

软件集成专家

Experiences & Focus 经验和专长

基于Informix、UNIX、
Client/Server开发工具的
培训、信息系统战略
规划、系统分析与设
计、性能优化、开发
指导、资源外包、软
件集成、高档数据库、
Client/Server计算、面
向对象的开发方法、
大型应用系统的开发
管理、分布式计算、
Internet应用

Business Scope 业务范围

高级培训

系统集成

顾问咨询

增值再销

应用开发

Advanced Training

System Integration

Consulting

Value-added Reselling

Application Development



96年度Informix
最佳代理商



Solaris Base Server
中国唯一总代理



北京融海诚信咨询有限公司
System Consulting & Services (Ronghai), Ltd.

地址：北京市海淀区学院路甲38号长城电脑大厦四层B405/B407室
B405/B407, Great Wall Computer Building, No. A 38 Xueyuan Road, HaiDian District, Beijing China
电话：(8610)62354938/39/40/41/42/43 传真：(8610)62355002
E-mail: consult@scs.cn.net 邮编：100083



融海咨询
RongHai Consulting

INFORMIX-OnLine动态服务器 性能调整

北京融海诚信咨询有限公司

顾问咨询部

目标

当本课程结束时，你应该能够：

- 进行性能分析；
- 使用 INFORMIX-OnLine 动态服务器的系统监控工具；
- 建立数据分布；
- 采取合适的数据分割策略；
- 优化和管理 PDQ 的资源；
- 了解数据装入、索引和备份操作的原则
- 管理内存和 CPU 的系统资源。

课程要求

为使本课程收到最好的效果，我们要求学员具备以下知识：

- 《INFORMIX-OnLine 动态服务器系统管理》或相应知识；
- 使用 INFORMIX-OnLine 动态服务器数据库管理的经验。

产品版本

本手册包括 UNIX 上的 Informix-OnLine 动态服务器 7.10 版本。

注释

感谢你参加本次培训班！

如需订购产品资料，或获取关于 Informix 培训的进一步信息，请与北京融海诚信咨询有限公司顾问咨询部联系。

电话：(010)62354938/39/40/41/42/43

传真：(010)62355002

章节

1. 性能调整介绍
2. 性能的要素
3. 衡量当前环境
4. 磁盘布局
5. 数据分割策略和原则
6. 调整 I/O 性能
7. 数据库性能策略
8. 处理器和操作系统
9. 内存使用
10. Client/Server 和分布式通讯
11. PDQ 管理
12. 大系统中的性能问题
13. 性能调整研究实例
14. 性能调整研究实例：调整的实例

Chapter

1

INFORMIX-OnLine 动态服务器 性能调整介绍

学完本章后，你将能够：

- 列出可能与性能问题有关的各个部分。

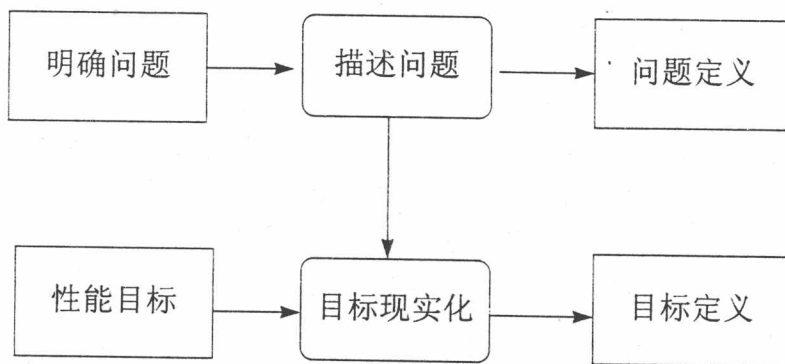
性能调整概述

提出性能调整要求的问题:

- 数据库和应用设计: 如何设计我们的数据库/应用, 以得到最佳的性能?
- 用户的抱怨: 我的交易太慢了!
- 系统容量: 如果我们的数据量增加50%, 现有的硬件和软件是否能够承受?

性能调整往往由一个要解决的问题而引发, 这个问题可能是用户或 MIS 人员的抱怨, 也可能是关于应用或数据库的设计的决策、硬件和软件的需求, 或者其他的容量规划问题。

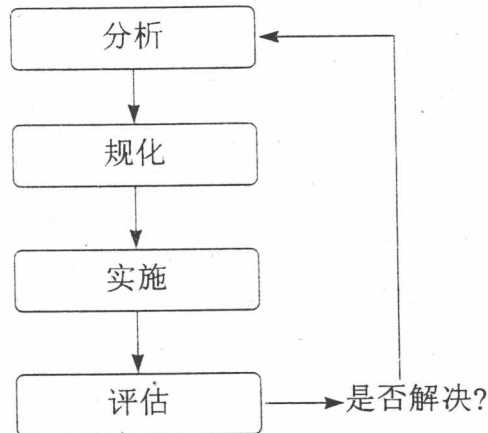
解决问题的模型



当接到有关性能问题的报告时首要的步骤是确认所反映的是问题的本质。经常会出现这样的情况，在非本质的问题上花费的许多人力物力，到头来没有解决问题。不少管理员都有过这样的经验，更换了一个速度更快的 CPU 后才发现真正的问题是出在磁盘配置不足。所以，首要的问题是研究并定义问题的本质。

“反应时间太慢了，它要两秒钟。我想它应在一秒内反应。”给定要执行的工作量后，两秒也许是能达到的很好的指标。问题的本质是对性能的期望过高而不是真正的性能问题。所以，关键是确定实际的性能目标，并与有关的用户进行沟通讨论。

解决问题的模型(续)



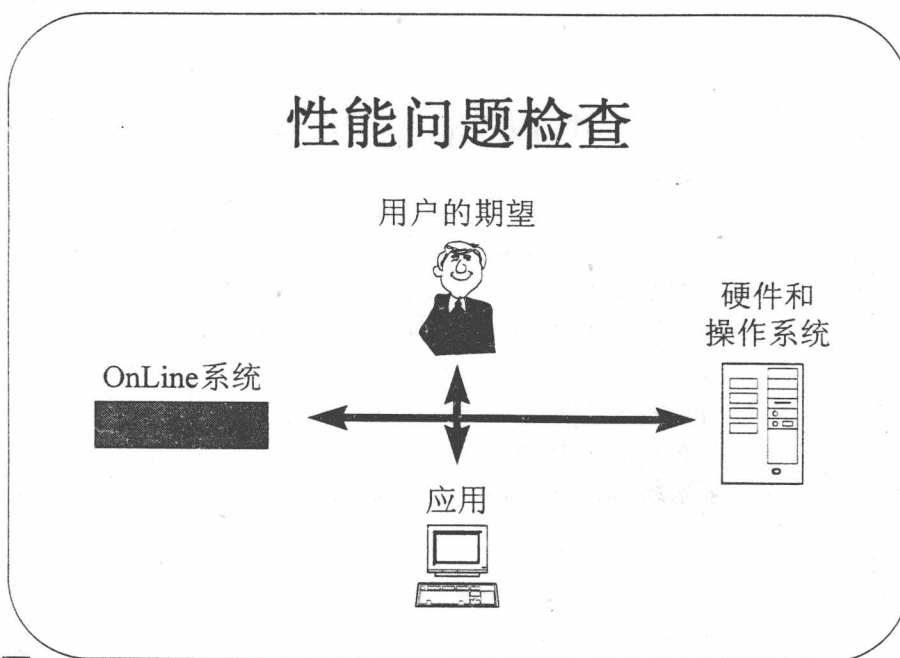
分析阶段以测量并报告组成部分的使用情况而告开始。应在系统的组成部分这一层次应用精确且合适的工具来收集性能数据。决不要使用道听途说的信息或来源不清的数据。

要估计并规划性能调整的目标。要把组成部分的性能和已知标准想比较。要确定那些最有助于提高性能的资源。

完成调整的各项动作。一次只集中注意一项资源。为提高给定资源的性能，建立一系列的调整步骤。确认按计划实施这些步骤。

对调整步骤的结果进行评测。把调整后的情况与先前定义的目标相比较。目标达到，调整也就完成了。如果目标没有达到，应从分析步骤入手重新开始调整过程。

性能问题检查



性能问题可能来自多个方面：

- 用户的期望。如果用户的期望过高，以至于最优情况下的实际性能都不能满足要求，那么管理员什么都不能做。关键是要是用户对性能的期望尽量与实际的可能相吻合。
- 硬件和操作系统。这一大类包括磁盘、操作系统、处理器和网络。对硬件和操作系统进行评测特别重要。
- 应用。应用优化是一个常被忽略的方面，当 OnLine 和系统管理员与应用设计编程人员是不同的人时尤其如此。管理员应确认应用开发人员所写的程序性能是优化的。
- OnLine 系统。最后，必须监控 OnLine 系统，确保其优化运行。

本课程的内容

在本课程中将要介绍:

- 硬件和操作系统性能调整的要点
- INFORMIX-OnLine 动态服务器性能调整

以下内容不包括在本课程之内:

- 数据库和应用设计要点
- 操作系统和硬件的深入性能调整

这门课仅讨论了整个性能调整领域的一部分。性能调整是一个广阔而复杂的领域。

这门课程讨论以下题目:

- 硬件和操作系统调整的概览
- INFORMIX-OnLine 动态服务器性能调整

然而, 本课程不涉及以下题目:

- 数据库和应用设计的问题。 Informix 提供其它课程深入研究这些问题。

操作系统和硬件调整的深入的概念。应从硬件供应商处得到关于硬件环境调整的详细信息。大多数硬件供应商提供特定的性能调整课程。

Chapter

2

**INFORMIX-OnLine 动态服务器
性能的要 素**

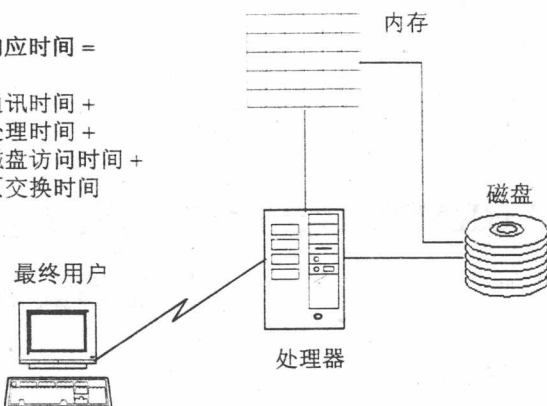
本章的目标是理解对应用的性能的影响因素。要提高性能，必须对所使用的系统资源以及它们各自如何影响应用的运作有所认识。本章中，我们将定义性能的要 素和测量单位。学完本章后，你将能够：

- 把应用的性能和系统资源联系起来；
- 描述影响性能的要 素；
- 认识与性能有关的 UNIX 和 INFORMIX 测量工具。

性能的要素

响应时间 =

通讯时间 +
处理时间 +
磁盘访问时间 +
页交换时间



运行给定的应用所花费的时间，或称反应时间，基于以下活动各自所需花费的时间：

- 通讯时间是消息往来于用户终端和服务器系统间所花费的时间之和。它不包括任何服务器系统中的处理时间。
- 处理时间是应用执行机器指令所花费的时间加上等待可用的CPU资源的时间。
- I/O 时间是磁盘 I/O 时间所持续的时间，不包括换页 I/O 。
- 换页时间是应用处于页等待状态的时间之和。换页直接与系统中内存的使用有关。

CPU要素

影响性能的因素:

- 指令执行的速度
- 处理时间:
 - 用户态时间(user state time)
 - 系统态时间(system state time)
- 分配策略
- CPU利用率

$$\text{CPU日用率} = \text{CPU繁忙时间} / \text{过去的时间} \times 100$$

CPU 要素控制了整个应用过程。

指令执行速度

无须多言，内部的处理速度极大地影响着应用的性能。有多种技术可用于评测 CPU 间的相对速度。描述内部 CPU 速度的最常用的单位是每秒百万条指令数(MIPS)、每秒百万个浮点操作数(MFOLPS)、每秒 Dhrystones 数、每秒 Whetstones 数和每秒 TPC 交易数等。要清楚一点，每种测量都集中针对很窄的领域，也许并不代表你要做的工作。

用户态时间

用户态时间是 CPU 以用户态运行应用程序所花费的时间的总和。它是在程序员的控制之下的，反映的程序语言的有效性和程序员的效率。

系统态时间

系统态 CPU 时间是 CPU 代表应用程序执行系统(核心)代码的时间

间之和。这包括了为应用程序执行系统调用和执行管理功能的时间。

调度优先权

调度优先权是每个 UNIX 进程的优先权。当 CPU 空闲时, UNIX 必须决定下面运行哪一个进程。准备运行的进程中优先权最高的将获得 CPU 控制权。系统公平地调度, 高优先权的任务比低优先权的任务完成得快。

调度算法

调度算法是操作系统用于为各个进程提供 CPU 的受控存取的技术。调度算法的例子如:

- 轮转法--各个进程轮流获得 CPU。
- 时间片法--每个进程获得一定的 CPU 时间。分配的时间用完后, 进程被中断, 调度另一个进程运行。
- 抢占法--根据优先权调度进程。把 CPU 给予具有最高优先权的任务, 直到它等待一个 I/O 操作或自愿等待。然后启动准备运行队列中下一个具有相等或较低优先权的进程。它一直运行直到有一个具有较高优先权的任务准备运行(例如, I/O 完成了)。低优先权任务的控制权被抢占, 控制交给高优先权任务。

UNIX 操作系统使用了以上各种算法。它是一个分时体系, 系统尽量在各个进程间平等地分配 CPU 时间。然而, 每个进程都被赋予一个优先权。高优先权的进程比低优先权的进程获得更多的注意, 但即使是优先权最低的进程也能得到一些 CPU 时间。具有相同级别优先权的进程间以轮转方式调度。

CPU 利用率

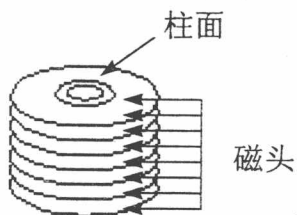
CPU 利用率是 CPU 忙(无论处于用户态或系统态)的时间比例。通常, 只有当没有要执行的工作或所有进程都在等待 I/O 的完成时, CPU 才会空闲。

CPU 利用率等于 CPU 忙的时间除以总的时间。如果在 100 秒的时间间隔中 CPU 忙碌了 30 秒, 利用率为 30%。

随着 CPU 利用率(或其它任意设备的利用率)的上升, 对一个请求进行服务的总的时间也会急剧上升。

磁盘要素

访问时间 = 寻道时间 + 延迟时间 + 传输时间



寻道时间 - 磁头移动相应柱面的时间

延迟时间 - 等待所需数据旋转到磁头下面的时间

传输时间 - 向磁盘读或写数据的时间

一个磁盘由盘面组成，通过磁盘头来存取盘面。磁盘头在内外磁道间往复移动。这种移动被称为寻道，磁头移动到一个磁道上的时间称为寻道时间。另外，盘面高速旋转，磁头不动，等待放有所需数据的扇区旋转到磁头下。这段时间称为等待时间。数据从磁盘传送出来的速度用传输率来衡量。某些厂商把传输率从寻道时间和等待时间等其他存取时间要素中分离出来。大多数磁盘厂商给出平均寻道时间和等待时间的统计数据。

物理位置

使寻道时间最小的一种方法是把常用数据安排放置在中间的磁道(磁盘的中心)。当用操作系统工具创建一个原始设备时，通常可以选择指定所使用的磁道号。

内存要素

影响性能的因素:

- 实际内存的需求量
- 换页率
- 对换操作
- I/O的访问时间

对给定的程序，工作集尺寸是指为了完成它的处理而必须获得的实际内存总和。工作集不一定与程序可执行模块的尺寸或分配给进程的共享内存总量相对应。如果一个程序主要由错误处理子程序组成，主模块很小，工作集也可能很小。

换页率

当前正在执行的程序的工作集超过实际可用内存时，将用当前正在执行的进程所需的信息来替代空闲进程的内存页。所需的信息存放在磁盘上，需要读磁盘检索数据。而且，如果内存中选定用以接收页以前曾被空闲程序修改过，修改后的数据必须首先写回到磁盘中，引起一次磁盘写操作。需要这样的页的频率称为换页率。

对换

如果所需的实际内存远大于系统中可用的内存总量时，操作系统会开始把整个进程对换到磁盘上。一般而言，系统中发生对换是严重内存短缺的一个标志。

换页 I/O 存取时间

频繁地换页导致磁盘 I/O 率的升高。当发生大量换页时，磁盘的速度对整个系统性能产生显著的影响。