



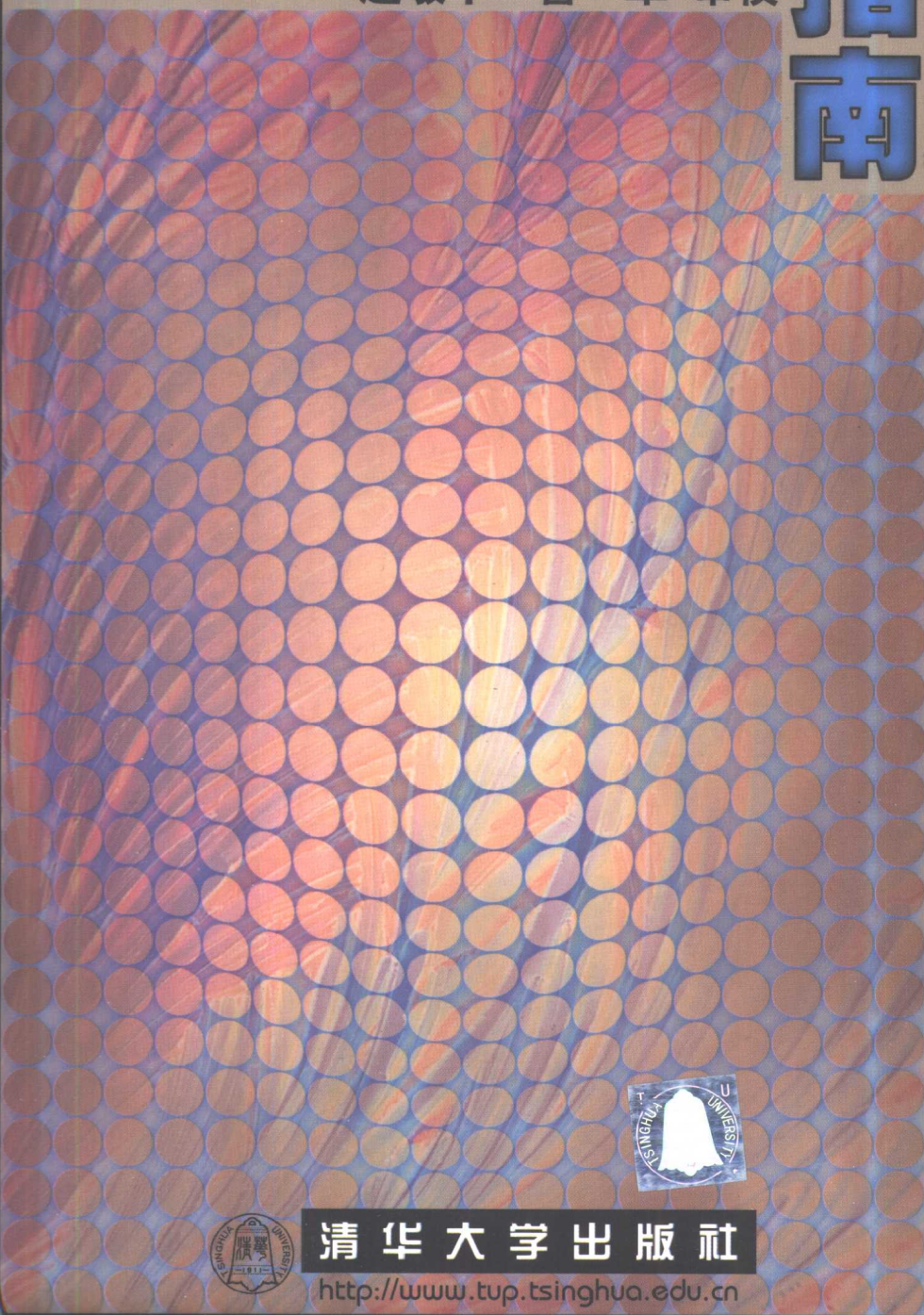
鲍居武 陆晓 郭进 等编著

Informix

高级编程

赵敬中 曹军 审校

指南



清华大学出版社

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

(京)新登字 158 号

内 容 简 介

Informix 数据库是当前比较流行的 UNIX 平台下的三大数据库 (Oracle, Informix, Sybase) 之一, 本书从 Informix 用户的角度出发, 系统地介绍了 Informix 数据库的知识, 全面地阐述了 Informix 数据库的各项实用技巧和应用软件开发技术。本书共分三部分, 分别为 Informix-OnLine、Informix-SQL 和 Informix-ESQL/C 语言。

本书总结了作者长期从事 Informix 研究和应用开发的实践经验, 相信会对广大用户有所帮助。通过对本书的学习, 读者能够熟练掌握 Informix 的使用技巧, 了解该软件发展状况, 并能够进行应用程序开发。

本书实用性强, 既可作为广大计算机用户的重要参考书, 也可作为各大中专院校相关专业的参考教材。

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: Informix 高级编程指南

作 者: 鲍居武 陆 晓 郭 进 等

出版者: 清华大学出版社(北京清华大学校内, 邮编: 100084)

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

印刷者: 北京市清华园胶印厂

发行者: 新华书店总店北京科技发行所

开 本: 787 × 1092 1/16 印张: 25.25 字数: 628 千字

版 次: 2000 年 10 月第 1 版 2000 年 10 月第 1 次印刷

书 号: ISBN 7-302-01215-6/TP·456

印 数: 0001 ~ 3000

定 价: 38.00 元

前 言

Informix 数据库是当前比较流行的 UNIX 平台下的三大数据库 (Oracle, Informix, Sybase) 之一, 本书从 Informix 用户的角度出发, 全面阐述了 Informix 数据库的各项实用技巧和应用程序开发技术。本书共分三部分, 分别阐述了 Informix-OnLine、Informix-SQL 和 Informix-ESQL/C 语言。

Informix-OnLine 是一个数据库服务器, 它把高可用性、联机事务处理 (OLTP) 性能和多媒体能力结合起来。OnLine 通过管理自己的共享内存资源支持并发进程, 并同时维护事务的隔离性。OnLine 通过管理自己的磁盘 I/O, 支持数据库的数据存储在多个磁盘上。管理人员还可以根据需要, 通过 Informix-OnLine/Optical 把数据存储在光盘上。

Informix 软件提供了多种多样的应用开发工具、CASE 工具、数据库服务器、实用程序和客户/服务器网络产品。Informix 的应用开发工具包括 Informix-SQL、Informix-4GL、Informix-ESQL/C 等。DB-Access 是一个功能齐全的实用程序, 它可以完成存取、修改、删除、检索数据库信息的工作。Informix-Star 提供了用户在网上运行应用系统的工具, 支持多个 OnLine 数据库服务器的分布式系统。

本书总结了作者长期从事 Informix 研究和应用开发的实践经验, 相信会对广大用户有所帮助。通过对本书的学习, 读者能够掌握数据库的基本概念和 Informix 的使用, 了解该软件发展状况, 并能够进行应用程序开发。

本书系统地介绍了 Informix 数据库的知识, 读者不必具有数据库管理的经验也能顺利阅读本书, 它可作为初、高级 Informix 数据库用户的教材或参考书, 也可作为各大中专院校相关专业的参考教材。

在本书编写过程中, 王存山、鲍立芳、张玉生、郭永、曹军、鲍居源、鲍居仕、次光、鲍清芳、鲍元芬、金瑞玲、丁平仁、张瑞芳、李云东、鲍文瑄、段妙珍、鲍卫仁、宫日明、王振华、贺映文、孙颖霞、赵敬中老同志也参加了编写工作。

由于我们水平有限, 书中难免有缺点和错误, 敬请广大读者批评和指正。

编 者

2000 年 8 月于北京

目 录

第 1 章 Informix 简介	1
1.1 Informix 公司简介	1
1.2 Informix 产品介绍	1
1.3 Informix-OnLine 动态服务器 7.1 版介绍	4
1.4 TPC 商业基准测试	18
1.5 Informix 与其他数据库比较	22
1.6 Informix 典型应用	23

第一部分 Informix-OnLine 系统

第 2 章 Informix-OnLine 系统概述	24
2.1 Informix-OnLine 的特点	24
2.2 Informix-OnLine 的安装	26
2.3 OnLine 的初始配置	27
2.4 初始化 OnLine	44
2.5 设置环境变量	45
2.6 附录	47

第 3 章 Informix 数据库结构	51
3.1 系统初始化	51
3.2 OnLine 的用户进程	57
3.3 共享内存结构	59
3.4 磁盘数据结构	70
3.5 物理日志	90
3.6 逻辑日志	91
3.7 消息日志	95

第 4 章 数据库管理	108
4.1 DB-Monitor(监控器)	108
4.2 OnLine 运行的方式	110
4.3 逻辑日志管理	114
4.4 archive 管理	129
4.5 数据库的一致性、备份和恢复	138
4.6 监控数据库的运行	156

4.7 数据库性能调整	169
4.8 修改 OnLine 配置	177
第 5 章 实用程序	192
5.1 dbexport/dbimport	192
5.2 dbload:用命令文件装入数据	197
5.3 tbcheck	201
5.4 tbinit	205
5.5 tbload	206
5.6 tbmode	207
5.7 tbstat	209
5.8 tbtape	220

第二部分 Informix-SQL 语言

第 6 章 SQL 语言概述	222
6.1 SQL 概貌及特点	222
6.2 SQL 数据定义功能	224
6.3 SQL 数据操纵功能	227
6.4 SQL 中视图的定义和操作	237
6.5 SQL 数据控制功能	241
第 7 章 SQL 语言程序优化	244
7.1 引论	244
7.2 查询优化	245

第三部分 Informix-ESQL/C 语言

第 8 章 使用 RDSQL	253
8.1 引言	253
8.2 关系数据库和数据类型	253
8.3 RDSQL 语句概要	254
8.4 加锁	264
8.5 用户的地位和权限	266
8.6 索引策略	267
8.7 NULL 值	269
8.8 视图	271
8.9 外连接	274
8.10 通过记录标识符访问表	275
8.11 函数 TODAY 和函数 USER	275

第 9 章 使用嵌入 SQL 的 C 程序	276
9.1 概述	276
9.2 前导文件	276
9.3 蕴含(include)文件	277
9.4 宿主变量	277
9.5 指示变量	279
9.6 C 例行程序中嵌入的 RDSQL 语句	280
9.7 出错处理和 sqlca 结构	280
9.8 动态 RDSQL 语句和 sqllda 结构	281
9.9 编译 ESQL/C 程序	286
9.10 范例	288
 第 10 章 RDSQL 语句	293
10.1 RDSQL 语句	293
10.2 SELECT 语句	328
 第 11 章 RDSQL 数据类型	346
11.1 概述	346
11.2 CHAR 类型	349
11.3 SMALLINT 和 INTEGER 类型	349
11.4 SERIAL 类型	350
11.5 SMALLFLOAT 和 FLOAT 类型	350
11.6 DATE 类型	350
11.7 MONEY 类型	355
11.8 DECIMAL 类型	355
 第 12 章 库函数	364
12.1 概述	364
12.2 函数说明	364
 附录 Informix-ESQL/C 错误信息	371

第 1 章 Informix 简介

1.1 Informix 公司简介

Informix 软件公司成立于 1980 年,其总部设在美国加利福尼亚的 Menlo Park,在 Menlo Park、伦敦、新加坡,分别设有 Informix 的美国、欧洲、国际销售总部。此外,在全美各地和世界的 28 个国家中还设有分公司和办事处。Informix 是国际软件市场上主要的数据库厂商之一。

1983 年,Informix 在全球首家推出了在 UNIX 操作系统上运行的关系型数据库产品。以后的十多年中,Informix 公司继续致力于 UNIX 和其他开放平台上关系型数据库软件的开发,为用户提供了包括数据库服务器、网络连接软件、应用开发工具、最终用户工具等在内的一系列数据库软件产品,以及相应的用户服务。

据调查统计,Informix 公司的产品全球销售量占据 UNIX 数据库市场的重要份额,其技术先进、功能强大、界面友好易用而深得广大计算机用户的喜爱。

Informix 公司十分重视软件产品的本地化工作,致力于对亚洲语言的支持,包括对简体汉字的支持。

1.2 Informix 产品介绍

Informix 公司开发产品的宗旨是:为用户提供高生产率的,贯穿应用系统整个生命周期的数据库技术。Informix 软件使用户能够方便地开发、维护和扩展应用系统,并使应用系统软件具有高效性、高安全可靠性和有效的恢复特性。

Informix 产品具有很好的开放性,它们都基于 ANSI 标准 SQL,同时 Informix 也是第一家通过 NIST(美国国家标准和技术协会)检验的数据库软件公司。它可以在 UNIX, Windows, Windows NT, NETWARE, Macintosh 等多种操作系统环境下运行。

1.2.1 数据库服务器

Informix 数据库产品采用客户/服务器体系结构。应用开发工具(或称做客户)提供开发和运行应用系统的用户界面,数据库核心(或称做服务器)进行数据管理,包括数据存储、数据跟踪、数据检索、事务管理、数据库保护等。Informix 提供两个主要的数据库服务器: Informix-OnLine 和 Informix-SE。

■ Informix-OnLine

这是适用于大型应用的、功能强、效率高的 OLTP(联机事务处理)数据库服务器,支持存储过程、触发器、参照完整性、多媒体数据存储等。

■ Informix-SE

这是一个易安装、易使用、易维护的数据库服务器。它提供 SQL 的数据处理能力,而所需要的数据库管理工作很少,因此很适合于中小型企业使用。Informix-SE 可在 UNIX、DOS、Windows、NETWARE、OS2、XENIX 环境下运行。

1.2.2 网络连接软件

Informix 提供网络环境下运行的数据库软件,支持分布式数据库。Informix 的客户/服务器结构中,客户和服务端可以在同一台机器上,也可以在计算机网络中的不同机器上;在一个计算机网络中可以有多个数据库服务器,一个客户上的应用程序可以访问多个服务器中的数据。Informix 网络连接软件支持的网络协议主要有 TCP/IP 和 AT&T 的 STARGROUP。

Informix 网络连接软件主要有:

■ Informix-STAR

这是一个分布式数据库产品,它使得用户能够对多个场地上的多个数据库中的数据进行连接、查询、更新等操作。

■ Informix-NET

这是一个网络通信产品,它使得客户计算机上的 Informix 应用开发工具能够访问远程的 Informix 数据库服务器中的数据。

■ Informix-GATEWAY WITH DRDA

DRDA 是 IBM 公司 1990 年 6 月公布的分布式关系数据库结构。它规定了应用系统和远程的关系数据库管理系统之间连接的一系列协议。IBM 把 DRDA 作为数据库间互操作的标准。Informix 是首家推出支持 DRDA 产品的非 IBM 厂商。使用 Informix-GATEWAY WITH DRDA 这个网络连接产品,不需要在 IBM 主机上再增加软件,Informix 应用就可以访问和修改 IBM 关系数据库 DB2 和 SQL400 中的数据。

■ Informix-TP/XA

这个产品将 Informix-OnLine 连接到符合 X/OPEN, TP/XA 的事务管理器,以支持跨多个数据库或多个计算机系统的全局事务。

1.2.3 应用开发工具

Informix 的应用开发工具支持用户高效率的开发应用。开发工具中既有支持字符方式的,又有支持图形方式的;既有过程化的高级语言,又有非过程化的描述工具。从专业的应用开发人员到最终用户,各类人员都可以找到适合自己的工具。

Informix 的应用开发工具主要包括:

■ Informix-4GL

这是 Informix 公司针对数据库应用而设计的整体性的、功能齐全的第四代语言,它提供了开发数据库应用的全部工具,使用户能够快速地开发出高级的数据库应用系统。

■ Informix-4GL FORMS

这是一个为快速建立数据录入应用而提供的代码生成器和屏幕表格描绘器,它能根据用户对屏幕格式的简单描绘,得到屏幕格式说明文件,并自动生成数据库录入应用程序的 Informix-4GL 代码。

■ Informix-OPENCASE、TOOLBUS

Informix-4GL FOR TOOLBUS 是 Informix 的 CASE 工具。OPENCASE、TOOLBUS 为 4GL 应用软件开发者提供了一个集成的图形开发环境,各种 CASE 工具被集成在一个基于 MOTIF 的界面环境下。4GL 和 TOOLBUS 是基于 Informix-4GL 的集成开发环境,它将 4GL 集成到 OPENCASE、TOOLBUS 中,并提供编辑、编译、调试与运行等手段,使基于 4GL 的应用开发周期大大缩短。

■ Informix-SQL

这是一个完整的数据库应用开发工具集,包括模式编辑器、菜单构造器、SQL 编辑器、屏幕表格建造器、报表书写器等。

■ Informix-ESQL/C

嵌入式 SQL,将 SQL 语句嵌入到第三代语言程序中。

■ Informix-4GL/GX

这是一个图形界面运行工具,使得在字符方式下的 4GL 软件能在图形环境下运行,并以图形环境的界面形式出现。

■ Informix-HYPERSCRIPT TOOLS

这是一个面向客户/服务器应用的多平台、可视化的编程环境,使语言开发人员可以很方便地设计基于图形的、事件驱动的应用系统。

■ Informix-DBA

这是一个专为数据库管理员提供的基于图形用户界面的系统维护工具,使用它可以方便地定义和修改数据库结构,并可建立和维护最终用户使用的超级视图 (SUPERVIEW)。

1.2.4 最终用户工具

最终用户工具将数据库的复杂性隐藏起来,为最终用户提供鼠标指示方式的数据库访问。Informix 的最终用户工具主要有 Informix-VIEWPOINT,这是专为最终用户提供的对数据进行存取并将所获得的数据以某种方式表现出来的图形开发工具。它建立在超级视图的概念之上,使最终用户看到的是一个关于他所要存取数据的简单、明了的描述。

除上述 Informix 公司的产品外,许多第三方厂商的开发工具也可以在 Informix 环境中运行,从而进一步增加了对用户的支持。

1.3 Informix-OnLine 动态服务器 7.1 版介绍

Informix-OnLine 动态服务器是 Informix 的强大的多线程数据库服务器,其设计目的在于发挥对称多处理器(SMP)和单处理器结构的能力,以提供数据库可伸缩性结构(DSA);其目的在于提供有效的并行数据库结构,以帮助在管理日益增大和复杂化的数据库的同时改善整个系统的性能和可伸缩性。

OnLine 动态服务器提供高级事务处理,通过新的并行数据查询(PDQ)技术而达到最优的决策支持和数据完整性。大型主机能力的系统管理,图形化的监控工具,以及多媒体的能力,所有这些都在一个单一的,已按客户/服务器包装的软件包中。OnLine 动态服务器支持 Informix 整个系统的基于 SQL 的应用开发工具,以及相当大数量的第三方工具,并可在许多基于 UNIX 独立或网络环境下的计算机系统中运行。

1.3.1 关于 Informix-OnLine 动态服务器

当今复杂的、关键任务的数据库管理应用通常需要联机事务处理(OLTP)、批处理和不断增长的决策支持的结合。为满足这些需求,你就需要有可伸缩的数据库性能,以动态地适应于容纳更大的数据库和更多的并发用户。你还需要为了尽量增大你现存的软硬件配置(包括单处理器和多处理器结构),以及新出现的硬件结构(例如松散耦合和大规模并行结构)的能力而设计的数据库技术。

Informix-OnLine 动态服务器提供最有效的可行解法——应用下一代并行数据库结构以提供大型主机能力的可伸缩性、可管理性和性能、最小的操作系统开销,以及负载的自动分布。OnLine 动态服务器的多线程结构的设计目的在于充分利用可用的硬件资源(CPU、磁盘、存储器、备份磁带设备等),并且可被联机地动态再装配以适应系统负载不断变化的要求。OnLine 动态服务器将这些优点提供给 SMP 和单处理器结构,并且在未来的版本中将进一步支持松散耦合的聚集和大规模并行硬件结构。

1.3.2 OnLine 动态服务器的优点

这种联机的动态调整数据库参数的能力,主要目的在于关键任务数据库应用的高可用性和可伸缩性。OnLine 动态服务器允许监控、分配与撤消数据库服务器进程(而不需要将系统停下来),以达到在保持高可用性的前提下保证对资源的最有效的分配。所有这些处理都是透明地进行的,不会影响应用系统和系统上的用户。

这样做的好处如下:

- 通过动态分配和配置数据库系统资源而达到最大限度的可伸缩性。
- 对于 OLTP 和决策支持应用的高级、多线程的并行处理性能。
- 动态的、联机的系统管理,以对任务进行监控和对负载进行分配。

- 为高级的并行 I/O 操作和高可用性的数据库管理而进行的局部的表的分割。
- 通过动态分配共享存储器,高速缓存和其他资源而达到的高可用性。
- 通过绕过操作系统的限制而达到的降低了的操作系统开销。
- 任务的优先级化,调度和数据库级的加锁(不同于操作系统级的加锁)。
- 跨所有可用的数据库服务器进程的动态负载平衡。

1. 最大限度的可伸缩性

使用 OnLine 动态服务器,在一个 OLTP 环境中的并发用户数目可大量增长,同时又保持相当高的性能水平和可接受的代价。这是因为 OnLine 动态服务器使用数据库服务器进程(称为虚拟处理器)来动态配置缓冲池和多个并发的线程(MCT),为多个客户的请求服务,其中唯一的限制因素是硬件资源的可用性和配置(例如 CPU、磁盘、存储器、备份磁带设备等的数目)。因为这种结构使用所有可用的处理器,所以当增加系统资源时吞吐量是恒定的,可伸缩性是有保证的。

2. 优良的性能

OnLine 动态服务器提供智能化的优化性能和充分发挥硬件固有处理能力的独特功力。这是因为许多主要的数据库活动是并行执行的(例如 I/O,复杂查询,索引建立,日志恢复,备份和恢复)。OnLine 动态服务器的虚拟处理器可以并发地服务于多个线程,于是所有的可用系统资源可发挥其最大的潜能。

这种多线程结构不仅对于 OLTP 是可伸缩的,而且对于运行决策支持查询极其高效。Informix 的 PDQ 技术冲破了以往的 UNIX 数据库结构的可伸缩性界限,以往的数据库结构不能够处理企业级决策支持应用的数据量、系统管理需求和性能需求。PDQ 技术将用户任务/查询分解为多个子任务,并把它们分配给多个处理器。(在下一节“OnLine 动态服务器是如何工作的”中对 PDQ 进行了详细的讨论。)

3. 动态体系管理

当数据库变得越来越大,并且跨不同的地点分布时,就非常需要动态系统管理工具,以控制和紧密调试各种系统参数,例如 CPU 和内存使用,异步 I/O 队列,决策支持和批队列,可利用的磁盘空间,高效的分割模式等。

与此同时,运行关键任务,每周 7 天 24 小时应用的公司需要连续不断地运行他们的系统。OnLine 动态服务器主机能力的管理工具使得数据库管理员(DBA)能够联机执行大多数或全部系统管理功能,用不着把系统停下来。这些工具包括系统监控界面(SMI),内存授权管理器(MGM),DB/Cockpit 和 Onperf。

4. 局部的表分割

OnLine 动态服务器的分割(亦称分片)能力使得表能够灵活地分布在多个硬盘上,这就大大地提高了处理性能。这样对于数据分布模式提供了更大的灵活性,使得可以并行地执行 I/O 操作,同时也提高了系统的可管理性和高可用性。因为数据库的规模不断增大,因此

数据库的局部表分割非常重要。它使得上百 GB 范围的非常大的数据库 (VLDBs) 变得更好管理。

5. 通过共享内存而达到的高可用性

并行性有时需要更多的内存以对请求进行处理。OnLine 动态服务器不是配置好一个内存界限,而是依据“按需要 (on-demand)”动态地分配共享内存,这就进一步提高了可用性,因为你不会为了重新配置内存的使用而被迫将数据库系统停下来。而且数据库释放内存时,可以被操作系统回收使用。

6. 减少操作系统的开销

通过要求较少的为数据库工作的 UNIX 进程来支持大量并发用户,OnLine 动态服务器尽可能地减少了不必要的操作系统资源。实际上,是由数据库服务器线程而不是数据库服务器进程来为客户的请求提供服务的。OnLine 动态服务器的多线程算法智能地进行上下文切换(数据库服务器线程可以从一个虚拟处理器移到另一个)和加锁,这就使得 OnLine 动态服务器比单线程的数据库服务器能更有效地处理数据库请求。

7. 任务的优先级化、调度和封锁

OnLine 动态服务器能提供最大限度的响应能力,因为对数据库任务的优先级处理,调度和封锁已由操作系统移至数据库服务器中。因为 OnLine 动态服务器知道哪一个数据库操作是最紧急的,所以它能更明智地调度 CPU 和对任务进行优先级划分。例如,一个高优先级的日志书写任务(用于保证在非正常系统关闭情况下的实际完整性)决不会被排在一个向临时工作文件写入的操作之后。

8. 动态负载平衡

OnLine 动态服务器能够为你的应用系统创建一个非常高效的环境,因为任务在所有的可用的资源上进行动态分配。线程静态地分配到服务器,比较突出的要求则由第一个可用的虚拟处理器提供服务。

1.3.3 OnLine 动态服务器是如何工作的

OnLine 动态服务器核心结构的设计完全是为了提供内部构造的多线程和并行处理能力。这种作为基础的 DSA 框架结构能保证对所有系统资源最有效的利用。

OnLine 动态服务器提供了一种独特的能力,使你能够通过利用独特分配的数据库服务器进程(虚拟处理器)缓冲和为了加快处理器将大任务划分为子任务来调整你的数据库系统。这些虚拟处理器调度和管理用户请求,并通过 MCT 来并行地执行多个子任务。

1. 多线程

OnLine 动态服务器不是通过启动 UNIX 进程来响应客户请求,而是创建一种叫做线程

的轻便机制。一个线程就是一个顺序的控制流,它代表数据库服务器进程中的离散任务。系统为每个用户对话建立一个线程,多条线程可以在虚拟处理器缓冲池中并行执行(在这一点上,虚拟处理器与物理处理器非常相似)。线程与多线程处理器的关系就如同数据库服务器进程和操作系统的关系一样。

2. 虚拟处理器

OnLine 动态服务器使用虚拟处理器来管理活动的线程,并高效地在它们中间进行切换。线程产生、排队,并由第一个可用的虚拟处理器为其服务,以保证高效的硬件使用和无瓶颈。OnLine 动态服务器的虚拟处理器设计成具有内部的智能,可以高效地协调多个 MCT。“线程调度器”可以充分利用数据库对象和算法的深层知识,因此它能够比通常的操作系统提供更“明智”的调度。

当一条线程正在等待资源时,虚拟处理器能够为另一条线程工作。通过提供这种灵活性,OnLine 动态服务器允许少量 UNIX 进程管理大量的用户,因为一个虚拟处理器可响应多个用户的要求。尽管系统可能产生多个虚拟处理器用于响应用户的要求,但在用户看来,则仅有一个数据库服务器。

因为所要求的 UNIX 服务器进程的数目显著下降,从而所要求的操作系统内部的上下文切换也就变少了。这就使 OnLine 动态服务器能够绕过操作系统强加的大多数性能限制和负担。另外,使用 OnLine 动态服务器,你还可以充分利用由硬件销售商提供的特殊的调度特性(例如 Processor affinity)。

3. 虚拟处理器分类

为了执行高效性和多功能的协调,虚拟处理器被按需分组。每一类被优化以完成一种特定的功能,例如 CPU 操作、磁盘 I/O、客户/服务器通信和管理任务。通过将虚拟处理器分类,OnLine 动态服务器能够高效地调度操作系统和将操作优先级化。

线程在有关类型的虚拟处理器之间透明地进行调度,你可以用每种类型的足够的虚拟处理器来配置你的系统,以处理系统上的特殊类型的负载。鉴于虚拟处理器缓冲池可以极简便地联机调整,所以你可以很快地调整每一个类型或进行改变以适应阶段性的大任务和负载高峰。

这些虚拟处理器类型中,最重要的是 CPU 类型,它可以在 CPU 处理需求改变时增加或减少。如果有一个长队列需要 CPU 处理,那么你可以很快动态地启动一个 CPU 虚拟处理器以便消除瓶颈,而无需中断系统上的任何用户。

鉴于处理客户需求所需要的虚拟处理器的数目常常是不可预料的,因此可以提供几个用于客户/服务器的通信协议(例如 IPX/SPX、TCP/IP,和共享内存)的虚拟处理器类型来处理客户与 OnLine 动态服务器之间的通信。如果需要向系统添加光媒体存储设备的话,甚至可以增加一个光盘虚拟处理器类型来处理 I/O。

4. 动态共享存储器

OnLine 动态服务器用到的所有存储器都是由虚拟处理器缓冲池共享的。这样,为了更

有效地处理客户需求,就可以配置 OnLine 动态服务器,向共享存储器缓冲池中增加更多的存储空间。

OnLine 动态服务器的共享存储器最大限度地减少了碎片。这样,存储器的利用率可以不随时间而下降。除了初始分配之外,共享存储器片段自动地按需要大块地添加进来。而且,共享存储器片段还可以在运行数据库时由管理员添加。存储管理系统还在存储空间用完时自动增长存储器片段。当用户对话结束时,它所使用的存储空间被释放,并且被另一个对象重新使用。通过释放分配给数据库的存储空间,操作系统可以将存储空间收回。这样,用户线程可以很容易地在虚拟处理器间移动,当用户数目增加时,这样就增强了 OnLine 动态服务器的可伸缩性。

OnLine 动态服务器在共享存储器的虚拟部分分配一个区域,用来存储线程执行的函数的非共享数据,使其不被并行地执行相同的代码的其他线程抹去。OnLine 动态服务器将为某些特定的操作(例如递归的存储过程)动态地增长栈的大小。

1.3.4 OnLine 动态服务器的特性和功能

OnLine 动态服务器给予依据当前需要调整数据库性能,以及更有效地管理整个系统的能力,本节详细介绍这些性能。OnLine 动态服务器利用这些特性来提供性能上的优越性。

1. 异步 I/O

通常 I/O 是数据处理中最慢的部分。OnLine 动态服务器使用它自己的异步 I/O(AIO)软件包(或可利用的操作系统核心的 AIO)来加速 I/O 处理。因为 OnLine 动态服务器的虚拟处理器不必等待一个 I/O 请求进行服务,所以一个虚拟处理器不必等待一个 I/O 结束就可以开始处理另一个服务请求。

I/O 虚拟处理器有三种特定类型:逻辑日志 I/O,异步 I/O,核心异步 I/O(KIO),将 I/O 划分为这些类型使得可以实施一种高效的优先级划分方案。另外,鉴于 I/O 请求被统一调度,OnLine 动态服务器就能有效地使所有可用的磁盘处于忙状态。实际上,你可以使用 OnLine 动态服务器系统监控界面来检测从磁盘读写的长队列,然后,启动附加的特定于 I/O 的虚拟处理器来消除 I/O 瓶颈。

2. 提前读

为了顺序地扫描或索引,可以将 OnLine 动态服务器配置成在处理共享内存中的前页面集的同时,能够异步地从硬盘提前读入若干页。具有了这种能力,应用系统就可花费较少的时间用于等待磁盘存取的完成,这样就保证了吞吐量。

3. 并行的装入

Informix 的并行装入实用程序能够非常快速地装入和卸出数据,这是由于它能够从多个来源读入数据,并且能够并行地将数据装入到 OnLine 动态服务器中。图形化的用户界面使得 DBA 能够:

- 为多种类型的普通文件(例如 ASCII、COBOL 或 EBCDIC)配置装入,并执行必要的转换(例如 EBCDIC 到 ASCII)。
- 执行装入文件和 Informix 模式之间的映像。
- 执行有选择的装入。
- 通过浏览器观察和装入文件。

有两种不同的方式使得 DBA 可以决定那些通常的装入任务(如检查参照完整性,日志建立索引等)是在数据装入的过程中进行,还是在装入完成之后进行(这样可以加快装入的速度)。

4. 并行的索引建立

使用 OnLine 动态服务器,SQL 的 CREATE INDEX 语句比使用基于系列的结构要快得多。这是因为算法的改进和并行地建立索引的能力。

在一个并行的索引建立过程中,OnLine 动态服务器对可用的数据进行采样,并确定扫描它所需的线程的数目。这些数据在可能的情况下使用提前读被并行地扫描进来,之后被放在共享存储器“bin”中。然后,这些“bin”被并行地排序,并行地建立排好序的子任务中的子树,最后并入最终的索引中。使用 OnLine 动态服务器,排序线程无需等到扫描线程结束就可以开始另一个排序。与此同时,索引建立线程不必等到排序结束就可以开始建立索引子树——所有这些都是同时并行进行的。实际上,基于数据量、使用的磁盘的数目和可用的存储器,这种并行机制与顺序的索引建立相比较,最多可以十倍地提高索引建立的性能。

5. 并行的数据查询

随着硬件发展到对称多处理器,松散耦合的聚集,以及大规模并行处理器结构,PDQ 将充分利用做为基础的 CPU 的优势,以比以往的数据结构快若干倍的速度去执行任务。在某些情况下,原来需要数小时的查询可以在数分钟内完成。这种性能上的收益适用于所有那些需要排序、扫描大量数据、进行表连接和执行聚合运算的复杂的数据库操作。

例如,让我们看一个简单的连接。顺序地进行连接是,用户必须等待,数据库完成了一个任务才能开始下一个任务。换一句话说,数据库必须首先扫描,然后连接,然后排序,然后将结果送给用户,PDQ 在处理以上两个方面获得了节省。首先,它并行地同时处理这些任务;另外,它将各个任务分解为子任务,充分利用 DSA 内部构造的核心并行机制,远远节省了总的处理时间。

6. 并行排序

排序是建立索引。排序——合并连接,SQL 中的 ORDER BY 查询等常用数据库作所需的基本活动。排序速度的提高将对许多应用程序的性能提高发挥作用。

并行查询将一个清单分解为多个节,并将每一节送到一个处理器上处理。按邮政编码去排序一个长长的邮递清单,或按生产成本将一个长长的零件号码清单重新排序,这都是典型的能够从并行排序中直接得到好处的任务。

实际上,排序操作的速度受表中扫描数据所需时间的限制。这一限制由于 PDQ 的另一

个特性——并行扫描而大大缩减了。

7. 并行扫描

在当今大多数的应用中,当在大型数据库上工作时,报表、连接、索引建立都需要很长时间的扫描操作。Informix 的 PDQ 技术大大缩减了扫描时间。表分割的作用是将表分成几个逻辑片断,并将它们分布到几个硬盘上,于是多个处理器可以并行地扫描这些硬盘,所获得的性能增长与磁盘的数目成比例。对于顺序表或索引扫描,OnLine 动态服务器可以设置成在当前页正被处理时异步地提前读入若干页;这样就可以维持较大的吞吐量,因为应用系统花在等待磁盘完成存取操作上的时间减少了。

8. 并行连接

复杂的查询通常需要从两个或更多个表中检索数据。然后将两个表中的记录进行匹配,并将结果返回给请求者。这一任务称做连接。DSA 采用先进的算法,让多个处理器去扫描数据,然后并行地连接几个片段,从而使连接操作并行地执行。

9. 并行聚合

复杂查询可能经常需要总和或聚合。由于 OnLine 动态服务器具有许多并行处理能力,包括 GROUP BY 命令,SUM,AVERAGE,MIN,MAX 等聚合函数的查询操作执行起来要快得多。

10. 内存授权管理器

内存授权管理器(MGM)使得 DBA 和程序员能够对于用户对可利用的系统资源请求的优先级进行平衡,从而控制并行的程度。MGM 执行下述任务:

- 调整 PDQ 类任务所需的系统资源数量。
- 设置每个查询的优先级。
- 调整可以同时运行的复杂查询的数目。
- 调整决策支持和 OLTP 使用的最大的内存量。
- 与基于代价的优化器协同工作,以保证各个层次上的最大的并发度。

MGM 的高效的内存空间利用是优化并行处理的关键。MGM 还强制 PDQ 处理的聚合资源界限。使用了 MGM,整个系统范围的优先级设置都可以动态地改变。

11. 表分割

正如前面所提到的,表分割可以提高并行处理性能和高可用性。通过将数据分解为较小的部分,使非常大的数据库变得易于管理了。

OnLine 动态服务器使得不中断数据库服务器就能设置表的分割和改变数据的分布。而且,分割模式对于应用系统和最终的用户是透明的,分割模式的设置可以采用简单轮循法(顺序地将每一个记录分配到下一个分片中),散列法(将一个算法作用到记录的关键码上,以确定它的分片号),或表达式法(基于关键码的,将一组记录分配到一个分片中)。分割模

式的设置通过 CREATE TABLE 和 ALTER TABLE 这样的 SQL 语句,必要时可对分片进行修改和调整。

要获得并行处理的好处,很关键的是要具有将单个的表分片(这是造成系统瓶颈的地方)划分为两个或三个新的表分片的能力。例如,由于数据库了解数据分片的模式,数据库服务器就知道哪些分片与特定的查询无关,从而可以跳过它们。同时,数据库服务器可以跳过由于系统故障而不可用的那些分片,以保持高可用性。

除了将数据表分割之外,OnLine 动态服务器还可以将索引分割,以获得最大的数据灵活性,从而达到最优的并行处理性能。DBA 可以把分割了的索引放到与数据不同的分片上去。分割了的索引可以有自己的分片模式,与数据表相分离。

分割对于有效的系统管理也是很关键的。数据库越大,就越需要系统管理员能够在表分片级进行备份和恢复,批量装入和卸出等操作,而不是必须备份或恢复整个数据库或表。这些操作可以并行地进行,从而大大地减少了装入、卸出和恢复数据所需的时间。

12. 基于代价的优化器

OnLine 动态服务器是基于代价的优化器,基于表的列中数据分布的详细信息可以自动确定从数据库表中检索数据的最快方法。优化器收集和计算有关数据分布的统计情况,并选取对资源影响最小的返回路径——在某些情况下,这可能是一个并行的返回路径。但是在其他情况下,可能是顺序处理。要控制并行的程度所需的是内存授权管理器。

13. 磁盘镜像

OnLine 动态服务器使用镜像来提高读的性能,因为数据的两个版本存放在不同的硬盘上。数据页面或者从主 chunk(组块)读入,这由哪半个 chunk 中包含数据页面的地址来决定。这称做分裂读。由于减少了磁盘搜索的时间,所以分裂读能提高性能。磁盘搜索时间的减少是由于磁头移动的最大距离减半了。对主块和镜像块写操作的异步提高,导致性能进一步改进。这些写操作不是顺序地而是并行地完成的。

14. 控制/管理功能

使用 OnLine 动态服务器,大多数数据库管理特性被完全并行,以提供大型主机能力的性能。

15. 系统监控界面(SMI)

当 OnLine 动态服务器第一次初始化时,将创建一个主 SMI 数据库,它包含能够跟踪下列类型的信息的表:

- 数据库名、所有者、日志状态。
- 正在等待数据库资源的用户状态。
- 性能概要信息。
- 用户和虚拟处理器使用系统 CPU 的情况。
- 有关磁盘空间的信息。