路由器而设计的一个框架。一个 CLICK 程序就是一个有向图。图的每个节点是一个被称为 Element 的程序模块，它实现某个具体功能。图的边代表 IP 报文在结点之间可能的流通路径。CACF 借鉴了 CLICK 把各种队列模型也作为独立模块实现，然后在完成实际报文处理的模块之间插入各种队列模块，从而实现 QoS、流控、带宽分配的思想。但 CLICK 缺乏消息复合机制，并且只支持协作过程这单一的并行模型。

SEDA 是一个为开发高度并行的网络服务器而设计的框架。在 SEDA 中，程序的功能被划分到多个级连的 Stage。每个 Stage 由一个队列和多个从该队列中读取请求的线程构成。SEDA 能支持很高的并发度，同时保证服务器在超载情况下的稳定性。但由于线程调度等原因，它不适用于对服务实时性要求比较高的应用。

结论 本论文提出了一个支持 P2P 客户端程序开发的应用程序框架 CACF，并讨论了使用它开发的具有复杂通信需求的应用程序 JCVIEWER。程序的开发实践证明，CACF 灵活的扩展机制，水平、垂直和协作过程等多种并行模型，复合的消息结构，通过消息优先级和多种队列模型实现的流控和 QoS，以及多路复用等机制能有效地降低开发通信程序的复杂度，构造出健壮高性能的 P2P 客户端程序。

参 考 文 献

1 Schmidt D. Applying Patterns and Frameworks to Develop Object-Oriented Commnunication Software. Handbook of Programming Languages, Vol. 1 MacMillan Computer Publishing, 1997
2 Schmidt D, Suda T. Transport System Architecture Services for High-Performance Communication Systems. IEEE Journal on Select Areas in Communication, 1993
3 Java 2 Platform, Standard Edition, v 1.4.1 API Specification, Sun Microsystem Press
4 STREAMS Programming Guide. Sun Microsystem Press
5 Welsh M, Culler D. SEDA: An Architecture for Well-Conditioned, Scalable Internet Services. The 18th Symposium on Operating Systems Principles, 2001
6 Morris R, Kohler E. The Click Modular Router. The 17th ACM Symposium on Operating Systems Principles, 1999

（上接第 32 页）

在实际应用中采用全局相似度来全面评估两个概念相匹配的程度，它是上面相似度计算值的线性组合。对每种相似度引入一个适当的权值来计算全局相似度。在应用中可采取灵活的匹配策略。利用全局相似度的值，可以在一定程度上控制匹配目标概念的概念数量。给定目标概念 C 和对等体本体论 PO，用 Mc 表示匹配 C 的 PO 的概念集合，即相对 C 有一个全局相似度值 $GA\neq 0$ 的概念。这里定义两种匹配策略用于选择匹配概念：

(1)相似：在相似策略里，其目的是发现与概念 C 语义对应的概念。所有全局相似度值大于等于相似域值 $t_s>0$ 的概念被检索，即

$Msc=\{C'\in PO|GA(C,C')\geqslant t_s\}$。

(2)等价：在等价策略里，其目的是发现等价的概念。所有全局相似度值大于等于等价域值 t_e，且 $t_e>>t_s$，这样的概念被检索，即

$Mec=\{C'\in PO|GA(C,C')\geqslant t_e\}$，其中 $Mec\leqslant Msc$

等价策略是一个更严格的策略，其返回的值是相似策略检索概念的子集，这些概念与 C 有极高的相似度。

继续上面的查询例子，考虑图 2 的查询 B，其指定了概念名字和概念属性，将其转换为 $BOOK$ 的本体论的描述。然后与对等体的本体论 PO_B 的每一个概念进行比较，估计名字和结构的相似度的系数。通常在 *Book* 和 *Volume* 之间会有一个较高的相似度值，因为它们的名字是同义关系，而且它们有两个公共的属性。而在 *Book* 和 *Library* 之间，由于没有公共的属性，其相似度就低。这里设定等价域值 $t_e=0.9$ 和相似域值 $t_s=0.5$ 用于全局相似度概念的选择。按照相似匹配策略，则返回 *Volume* 和 *Journal*。按照等价匹配策略，则只返回 *Volume*。

结束语 由于本体论具有良好的概念层次结构和对逻辑推理的支持，在信息检索，特别是基于知识的检索中得到了广泛的应用。本文利用本体论较好地解决了 P2P 系统中对等体相互协作时的知识共享和积累。基于本文提出的本体论知识管理的框架结构，不仅能在纯 P2P 系统相互协作的对等体间实现知识共享和积累，而且在混合的 P2P 网络结构中，这种基于协作的方法也能用来实现超级节点之间的知识共享。

当然，该框架的有效实施还取决于对等体本体论知识的建立，这涉及到两个方面，一是对等体本体论的建立(这是一个由领域专家不断修改、完善的过程)，另一个是如何用本体论对对等体的资源进行有效的描述。

参 考 文 献

1 Castano S, Antonellis V D, Vimercati S D C D. Global viewing of hetero- geneous data sources[J]. IEEE Transactions on Knowledge and Data Engineering, 2001, 13(2)
2 Castano S, Ferrara A, Montanelli S, Pagani E, Rossi G. Ontology-addressable contents in P2P networks[J]. In: Proc. of WWW'03 1st SemPGRID Workshop, Budapest, Hungary, May 2003
3 刘群，李素建. 基于《知网》的词汇语义相似度计算[J]. Computational Linguistics and Chinese Language Processing, Aug. 2002, 17(2): 59～76
4 Perez A G, Benjamins V R. Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods [J]. In: Stockholm V R, Benjamins B, Chandrasekaran A, eds. Proc. of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5) 1999. 1～15
5 Heflin J, Hendler J. A portrait of the Semantic Web in action[J]. IEEE Intelligent System, 2001, 16: 54～59
6 Berners-Lee T, Hendler J, Lassila O. The Semantic Web [J]. Scientific American, May 2001