

RUANJIAN PINGTAI YU  
ZHONGJIANJIAN JISHU

RUANJIAN PINGTAI YU  
ZHONGJIANJIAN JISHU

# 软件平台 **与**

# 中间件技术

▶ 马亨冰 叶东毅 编著



厦门大学出版社

31  
3

# 软件平台与中间件技术

ISBN 7-309-03969-9

马亨冰 叶东毅 编著

厦门大学出版社出版发行

(地址:厦门大学 邮编:361005)

<http://www.xmupress.com>

[xmup@public.xm.fj.cn](mailto:xmup@public.xm.fj.cn)

厦门大学印刷厂印刷

(地址:厦门集美有教路9号 邮编:361021)

2004年8月第1版 2004年8月第1次印刷

厦门大学出版社

定价:18.00元

本书如有印刷质量问题请与厦大印刷厂调换

图书在版编目(CIP)数据

软件平台与中间件技术/马亨冰,叶东毅编著. —厦门:厦门大学出版社,2004. 8

ISBN 7-5615-2246-0

I. 软… II. ①马…②叶… III. 系统软件 IV. TP31

中国版本图书馆 CIP 数据核字(2004)第 074890 号

叶东毅, 1964年出生, 1982年7月本科毕业于南开大学数学系计算数学专业, 1985年4月毕业于福州大学计算机系, 获硕士学位, 1992年毕业于法国里昂大学应用数学与计算机系, 获博士学位。现任福州大学数学与计算机科学学院院长, 教授, 硕士生导师, 并兼任中国数学学会理事, 福建省信息协会副会长, 福建省软件高级职称指导委员会副主任, 福建省软件行业协会理事及专家组组长, 福建省计算机学会理事, 福建省青年联合会常务委员。主持或参与完成了多项国家或省部级科研项目, 科研成果曾获国家(1项)和省部级(2项)科技进步奖, 在国内外发表学术论文50余篇, 主(参)编著作3部。1996年入选“福建省百千万人才工程”, 1999年获“福建青年五四奖章”, 2002年获“第六届福建青年科技奖”, 2003年获“福建省第十届优秀青年科技奖”。

厦门大学出版社出版发行

(地址:厦门大学 邮编:361005)

<http://www.xmupress.com>

[xmup@public.xm.fj.cn](mailto:xmup@public.xm.fj.cn)

集美大学印刷厂印刷

(地址:厦门集美石鼓路9号 邮编:361021)

2004年8月第1版 2004年8月第1次印刷

开本:787×1092 1/16 印张:10.25

字数:262千字 印数:1—3100册

定价:18.00元

本书如有印装质量问题请直接寄承印厂调换

## 作者简介

马亨冰, 1956年出生, 1982年1月毕业于上海工业大学(现上海大学)电子计算机工程系, 获学士学位; 1982年2月起在福建省经济信息中心(原福建省计委电子计算中心)长期从事计算机软件研究与开发工作, 主持或参与研制的十几项计算机科研成果荣获国家级或省、部级科技进步奖或科技成果奖, 在省级及以上级别的期刊和学术交流会上发表论文共38篇, 参与编辑出版《数字福建300问》; 现任福建省经济信息中心副主任、高级工程师, 福州大学数学与计算机科学学院硕士生导师; 并兼任福建省计算机学会理事, 福建省信息协会理事, 福建省系统工程学会理事, 福建省软件行业协会理事及专家组副组长等职。1991年10月获“第一届福建青年科技奖”; 1994年5月获“福建省首届运盛青年科技奖”; 1995年6月首批入选“福建省百千万人才工程”; 1997年8月被福建省委、省政府授予“福建省优秀专家”称号; 1997年10月被国务院评为“享受政府特殊津贴”的专家; 2001年3月被福建省人民政府聘任为“数字福建”专家委员会委员。

## 作者简介

**叶东毅**，1964年出生，1982年7月本科毕业于南开大学数学系计算数学专业，1985年4月毕业于福州大学计算机系，获硕士学位，1992年毕业于法国图卢兹大学应用数学与计算机系，获博士学位。现任福州大学数学与计算机科学学院副院长、教授、硕士生导师，并兼任中国运筹学会理事，福建省信息协会副会长，福建省软件高职教学指导委员会副主任，福建省软件行业协会理事及专家组组长，福建省计算机学会理事，福建省青年联合会常委等职。主持或参与完成了多项国家或省部级科研课题，科研成果曾获国家（1项）和省部级（2项）科技进步奖，在国内外学术刊物上发表论文50余篇，主（参）编著作5部。1996年入选“福建省百千万人才工程”，1999年获“福建青年五四奖章”，2002年获“第六届福建青年科技奖”，2003年获“福建省第十届运盛青年科技奖”。

# 内 容 简 介

中间件的产生只有短短的 10 年时间,但其发展速度却相当惊人,已经成为构建网络分布式异构信息系统不可缺少的关键技术,与操作系统、数据库管理系统并列为基础软件体系的三大支柱。本书共分 8 章,分别阐述了中间件产生的背景,中间件概念、定义及发展情况,中间件的功能、作用、特点、分类、优越性、面临的问题及发展趋势,中间件基本框架、工作原理和实现的关键技术,五大类中间件的工作机理,当前支持服务器端中间件的平台技术,构件技术、XML 技术、Web Service 技术、ERP 技术、网格计算技术与中间件的关系,中间件技术在电子政务、电子商务平台中的应用。

本书可作为高等院校计算机软件、计算机应用专业大学生、研究生的教材,也可供从事计算机软件研究、开发的技术人员和管理人员参考。

# 前 言

从20世纪70年代至今,计算机软件开发技术从面向过程到面向对象,直至面向软件代理(Agent),无论从开发方法还是从开发工具上都发生了翻天覆地的变化,人们的开发模式也已经从简单的单机模式发展到了复杂的分布式大规模软件集成模式。因此,当前计算机信息系统和应用软件系统的有效设计和成功开发需要强有力软件平台和软件开发技术的支撑。中间件技术正是为了适应这一需求而产生的一个新的支撑软件开发技术。在短短10年左右的时间里,中间件技术的发展速度相当惊人,并得到愈来愈广泛的成功应用,已成为构建网络分布式异构信息系统不可缺少的关键技术,与操作系统、数据库管理系统并列为基础软件体系的三大支柱。因而,中间件技术的重要性不言而喻。

尽管中间件技术已经得到相当广泛的应用,有关的技术资料已陆续公开出版,部分高校计算机及相关专业本科生或硕士生已开始将中间件技术作为选修课以培养相关的人才,但是,完整介绍中间件技术及其应用,特别是结合具体的应用实例展开介绍的书籍却比较缺乏,适合于教学的有关教材也鲜有问世,我们在福州大学软件工程硕士专业的教学过程中也经常被这个问题所困扰。为此,我们将平时收集的有关资料、讲义以及在信息化建设过程中的实践和体会进行整理,按照教材的方式编写本书,期望能够起到抛砖引玉的作用。本书共分8章,分别阐述了中间件产生的背景,中间件概念、定义及发展情况,中间件的功能、作用、特点、分类、优越性、面临的问题及发展趋势,中间件基本框架、工作原理和实现的关键技术,五大类中间件的工作机理,当前支持服务器端中间件的平台技术,构件技术、XML技术、Web Service技术、ERP技术、EAI技术、网格计算技术与中间件的关系,中间件技术在电子政务、电子商务平台中的应用。

本书既可作为高等院校计算机软件、计算机应用专业大学生、研究生的教材,也可作为从事计算机软件研究、开发的技术人员和管理人员的参考资料。

承蒙厦门大学出版社的大力支持,使本书得以顺利出版,对此我们表示由衷的感谢。

鉴于作者的水平,书中的错误、遗漏和缺陷在所难免,恳请广大读者批评指正。

马亨冰、叶东毅

2004年7月于福州





2.4.1 国际中间件技术的发展 .....	(19)
2.4.2 中国中间件市场的发展 .....	(21)
<b>第3章 中间件的特性与发展趋势 .....</b>	<b>(24)</b>
3.1 中间件的功能 .....	(24)
3.2 中间件的作用 .....	(25)
3.3 中间件的特点 .....	(25)
3.4 中间件的分类 .....	(26)
3.4.1 中间件的技术分类 .....	(26)
3.4.2 中间件的应用分类 .....	(32)
3.4.3 由底向上的中间件分类 .....	(33)
3.5 中间件的优越性 .....	(33)
3.6 中间件面临的问题 .....	(34)
3.6.1 中间件发展面临的问题 .....	(34)
3.6.2 中间件选用存在的误区 .....	(34)
3.7 中间件技术的发展趋势 .....	(35)
3.8 各类通用中间件的技术与应用走势 .....	(37)
<b>第4章 中间件基本框架、工作原理和实现的关键技术 .....</b>	<b>(43)</b>
4.1 中间件基本框架 .....	(43)
4.2 中间件工作原理 .....	(44)
4.3 基于对象请求代理的中间件模型 .....	(45)
4.4 中间件实现的关键技术和中间件集成框架 .....	(51)
4.4.1 远程过程调用 RPC .....	(51)
4.4.2 发报文和排队 .....	(52)
4.4.3 大规模软件构架中的中间件集成框架 .....	(52)
<b>第5章 五大类中间件的工作机理 .....</b>	<b>(54)</b>
5.1 远程过程调用中间件(RPCM) .....	(54)
5.1.1 远程过程调用概念 .....	(54)
5.1.2 远程过程调用中间件工作机理 .....	(54)
5.1.3 远程过程调用的特点 .....	(55)
5.2 消息中间件(MOM) .....	(56)
5.2.1 消息中间件(MOM)产生 .....	(56)
5.2.2 消息中间件(MOM)的定义 .....	(57)
5.2.3 什么是消息 .....	(57)
5.2.4 消息中间件工作机理 .....	(57)

5.2.5 队列管理器 .....	(58)
5.2.6 触发机制 .....	(59)
5.2.7 接口函数 .....	(60)
5.2.8 消息中间件的功能 .....	(61)
5.2.9 消息中间件的实现 .....	(62)
5.2.10 消息代理中间件 .....	(63)
5.3 数据库访问中间件(DAM) .....	(66)
5.3.1 在 Internet 上实现数据库访问的方式 .....	(66)
5.3.2 数据库访问中的 3 层结构和数据库访问中间件概念的提出 .....	(68)
5.3.3 数据库访问中间件 .....	(70)
5.3.4 数据库访问中间件的类型 .....	(70)
5.3.5 几种常见的数据库访问中间件描述 .....	(71)
5.3.6 数据库访问中间件的优、缺点 .....	(74)
5.3.7 一种数据库访问中间件原型的构造方法 .....	(75)
5.3.8 通用数据库访问中间件的概念模型 .....	(76)
5.4 交易中间件 .....	(77)
5.4.1 事务的概念 .....	(77)
5.4.2 事务管理系统 .....	(77)
5.4.3 交易中间件 .....	(78)
5.4.4 TPMonitor .....	(79)
5.5 J2EE 中间件技术 .....	(79)
5.5.1 J2EE 体系 .....	(79)
5.5.2 J2EE 应用模型 .....	(80)
5.5.3 J2EE 应用服务器中间件的体系结构 .....	(82)
<b>第 6 章 当前支持服务器端中间件的平台技术 .....</b>	<b>(86)</b>
6.1 Microsoft DNA 2000 .....	(86)
6.2 SUN 的 J2EE .....	(87)
6.3 OMG 的 CORBA .....	(88)
6.4 三种技术支持下的分布式构件技术 .....	(90)
6.5 三种分布计算平台技术评述 .....	(91)
6.6 从商业角度分析 J2EE 与 DNA 2000 .....	(92)
<b>第 7 章 构件、XML、Web Service 等技术与中间件的关系 .....</b>	<b>(96)</b>
7.1 构件技术与中间件 .....	(96)
7.1.1 构件技术产生的背景 .....	(96)
7.1.2 构件技术的基本思想 .....	(96)
7.1.3 中间件:构件存在的基础 .....	(97)

7.1.4 构件思想对中间件的作用 .....	(98)
7.2 XML 技术与中间件 .....	(99)
7.2.1 XML 中间件的产生 .....	(99)
7.2.2 中间件与 XML 的关系 .....	(100)
7.3 中间件与 Web Service .....	(101)
7.3.1 何谓 Web Service .....	(101)
7.3.2 Web Service 中使用的消息模型 .....	(101)
7.3.3 Web Service 的标准和互操作性 .....	(102)
7.3.4 典型的 Web Service 工作方式的步骤 .....	(102)
7.3.5 从中间件到 Web Service .....	(103)
7.4 中间件与 ERP .....	(104)
7.4.1 中间件在 ERP 中的作用 .....	(104)
7.4.2 ERP 对中间件的新需求 .....	(105)
7.4.3 携手合作是必然趋势 .....	(106)
7.5 从中间件到 EAI .....	(107)
7.5.1 EAI, 从狭义到广义 .....	(107)
7.5.2 把开发引入集成 .....	(107)
7.5.3 标准——EAI 的基础 .....	(108)
7.5.4 中间件难当重任 .....	(108)
7.5.5 EAI 实现快速实施 .....	(109)
7.5.6 支持业务流程管理 .....	(109)
7.6 网格技术与中间件 .....	(109)
7.6.1 计算网格的概念 .....	(109)
7.6.2 计算网格的特征 .....	(110)
7.6.3 网格的体系结构 .....	(110)
7.6.4 网格与中间件的关系 .....	(113)
<b>第 8 章 中间件技术在电子政务、电子商务平台中的应用 .....</b>	<b>(114)</b>
8.1 中间件技术在电子政务平台中的应用 .....	(114)
8.1.1 概述 .....	(114)
8.1.2 电子政务系统的功能需求 .....	(115)
8.1.3 电子政务系统的技术需求 .....	(118)
8.1.4 电子政务系统中中间件的作用 .....	(119)
8.1.5 中间件与电子政务的关系 .....	(120)
8.1.6 电子政务软件系统采用的关键技术 .....	(120)
8.1.7 电子政务系统平台架构 .....	(121)
8.1.8 电子政务系统的中间件平台 .....	(124)
8.1.9 电子政务的基础组件平台 .....	(127)
8.1.10 电子政务应用系统 .....	(130)

---

8.1.11 网上联合审批中的互联互通应用集成平台 .....	(132)
8.2 中间件技术在电子商务平台中的应用 .....	(136)
8.2.1 电子商务概念 .....	(136)
8.2.2 电子商务特点 .....	(136)
8.2.3 电子商务分类 .....	(137)
8.2.4 电子商务必须面对的四大核心问题 .....	(137)
8.2.5 中间件——实现电子商务的基础软件 .....	(138)
8.2.6 电子商务中间件构架 .....	(139)
8.2.7 电子商务应用服务器 .....	(140)
8.2.8 应用服务器的主流产品 .....	(143)
8.2.9 东方通中间件应用于中国联通电子商务平台 .....	(145)
<b>参考文献</b> .....	(149)

# 第 1 章

## 中间件产生的背景

计算机软件的发展从面向过程到面向对象,直到面向软件代理(Agent),无论从开发方法上,还是从开发工具上都发生了翻天覆地的变化,从而使人们的开发模式从简单的单机模式发展到了复杂的分布式大规模软件集成模式。

中间件,英文名称是 Middleware,这个名词是 IBM 公司于 1994 年提出的。作为一个新兴的软件技术与品种,虽说中间件的产生只有短短 10 年时间,但它的发展速度却相当惊人,已经成为构建网络分布式信息系统不可缺少的关键技术,与操作系统、数据库管理系统并列为基础软件体系的三大支柱。本章将从计算模式的发展过程简要地介绍中间件产生的背景。

### 1.1 计算模式的发展

在计算技术领域,随着微处理器技术和计算机网络技术的不断发展,计算模式经过了以下几次变迁。

#### 1.1.1 集中式计算模式

从 1945 年现代计算机时代开始到 1985 年前后,计算机设备既庞大又昂贵,即使是小型机,每台也价值数万美元。因此,大多数机构也只有有限的几台计算机。为了节省成本,在一个系统中往往以一台主机(Mainframe)为主,连接着若干个终端设备。所有的数据存储和计算都在主机上进行,终端设备只负责为用户发出计算请求和显示计算结果。我们称这种方式为集中式计算模式。

#### 1.1.2 桌面计算模式

随着集成电路技术的不断进步,20 世纪 80 年代中期开始出现了微型计算机。从 8 位、16 位、32 位,到今天 64 位的 CPU 机型,发展非常迅速。许多 PC 机和工作站具备了以前大型计算机的能力,可以存储大量的数据且能进行相对复杂的计算,而价格却非常便宜,可以被一个机构大量采用。计算机也由此脱下了高贵的外衣,走入了寻常百姓之家。因此计算模式的主流从主机转移到了用户桌面。我们称这个阶段的模式为桌面计算模式。

### 1.1.3 分布式计算模式

进入 20 世纪 90 年代,计算机技术最显著的进步之一就是高速计算机网络技术的飞速发展。局域网 LAN 使得同一建筑内的数十甚至上百台计算机连接起来,使大量的信息能够以 100~1 000 Mbps 或更高的速度在计算机间传送。广域网 WAN,尤其是 Internet 的迅速普及,使得全球范围内的数百万台计算机连接起来得以进行信息交换,改变了人们传统的获取和处理信息的方式。随着计算资源的网络化,拥有个人计算机或工作站的广大用户,迫切需要共享分布于网络上的丰富信息资源,以廉价获得超出局部计算机能力的高品质服务,并逐步实现具有计算机支持的协同工作,因此在多个资源上进行分布式处理就变得越来越迫切。从简单的数据共享到多个服务的先进系统,大量的计算转移到了网络环境下的各种资源和个人桌面,这就是分布式计算模式。分布式计算技术成为影响当今计算机技术发展的关键技术。

## 1.2 分布式计算模式的特征

分布式计算(Distributed Computing)技术是在近 20 年来影响计算技术发展的最为活跃的因素之一。其发展经历了两种不同的技术路线:理想路线和现实路线。

### 1.2.1 理想路线(传统意义上的理解)

对于分布式计算或者称分布式系统、分布式应用,不同的学者给出了不同的定义。A. S. Tanenbaum 认为一个分布式系统可以看作是一些独立的计算机集合,但是对这个系统的用户来说,系统就像一台计算机一样。这个定义有两个方面的含义:第一,从硬件角度来讲,每台计算机都是自主的;第二,从软件角度来讲,用户将整个系统看作是一台计算机。Carl L. Hall 则将分布式计算定义为通过多个独立的计算机处理来完成一个特定的任务,每个处理可以在相同或不同的计算机平台上以并行或串行的方式进行,通过通信协议相互协作完成任务,从而实现了把计算负担分散到多个能通信的计算机上。对于分布式应用系统,P. H. Enslow 作了如下描述:包含许多物理资源和逻辑资源;通过网络通信实现信息交换;有一个高层的操作系统能够对整个系统进行管理;系统对用户透明;系统中各部分资源既相互独立又相互配合。

从上面的定义不难看出,系统中计算机的互连和各部分在网络中的分布仅仅是分布式计算的必要条件,分布式系统的统一的逻辑特性才是其充分条件。

其主要特征有以下几点:

(1)透明性 就是让用户将一些机器集合的协同工作看作是在一台机器上完成的。透明性包括:

- 位置透明 即用户不需知道软、硬件资源(如 CPU、文件和数据库)的位置,资源的名字不应含有资源的位置信息;

- 迁移透明 即资源无需更名就可自由地在系统中流动,外界不需要知道系统为使资源均衡而改变对象的位置;

- 复制透明 即系统可以随意地为文件和其他资源进行附加拷贝而无需用户知道;

- 并发透明 即多个用户可以自动共享资源;

- 并行透明 即系统活动可以在用户没有感觉的情况下并行发生；
- 存取透明 即隐藏数据表示和调用机制的异同；
- 失败透明 即将出错和恢复事件隐藏在对象内部,以达到容错的目的；
- 持久性透明 即对象里隐藏着所用资源的变化,如处理器资源、存储资源的冻结与解冻；
- 重定位透明 即改变一个接口的位置不影响与之编联的其他接口；
- 提交透明 即一组对象发生作用的次序不影响结果的一致性。

(2) 灵活性 可以根据不同的情况,用最有效的方式将工作负荷分配到可用的机器上,最大限度地合理利用资源。

(3) 可靠性 即系统可以屏蔽错误。通过把工作负载分散到众多的机器上,单个芯片的故障最多只会使一台机器停机,而其他机器不会受任何影响。理想条件下,某一时刻如果有5%的计算机出现故障,系统仍将能继续工作,只不过损失5%的性能。对于某些关键性应用,如证券交易或核反应堆控制系统,采用分布式系统可以保证其高可靠性。

(4) 可伸缩性 系统可在需求增长的时候,通过增加资源,对系统能力进行灵活地扩充。

由此可见,理想路线试图在互连的计算机硬件上部署全新的分布式操作系统,全面管理系统中各自独立的计算机,呈现给用户单一的系统视图。在20世纪80年代,学术界普遍追求这一目标,尽管产生了许多技术成果和实验系统,但却没有被用户和市场接受。这些技术总称为分布式计算机系统技术。对于分布式计算机系统的定义一直不太明确的一个主要原因是,不知采用什么样的模型来构造、管理和协调使用基于网络(包括局域网、城域网和广域网)中的各个计算机系统。但是,对于分布式计算机系统,当时追求的一个共同的特点是其含有多个处理单元,每个处理单元都既能从事自己的活动,又能协同处理一个大任务。

### 1.2.2 现实路线( Internet/Intranet 下的理解)

随着人们对 Internet/Intranet 的进一步使用,人们的生活方式和获取信息的方式发生了根本性的变化。在分布式软件的构架和研究方面,那种一味地追求逻辑上完全统一的研究方式已经失败。经验告诉我们,统一协议下的松散式协同工作模式更容易满足人们的需求。目前的分布式研究,不再追求用严格的、逻辑上完全统一的系统对分布在各个角落的计算机进行管理,而是用像 CORBA 或 DCOM 或 CCM(CORBA Component Model) 等这样的分布式构件模型,结合软件 Agent 建立基于网络的全方位分布式框架和环境的构架。

由此可见,现实路线是在网络计算机平台上部署开放分布式计算环境(也称中间件),提供开发工具和公共服务,支持分布式应用,实现资源共享和协同工作。在20世纪90年代,工业界普遍追求这一技术路线,产生了一系列行之有效的技术和广为用户接受的产品。当前,人们所说的分布式计算技术通常是指在网络计算机平台上开发、部署、管理和维护以资源共享和协同工作为主要应用目标的分布式应用技术。也称为开放分布式技术。

无论如何,开放分布式技术像分布式技术一样也在不断地发展之中,要给出一个确切的定义是不可能的,但概括起来是指在独立的计算机集合系统中,通过网络通信来开发、部署、管理和维护以资源共享和协同工作为主要应用目标的分布式开发环境。相对于集中式计算模式,开放分布式计算模式在性能价格比、计算能力、可靠性、伸缩性和解决问题固有的分布性上占有明显的优势;而对于桌面计算,分布式计算在数据共享、设备共享、通信、灵活性等方面则显示了无可比拟的优势。当然,事物总是一分为二的,开放分布式计算技术也面临着一些急需

解决的问题,如没有一个权威统一的标准、缺乏开发分布式系统的支撑软件、网络负载饱和和引起的问题、数据安全等问题。

## 1.3 开放分布式计算模型

### 1.3.1 客户机/服务器(C/S)计算模型

进入 20 世纪 90 年代,开放分布式计算的实现主要依赖于经典的客户机/服务器计算模型。它将分布式应用分解为客户机和服务器两大部分,客户机首先发出请求,服务器在接到客户机的请求后提供服务。两层 C/S 结构(Client/Server)的计算模型如图 1-1 所示。

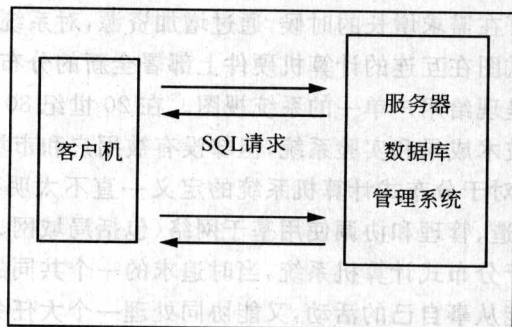


图 1-1 两层 C/S 结构的计算模型

这种方式不同于主机的计算模型在于:充分利用了客户端计算机的计算能力,每一个客户机都是一个独立的计算单元,有自己的处理器和存储器,负责处理应用系统的显示逻辑,如图形用户界面、信息预处理等,而把复杂的计算,如业务逻辑、数据处理,交给了服务器。通过平衡客户机和服务器的负载以实现分布式资源和信息的共享。目前 Internet 上最流行的 Web 应用就是一种开放分布式的客户机/服务器结构。

### 1.3.2 B/S 与 C/S 的关系

两层 C/S 结构的计算模型,在 20 世纪 80 年代及 90 年代初得到了大量应用,最直接的原因是可视化开发工具的推广。之后,它开始向三层(多层)结构发展,图 1-2 是三层 C/S 结构的计算模型。

近来,随着网络技术不断发展,尤其是基于 Web 的信息发布和检索技术、Java 计算技术以及网络分布式对象技术的飞速发展,导致了很多应用系统的体系结构从 C/S 结构向更加灵活的多级分布结构演变,使得软件系统的网络体系结构跨入一个新阶段,即 B/S 体系结构(Browser/Server 的简称,浏览器/服务器模式)。基于 Web 应用的 B/S 模式其实也是一种客户机/服务器模式,只不过它的客户端是浏览器。为了区别于传统的 C/S 模式,才特意将其称为 B/S(浏



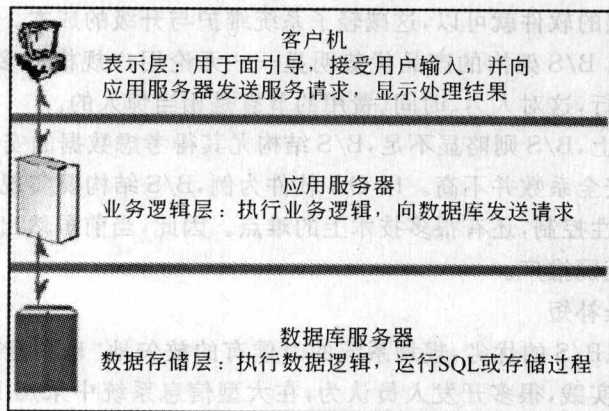


图 1-2 三层 C/S 结构的计算模型

览器/服务器)模式。认识到这些结构的特征,对于系统的选型而言是很关键的。

### (1)系统的性能

在系统的性能方面,B/S 占有优势的是其灵活性。任何时间、任何地点、任何系统,只要可以使用浏览器上网,就可以使用 B/S 系统的终端。近年来,智能手机、智能家电、PDA 等传统电脑之外的上网方式发展迅速,这在一定程度上为 B/S 结构增加了砝码。

不过,采用 B/S 结构,客户端只能完成浏览、查询、数据输入等简单功能,绝大部分工作由服务器承担,这使得服务器的负担很重。采用 C/S 结构时,客户端和服务器端都能够处理任务,这虽然对客户机的要求较高,但因此可以减轻服务器的压力。而且,由于客户端使用浏览器,使得网上发布的信息必须是以 HTML 格式为主,其他格式文件多半是以附件的形式存放。而 HTML 格式文件(也就是 Web 页面)不便于编辑修改,给文件管理带来了许多不便。

### (2)系统的开发

最新的 C/S 结构是建立在现在被称为中间件的产品基础之上的,严格来讲这些产品还缺乏作为企业级应用平台的一些特性,难以扩展到互联网这样的环境上去,而且要求应用开发者自己去处理事务管理、消息队列、数据的复制和同步、通信安全等系统级的问题。这对应用开发者提出了较高的要求,而且迫使应用开发者投入很多精力来解决应用程序以外的问题。这使得应用程序的维护、移植和互操作变得复杂,成了 C/S 的一大缺陷。如果客户端是在不同的操作系统上(比如 Windows 2000 以及不同版本的 Unix),C/S 结构的软件需要开发不同版本的客户端软件。如果产品经常需要更新换代,那么升级系统时候需要付出的高代价以及工作的低效率会在一定程度上制约企业的应用。

但是,与 B/S 结构相比,C/S 技术发展历史更为“悠久”。从技术成熟度及软件设计、开发人员的掌握水平来看,C/S 技术应是更成熟、更可靠的。和佳软件总裁陈佳认为:对于 ERP 软件而言,采用 100%的 B/S 方式将造成系统响应速度慢、服务器开销大、通信带宽要求高、安全性差、总投资增加等问题。而且,对于复杂的应用,B/S 方式目前尚没有合适方式进行开发。

### (3)系统的升级维护

C/S 系统的各部分模块中有一部分改变,就要关联到其他模块的变动,使系统升级成本比较大。B/S 与 C/S 处理模式相比,则大大简化了客户端,只要客户端机器能上网就可以。对于 B/S 而言,开发、维护等几乎所有工作也都集中在服务器端,当企业对网络应用进行升级