

一种用户可扩展消息中间件的设计与实现

魏 巍, 王前震, 蔡瑞英

(南京工业大学 信息科学与工程学院, 江苏 南京 210009)

摘 要: 由于消息中间件不能适应需求变化、功能复杂度增加等问题, 文中提出了消息中间件内核(MOMkernel)的概念, 将消息中间件分成内核部分和用户可扩展部分. 这种用户可扩展消息中间件(UMOM)伸展方便、使用简单, 可适应各种需求. 具体介绍了消息中间件内核的基本组成, 以及用户可扩展消息中间件(UMOM)的设计思想与实现方法, 并且系统已经应用到工程实践中.

关键词: 中间件; 消息中间件; 中间件内核; 用户可扩展消息中间件

中图分类号: TP311

文献标识码: A

文章编号: 1000-7180(2010)12-0098-04

Design and Implement of User Extensible Message—Oriented Middleware

WEI Wei, WANG Qian-zhen, CAI Rui-ying

(College of Information Science and Engineering, Nanjing University of Technology, Nanjing 210009, China)

Abstract: Based on problems of the current Message—Oriented Middleware can not adapt to changes in demand and functional issues such as complexity increases, this paper presented the concept of Message—Oriented Middleware kernel (MOMkernel) and introduced the basic functions of MOMkernel in detail. Finally an User Extensible Message—oriented middleware was designed and implemented. This design has been applied in the engineering project.

Key words: middleware; message—oriented middleware; MOMkernel; UMOM

1 引言

所谓中间件(Middleware)就是一种独立的系统软件或服务程序. 中间件位于客户机服务器的操作系统之上, 管理计算机资源和网络通讯^[1], 是客户方与服务方之间的连接软件, 是基于分布式处理的系统软件^[2]. 另外, 中间件提供的可复用的、构件化的服务, 使开发者在应用开发过程中充分利用前人的经验, 以及一些比较成熟的解决方案, 从而不必从开始构建应用系统^[3].

2 消息中间件技术

2.1 消息中间件

消息中间件(Message—Oriented Middleware, MOM)是中间件中发展较为迅速, 也是比较成熟的一类中间件.

消息中间件提供的异步通信模型更适合在开放协同的软件环境中使用^[4], 从而满足企业内部对异构系统和不同应用的扩展和兼容, 使企业能够以统一的方式实现企业的资源集成. 同时消息中间件具有可扩展性, 能够为企业业务需求的更新变化提供及时的解决方案.

2.2 核心技术

消息中间件以消息的形式收发不同系统中的应用程序数据, 解决不同系统或应用间的通讯问题. 通信的双方以异步、非阻塞、确定的方式收发消息, 不需要实际的物理通路, 因此消息中间件很适合松散耦合系统^[5].

消息传递是消息中间件的关键技术. 消息中间件采用多种机制以保证消息可靠、高效、安全的传输, 如核心的异步机制、多路由的管理、优先级机制等.

2.3 消息中间件的总体架构

消息中间件是用于消息传递的中间件,消息中间件的作用在于,在分布式环境中,在不同的计算机之间采用异步通信模式传递消息,发送消息者不用等待接受者的回应,而接受者也不必在接到消息后立即对发送者的请求进行处理,双方是松散耦合关系,消息之间的逻辑关系和结构由程序本身定义^[6].消息中间件通过提供消息传递和消息排队模型,它可在分布环境下扩展进程间的通信,并支持多种通讯协议、多种语言、多种平台.

消息中间件从总体上讲具有消息管理和消息传送两个部分,应用程序(APP)通过调用消息中间件的应用接口(API)来获得消息中间件提供的服务.消息中间件总体结构^[1]如图1所示.

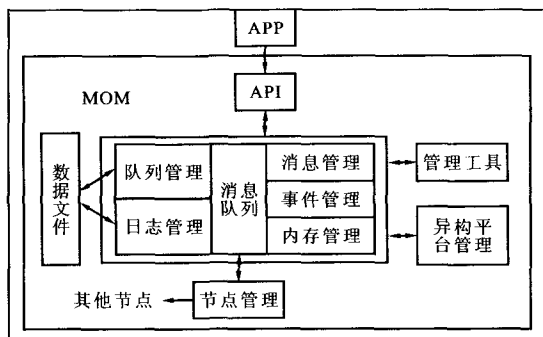


图1 消息中间件的结构

为了满足需求的变化,消息中间件厂商不断为自己的消息中间件添加扩展功能,使得消息中间件产品变得十分臃肿,并且随着功能模块的不断加入,消息中间件的功能修改也变得相当困难.其次,即使消息中间件的功能不断添加,消息中间件也不能适应需求的变化,使得用户不得不修改自己的软件设计来适应消息中间件的要求^[6].另外,随着消息中间件复杂度的增加,一些不必要的服务和消息转存将直接降低消息传递的效率.为此文中提出了消息中间件内核(MOMkernel)概念,这样消息中间件可以分成两部分:消息中间件内核和用户可扩展部分.消息中间件内核完成核心功能,用户可扩展部分完成扩展功能.

2.4 消息中间件内核

消息中间件内核(MOMkernel)完成消息中间件的核心功能,包括消息路由、链路维持、存储管理(含基本的队列管理)、必要的进程/线程管理和信息安全控制,传输的消息是字节流信息.这种字节流信息方便用户能够按照自己的需要进行功能扩展.

(1) 消息路由:路由是消息发送结点和消息接

收结点之间的连通路程,消息路由是消息中间件的核心功能.如果没有消息路由功能就不能称之为消息中间件.

(2) 链路维持:维持整个网络工作结点的连通及连接信息的完整,也是中间件系统的核心功能,链路维持包括结点间的注册、连通维护两大功能.

(3) 内存管理:主要指对消息队列的管理,内存管理就是对软件运行时所需要内存的分配、回收等功能,这里的内存管理主要对象是消息队列内存的分配和回收.

(4) 必要的进程/线程管理:对于一些服务进程/线程应该对用户透明,比如路由结点的工作进程/线程、一些内存(消息队列)管理进程/线程.对于有用户可以扩展和控制的进程/线程应该提供API,而对于链路维持进程/线程应该对用户透明.

(5) 字节消息:消息中间件内核传输消息的API应该基于字节流的,用户可以根据需要构建自己的消息体.

(6) 安全控制:对于消息传输过程中,在保证合法用户正常获取与理解消息的基础上不但要保证消息不被窃听,或窃听者不能了解消息的真实含义,而且要对消息的来源进行判断,能对伪造来源的消息予以鉴别.

3 用户可扩展消息中间件(UMOM)

基于消息中间件内核思想提出了用户可扩展消息中间件,该消息中间件为UMOM(User Extensible MOM),UMOM按照消息中间件内核模式构建,只保留消息中间件的核心功能,提供必要的用户添加扩展功能的调用接口,用户能够方便地在UMOM之上添加扩展功能.

3.1 UMOM的特点

(1) UMOM用户扩展性强

UMOM仅提供消息中间件的核心功能,UMOM的功能包括:消息路由、链路维持、内存管理(含基本的队列管理)、必要进程/线程管理及安全控制.UMOM传送的数据是字节流信息,用户可以根据自身需要对UMOM进行功能扩展.

(2) UMOM能很快适应用户需求的变化

UMOM不但能为用户减少开发工作量,而且能很快满足用户需求的变化.UMOM目标是在保持消息中间件通信代码透明的基础上,强化消息中间件的编程能力.用户可以根据自身需要扩展(UMOM)相应的功能.

(3) 平台移植方便

当对 UMOM 进行平台移植时只需要改写很少的与平台相关的核心功能代码,而无需更改 UMOM 的扩展功能代码和用户应用代码.

(4) 消息传输效率高、配置简单、部署方便

在 UMOM 中只给出了消息中间件的核心功能,而更多的功能由用户扩展,所以减少了不必要的消息周转与存储环节,提高了通信效率. UMOM 系统仅提供一个路由结点和一个构建工作结点的动态链接库,基本不需要复杂的配置,部署方便.

3.2 UMOM 层次结构

基于 UMOM 构建的应用软件系统如图 2 所示. 可以看出应用程序基于 UMOM 构建,而 UMOM 构建在 TCP/IP 等协议之上,这样保持了应用程序的相对独立性,即使底层的通信协议发生变化,也无需变更用户应用程序,另外该 UMOM 能被用户扩展,仅保留消息中间件的核心功能,用户可以根据需要添加 UMOM 的扩展功能.

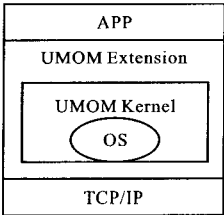


图 2 UMOM 在应用系统中的层次

3.3 UMOM 的总体架构

UMOM 总体上分为两部分:消息中间件内核和用户可扩展部分. 消息中间件内核实现了消息中间件的核心功能,用户可以根据自身需要扩展 UMOM 的功能. UMOM 的总体架构如图 3 所示.

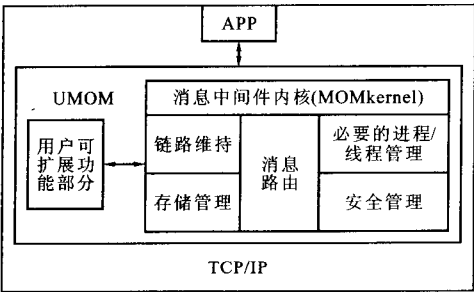


图 3 UMOM 的总体架构

3.4 UMOM 功能模块设计

3.4.1 消息中间件内核设计

消息中间件内核主要由三大部分组成:消息队列、队列(含存储)管理,进程/线程池管理、接口管理,这些基本模块是 UMOM 的核心,而实现又依赖

于具体的操作系统,此软件是在 Windows 操作系统上实现的.

消息队列是在多个不同的应用之间实现相互通信的一种异步传输模式,相互通信的应用可以分布于同一台机器上,也可以分布于相连的网络空间中的任一位置. 消息队列可驻留在内存或磁盘上直到它们被应用程序读走. 消息队列包括这样几类,用户队列、发送队列、死信队列和事件队列. 用户队列是用户自己通过队列管理器创建的,发到应用程序的消息都是先放到用户队列里;发送队列也就是传输队列,是要发送消息的缓冲池;通信管理模块从发送队列取发送的消息;死信队列用来存放发送和接收失败的消息;事件队列存放系统产生的所有的事件,并等待事件处理器来处理. 它的实现原理是:消息的发送者把自己想要发送的信息放入一个容器中(称为 Message),然后把它保存至一个系统公用空间的消息队列中;本地或是异地的消息接收程序再从该队列中取出并进行处理. 消息队列模块(包括相应的消息队列管理器)用于接收外来的服务请求消息,逻辑上每个服务器都有自己的消息队列,但不要求消息队列与相应的服务器进程位于同一台物理机器上.

队列(含存储)管理、进程/线程池管理模块(包括服务线程(进程)组、消息路由例程、线程管理器和性能监视器)完成消息的传递,每个服务线程独立地从消息队列中读取服务请求消息,执行相应的服务操作;消息路由例程被服务线程调用,用于把服务线程生成的消息传递给下级其他服务器作进一步处理;性能检测器检测统计消息队列内消息数量变化情况,估算服务器的负荷;线程管理器则根据性能检测器的统计信息控制服务线程组内各种线程的数目和优先级,实现系统的性能优化.

接口管理模块负责处理来自应用的服务请求,根据应用请求的类型(数据或控制)分别进行不同的处理.

3.4.2 用户可扩展部分设计

提供必要的用户添加扩展功能的调用接口,用户能够方便的在 UMOM 之上添加扩展功能,减少了不必要的消息周转与存储环节,提高了通信效率. 为了用户扩展的需要,工作结点并不以服务程序的形式给出,只提供实现工作结点的动态链接库. 用户基于该动态链接库可以方便地构建工作结点和用户应用程序(如果工作结点以独立的可执行程序给出,此时的工作结点称为工作结点服务程序,如果工作

结点以应用程序的功能模块给出,此时的工作结点称为用户应用程序的服务模块)。

UMOM 的消息路由方式、链路维持方式、结点注册流程、消息传递的方式是相同的,而和用户应用程序相关的进程/线程、消息队列的构建方式是有区别的,这样,即保持了工作结点和路由结点间的工作方式的独立性,又使用户能够在工作结点和应用程序之间进行灵活扩展。

3.5 UMOM 流程设计

(1) 用户注册

首先用户应用程序向工作结点注册,然后工作结点将用户应用程序地址向路由结点注册。用户应用程序可以向路由结点注册,也可以只在本地工作结点注册,对于只在本地工作结点注册的应用程序只能在本机通信。而向路由结点注册的应用程序可以在结点间通信。

(2) 用户数据发送

对于只在本地工作结点注册的应用程序,只能完成本机进程间通信。只有在路由结点注册的应用程序才能在结点间通信,其数据发送流程是应用程序将消息发送到本地工作结点。工作结点的发送数据流程如图4所示。当工作结点接收到用户发送的消息后,首先判断消息接收方是否在本机,如果在本机,将消息发送给本地接收者,否则,根据本地存储的用户应用程序在线信息来判断远程接收方是否在线,如果远程接收方在线,则将消息发送给路由结点,路由结点将此消息转发给消息的接收方,如果远程接收方不在线,则向路由结点询问接收方是否在线,如果接收方在线,则将消息发送给路由结点,路由结点将此消息转发给消息的接收方,否则消息发送失败。

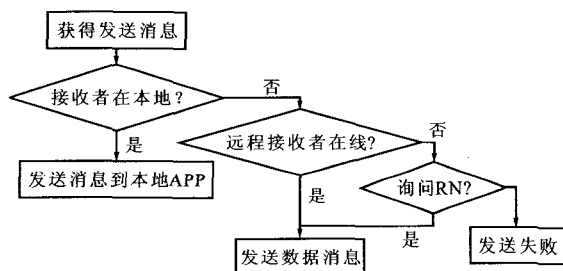


图4 工作节点发送消息流程

(3) 用户数据接收

工作结点接收到路由结点传递过来的消息后,将消息转发给本地的应用程序,实际上,工作结点接

收到路由结点传递过来的消息后,只是对消息进行简单的缓存,剩下的流程由用户扩展。

(4) 链路维持

路由结点周期性的发送维持链路信息,工作结点靠接收到的路由结点发送的数据包来维持链本地的地址表与路由结点地址表的一致。

3.6 UMOM 的实现

按照以上 UMOM 的设计,在 Windows 操作系统上、Microsoft .NET 开发环境下,采用 C#、C++ 语言编码实现 UMOM,编码方式采用面向对象的软件开发方法。在开发过程中,为了选取最优实现技术,借鉴了近年来流行的测试驱动开发(Test Drive Development),在项目进行期间注重测试,根据测试结果及时做出技术上甚至设计上的调整,在迭代与调整中力争获取最优实现方案。

4 结束语

文中提出了一种基于消息中间件内核思想的用户可扩展消息中间件。该消息中间件用户可扩展能力强、操作简单,不但减少了用户的开发工作量,而且能很快满足用户需求的变化,平台移植方便,在 UMOM 中只给出了消息中间件的核心功能,而更多的功能由用户扩展,所以减少了不必要的消息周转与存储环节,提高了通信效率。

参考文献:

- [1] 张志标. 北京: 中间件技术产品应用[M]. 北京: 中国石化出版社, 2002.
- [2] 黄崇本, 吴朝晖. 中间件技术规范与应用[J]. 计算机应用与软件, 2004, 21(7): 24-26.
- [3] 郭胜, 许平, 王颖, 等. 中间件技术的研究[J]. 计算机科学, 2004, 31(2): 155-159.
- [4] 张志伟, 隋品波, 郭长国. 分布对象中间件异步消息的研究与实现[J]. 计算机学报, 2004, 27(12): 1626-1632.
- [5] 齐艳珂, 徐宏炳. 面向 Web 服务的消息中间件模型[J]. 电脑知识与技术: 学术交流, 2007, 11(18): 1491-1492.
- [6] 秦景, 高文, 储方杰. 中间件技术研究[J]. 计算机应用研究, 2003, 11(8): 36-37.

作者简介:

- 魏巍 女, (1985-), 硕士研究生. 研究方向为人工智能.
 王前震 男, (1982-), 硕士研究生. 研究方向为人工智能.
 蔡瑞英 女, (1941-), 教授. 研究方向为人工智能与网络服务.