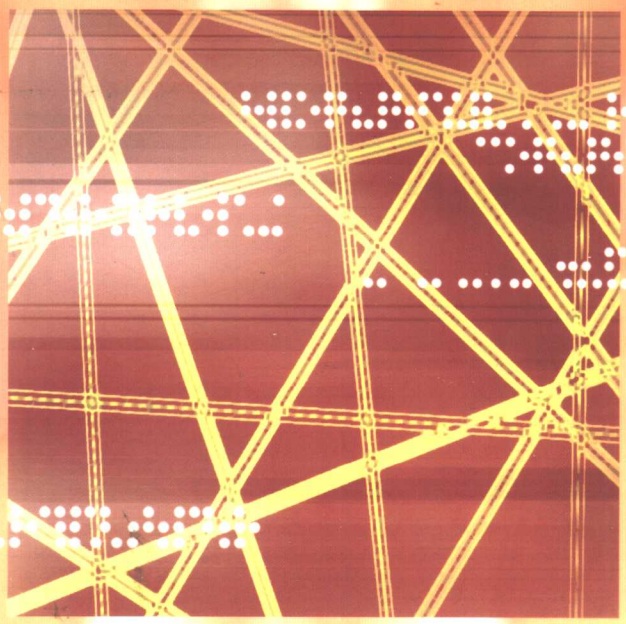
高等学校研究生系列教材

分布式操作系统

Distributed Operating Systems

何炎祥



高等教育出版社
HIGHER EDUCATION PRESS

内 容 提 要

分布式操作系统是为分布式计算机系统配置的一种操作系统。本书主要介绍设计和构造分布式操作系统的基本原理和典型实现技术，内容包括：分布式计算机系统的拓扑结构，分布式操作系统的结构模型、层次划分、通信机制、事件定序、并发控制与协同处理、资源管理、进程调度、处理机分配、死锁处理、文件系统、命名与透明性、任务分配和负载共享、故障检测与容错以及分布式事务处理，分布式共享内存，CORBA体系结构与中间件技术，面向对象的分布式操作系统的设计方法等。并分析、比较了三个有代表性的分布式操作系统实例，还讨论了一种新型分布式操作系统设计模型。

本书可作为高等院校高年级本科生、研究生和教师的教学用书，也可供从事分布式计算机系统体系结构、分布式操作系统、分布式数据库、分布式程序设计语言以及计算机网络等方面研究和开发的科技工作者阅读和参考。

图书在版编目（CIP）数据

分布式操作系统 / 何炎祥. —北京：高等教育出版社，2005.1
ISBN 7-04-016170-2

Ⅰ.分… Ⅱ.何… Ⅲ.分布式操作系统 Ⅳ.TP316.4

中国版本图书馆 CIP 数据核字（2004）第 126269 号

策划编辑 武林晓 **责任编辑** 武林晓 **市场策划** 陈 振
封面设计 王凌波 **版式设计** 史新薇
责任校对 朱惠芳 **责任印制** 朱学忠

出版发行	高等教育出版社	**购书热线**	010-64054588
社 址	北京市西城区德外大街 4 号	**免费咨询**	800-810-0598
邮政编码	100011	**网 址**	http://www.hep.edu.cn
总 机	010-58581000		http://www.hep.com.cn
经 销	新华书店北京发行所		
印 刷	河北省财政厅印刷厂		
开 本	787×1092 1/16	**版 次**	2005 年 1 月第 1 版
印 张	17.75	**印 次**	2005 年 1 月第 1 次印刷
字 数	380 000	**定 价**	29.50 元

物料号：16170-00

序　言

分布式计算机系统是并行处理系统的一种新形式，是计算机网络的高级发展阶段，是近年来计算机科学技术领域中备受青睐、发展迅速的一个方向。分布式操作系统是为分布式计算机系统配置的一种操作系统。一般认为，分布式操作系统是多机操作系统的典型代表。它在这种多机系统环境下，负责控制和管理以协同方式工作的各类系统资源，负责分布式进程的同步与执行，处理机间的通信、调度与分配等控制事务，自动实行全系统范围内的任务分配和负载平衡，并具有高度并行性以及故障检测和重构能力。它与单机操作系统和网络操作系统都有不同程度的区别，其复杂程度也明显高于它们。

本书主要讨论设计和构造分布式操作系统的基本原理和典型实现技术。全书共分十五章。第一章简述分布式计算机系统的拓扑结构与计算机网络；多机操作系统的基本结构，分布式操作系统的结构模型、层次划分、控制算法、设计途径及在设计时应着重考虑的一些问题。第二章介绍分布式通信机制，包括消息传递、远程过程调用（RPC）以及基于 Agent 的异步分布进程通信模型。第三章讨论分布式协同处理，包括事件定序与时戳，Lomport 算法、Ricart & Agrawala 算法和令牌传递算法，以及当协调者故障时选择新的协调者的算法。第四章介绍分布式系统中的资源管理策略，及其相关的死锁预防和死锁检测的有效方法。第五章专门讨论分布式进程管理以及处理机分配的有关问题。第六章集中讨论分布式系统中的多种任务分配与负载平衡方法，在此基础上，引入了智能型任务调度算法的模型及实现方法。第七、八章主要介绍分布式文件系统和命名服务的有关策略及分布式系统的透明性。第九章讨论分布式事务的并发控制问题，包括锁机制、两阶段提交协议及乐观并发控制方法等。第十章介绍分布式系统中的故障恢复和容错技术。第十一章结合 Ivy 系统讨论分布式共享内存的基本原理，重点在于一致性模型。第十二章专门介绍面向对象的分布式操作系统设计方法，讨论对象的权限和对象的同步，以及利用对象构造分布式操作系统的基本方法和步骤。第十三章通过对 Mach、Chorus 和 Amoeba 三个典型的分布式操作系统实例的分析和比较，将前面各章介绍的设计原理和方法进一步具体化。第十四章简述 CORBA 体系结构和中间件技术。实际上，分布式操作系统作为多机操作系统的高级表现形式，仍处于研究和发展阶段，在理论和研制方法上仍存在有待进一步解决和探索的问题，因此，在最后的第十五章提出了一种集智能型、集成化和可塑性于一体的新型分布式操作系统设计模型及其实现思路，以期加速有关的研究和探索过程。书后附有丰富的参考文献，可供有兴趣的读者进一步参阅。

本书力求叙述简明、深入浅出，并努力反映分布式操作系统研究方面的一些新概念、新

观点、新方法和新成果，可供学习和讲授分布式操作系统的计算机工作者以及高等院校本科高年级学生、研究生和教师作为教学用书，也可供从事分布式计算机系统体系结构、分布式操作系统、分布式数据库、分布式程序设计语言和计算机网络等方面研究和开发工作的科技人员阅读和参考。

书中还引用了一些专家学者的研究工作和论著。软件工程国家重点实验室、武汉大学研究生院、武汉大学计算机学院、高等教育出版社等对本书的编写给予了关注和支持，在此一并表示诚挚的感谢。

限于水平，书中难免存在错误，敬请各位赐教。

何炎祥

2005 年 1 月于武昌珞珈山

目　录

第一章　分布式计算机系统 …… 1
1.1　分布式系统的特征 …… 2
1.1.1　资源共享 …… 2
1.1.2　开放性 …… 2
1.1.3　并发性 …… 3
1.1.4　容错性 …… 3
1.1.5　透明性 …… 3
1.2　分布式系统的总体评价 …… 4
1.2.1　优点 …… 4
1.2.2　不足 …… 4
1.3　分布式系统的结构 …… 5
1.4　分布式系统的资源管理 …… 5
1.5　分布式系统的拓扑结构 …… 6
1.5.1　全互连结构 …… 6
1.5.2　部分互连结构 …… 6
1.5.3　层次结构 …… 7
1.5.4　星形结构 …… 7
1.5.5　环形结构 …… 7
1.5.6　多存取总线结构 …… 8
1.5.7　环－星形结构 …… 8
1.5.8　有规则结构 …… 9
1.5.9　不规则结构 …… 9
1.5.10　立方体结构 …… 9
1.6　计算机网络 …… 9
1.6.1　远程网 …… 10
1.6.2　局域网 …… 11
1.6.3　网络分层结构及通信协议 … 11
1.7　分布式操作系统 …… 13
1.7.1　多机操作系统的基本结构 … 13
1.7.2　设计分布式操作系统时应考虑的问题 …… 14
1.7.3　构造分布式操作系统的途径 …… 15
1.7.4　分布式操作系统的结构模型 …… 16
1.7.5　分布式操作系统的层次划分 …… 17
1.7.6　分布式操作系统的控制和管理策略 …… 18
1.7.7　分布式系统与计算机网络 … 19
1.7.8　分布式操作系统的设计方法 …… 19
1.8　小结 …… 20
第二章　分布式通信机制 …… 21
2.1　概述 …… 21
2.1.1　发送策略 …… 21
2.1.2　连接策略 …… 22
2.1.3　争夺处理 …… 22
2.1.4　保密 …… 23
2.2　消息传递 …… 24
2.2.1　消息传递原语 …… 24
2.2.2　同步消息传递方式的应用 … 26
2.2.3　组通信 …… 28
2.2.4　组通信的实现 …… 30
2.2.5　组通信的一个实例 …… 32
2.3　远程过程调用 …… 33
2.3.1　RPC 的功能 …… 35
2.3.2　RPC 的通信模型 …… 35
2.3.3　RPC 的结构及实现 …… 36
2.3.4　RPC 的语义 …… 39
2.4　异步分布进程通信模型 …… 40
2.4.1　PCAP 模型 …… 41
2.4.2　通道语法规则 …… 41
2.4.3　PCAP 模型的基本算法及其

改进 …… 42
2.4.4 一个层次－F通道应用 …… 43
2.4.5 性能分析 …… 45
2.5 小结 …… 45
第三章 分布式协同处理 …… 46
3.1 事件定序与时戳 …… 46
3.2 分布式互斥算法 …… 47
3.2.1 分布式互斥算法的基本假定 …… 48
3.2.2 集中式算法 …… 48
3.2.3 Lamport 算法 …… 48
3.2.4 Ricart 和 Agrawala 算法 …… 49
3.2.5 令牌传递算法 …… 51
3.3 选择算法 …… 53
3.3.1 Bully 算法 …… 53
3.3.2 基于环结构的算法 …… 54
3.4 小结 …… 55
第四章 分布式资源管理 …… 56
4.1 资源共享 …… 56
4.1.1 数据迁移 …… 56
4.1.2 计算迁移 …… 56
4.1.3 作业迁移 …… 57
4.2 资源管理策略 …… 57
4.2.1 局部集中管理 …… 58
4.2.2 分散式管理 …… 58
4.2.3 分级式管理 …… 59
4.2.4 一种分散式资源管理算法 … 59
4.2.5 招标算法 …… 60
4.3 死锁处理 …… 61
4.3.1 资源分配图 …… 62
4.3.2 进程等待图 …… 64
4.3.3 利用时戳预防死锁 …… 64
4.3.4 死锁检测方法 …… 65
4.3.5 集中式死锁检测方法 …… 66
4.3.6 层次式死锁检测方法 …… 67
4.4 小结 …… 68
第五章 分布式进程与处理机管理 …… 69
5.1 进程管理 …… 69
5.1.1 分布式进程 …… 69
5.1.2 分布式进程的状态与切换 … 69
5.1.3 分布式进程的同步与互斥 … 70
5.2 处理机管理 …… 70
5.2.1 处理机的状态及其转换 …… 70
5.2.2 处理机通信 …… 71
5.2.3 处理机分配与调度 …… 72
5.3 小结 …… 73
第六章 任务分配与负载平衡 …… 74
6.1 任务分配 …… 74
6.1.1 任务分配环境 …… 75
6.1.2 影响系统性能的因素 …… 75
6.1.3 基于图论的分配策略 …… 76
6.1.4 数学规划策略 …… 78
6.1.5 “合一－阈值”启发式分配算法 …… 79
6.1.6 一个改进的启发式算法 …… 81
6.1.7 基于遗传算法和模拟退火算法的任务分配策略 …… 85
6.1.8 基于非循环有向任务图的任务调度策略 …… 88
6.2 负载平衡 …… 94
6.2.1 概述 …… 94
6.2.2 负载平衡算法分类 …… 95
6.2.3 负载平衡算法的组成 …… 95
6.2.4 发送者主动算法 …… 96
6.2.5 接收者主动算法 …… 97
6.2.6 双向主动算法 …… 98
6.2.7 梯度模型 …… 98
6.2.8 接收者主动的渗透算法 …… 98
6.2.9 预约策略 …… 99
6.2.10 投标策略 …… 99
6.2.11 广播策略 …… 99
6.3 智能型任务调度算法 …… 99
6.3.1 任务调度中的知识及其表示 …… 100
6.3.2 任务调度程序的结构 …… 100
6.3.3 任务调度算法的实现 …… 101
6.4 小结 …… 102
第七章 分布式文件系统 …… 103

7.1 分布式文件系统的要求 …… 103
7.2 分布式文件系统的组成 …… 105
7.3 设计策略 …… 106
7.4 接口 …… 107
7.4.1 展开文件服务 …… 108
7.4.2 与 UNIX 的比较 …… 109
7.4.3 目录服务 …… 111
7.5 文件系统实现技术 …… 112
7.5.1 文件组结构 …… 112
7.5.2 权限和存取控制 …… 113
7.5.3 文件定位 …… 116
7.5.4 高速缓存 …… 117
7.6 NFS 分析 …… 119
7.7 小结 …… 125
第八章 命名服务与透明性 …… 127
8.1 概述 …… 127
8.1.1 名字与属性 …… 127
8.1.2 命名服务系统 …… 128
8.1.3 命名服务的一般要求 …… 129
8.2 一般的命名方式 …… 129
8.3 分布式系统中的命名方式 …… 131
8.3.1 名字管理器的主要功能 … 131
8.3.2 分布式系统中的命名方案 …… 131
8.3.3 惟一标识符和字符串名 … 132
8.4 名字服务器的设计 …… 133
8.5 分布式系统的透明性 …… 134
8.5.1 透明性 …… 134
8.5.2 与透明性相关的几个问题 …… 135
8.6 实例分析 …… 136
8.6.1 SNS …… 136
8.6.2 Internet 域名系统(IDNS) … 140
8.7 小结 …… 144
第九章 分布式事务处理 …… 145
9.1 概述 …… 145
9.2 简单分布式事务和嵌套事务 …… 145
9.3 原子提交协议 …… 147
9.4 分布式事务的并发控制 …… 150
9.4.1 分布式事务的锁机制 …… 150
9.4.2 分布式事务中的时戳定序并发控制 …… 150
9.4.3 分布式事务中的乐观并发控制 …… 151
9.5 分布式事务的死锁 …… 152
9.6 带复制数据的事务 …… 157
9.6.1 复制事务的体系结构 …… 158
9.6.2 有效副本复制 …… 160
9.6.3 网络分割 …… 161
9.6.4 带验证的有效副本 …… 162
9.6.5 定数一致方法 …… 162
9.6.6 虚拟分割算法 …… 165
9.7 小结 …… 167
第十章 故障恢复与系统容错 …… 169
10.1 概述 …… 169
10.2 事务恢复 …… 170
10.2.1 登录 …… 171
10.2.2 影子版本 …… 173
10.2.3 恢复文件中的事务状态表及意向表表目 …… 175
10.2.4 事务的故障模型 …… 177
10.3 容错 …… 178
10.3.1 故障特征 …… 179
10.3.2 Byzantine 故障 …… 180
10.4 分层故障屏蔽和成组故障屏蔽 … 182
10.4.1 分层屏蔽 …… 182
10.4.2 成组故障屏蔽 …… 182
10.4.3 稳定存储器 …… 184
10.4.4 主服务器与备份服务器 …… 185
10.5 小结 …… 187
第十一章 分布式共享内存 …… 188
11.1 概述 …… 188
11.1.1 消息传递与 DSM 的比较 …… 189
11.1.2 DSM 的主要处理方式 … 190
11.2 设计和应用 …… 191
11.2.1 数据结构 …… 191

11.2.2 同步模型 …… 191
11.2.3 一致性模型 …… 192
11.2.4 修改问题 …… 194
11.2.5 颗粒性 …… 194
11.2.6 抖动问题 …… 195
11.3 有序一致性与 Ivy 系统 …… 195
11.4 自由一致性与 Munin 系统 …… 201
11.4.1 自由一致性 …… 202
11.4.2 Munin 系统 …… 203
11.5 其他一致性模型 …… 204
11.6 小结 …… 205
第十二章 面向对象的分布式操作系统设计 …… 207
12.1 对象概念 …… 207
12.2 利用对象构造分布式操作系统的基本方法 …… 208
12.3 对象的保护域和权限 …… 210
12.4 对象的同步 …… 211
12.5 进程管理 …… 213
12.6 存储管理 …… 214
12.7 设备管理 …… 214
12.8 I/O 管理 …… 216
12.9 通信管理 …… 216
12.10 小结 …… 217
第十三章 分布式操作系统实例分析 … 219
13.1 Mach 系统 …… 219
13.1.1 设计目标和主要设计特性 …… 220
13.1.2 Mach 的主要概念 …… 221
13.1.3 端口、命名和保护 …… 222
13.1.4 任务和线程 …… 223
13.1.5 通信模型 …… 224
13.1.6 通信实现 …… 226
13.1.7 内存管理 …… 229
13.1.8 外部页面 …… 231
13.1.9 Mach 的主要特征 …… 233
13.2 Chorus 系统 …… 233
13.2.1 设计目标和主要设计特性 …… 234
13.2.2 Chorus 的主要概念 …… 235
13.2.3 进程管理模型 …… 235
13.2.4 命名和保护 …… 238
13.2.5 资源的群组管理 …… 238
13.2.6 通信模型及其实现 …… 241
13.2.7 Chorus 的主要特征 …… 243
13.3 Amoeba 系统 …… 243
13.3.1 设计目标和主要设计特征 …… 244
13.3.2 保护和权限 …… 244
13.3.3 进程与通信 …… 245
13.3.4 通信实现 …… 247
13.3.5 Amoeba 的主要特征 …… 250
13.4 Mach, Chorus 和 Amoeba 三者的比较 …… 251
第十四章 中间件技术与 CORBA 体系结构 …… 253
14.1 中间件技术 …… 253
14.2 CORBA …… 254
14.2.1 CORBA 简述 …… 254
14.2.2 CORBA 体系结构 …… 254
14.3 基于 Agent 和 CORBA 技术的分布式多媒体数据挖掘系统 …… 256
14.3.1 系统简介 …… 256
14.3.2 系统体系结构 …… 257
14.3.3 系统工作流程 …… 259
14.4 小结 …… 260
第十五章 新型分布式操作系统及其研制方法研究 …… 261
15.1 问题的提出 …… 261
15.2 新型分布式操作系统自动生成系统模型 …… 263
15.3 需要解决的关键问题 …… 266
参考文献 …… 267

第一章　分布式计算机系统

20 世纪 80 年代以来，高速计算机网络发展非常迅速。局域网（Local Area Network，LAN）可以连接小范围内的数十台甚至上百台计算机，信息传输的速度达 100 Mb/s 到 1 000 Mb/s，甚至更高。而广域网（Wide Area Network，WAN）则可以将全世界的计算机连接起来。

网络技术的发展使一些计算机系统从集中式走向分布式，那么什么是分布式系统呢？

分布式计算机系统（distributed computer systems）是由多台分散的计算机经互连网络连接而成的计算机系统。其中各个资源单元（物理的或逻辑的）既相互协同又高度自治，能在全系统范围内实现资源管理，动态地进行任务分配或功能分配，并能并行地运行分布式程序。

分布式计算机系统是多机系统的一种新形式，它强调资源、任务、功能和控制的全面分布。就资源分布而言，既包括处理机、输入/输出设备、通信接口、后援存储器等物理设备资源，也包括进程、文件、目录、表、数据库等逻辑资源。它们分布于物理上分散的若干站点中。而各站点经互连网络沟通，彼此通信，构成统一的计算机系统。

分布式计算机系统的工作方式也是分布的，其中各站点之间可根据两种原则进行分工，一种是把一个任务分解成多个可并行执行的子任务，分散给各站点协同完成，这种方式称为**任务分布**。另一种是把系统的总功能划分成若干子功能，分配给各站点分别承担，这种方式称为**功能分布**。不论是任务分布还是功能分布，分配方案均可依处理内容动态地确定。在分布式操作系统控制下，各个站点能较均等地分担控制功能，独自地发挥自身的控制作用，但又能相互配合，在彼此通信协调的基础上实现系统的全局管理。

然而，分布式系统有别于人们常说的网络系统。从操作系统的角度来看，分布式操作系统和网络操作系统是有很大区别的。

网络操作系统是为计算机网络配置的操作系统，网络中的各台计算机配置各自的操作系统，由网络操作系统把它们有机地联系起来。因此，它除了具有一般操作系统所具备的存储管理、处理机管理、设备管理、信息管理和作业管理等功能外，还应具有以下网络管理功能：

- 高效可靠的网络通信能力；
- 多种网络服务功能，包括远程作业录入、分时系统服务和文件传输服务等。

分布式操作系统（Distributed OS，DOS）则是为分布式计算机系统配置的操作系统，除了最低级的 I/O 设备资源外，所有的系统任务都可以在系统中任何别的处理机上运行。并提

供高度的并行性和有效的同步算法和通信机制，自动实现全系统范围的任务分配，并自动调度各处理机的工作负载，为用户提供一个方便、友善的使用环境。其主要特点是：

- 进程通信不能借助公共存储器，因而常采用信息传递（message passing）方式；
- 系统中的资源分布于多个站点，因而进程调度、资源分配及系统管理等必须满足分布处理要求，并采用保证一致性的分散式管理方式和具有强健性的分布式算法；
- 不失时机地协调各站点的负载，使其达到基本平衡，以充分发挥各站点的作用；
- 故障检测与恢复及系统重构和可靠性等问题的处理和实现都比较复杂。

1.1 分布式系统的特征

分布式系统应具有资源共享（resource sharing）、开放性（openness）、并发性（concurrency）、容错性（fault tolerance）和透明性（transparency）五个主要特征，这也是分布式系统的设计目标。

1.1.1 资源共享

资源共享分为两个方面：一是硬件资源共享，包括 CPU、存储器、大容量硬盘、打印机及其他设备；二是软件资源共享，包括软件工具、软件平台、应用软件、商用软件等。

为完成资源共享，必须进行管理，这就需要提供资源管理程序。提供资源管理程序的方法可以分为两种：

（1）客户机/服务器（client/server）模型：在客户机/服务器模型中，服务器提供各种资源共享的服务，如文件服务、打印服务或数据库服务等；客户机由用户直接使用，处理与用户的交互，负责向服务器发送服务请求，等待并接收服务器发回的应答信息，处理后显示给用户。在本模型中，客户机与服务器不一定是计算机，例如，可以是数据库中的数据库客户端和数据库服务器。因此，该模型既可作为硬件模型，也可作为软件模型。

（2）面向对象（object-oriented）模型：这种模型将分布式系统中可独立存在的资源作为对象处理。在这种模型中，任何共享资源及对于该资源的访问服务均被看作对象，其优点在于处理过程与资源封装在一起，不会随着对象的移动而改变对对象的访问模式，无论一个进程何时访问共享资源，只要向相应的对象发送一个消息即可，对象接到消息后，再分发到执行相应请求的过程或进程，然后将处理结果发回请求者。

1.1.2 开放性

（1）可伸缩性：分布式系统的可伸缩性，从硬件上看是指大可大到通过 Internet 连接的成千上万台主机，小可小到由局域网连接的几台机器。从软件上看是指系统软件可以根据需要扩充和裁剪。

（2）可移植性：分布式系统应该提供一个开放的服务集合，可能有许多种服务，一个服务也可能有许多不同的版本。例如，不同的文件命名和访问方法，可能在一种工作站上提供

Windows 的设施，在另一种工作站上提供 UNIX 的设施。每个客户程序可方便地、不需花太多开销地选择并装载合适的设施进入其执行环境，这就是可移植性。

（3）互操作性：不同厂家的软硬件，即使不是按标准生产或提供，但如果对外接口中的数据格式相互之间可转换，也能满足开放系统的要求，这就是互操作性。可以这样说，开放系统的数据是可交换的；开放系统的对外接口是公开的；开放系统必须提供统一的通信机制；开放系统必须有能力处理硬件的异构性，提供统一的用户界面。

因此，可以这样简单理解：

开放性 = 可伸缩性 + 可移植性 + 互操作性

1.1.3 并发性

并发性和并行性在分布式系统中是一种内在的特征。在分布式系统中，有许多计算机，每台计算机都有自己的 CPU 和存储器。若有 m 台计算机，每台计算机中有一个 CPU，那么，就会有 m 个进程并行执行。从分布式系统对于资源共享的基本要求来看，可以有以下两种并发性：

（1）许多用户同时发出命令，并与机器交互。在这种情况下，应用进程都在用户工作站上（并行）运行，相互之间没有冲突。

（2）许多服务器进程并发运行，每个进程响应不同的客户要求。在这种情况下，服务器之间存在并行进程，每台服务器中又存在并发进程，这些并发进程要响应不同的请求，但有可能要共享同一资源，因此必须解决并发控制问题。

分布式系统中分散的资源单元可以相互协作，共同解决同一个问题，在分布式操作系统控制下，实现资源重复（按任务）或时间重叠（按功能）等不同形式的并行性。

1.1.4 容错性

首先要承认计算机是会出错的，现在要讨论的问题是出错后怎么办？那就是容错。容错有两个基本方法，即硬件冗余和软件恢复。这些方法同样适用于分布式系统。硬件冗余如服务器镜像，即使用两个以上的完全相同的服务器，其中一个出现故障后，可以立即用另一个来提供对用户的服务；软件恢复则是指根据备份和备份后的操作日志将数据恢复到故障前的状态，使系统得以继续运行，数据库的故障恢复就是一个典型的例子。

1.1.5 透明性

分布式系统的透明性涉及到许多不同的方面，主要包括：

（1）位置透明性：用户不必知道待访问的资源在何处；

（2）迁移透明性：系统中的对象可以迁移，而不必改名；

（3）副本透明性：用户可以不知道他访问的对象是否有副本；

（4）并发透明性：多个用户可以自动共享资源，互不干扰；

（5）并行透明性：用户可以不必知道系统当前到底有多少项活动同时发生。

其中，最重要的透明性是位置透明，这种透明性的实现与否直接影响到分布式系统的表现，特别影响到分布资源的利用，最终影响到分布式系统的成功与否。

1.2 分布式系统的总体评价

任何事物都有两面性，分布式系统也有它的优点和缺点。

1.2.1 优点

因为分布式系统是表现为单机特征的多机系统，因此，它与集中式系统和分散式工作站都不同。下面分别将其与这两类系统进行比较。

与集中式系统相比，分布式系统具有以下几个方面的优点：

（1）经济：通过网络连接的个人计算机比起大型主机而言，具有很高的性价比。

（2）速度：在用户很多的时候，分布式系统在平均响应时间上要比大型主机短。分布式系统对于任务分散、交互频繁并需要大量处理能力的用户来说特别适合，由于各台机器支持单用户的处理能力，从而保证了执行交互任务时的快速响应。

（3）内在的分布性：这一优点对于许多新型应用提供最直接的支持。

（4）可扩充性：分布式系统的设计者或管理员可以根据需要扩充系统，而不必替换现有的系统成分。

（5）可靠性：当系统的一部分单元出现故障时，系统大部分工作仍可以继续，无须停机。

（6）适应多种应用环境：分布式系统中每个站点上的资源配置都能灵活地与当地用户的需求相吻合，因而特别适用于经济管理、事务处理、过程控制等这样一些具有分散用户又要求相互协同的应用场合。

与分散式工作站或个人计算机相比，分布式系统具有以下优点：

（1）资源共享：这是分布式系统最重要的目标。

（2）通信得到加强：分布式系统使原来分散的用户可以方便地通信，既方便了日常工作，也为协同工作提供了便利条件。

（3）可扩充的能力：通过网络互连形成的分布式系统可以均衡负载，提高系统效率。

1.2.2 不足

虽然分布式系统具有许多优点，但也存在下述不足：

（1）在分配资源时缺乏灵活性：在集中式系统中，所有的资源都由操作系统管理和分配，但在分布式系统中，资源分属于局部工作站或个人计算机，所以在调度的灵活性上不如集中式系统。

（2）性能和可靠性过于依赖网络：局域网的故障将会引起对用户服务的中止，网络的超负荷会导致性能的降低，增大用户的响应时间。

（3）安全保密性不足：为了获得可扩充性，分布式系统中的许多软件接口都提供给用

户，这样的开放式结构对于系统开发人员非常有价值，但同时也为破坏者打开了方便之门。

（4）基于分布式系统的应用软件太少。这也是目前难以建立完全分布式系统的原因。

1.3 分布式系统的结构

分布式计算机系统的结构可用机间耦合度作为主要标志来加以描述。**耦合度**是系统模块间互连的紧密程度，它是数据传输率、响应时间、并行处理能力等性能指标的综合反映，它主要取决于所选用的互连拓扑结构和通信链路的类型。

远程计算机网络采用串行数据传输，且受复杂的通信协议制约，故其耦合度最低，属于松散耦合系统。多处理机追求尽可能高的并行处理速度，故其耦合度最高。分布式计算机系统是两者发展的产物，它既考虑地理上分散环境的限制，又满足一定并行性的要求，故其耦合度介于前两者之间，一般属于中等耦合系统。

按地理环境衡量耦合度，分布式计算机系统可以分为机体内系统、建筑物内系统、建筑物间系统、不同地理范围的区域系统等，它们的耦合度依次由高到低。

按应用领域的性质决定耦合度，可以区分为三种不同的类型：一种是面向计算任务的分布并行计算机系统和分布式多用户计算机系统，它们要求尽可能高的耦合度，以便发展成为能分担大型计算机和分时计算机系统所完成的工作。第二种是面向管理信息的分布式数据处理系统，耦合度可以适当降低。第三种是面向过程控制的分布式计算机控制系统，耦合度要求适中，当然对于某些实时应用，其耦合度的要求可能很高。

分布式计算机系统可看作是并行处理系统的一种常见形式。

1.4 分布式系统的资源管理

分布式计算机系统有多种资源，每种资源可能有多个设备（包括逻辑设备），每个设备又可能有多种活动。设备分散在多台处理机中，因此，为了适应模块性、自治性和强健性的要求，系统设置了多个控制机构来管理，这样就形成了多个资源与多个控制机构之间比较复杂的关系。对于资源管理的分类有两种观点，一种是从单个资源与多个管理机构相互关系的角度进行分析，称为单个资源管理；另一种是从多个资源与多个管理机构相互关系的角度进行分析，称为多个资源管理。前者是后者的基础，后者是前者的提高，这是因为整个系统的资源是由单个资源组成的，但只有从整个系统的总体要求来考虑资源管理，才可能获得最佳的系统性能。

单个资源管理有四种方式：

（1）全集中管理方式，即一个资源仅能由一个管理机构管理；

（2）分担管理方式，即一个资源虽由几个管理机构管理，但各分担一种管理职能；

（3）轮流管理方式，即一个资源可由几个管理机构管理，但轮流执行管理职责；

（4）全分散管理方式，即一个资源由多个管理机构在协商一致的原则下共同管理。

在分布式计算机系统中，究竟使用哪种资源管理方式，须根据总体要求和资源性质来决定。在具体设计有关的资源管理算法时，除考虑死锁问题外，还须考虑公平性，并应防止饥饿现象发生。

1.5 分布式系统的拓扑结构

可以用不同的方式将分布式系统中的站点从物理上连接起来，每种方式都有其优缺点，下面简单讨论几种常用的连接方式并按以下标准来比较它们的性能：

- 基本开销：连接系统中的各个站点要多少花费？
- 通信开销：从站点 A 发送信息到站点 B 需要多少时间？
- 可靠性：若系统中某站点或通信链路出现故障，余下的站点是否仍能彼此通信？

为方便讨论，这里把各种拓扑结构用图形表示出来，其中的节点对应于站点，从节点 A 到节点 B 的连线对应于这两个站点之间的直接链路。如果一个系统已被分划成两个或多个子系统，且不同子系统中的站点已不再能彼此通信，则这个系统被称为分割的。

1.5.1 全互连结构

在一个全互连结构中，每个站点都直接与系统中所有其他的站点相连（如图 1.1 所示），这种结构的基本开销很高，因为每对站点之间都必须有一条直接通信链路。但在这种环境中，站点间的消息传递非常快，因为任何两站点间的消息传递只需要经由一条通信线路就可以直达。此外，这种结构是很可靠的，因为只有在相当多的通信链路故障的情况下，才可能分割该系统。

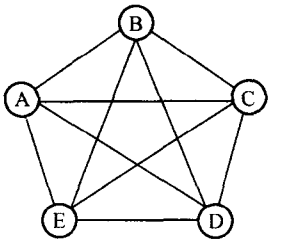


图 1.1 全互连结构

1.5.2 部分互连结构

在一个部分互连结构中，有些站点间存在直接通信链路，但有些则没有，如图 1.2 所示。因此这种结构的基本开销比全互连结构要低，但站点间的消息传递可能经由若干中间站点，以致延缓了通信速度。例如，在图 1.2 中，从站点 A 发送消息到站点 D 必须经由站点 B 和 C。

此外，部分互连系统也不如全互连系统可靠，因为其中的一个通信链路出现故障就可能分割该系统。例如，在图 1.2 中，若从站点 B 到站点 C 的通信链路出现故障，则该系统便被分割成两个子系统，一个包括 A、B、E；另一个包括 C 和 D，而且这两个子系统中的站点彼此不再能通信。为了减少这种情况的发生，通常让每个站点至少与另外两个站点连接。例如，如果在图 1.2 中增加一条从站点 A 到站点 D 的通信链路，那么任何单条通信链路故障都不可能导致对该系统的分割。

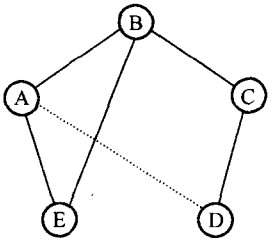


图 1.2 部分互连结构

1.5.3 层次结构

层次结构中的各站点组织呈树形结构，如图 1.3 所示，其中，每个站点（根除外）有一个惟一的父亲和若干个（或 0 个）孩子。这种结构的基本开销一般小于部分互连结构。在这种环境中，父子之间可以直接通信，孩子之间只能经由它们的共同父亲进行通信，从某个兄弟向另一个兄弟发送消息，需先向上发送给它们的父亲，然后再由其父亲向下发送给相应的兄弟。类似地，堂兄弟姐妹之间只能经由其共同的祖父进行通信。

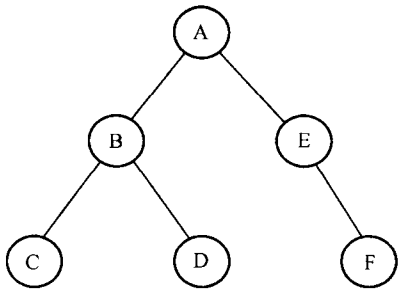


图 1.3　树形结构

若父站点故障，那么，它的孩子们彼此就不能相互通信，也不能与其他进程通信。一般而言，其中的任何中间节点故障（末端节点除外）都可能将这种结构分割成若干不相交的子树。

1.5.4 星形结构

在星形结构中，系统中的站点之一与系统中所有其余站点相连，其他的站点之间彼此不直接相连，如图 1.4 所示。这种结构的基本开销是站点个数的线性函数，其通信速度看起来也不会很慢，因为从站点 A 向站点 B 传递消息至多需要两次转接（从 A 到中央站点，再从中央站点到 B），但这种通信速度却是难以预测的，因为中央站点可能变成瓶颈，虽然传递消息所需的转接次数不多，但传递消息所花的时间可能不少。在一些星形结构系统中，中央站点完全担负着消息转接的任务。

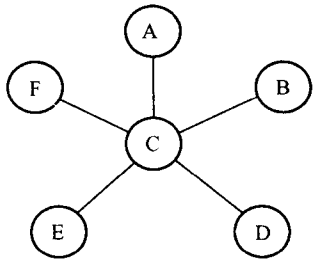


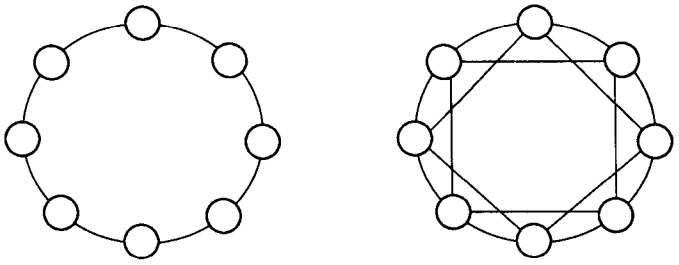
图 1.4　星形结构

如果中央站点出现故障，那么该系统就完全地被分割了。

1.5.5 环形结构

在环形结构中，每个站点物理上恰好与另外两个站点相连，如图 1.5(a)所示。这样的环形结构可以是单向的，也可以是双向的。在单向环形结构中，其中的一个站点只能给它的邻近站点之一直接传递消息，且所有的站点必须按相同的方向传递消息。在双向环形结构中，其中的一个站点可将信息传递给它的两个邻近站点。这种结构的基本开销不会很高，但通信代价可能较高，因为从一个站点向另一站点传递消息需沿环按预定方向传递直至到达目的地。在单向环结构中，这最多可能需要 n－1 次转接，而在双向环结构中，则最多可能需要 n/2 次转接，其中 n 是网络中站点的个数。

在双向环形结构中，其中两条通信链路故障就可能导致分割整个系统。在单向环形结构中，单个站点或单条通信链路故障，就可能分割整个系统。一种补救的办法是通过提供双通信链路来扩充这种结构，但这显然会增加基本开销，如图 1.5(b)所示。



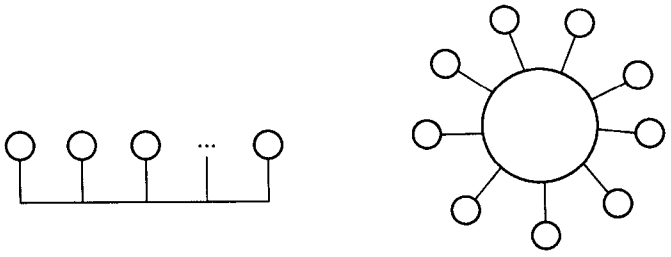
(a) 单通信链路　　(b) 双通信链路

图 1.5　环形结构

1.5.6　多存取总线结构

在多存取总线结构(简称总线结构)中,有一条共享的通信链路(即总线)。系统中所有的站点都直接与这条通信链路相连,它可以组织成直线状,如图 1.6(a)所示,也可以组织成环形,如图 1.6(b)所示,其中的站点可以经由这条总线彼此直接进行通信。这类结构的基本开销是站点个数的线性函数,通信代价也很低,除非这条总线变成了瓶颈。这类结构类似于带有一个中央站点的星形结构,其中某个站点故障不会影响其他站点间的通信,但是,若这条总线出现故障,那么该结构就完全地被分割了。

这只是单总线结构,另外还有多总线结构。



(a) 线形总线　　(b) 环形总线

图 1.6　总线结构

1.5.7　环 - 星形结构

环 - 星形结构由环形与星形结构叠加而成,其优缺点介于星形与环形结构之间(如图 1.7 所示)。

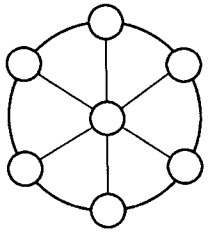


图 1.7　环 - 星形结构

1.5.8 有规则结构

有规则结构(如图 1.8 所示)中的每个站点都和与它相邻的上、下、左、右站点相连,因而具有高性能、高速度和高可靠性。不过,这种结构比较复杂,且一般要求各站点是完全一致的,构造这种系统的费用也较高。

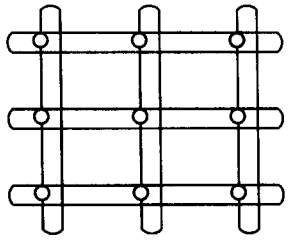


图 1.8 有规则结构

1.5.9 不规则结构

不规则结构(如图 1.9 所示)中的各站点间的连接关系无一定规则可依。其优点是:可随意增加不同类型的节点,各节点互连起来比较方便,还可提供任意冗余和重组能力;其缺点是运行时需要较复杂的路径选择算法。

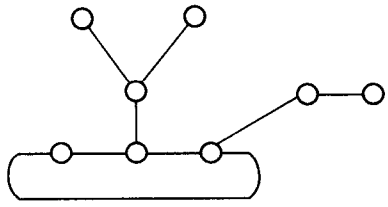


图 1.9 不规则结构

1.5.10 立方体结构

立方体结构又称 **n 维立方体分布式网络结构**。这种结构把 $2^n = N$ 个计算机互连起来,各计算机分别位于该立方体的角顶。立方体的每条边把两个站点连接起来,而每个站点则由 n 个全双向通路把它和 n 个其他站点相连。例如,n = 3、n = 4 时立方体互连结构如图 1.10 所示,其中,n 为立方体的维数。

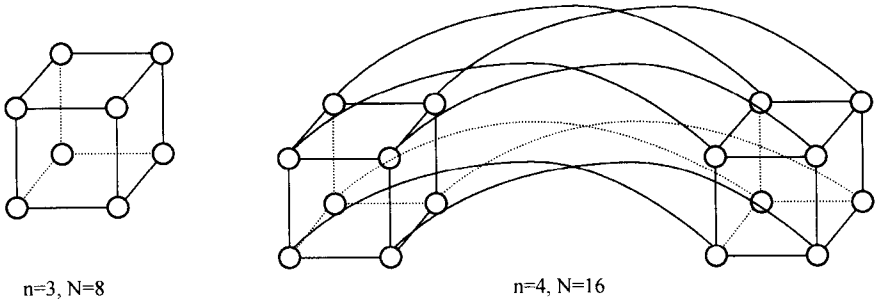


图 1.10 立方体互连结构

分布式系统的拓扑结构除上述 10 种外,还有交叉开关网、树形网、网状网、立方体网及超立方体结构等。

1.6 计算机网络

计算机网络分为两大类:远程网(广域网)和局域网。两者的主要区别在于它们所处的地理区域的大小不同。**远程网**(remote net)由分散在一个较大地理区域(如全国,甚至全球)的若干自治的处理机组成,而**局域网**(local net)则由分散在一个较小地理区域(如一座楼房