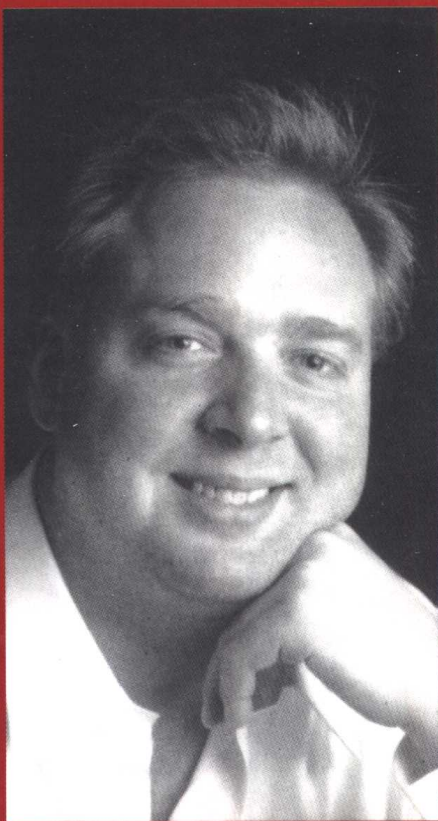
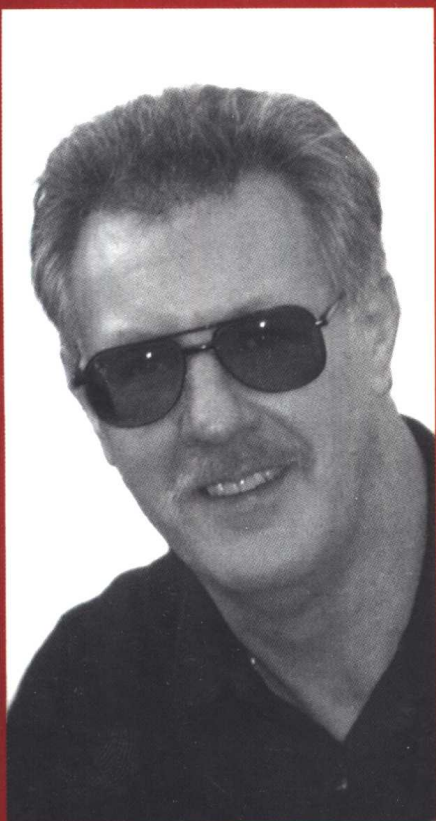


PROGRAMMER TO PROGRAMMER™



PROFESSIONAL Application Center 2000

[美] Alex Homer Matt Odhner 等著
袁勤勇 王福华 等译

Application Center 2000 专家指南



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



Application Center 2000 专家指南

[美] Alex Homer Matt Odhner 等著

袁勤勇 王福华 等译

清华大学出版社

(京)新登字 158 号

北京市版权局著作权合同登记号: 01-2001-4060

内 容 简 介

本书详细剖析了如何使用 Application Center 2000 来创建易管理、可伸缩和高可用的 Web 网站和 Web 应用程序,并对其性能进行监控,以获得最佳性能。全书共分 9 章和两个附录。第 1 章介绍可扩展的 Web 网站和 Web 应用程序;第 2 章介绍如何创建和管理 Web 群集;第 3 章介绍 Web 网站和 Web 应用程序的部署;第 4 章介绍如何管理客户会话状态;第 5 章介绍 COM+ 应用程序群集和路由群集;第 6 章介绍 Web 压力测试;第 7 章介绍 Web Application Stress 工具的使用;第 8 章介绍如何监控服务器和应用程序;第 9 章介绍监控器、警告以及监控数据库的定制。附录 A 介绍命令行管理实用程序(ac.exe);附录 B 介绍 Health Monitor 的默认选项。

本书内容新颖、丰富、全面,具有很强的技术通用性、实用性和指导性。本书是有关 Application Center 2000 的权威性著作,对网站建设具有很强的指导意义。

本书不但是从事网站建设、管理的广大从业人员的重要的工作指导书,而且也是高等院校师生教学、自学用的参考书。

Alex Homer, Matt Odhner: PROFESSIONAL Application Center 2000

EISBN: 1-861004-47-8

Copyright© 2001 by Wrox Press Ltd.

Authorized translation from the English language edition published by Wrox Press Ltd.

All rights reserved. For sale in the People's Republic of China only.

Chinese simplified language edition published by Tsinghua University Press.

本书中文简体字版由清华大学出版社和英国乐思出版公司合作出版。未经出版者书面许可,不得以任何方式复制或抄袭本书的任何部分。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: Application Center 2000 专家指南

作 者: [美] Alex Homer Matt Odhner 等著 袁勤勇 王福华 等译

责任编辑: 于 平

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印 刷 者: 北京通州区大中印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 23.5 字数: 557 千字

版 次: 2002 年 1 月第 1 版 2002 年 1 月第 1 次印刷

书 号: ISBN 7-302-05000-7/TP·2820

印 数: 0001~4000

定 价: 39.00 元

出版者的话

近年来，国内计算机类图书出版业得到了空前的发展，面向初级用户的应用类软件图书铺天盖地，但是真正有深度和内涵的高端图书不多。已经掌握计算机和网络基础知识的人们，尤其是 IT 专业人士迫切需要“阳春白雪”。IT 图书市场呼唤精品！

为了满足这种市场需求，清华大学出版社从世界出版业知名品牌 Wrox 出版公司引进了受到无数 IT 专业人士青睐，被奉为 IT 出版界经典之作的 Professional 系列丛书。这套讲述最新编程技术与开发环境的高级编程丛书，从头到尾都贯穿了 Wrox 出版公司“由程序员为程序员而著(Programmer to Programmer)”的出版理念，每一本书无不是出自软件大师之手。实际上，Wrox 公司的图书作者都是世界顶级 IT 公司(如 Microsoft, IBM, Oracle 以及 HP 等)的资深程序员，他们的作品既深入研究编程机理，传授最新编程技术，又站在程序员的角度，指导程序员拓展编程思路，学习实用开发技巧，从而风靡世界各地，被 IT 专业人士和程序员视为职业生涯中的必读之作。

作为国内 IT 出版社中最知名品牌，清华大学出版社与 Wrox 公司合作引进了这套 Professional 系列，然后迅速组织了一批相关领域的知名专家学者进行翻译，经过编辑人员认真细致的加工后，现陆续奉献给广大读者。

读者可以从 www.wrox.com 网站下载所需的源代码并获得相关的技术支持。同时，也欢迎广大读者参与 p2p.wrox.com 网站上的在线讨论，与世界各地的编程人员交流读书感受和编程体验。

前言

用户建立 Internet 连接,安装了 Windows 2000,草草建立了一个简单的电子商务网站,就可以在互联网上公开露面。或许还有点幸运,一些网站决定将您的 URL 加入到几种主要搜索引擎,您的网站将有访问者。您就可以出售产品,发布产品和服务信息,汇集联系信息和订单,然后就涉足到电子商务业务。

然后将会是一种微妙的变化,或许还需要一些时间,但是肯定会发生——您开始越来越多地依赖网站。它花费您更多的时间、更多的公司投资。同时,带来更多的公司销售额和市场份额,在您意识到这些之前,您的公司已经成为一个完整意义的“.com”公司。您的未来已经与 Internet 全球市场关联到一起。

有一天服务器死机了,所有的销售信息、所有的市场份额、所有顾客的联系信息都随之而去。当顾客再次定购时,他们发现除了“404 - page not found”外,什么也找不到。这条信息不像一条门上写着“去吃午餐,五分钟后返回”的便笺,至于网站的访问者,它就好像说“倒闭”了。他们将寻找其他的厂家购买产品。

此时您需要考虑应该成为一个真正的在线公司。不再将网站作为公司的副业。一旦开始依赖 Internet 提供个人和公司的收入,就需要慎重考虑如果公司的系统失败,将会带来怎样的严重后果。

0.1 本书内容

或许您现在已经猜到,本书就是介绍关于如何保护系统免受一些灾难性的失败,这些失败将潜在地导致您的业务中止。我们将不准备讨论可能发生的所有问题,这些问题包括在网站实现过程中的每一个层次上的不同问题,甚至导致不能保证 100% 的计算机正常运行时间。

但是,我们集中讨论两个具体问题:

- 当服务器失败时,网站或者网络应用程序怎样才能继续正常运行?
- 怎样应付来自拜访者日益增长的和经常变化的负载?

微软最近发布了一个新产品,将主要解决这两个问题:Microsoft Application Center 2000 (MSAC)。本书介绍 MSAC,怎样使用这个软件使网站更具有扩展性、更健壮、更可靠且更容易维护。

事实上,这 4 个问题是我们将要讨论的核心内容,我们将讨论每一个 MSAC 组件怎样实现这些目标,并有助于解决建立、运行与管理网站或者应用程序的问题。

那么,Application Center 究竟是什么?它是一组用于使建立、管理与监控多服务器网站和网络应用程序平台更容易的工具和功能。该产品提供的功能如下:

- 通过用多个服务器分担负载量,简易地实现可扩展性;

- 通过负载分担和自动的故障恢复实现较高的可用性；
- 在群集上对内容和计算机配置实现自动同步化；
- 简易部署使用组件的 Windows 2000 COM + 应用程序；
- 对单机和整个群集实现单点监控。

在本书中,将从一个程序员的角度讨论所有这些问题。当然,创建和管理一个 Web 是一个技术性和管理性的问题,而不仅仅是一个程序问题。我们将提供解决这些问题的基础性信息。

我们将介绍用户开始使用 Application Center 时需要配置和管理的选项,然后将讨论一个怎样形成网站的网页、组件和整个应用程序的例子。我们将讨论怎样测试网站和应用程序、怎样测量网站可支持的负载等内容。

注意:

或许您认为不得不花费成千上万的美元来使用 Application Center,但是我们将指出测试系统(如第一章所述)的花费少于 4000 美元。像这样的简单工具对于作技术试验并使用户熟悉此软件是理想的,而且幸运的是,用户办公室或部门的计算机都可以安装此软件。

0.2 本书读者对象

很明显,我们致力于使本书可以用于那些需要实现可扩展网站或者网络应用程序的读者。包括应用程序设计者、程序员和管理员,所有这些人或许都需要为此投入工作。因此,无论是个人或者接入 Internet 的公司、Internet 服务提供商(Internet Service Provider, ISP)、甚至是应用程序服务提供商(Application Service Provider, ASP),本书都会向他们提供帮助。惟一的要求是必须使用 Microsoft Windows 2000。

当然,作为个人,您可能在开始时发现并不需要整本书的内容,而可能仅仅感兴趣的是此技术的编程、管理、监控或者一些其他的技术领域。然而,您或许发现学习了更多的关于 Application Center 的知识,还要更多地了解关于 Application Center,想要利用它完成更多的工作。例如,它内在的可扩展监控环境,可以按所有的新方式使用。

即使您从来不想要建造自己的 Web 集,也可能考虑使用可以完成构建任务的一个主机。在不远的将来,您可能看到 ISP 为其客户实现 Application Center。在这种情况下,理解 Application Center 的概念是很重要的,尤其重要的是应该了解 Application Center 是如何影响程序和 Web 应用程序运行的环境的。

0.3 使用本书需要的环境

如果您想要试用 Application Center,所需要的不仅仅是一些不使用的单台服务器硬件。

为了建立一个测试群集,您需要至少两台同样配置的计算机,在这两台计算机上安装 Windows 2000 Server(或者更高版本)。这将可以试验网络群集的负载平衡和群集的管理。您也可以在这两台机器上建立一个负载平衡的 COM+ 应用程序群集。然而,为了理解所使用的产品功能,尤其是想要一起试验网络群集和 COM+ 应用程序群集时,至少需要 4 台这样的计算机。每一台计算机应该有如下配置:

- 一个 400MHz(或者更高)的 Pentium 兼容处理器;
- MS Windows 2000 Server、Advanced Server 或者 Datacenter Server(带 Service Pack1);
- 至少 256MB 内存;
- 100MB 用于 MSAC 服务的硬盘空间,另外的空间用于存储网站内容和数据库;
- 一块光驱;
- 一块网卡(推荐使用两块网卡)。

在第 1 章中,将详细讨论要求的规范和网络结构。

为了使用 Windows Network Load Balancing(NLB,Windows 网络负载平衡),在每台计算机上需要安装两个网卡(最好是新型号的网卡,一些旧型号网卡不能很好地执行 Windows NLB);为了连接计算机,也需要两个网络集线器和两条网线。其他设备,例如 KVM(Keyboard/Video/Mouse)转换器是很有用的,它允许通过一个键盘、监视器和鼠标管理所有的计算机。当然,也需要 Application Center 软件。

我们假定读者熟悉基本的 Web 编程技术,例如:

- Visual Basic/VBScript;
- JavaScript /JScript;
- Microsoft Active Server Pages。

而且还要了解网络服务器和 Internet 是怎样工作的。在此我们不深入讨论复杂编程问题,因此您不需要是这方面的专家。相反的,我们将简单地使用它们演示 Application Center 是怎样友好地创建应用程序,或者利用特定的功能。

本书中的所有例子使用的源代码都可从以下网站下载:<http://www.wrox.com>。

0.4 作者的联系地址

作者尽可能使本书满足读者需要,因此我们需要知道读者的需求。作者渴望知道读者想要学习的知识。

作者会重视来自读者的反馈,不论是批评或者赞誉。在下一版中,将考虑这些建议。希望读者通过以下地址将建议反馈给我们:

feedback@wrox.com

或者连接网站:

<http://www.wrox.com>

目 录

第 1 章 可扩展的 Web 网站和 Web 应用程序	1
1.1 对可扩展性和可靠性的研究	1
1.2 使硬件更健壮	2
1.3 使软件更健壮	2
1.3.1 使系统更具可扩展性	2
1.3.2 系统监控和性能测量	3
1.3.3 应用程序开发的前景	3
1.3.4 应用程序服务提供商	4
1.4 网络的可靠性与可扩展性	4
1.4.1 通过多服务器增强可靠性	5
1.4.2 通过多服务器提高系统可扩展性	5
1.4.3 两种方法的比较	5
1.5 Application Center 的功能	6
1.5.1 易于实现的可扩展性	6
1.5.2 自动化的高可用性	8
1.5.3 内容和配置的自动同步化	9
1.5.4 COM + 应用程序的简易部署	9
1.5.5 单点监控	10
1.5.6 第三方负载平衡解决方案	12
1.5.7 实际性能	13
1.6 安装 Application Center	14
1.6.1 硬件要求	15
1.6.2 安装 Application Center 软件	17
1.6.3 配置网络连接	20
1.7 小结	27
第 2 章 创建和管理 Web 群集	28
2.1 Application Center 群集的类型	28
2.2 创建 Web 群集和 Stager 群集	29
2.2.1 创建新 NLB 群集	30
2.2.2 创建非 NLB 群集	35
2.2.3 创建 Stager 群集	35

2.3 查看新创建的群集	36
2.3.1 检查群集和服务器状态	36
2.3.2 查看性能信息	39
2.3.3 查看单个服务器的信息	40
2.3.4 查看事件记录目录	41
2.3.5 查看群集的监控器	42
2.3.6 改变群集的 IP 地址	43
2.4 在 Web 群集中添加服务器	44
2.4.1 启动添加群集成员向导	44
2.4.2 设置负载均衡选项	44
2.4.3 创建群集或向其中添加服务器时出现的错误	46
2.5 管理 Web 群集	47
2.5.1 连接到一个 Web 群集	47
2.5.2 服务器的在线或离线设置	48
2.5.3 强制同步化	49
2.5.4 改变群集控制器	49
2.5.5 重启动服务器和服务	49
2.5.6 设置群集属性	49
2.5.7 设置成员服务器属性	54
2.5.8 设置事件集属性	56
2.5.9 查看群集同步化	57
2.5.10 使用基于浏览器的管理工具	58
2.5.11 命令行管理选项	61
2.6 从群集中删除成员服务器	63
2.7 解散 Web 群集	64
2.8 小结	65
第 3 章 部署 Web 网站和 Web 应用程序	66
3.1 内容同步化处理过程	66
3.1.1 同步化模式	66
3.1.2 监控手动同步化	67
3.1.3 部署选项	67
3.1.4 简单内容的同步化	68
3.1.5 COM+ 应用程序的同步化	72
3.2 发布一个简单 Web 网站	74
3.2.1 连接到群集控制器	74
3.2.2 为网站添加一个新的 IP 地址	75

3.2.3	为网站创建一个新的磁盘文件夹	75
3.2.4	连接到 Internet Services Manager	75
3.2.5	创建新网站	76
3.2.6	发布 Web 网站内容	79
3.2.7	最后的结果	79
3.3	发布一个 COM + 应用程序	80
3.3.1	管理逐步进行的部署	81
3.3.2	样本网站和传统的 COM 组件	81
3.3.3	安装和测试组件	83
3.3.4	创建一个 COM + 应用程序	84
3.3.5	在 COM + 应用程序中安装组件	87
3.3.6	使用系统 DSN 来定义一个数据库连接	89
3.3.7	创建一个新的应用程序包	90
3.3.8	向包中添加一个 COM + 应用程序	91
3.3.9	向包中添加一个数据源	92
3.3.10	向包中添加其他资源	94
3.3.11	部署新的应用程序	97
3.3.12	应用程序包是否值得	108
3.4	同步化 ASP.NET 应用程序	108
3.4.1	ASP.NET 的配置和部署模式	109
3.4.2	Application Center 和 ASP.NET	110
3.5	小结	110
第 4 章	管理客户会话状态	112
4.1	管理客户状态	112
4.1.1	客户服务器应用程序和可执行应用程序	112
4.1.2	基于 Web 的分布式应用程序	113
4.1.3	管理 Web 集的客户状态	117
4.2	客户状态管理实例	125
4.2.1	表单控件和查询字符串	125
4.2.2	使用 cookie 维护状态	129
4.2.3	使用 IE5 的特定功能	133
4.2.4	使用 ASP 会话保存状态	141
4.2.5	访问一个后端数据库	149
4.2.6	在服务器上使用文本文件和 XML 文件	157
4.3	小结	159

第 5 章 COM + 应用程序群集和路由群集	161
5.1 什么是 COM + 应用程序群集	161
5.1.1 COM + 应用程序群集如何工作	162
5.1.2 COM + 应用程序群集包括什么	164
5.2 WroxBank 应用程序样本	164
5.2.1 数据库表	165
5.2.2 应用组件	165
5.2.3 ASP 页	166
5.3 创建一个 COM + 应用程序群集	168
5.3.1 连接到服务器	168
5.3.2 新建群集向导	169
5.3.3 在 COM + 应用程序群集中添加服务器	172
5.3.4 COM + 应用程序群集的属性	174
5.4 部署样本应用程序	175
5.4.1 创建 COM + 应用程序	176
5.4.2 创建 COM + 应用程序	176
5.4.3 创建应用程序包	177
5.4.4 将 BankWebApp 应用程序部署到 Web 群集	177
5.4.5 在向 COM + 应用程序群集中部署 BankCOMApp 应用程序	180
5.5 配置样本应用程序	182
5.5.1 为组件启动 CLB	183
5.5.2 指定 CLB 请求的路由信息	183
5.6 运行这个样本应用程序	185
5.7 什么是 COM + 路由群集	186
5.7.1 创建一个 COM + 路由群集	187
5.7.2 配置一个 COM + 路由群集	188
5.8 小结	189
第 6 章 Web 压力测试	191
6.1 Web 测试的重要性	191
6.2 服务器端 Web 压力测试	192
6.2.1 服务器端 Web 性能测试的目的	193
6.2.2 测试请求	194
6.3 找到瓶颈	197
6.3.1 内存瓶颈	198
6.3.2 处理器瓶颈	201
6.3.3 COM 组件瓶颈	203

6.3.4	线程的选通	204
6.3.5	网络瓶颈	204
6.3.6	硬盘瓶颈	205
6.4	建立一个性能度量	206
6.4.1	兆赫开销公式	207
6.4.2	常见网站的请求速度	209
6.4.3	群集容量计划	210
6.4.4	理解慢客户连接的影响	210
6.4.5	基线 IIS 性能	212
6.4.6	基线 Application Center 2000 性能	213
6.5	常见 ASP 性能故障	215
6.5.1	优化 ASP 脚本设计	216
6.5.2	运行到无阻塞 ASP	216
6.5.3	安全套接字层	217
6.5.4	优化 IIS 性能和稳定性	217
6.6	解决 Web 应用程序的问题	218
6.7	对网络压力测试有效的工具	220
6.7.1	Application Center 测试	221
6.7.2	性能管理工具	223
6.7.3	其他的资源	223
6.8	小结	224
第 7 章	Web Application Stress 的使用	225
7.1	WAS 测试平台	225
7.1.1	压力测试的目的	227
7.1.2	何处获取 WAS 工具	227
7.1.3	安装	227
7.1.4	首次打开此工具	227
7.1.5	报表	230
7.1.6	查询字符串和查询字符串编辑器	235
7.1.7	脚本节点	237
7.2	使用脚本向导创建脚本	252
7.2.1	手动创建脚本	253
7.2.2	创建浏览器记录脚本	253
7.2.3	日志文件脚本的创建	256
7.2.4	创建内容树脚本	257
7.2.5	使用对象模型创建一个脚本	258

7.2.6 关于脚本的一般准则	259
7.3 为压力测试准备好群集	259
7.3.1 对网络负载均衡群集的压力测试	259
7.3.2 Web 应用程序压力工具名称解析	262
7.3.3 得到高并行连接	263
7.3.4 确定你的需求	264
7.4 常见 Web 压力测试问题	264
7.4.1 Web 压力测试问题解决	265
7.4.2 压力测试环境	267
7.4.3 使用 Microsoft Web Application Stress 的局限性	267
7.5 小结	269
第 8 章 监控服务器和应用程序	270
8.1 监控的角色	270
8.2 监控结构	271
8.3 性能监控	272
8.3.1 时间周期	273
8.3.2 性能计数器	273
8.3.3 集合的危险	274
8.3.4 监控器	275
8.3.5 事件	276
8.3.6 远程监控	276
8.4 Health Monitor	277
8.4.1 Health Monitor 代理	277
8.4.2 Health Monitor——实用性	281
8.4.3 Health Monitor 控制台	283
8.5 小结	299
第 9 章 自定义监控器和警告,以及监控数据库	301
9.1 视窗管理工具(WMI)	301
9.1.1 WMI 提供器	302
9.1.2 WMI SDK	303
9.1.3 Application Center 和 Health Monitor WMI 的提供器	307
9.1.4 使用 WMI 事件提供器	307
9.2 创建一个监控工具	312
9.2.1 样本监控工具	312
9.2.2 Application Center 管理接口(ACAI)	313
9.2.3 WMI 事件	317

9.3	Application Center 数据库	321
9.3.1	连接到数据库	321
9.3.2	数据库结构	322
9.3.3	观察数据	325
9.4	添加性能计数器	326
9.5	Application Center 监控器的未来	329
9.6	小结	329
附录 A	命令行管理实用程序	330
A.1	应用程序命令	330
A.1.1	/create 参数	330
A.1.2	/delete 参数	331
A.1.3	/list 参数	331
A.1.4	/addresource 参数	332
A.1.5	/removeresource 参数	332
A.1.6	/listresources 参数	333
A.2	CLB 命令	333
A.2.1	/setclbmembers 参数	334
A.2.2	/loadclbmembers 参数	334
A.2.3	/addclbmember 参数	335
A.2.4	/removeclbmember 参数	335
A.2.5	/listclbmembers 参数	335
A.3	群集命令	336
A.3.1	/create 参数	336
A.3.2	/delete 参数	339
A.3.3	/add 参数	340
A.3.4	/remove 参数	340
A.3.5	/setcontroller 参数	341
A.3.6	/clean 参数	342
A.3.7	/listmembers 参数	342
A.3.8	/listnics 参数	342
A.4	部署命令	343
A.4.1	/start 参数	343
A.4.2	/listdeployments 参数	344
A.4.3	/terminate 参数	344
A.4.4	/enablesync 参数	344
A.4.5	/disablesync 参数	345

A.4.6	/status 参数	345
A.5	LOADBALANCE 命令	346
A.5.1	/online 参数	346
A.5.2	/offline 参数	346
A.5.3	/setnlbweight 参数	347
A.5.4	/status 参数	347
A.6	HELP 命令	348
附录 B	Health Monitor 默认选项	349
B.1	Data Groups	349
B.2	默认动作	349
B.2.1	E-mail Administrator 属性	350
B.2.2	Log to offline.log 属性	350
B.2.3	Log to Websitefailures.log 属性	351
B.3	默认的同步化监控器	351
B.3.1	Application Center 监控器(数据组)	351
B.3.2	Application Center Log Monitors(数据组)	354
B.3.3	在线/离线监控器(数据组)	356
B.3.4	系统监控器(数据组)	356
B.3.5	Web 站点监控器(数据组)	358

第 1 章 可扩展的 Web 网站和 Web 应用程序

利用现有的操作系统和软件建立一个网站相对容易。在微软系列产品中,只要安装 Windows 2000 Server 就能得到创建工作网站或 Web 应用程序所需要的全部服务。如果只有相对较少的访问者,这样就足够了。

然而,网站可能很快就会成为你事业的核心部分,此时就必须要考虑一些其他的问题。例如,你能确定自己的网站可以应付日益增加的访问者吗?如果由于技术问题影响了服务器,如何才能保证网站每周五天、每天九小时都是可用的?而且,如何才能保证为网站提供服务的后端事务服务,如数据库存取和事务服务,不会由于其复杂性日益增加而使数据处理降低为缓慢的蠕动。

本书为在 Windows 2000 下运行的网站和 Web 应用程序提供了一个解决问题的新方法,即 Microsoft Application Center 2000(通常称为“Application Center”,或者缩写为 AC 2K 或 MSAC)。

注意:

AppCenter 是另一个常用的“昵称”,实际上是来自另一个公司的一种与其不同的产品(见 <http://www.inprise.com/appcenter>)。

在这一章里,将涉及到的主要问题包括建立可扩展的网站,在紧接其后的一章里,将继续讨论如何使用 AC2K 实现这样的网站和应用程序。本章的主题如下:

- 对可扩展性、可靠性和有效性的追求;
- Web 上的可扩展性和可靠性;
- AC2K 的功能;
- 安装 AC2K。

下面就从“可扩展性”和“可靠性”开始。

1.1 对可扩展性和可靠性的研究

通常许多人认为“计算机”和“可靠性”是相互排斥的。计算机在统计电子账单时会出错;当从一个生产商那里订购货物时,会丢失详细资料、向用户索要错误的电话费……所有的这些情况发生得太频繁了,这些似乎是您的问题的根源。

当然,尽管许多错误是由于软件的或人为的因素造成的,但是硬件故障(如磁盘碎裂)可能是更常见的原因。问题就在于我们现在的信息技术(Information Technology, IT)环境的整个基础设施是建立在真实的物理设备的基础之上——不能期望这些设备在 100%的时间里

都能绝对正常运转。而且这决不可能,计算机也和其他一切事物一样有发生故障的时候。可靠性就是对一个系统在多大程度上会运行正常的一种测量标准。

1.2 使硬件更健壮

与以前的计算机相比,现代计算机更具可靠性,而且制造商也在不断使用新技术提高其产品可靠性。现在的服务器都具有热插拔磁盘组、双电源、内置的硬件监控系统和许多专为保持计算机持续运转的其他特性。对此一个常用的描述性术语就是“健壮性(robustness)”。

注意:

可靠性和健壮性两个概念的差异在于:可靠性是指试图将服务器故障对系统造成的损坏或使系统停止运行的可能性减少到最小——这种情况与您开着一盏亮着油压警示灯的汽车回家相似。而健壮性更多指系统根本不发生故障,或者尽管造成了微小的破坏,但仍能恢复。这很像永无故障的汽车减震器和那些自密封的轮胎被扎破后能自动充气一样……

1.3 使软件更健壮

这种方法正日益被现代软件所采用。例如,Windows 2000 操作系统已经增加了这一功能,可以自动保护系统文件(如 DLL)不被其他版本的系统文件修改,或不小心而被删除——这是造成软件错误的常见原因之一。对驱动器文件预先做出数字标识以便确定它们已经被充分检测,而且他们是(或尽可能地)和操作系统充分兼容的。设备驱动器上的错误是造成 Windows NT 操作系统出现蓝屏错误的最常见原因。无论如何,想要建立一个不出错或不会造成整个系统停止运行的不兼容的操作系统(或其他应用程序既无毛病,又能提供我们想要的所有功能)几乎是不可能的。

1.3.1 使系统更具可扩展性

除了硬件和软件都不可能 100% 健壮和可靠之外,还要考虑其他的问题。这些问题大部分可归于可扩展性这一类。换句话说,即使是硬件和软件无任何故障或错误,它们能支持加在其上的处理负载吗?

对扩展性的追求是用于各种各样应用程序(而不仅仅是 Web)分布式计算领域的崇高目标。随着公司规模扩大和雇员的增加,对现存资源,如域控制器、电子邮件服务器和整个网络基础设施的压力可能弱化其功能,并造成相关系统故障。

可扩展性的意义在于,在一个封闭的环境,如企业网或局域网中,用户至少能制定预期发展计划。在 Internet 上网站或应用程序是面向全世界的,精确估算下个月对某一网站或网络应用程序的访问人数,以及因此而造成的 Web 服务器和后端支持系统的负载,几乎是不可能的。