



客户/服务器系统 (Client/Server)

技术基础

[美] Carl L. Hall 著

邓召义 张云华 陈是力 译

沈翔 校



电子工业出版社

Publishing House of Electronic Industry

客户/服务器系统技术基础

〔美〕 Carl L. Hall 著

邓召义 张云华

陈是力 译

沈翔 审校

电子工业出版社

(京)新登字 055 号

内容提要

客户/服务器技术是 90 年代新发展起来的计算技术。随着人们对提高灵活性、计算能力和雇员工作能力要求的认识,客户/服务器技术及其在分布式和协作式计算领域的扩展将逐渐成为未来发展的技术基础。

本书为只有计算机基础知识并希望了解这种流行的新计算方法和基本原理的读者提供了了解和学习客户/服务器技术的必需资料。本书由浅入深、循序渐进地介绍了当前的客户/服务器技术,共包括了三个方面的客户/服务器系统范例的内容,即文件服务器、数据库服务器和由新的操作系统和事务处理系统支持的增强型客户/服务器的功能。同时,还介绍了这门新技术在分布式和协作式计算领域的扩展及其这些系统在成功的现代企业中的应用。

本书是为从事信息系统工作的读者而编写的,只要读者具备计算机的入门知识,便可轻松掌握本书介绍的新技术。本书对各企业的主管信息官员、信息系统管理员、应用程序员和系统程序员尤其有参考价值。

本书英文版书名为 Technical Foundations of Client/Server Systems,由美国 John Wiley & Sons, Inc. 出版。该书的中文版版权于 1994 年 12 月由原出版公司授权电子工业出版社,未经允许,任何人不得以任何手段复制或抄袭本书的任何内容。

Copyright © 1994 by John Wiley & Sons, Inc.

Chinese Copyright © 1995 by Publishing House of Electronic Industry, PRC

客户/服务器系统技术基础

[美] Carl L. Hall 著

邓召义 张云华

陈是力 译

沈翔 审校

责任编辑:王世忠

*

电子工业出版社出版(北京市万寿路)

电子工业出版社发行 各地新华书店经销

北京志达排版公司排版

北京科技大学印刷厂印刷

*

开本:787×1092 毫米 1/16 印张:8.75 字数:199 千字

1995 年 4 月第一版 1995 年 4 月第一次印刷

印数:4000 册 定价:16.50 元

ISBN 7-5053-2977-4/TP·1024

前 言

客户/服务器(Client/Sever)技术是90年代的新技术。随着人们对提高灵活性、计算能力、雇员工作能力要求的认识,客户/服务器技术及其在分布式和协作式计算领域的扩展将逐渐成为未来发展的技术基础,这一点已为人类所接受。

正如书名表示的那样,使用本书可以在客户/服务器技术知识方面打下良好的基础。为能给企业提供合适功能的客户/服务器平台,开发规范已不再是一项简单的工作。本书为用户能提出正确问题和评价所得答复提供了必需的资料。从某种意义上说,本书为用户能揭穿销售商的虚假广告提供了有力的武器,因此用户要好好使用本书来保护自己,避免上当受骗。

任何只有计算机基础知识并希望了解这种流行的新计算方法和基本原理的用户都应该阅读本书,它对主管信息官员、信息系统管理员、应用程序员和系统程序员尤其有用。

网络管理员和建筑师也会发现本书介绍的技术,对他们90年代及其以后的生活会产生有力的影响。数据库管理员和知识工程师也应该知道本书介绍的信息,以便在将来需要时使用这门新技术。

尽管客户/服务器技术得到普遍承认,但是还有许多IS技术人员并没有完全掌握它。贸易出版商和当前的书籍对这门新技术产生了一个错误的导向,甚至有一些人认为安装了数据库服务器以后就到了客户/服务器系统的终点。通过对增强型客户/服务器功能的介绍,我们将给出一些让许多人都吃惊的功能和能力。

小公司的老板和大公司的高级顾问都必须掌握这门新的技术。然后,还要发现一批适合于这种技术的人员来把本技术的魅力应用于公司的业务中从而为公司创造更多的收益。

本书介绍了当前的客户/服务器技术,它包括三个方面的客户/服务器系统范例的内容:即文件服务器、数据库服务器和由新的操作系统和事务处理系统支持的增强型客户/服务器的功能。本书还介绍了这门新技术在分布式和协作式计算领域的扩展及其这些系统在成功的现代企业中的应用。

因为出现低的性能和比预料更高的成本,许多人对客户/服务器这种工具产生了失望。本书在数据库服务器和增强型客户/服务器的章节中为用户解决上述问题提供了资料,也为计划购买这种激动人心的新技术工具的企业提供了预防问题出现的方法。

本书是为从事于信息系统的用户而写的,不具备编程或更高级技术知识的用户也能看懂本书中的大部分内容,某些章节介绍了一些较深的内容,它要求用户具备C语言的知识,但是只具备计算机入门知识的用户也能基本看懂这些内容。

本书的目的是定义和解释客户/服务器的基本原理,其中更详细地介绍了某些方面的内容,以便于评估和选择客户/服务器产品时使用。对于某些用户,书中的细节似乎是参差不齐;对另一些用户,更详细的内容会是非常有用的,这些用户应该参阅本作者的其他专著以便获得更多的信息。

客户/服务器技术在每章以概述开始,由浅入深地介绍了文件服务器到增强型客户/服

务器等方面的内容。

对于计划在多服务器中使用分布数据系统的任何用户来说,关于增强型客户/服务器技术的章节是最重要的部分。因为提供足够网络容量(域宽)来满足数据库的调用需要所需的花费是非常高的,所以在宽域网(WAN)中使用增强型客户/服务器几乎是最经济的方法。如果在系统中有多种数据库(异型数据库)并且(或者)数据表是分布式的,那么就绝对要使用增强型客户/服务器系统。

增强型客户/服务器处理的讨论使用了 UNIX System Laboratories 的 TUXEDO System/T 作为技术实例,包括使用 System/T 来解释技术的最有效使用过程。

一个完整的客户/服务器系统必须能管理已存信息和解释并处理信息的应用程序。在任何复杂的或大型的系统中,只有完全开发的事务处理系统才能提供全部要求的功能,在《事务处理:概念与方法》一书中,作者 Jim Gray 和 Andreas Reuter 说:“总之,没有事务,分布式系统就不可能应用于现实生活中”。

在 Oracle Integrator (第 3 卷,第 6 期,1992. 12)中的文章:“90 年代的分布式计算体系结构”中讲叙了如下一段内容:

在单个表格分布在多个系统的环境中,理想的做法是把进入任何系统的插页放至基于数据值系统的要插入行处。

用于支持体系结构的软件产品则开始在市场上出现。一个很好的例子就是支持 X/Open's XA 接口的软件,一个在 Oracle 7 中常被忽略的特性。XA 接口软件是一种工业标准方法,用于把两阶段确认数据库管理程序(例如 Oracle 7)与分布式事务处理(DTP)管理程序(例如 Transarc's Encina, UNIX International's TUXEDO 和 NCR'S TOPend)集成化。DTP 管理程序能互联多个异型硬件平台、操作系统和当前在 500 个公司及联邦机构中存在的数据库。

因为事务处理是成功进行客户/服务器操作的关键,所以有必要特别强调和解释一下事务处理的过程及其如此重要的原因。

本书中使用系统举例的目的是用来解释使用已讲原理设计和开发系统的方法。因为篇幅限制,并没有完整地介绍举例,只希望这些举例能足以帮助用户使用本书介绍的原理。

网络计算的介绍不够全面,而是非常简单的介绍(例如,没有包括网络协议数据的装入),原因是它们仅用于解释有关的一些观点。在实际情况中,网络计算必须包括对所有协议数据,某些网络上的通讯冲突及其他网络开销项目的综合研究。

在本书及其举例中更详细地介绍了一些产品的性能,但这并不意味着对这些产品的保证,仅用于代表这类产品的风格。本书介绍的产品都是很好的,在其功能的范围内可得到最有效的使用。而且在此声明:在本书中没有提到的产品也不一定质量差,本书的目的仅为用户提供一些基本性质的信息,并引用一些产品作为实例。本书不是一本产品目录,因此许多好的产品并没有在此提到。

开放式系统正在变得流行起来,特别是在支持客户/服务器方面更是如此。在此没有直接介绍开放式系统,但用户将会遇到许多讨论都是基于开放式系统原理的。计划安装客户/服务器的用户应该仔细研究使用符合开放式系统标准(例如 POSIX 和 X/Open 规范)的系统。

UNIX 常常被认为是开放系统的同义词。UNIX 是已经开发和将要开发操作系统的基

础,但 UNIX 不是唯一可用的开放系统。甚至大部分专用的系统也有很高的开放性。使 UNIX 及其派生体真正开放的是它们几乎可以在任何硬件上使用。

为一种 UNIX 平台开发的应用程序能很容易地移到任何其他 UNIX 平台上(注意:接口不是完全通用的,几乎总要做些修改!)

UNIX 型系统的另一个优点是大量系统工具可供使用。长时间使用 UNIX 的用户,改用一种旧的专用系统后不久就会怀念起 UNIX 的各种工具。这种情况还在不断出现。

将来,会出现更开放的系统。象 POSIX 一样的现存标准和象来自 X/Open 的 PG3 等规范为销售商开发新的系统提供了基础,符合上述标准的新系统内部是完全不同的,但它们保持开放以供使用。

本书附录列出了其他一些有关主题,例如标准、两阶段确认和词汇表等。尽管这些主题是很有用的,但在掌握本书内容的同时,不需要立即阅读它们。

商标说明

OS/2、RS/6000、IMS、IMS/DC 是 IBM 公司的注册商标。

ORACLE、ORACLE7、SQL * Net、SQL * Forms、Pro * C、PL/SQL 是 Oracle 公司的注册商标。

Microsoft、MS、MS-DOS、Windows 是 Microsoft 公司的注册商标。

TUXEDO 和 UNIX 是 UNIX System Laboratories 公司的注册商标。

Encina 是 Transarc 公司的注册商标。

NCR、TOPEND 是 National Cash Register 公司的注册商标。

TCP/IP 是 Defense Advanced Research Projects 公司的注册商标。

在本书中使用的其他一部分公司或产品名称都是相应公司的注册商标,本书仅用于信息介绍。

目 录

前言

第一章 绪论	(1)
1.1 概述	(1)
1.2 客户/服务器的一些基本定义.....	(2)
1.2.1 分布式计算	(2)
1.2.2 分布式处理	(3)
1.2.3 分布式数据库	(4)
1.2.4 客户/服务器的基本定义.....	(4)
1.2.5 硬件服务器	(5)
1.2.6 软件服务器	(6)
1.3 事务的完整性	(7)
1.4 服务器类型	(7)
1.4.1 文件服务器	(7)
1.4.2 数据库服务器	(9)
1.4.3 增强型客户/服务器(C/S)处理	(9)
1.5 RPC、同级式和对话式.....	(10)
第二章 客户/服务器的计划	(13)
2.1 基本的研究	(13)
2.2 使用客户/服务器的原因.....	(13)
2.3 系统应该提供什么	(14)
2.4 避免潜在的损失	(15)
2.4.1 网络概述	(15)
2.4.2 不能保持事务的完整性	(15)
2.4.3 安全性丢失	(16)
2.4.4 错误的平台规格	(16)
2.4.5 缺少跨越平台的能力	(17)
2.4.6 低的性能	(17)
第三章 网络和平台的基本信息	(19)
3.1 某些术语	(19)
3.2 MS-DOS 服务器	(19)
3.2.1 典型的硬件	(19)

3.2.2	操作系统	(20)
3.3	UNIX 服务器	(21)
3.3.1	典型的服务器	(21)
3.3.2	操作系统	(22)
第四章	文件服务器	(23)
4.1	典型的硬件	(23)
4.2	网络要求	(23)
4.2.1	概述	(23)
4.2.2	数据的性质	(24)
4.2.3	同时使用的用户数量	(24)
4.2.4	工作站类型	(24)
4.2.5	访问频率	(24)
4.2.6	网络特性	(24)
4.3	MS-DOS 文件服务器如何工作	(25)
4.4	UNIX 文件服务器如何工作	(26)
4.5	远程过程调用	(28)
4.5.1	远程过程调用的定义	(28)
4.5.2	RPC 如何工作	(28)
4.6	应用程序设计研究	(30)
4.6.1	性能研究	(30)
4.6.2	事务的完整性	(32)
4.7	安全性	(33)
4.8	文件服务器的优点	(33)
4.9	文件服务器的缺点	(33)
第五章	数据库服务器	(35)
5.1	数据库服务器如何工作	(35)
5.1.1	应用程序如何使用数据库	(35)
5.1.2	远程数据库接口	(36)
5.1.3	数据库服务器	(36)
5.1.4	单线服务器	(37)
5.1.5	多线专用服务器	(38)
5.1.6	Oracle 7 方法	(38)
5.1.7	并行服务器	(39)
5.1.8	Sybase 开放式服务器	(41)
5.2	分布式数据库技术	(41)
5.2.1	分布式数据库的定义	(41)
5.2.2	分布式数据库的功能	(42)

5.2.3	对分布式数据库的访问	(42)
5.2.4	连接	(42)
5.2.5	分布式数据字典	(43)
5.3	设计应用程序时应做的考虑	(44)
5.3.1	单个服务器的考虑	(44)
5.3.2	多服务器和分布式数据库	(45)
5.3.3	应用程序的驻留	(46)
5.3.4	网络装载	(46)
5.3.5	开发工具	(47)
5.3.6	备用过程	(47)
5.3.7	Oracle 的备用过程	(47)
5.4	可用的开发工具和最终用户工具	(48)
5.4.1	应用开发工具	(48)
5.4.2	互操作性工具	(49)
5.5	数据库的容量	(49)
5.6	客户器/服务器环境下的数据库管理	(50)
5.6.1	数据库和表的位置	(50)
5.6.2	定义数据库访问和安全性	(50)
5.6.3	应用程序和工具的物理地址	(50)
5.7	数据库服务器的优点	(51)
5.8	数据库服务器的缺点	(51)
第六章	数据库服务器应用举例	(53)
6.1	数据库	(53)
6.1.1	数据需求	(53)
6.1.2	数据库定义	(53)
6.1.3	系统设计的考虑	(55)
6.1.4	系统设计	(55)
6.2	系统是如何工作的	(57)
第七章	增强型客户/服务器处理	(59)
7.1	增强型客户/服务器基础	(59)
7.1.1	增强型客户/服务器特点	(59)
7.1.2	OLTP 和增强型客户/服务器	(59)
7.1.3	典型平台	(60)
7.1.4	增强型客户/服务器模型	(61)
7.1.5	现有产品	(61)
7.2	TUXEDO 概略	(62)
7.2.1	TUXEDO 的特点	(62)

7.2.2	TUXEDO 如何工作	(63)
7.2.3	客户、服务、服务器和服务器组	(64)
7.2.4	告示方式和对话方式	(65)
7.3	TUXEDO 的设计原则	(66)
7.3.1	客户程序	(66)
7.3.2	服务器及其服务	(67)
7.3.3	服务器组	(68)
7.3.4	TUXEDO 缓冲器	(69)
7.3.5	异型数据库	(70)
7.3.6	使用并行计算机	(70)
7.4	TUXEDO 和数据库	(71)
7.4.1	事务	(71)
7.4.2	设计防止不必要的程序修改	(72)
7.4.3	使用数据完整性特点	(72)
7.4.4	连接	(72)
7.4.5	数据库的分布式特点	(73)
7.4.6	数据复制	(73)
7.5	TUXEDO 和工作站	(74)
7.5.1	TUXEDO 工作站产品	(74)
7.5.2	UNIX 工作站	(75)
7.5.3	PC 工作站	(75)
7.6	TUXEDO 系统管理概述	(75)
7.6.1	系统管理工具	(76)
7.6.2	系统管理例子	(77)
7.7	在大型机上使用 TUXEDO	(77)
7.8	增强型客户/服务器系统的优点	(78)
7.9	增强型客户/服务器系统的缺点	(78)
第八章	TUXEDO 应用举例	(79)
8.1	数据库	(79)
8.1.1	数据需求	(79)
8.1.2	数据库定义	(79)
8.1.3	表格分割	(81)
8.1.4	数据复制	(81)
8.1.5	订货历史	(81)
8.2	系统设计	(81)
8.2.1	系统设计考虑	(81)
8.2.2	设计	(81)
8.2.3	WAN 上的装载	(82)

8.2.4	系统如何运行	(83)
8.2.5	实现语言	(83)
8.3	系统注解	(84)
第九章	遗留系统的帮助	(85)
9.1	使用遗留数据	(85)
9.2	COBOL	(86)
9.3	CICS	(86)
9.4	CICS/6000	(87)
第十章	客户/服务器环境的安全性	(89)
10.1	概述	(89)
10.2	特别安全考虑	(89)
10.2.1	一般讨论	(89)
10.2.2	网络安全要求	(89)
10.3	客户/服务器安全要求	(90)
10.4	如何看待	(91)
10.4.1	用户如何看待安全性	(91)
10.4.2	管理员如何看待安全性	(91)
10.4.3	应用程序员如何看待安全性	(91)
10.4.4	在程序中检查安全性	(91)
10.5	如何使用	(92)
10.5.1	证明	(92)
10.5.2	授权	(93)
10.6	需要探讨的一些问题	(94)
10.7	关于口令的注解	(94)
第十一章	要求的技巧	(97)
11.1	历史	(97)
11.2	任务结构	(99)
11.3	可取之处	(99)
11.4	资格和职位	(100)
11.5	关于职位层次的注解	(101)
11.6	为了运用客户/服务器人们应该如何准备	(102)
第十二章	混合标题	(105)
12.1	选择客户/服务器、4GL 或面向对象的工具	(105)
12.2	关于在哪里进行处理的更多讨论	(107)

附录 A 订单输入应用的要求	(109)
A.1 概述	(109)
A.1.1 公司	(109)
A.1.2 当前使用的订单系统	(109)
A.1.3 公司需要什么	(109)
A.2 特殊需要	(110)
A.2.1 用户需要什么	(110)
A.2.2 系统的容量	(110)
A.2.3 性能	(111)
A.2.4 数据完整性	(111)
A.2.5 技术需要	(111)
A.2.6 价格要求	(112)
A.2.7 开发计划	(112)
附录 B 为订单输入服务的网络	(113)
B.1 网络类型	(113)
B.1.1 局域网(LAN)	(113)
B.1.2 宽域网(WAN)	(113)
B.1.3 协议	(113)
B.2 网络结构	(113)
B.2.1 概述	(113)
B.2.2 分公司网络	(114)
B.2.3 总公司网络	(114)
B.3 网络带宽(BANDWIDTH)	(116)
附录 C 开放式系统标准的概述	(117)
C.1 国际标准化组织(ISO)的标准	(117)
C.1.1 ISO 开放式系统互连(OSI)参考模型	(117)
C.1.2 ISO 分布式事务处理	(117)
C.1.3 ISO 两阶段确认	(117)
C.2 IEEE-POSIX	(118)
C.3 X/OPEN 分布式事务处理规范	(118)
C.4 开放式软件协会	(118)
C.5 国际 UNIX 组织	(118)
附录 D 关于两阶段确认的简短讨论	(121)
词汇表	(125)

第一章 绪 论

1.1 概述

客户/服务器技术是强大的计算机硬件与可靠的、快速的和低成本通讯技术相结合的范例,它涉及到象分布式计算、协作式计算和分布式事务处理等一些概念。客户/服务器系统把计算从单一的集中管理系统中移动到每个用户都可以快速、经济地访问信息的分布式系统中。

图 1.1 所示的单一系统是主机的逻辑代表。

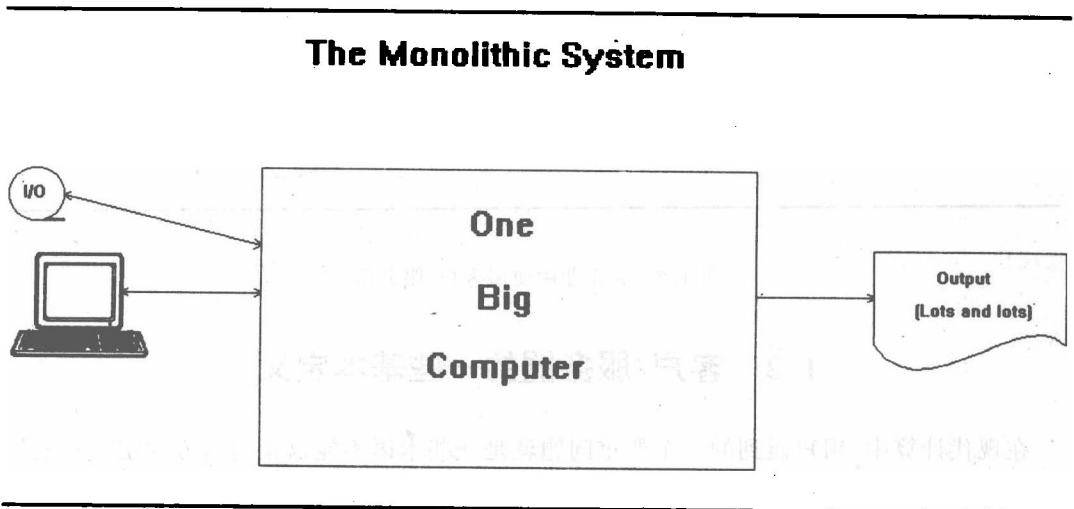


图 1.1 单一系统

图中“一个大的计算机”实际上是通过硬件和软件由几台中央处理器组成的逻辑单元。

图 1.2 说明了如何使用客户/服务器技术以提高用户的工作能力。每个分公司有一个带有必需数据的本地计算机,而总公司在一大组服务器上保存集中的文件。一个良好设计的 C/S(Client/Server)系统可高性能地支持几千个用户。

由于下述技术使客户/服务器技术流行起来:

- 允许使用低成本的硬件。
- 支持平台跨越能力。
- 较好地支持容错系统。
- 为分布式数据提供了更容易的管理方法。
- 支持使用户界面更好更快的图形用户界面前端。

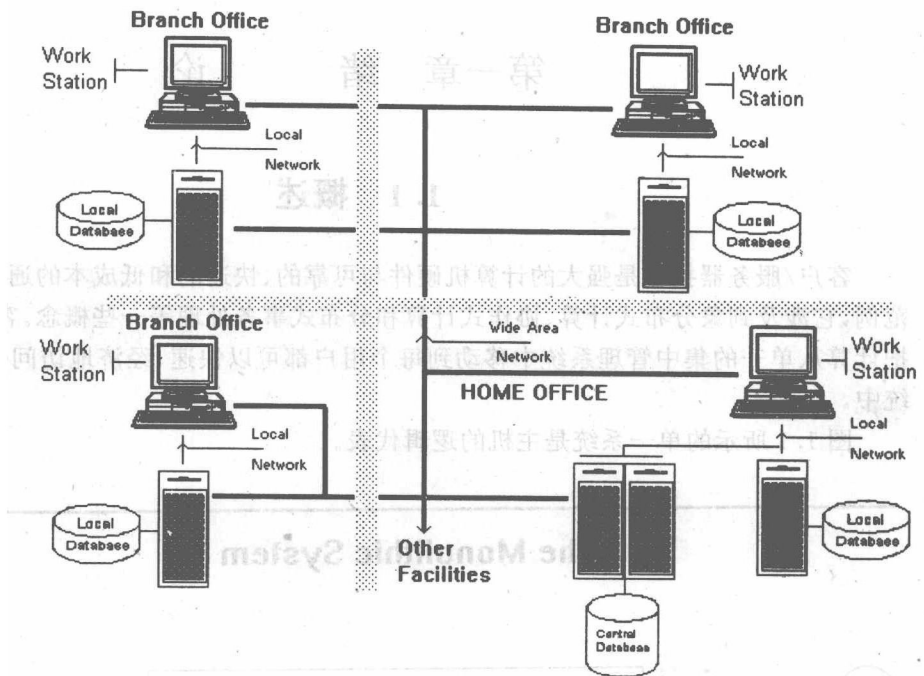


图 1.2 在企业中使用客户/服务器

1.2 客户/服务器的一些基本定义

在现代计算中,用户碰到的一个严重问题就是一些术语不能以相同的方式定义和使用,不仅仅在使用中有多种含义,而且用户根本就没有意识到这种不正确的术语,从而在一开始就导致混淆。在客户/服务器技术中使用含糊不清的术语甚至会更糟糕。即便是术语“Client/Server”(客户/服务器)也有几种意思。术语“Distributed computing”(“分布式计算”),“Cooperative computing”(“协作式计算”)和“Distributed transaction processing”(“分布式事务处理”)的使用也不是很精确。

本节精确地定义了上述术语和其他术语,附加的定义请见附录中的词汇表。

1.2.1 分布式计算

术语“分布式计算”的含义常被误认为是一种假定的含义,至今它对许多不同的人有不同的含义。通常定义它为下述含义:即使用多个独立的计算机处理来完成一个特定的任务。使用的处理可以在相同的或不同的计算平台上。这个定义不是很严密的,因为它允许包括多步批处理作为分布式计算系统,从直觉上说这也是不正确的。以批处理思维的人是用“job step”(任务级)代替了“task”(任务),而结果将偏离了正确的定义。

以上述方式定义的“分布式计算”允许包括许多计算范例,例如 C/S(客户/服务器)、分

布式处理和分布式数据库。本书将使用更精确的术语,以便区分分布式处理和分布式数据库。

1.2.2 分布式处理

M. Tamer Ozsü 和 Patrick Valduriez 认为,术语分布式处理(或分布式计算)…可能是近几年来计算机科学中最滥用的术语…,这种滥用已达到了如此程度,以致于术语分布式处理有时被认为是寻找定义和名字的概念”。

本书作者将不谈这些争论,也不作为同义词研究“分布式处理”和“分布式计算”。相反地,象本书使用的那样,分布式处理的意思是任务的一个应用单元的处理使用了一个以上的独立计算机处理。这里使用的处理是应用处理而不是一个操作系统、数据库或其他支持系统的部分。这种分布式处理就定义为应用能力,因为支持处理(包括数据库处理)的任何分布对于应用程序开发商来说是不明显的,仅当应用程序的处理被分配时,系统才是可跨越平台的并允许应用程序使用多计算机平台。

因为使用不同的处理来完成任务级,就象在主机上一样,所以不会发生分布式处理。仅当由多计算机处理完成一个单一的工作单元时,才是分布式处理。

根据应用程序要求的不同,分布式处理应用程序可以以并行或串行的方式使用各种处理。而且这些处理可以在不同的计算机平台上进行,但在不同平台上的处理不要求研究应用程序的分布。

图 1.3 说明了分布式处理的布置,单一的任务应用单元使用一个以上处理(在本例中为两个),这些处理通过通讯协议互相协作完成任务,每个应用处理都可以访问独立的数据库。这两个数据库可以是相同的类型,也可以是不同的类型。

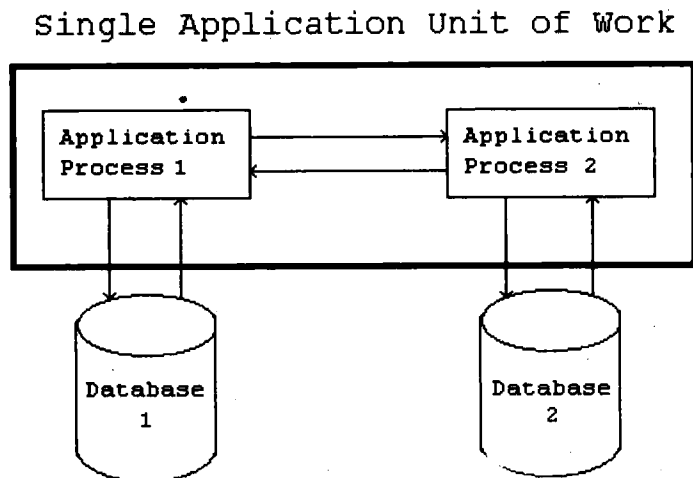


图 1.3 分布式处理

1.2.3 分布式数据库

数据库定义为计算机内永久存储的信息总和,它可以由计算机应用程序访问来支持应用处理。使用象数据库管理系统(DBMS)和更现代化的关系数据库管理系统(RDBMS)的指定软件后,上述的定义就更加具体化了。

在分布式数据库中,数据可以分布在一个以上的数据库中,但应用程序可以把它们当作一个逻辑数据库来处理。一个分布式数据库可以在相同的平台上,但更常见的是在多个平台上。分布式数据的概念对于理解事务管理的含义是很重要的。

大部分 DBMS 定义了数据库管理软件的多种情况。在本处,每一个情况就称为一个数据库。这些系统也常常允许定义多个空间,它们定义了数据的物理位置。当同一个数据库定义多个空间时,DBMS 管理这些空间,就象管理一个逻辑空间一样,因此,数据库不被认为是分布式的。

分布式数据库的概念与分布式处理是根本不同的。一个分布式数据库可以把数据分散到各个位置,而分布式处理把业务应用分散到各个位置进行处理。

图 1.4 说明了分布式数据库的布置。应用程序处理使用 DBMS 进行通讯,就好象数据库在同一个位置一样,DBMS 管理着数据在多个位置的分布。

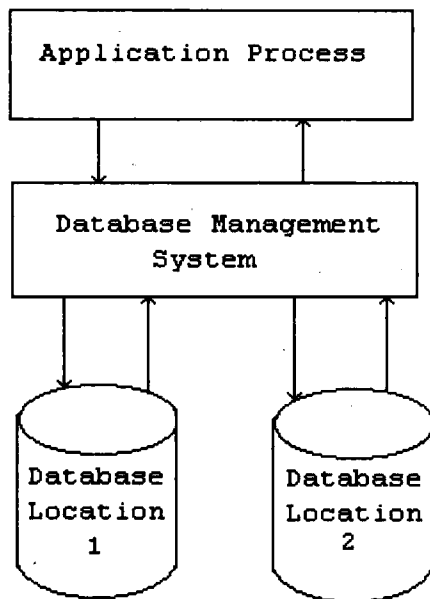


图 1.4 分布式数据库

1.2.4 客户/服务器的基本定义

术语客户/服务器是两个系统或两个处理之间的关系。客户要求服务器系统为之完成工

作。在大部分情况下,要根据要求者(客户)与服务器的关系来确定哪一个是客户和哪一个是服务器。

服务器提供客户要求的服务,区分服务器及其提供的服务是很重要的。因为单个服务器可以提供多个服务,因此应仔细区分名称“服务器”(Server)和“服务”(Service)。

把收到要求的系统或处理作为服务器并把发出要求的系统或处理作为客户是很方便的。当一个服务器向另一个服务器发出要求时,就会产生一些混淆。在以后的讨论中,可以使用上下文来确定正在处理的系统中哪个是客户,哪个是服务器。

图 1.5 说明了简单的 C/S(客户/服务器)的关系。客户发出服务要求到服务器,而服务器通过适当的答复响应客户的要求。

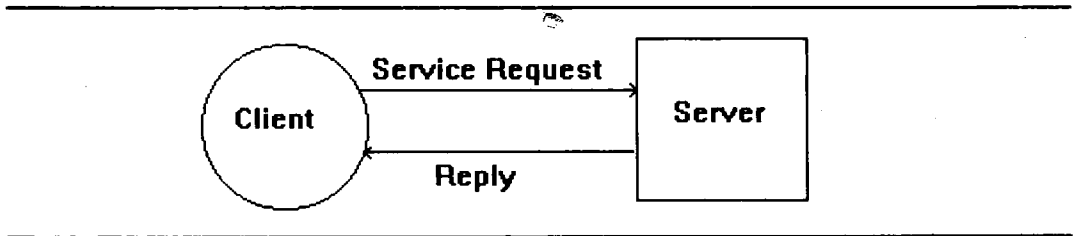


图 1.5 客户/服务器的关系

1.2.5 硬件服务器

目前,从一个或多个网络计算机系统中获得各种服务已是很平常的事情了。既然如此,提供服务的机器就称为服务器。通常,没有直接的终端用户与服务器交互,而是使用某些其他硬件(硬件客户)完成交互工作,这些硬件可能是完全不同类型的计算机,有时也可能是工作站。

操作系统和网络软件支持硬件服务器工作。换句话说,尽管计算机被认为是服务器,但它需要指定的软件作为服务器。

通常,硬件服务器有下述功能:

- 作为打印服务器,服务器上带有打印机可供网络上的任何用户使用。
- 作为文件服务器,服务器上存有公用文件。
- 作为存储机构,来支持带有很小或没有储存能力的工作站。
- 作为企业信息的集中式存储器。

对用户而言,提供的硬件服务器是显而易见的,但它通常对应用设计没有多少影响。本书后面的内容不再讨论这种服务器,因为软件服务器才是真正使客户/服务器技术如此强大的根源。

图 1.6 为一个硬件客户/服务器系统。一个或多个工作站与服务器相连并向服务器系统发出要求,然后服务器就通过回复来响应其要求。