

# 高负载环境下 INFORMIX 的调优

崔颖安

(西安理工大学计算机科学与工程学院, 西安 710048)

**摘要:** INFORMIX 是国内金融行业使用最多的大型数据库之一, 其配置参数对系统性能有重要影响。该文针对某省商业银行数据仓库系统性能存在的问题, 通过对 INFORMIX 性能监测, 确定系统瓶颈, 并在此基础上, 给出了在虚拟处理器、共享内存、查询优化器、数据抽取方面的优化措施。实践表明, 通过调优使系统性能得到了明显改善。该文中提出的分析与处理问题的方法具有一定的通用性, 所采取的优化策略对类似系统具有借鉴意义。

**关键词:** INFORMIX 数据库; 性能监控; 系统优化

## Performance Tuning for High-load INFORMIX

CUI Ying-an

(School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048)

**【Abstract】** INFORMIX is one of the most popular database in finance industry in China. Its configure parameters have important influence on its performance. According the result of inspect INFORMIX performance on the data warehouse system of the commercial bank, some faults in performance of INFORMIX data warehouse system are found out through a long time monitor. The paper takes some measures of system optimization in virtual process, share memory, and data extraction. The performance of the system is improved by optimization, because the method of analysis and process question is universality, and it is worthy of reference to similar system.

**【Key words】** INFORMIX; performance monitor; system optimization

某省商业银行完成全省数据大集中以后, 建成了银行数据仓库系统。该系统是日常使用频率最高的 IT 系统之一, 它部署在 Linux 操作系统上, 采用 INFORMIX 数据库。其功能是从综合业务系统、本级财务系统、信贷管理系统、现代化支付系统等多个生产系统中抽取业务数据, 经过数据清洗、转换后装入数据仓库, 供全省各业务部门使用。

### 1 系统存在的主要问题

近 2 年随着数据规模的不断扩大, 系统暴露出以下问题:

#### (1) 数据抽取效率低下

该系统所有数据都源于其他生产系统。随着数据规模的不断增大, 数据抽取效率明显下降, 抽取时间平均耗费 6 h 左右, 有时甚至超过 8 h, 直接影响次日的正常使用。

#### (2) 并发处理能力不足

在日常使用过程中, 如果并发访问用户超过 300 人, 系统性能急剧下降, 尤其在每日 9 时~16 时的使用高峰期, 性能问题更为突出; 系统长时间保持在高负载状态, 运行缓慢, 频繁出现各种数据库异常错误。

#### (3) 部分功能无法正常使用

系统中部分统计功能流程复杂, 数据计算量大, 即使在空载情况下, 也需要 20 min~30 min 才能完成统计。当多个类似功能在日间同时启动时, 系统资源十分紧张, 尤其在月末或年终时, 有些统计功能根本无法正常使用。

#### (4) 出现不规律的运行错误

在系统运行过程中, 数据库经常出现各种异常错误, 如索引失效、部分 SQL 语句非正常运行失败、库表数据损坏等。最严重时, 数据库由 online 模式转成 quisent 模式, 数据库服务器异常宕机。

## 2 系统性能诊断

### 2.1 性能监控方案

为了确定性能瓶颈, 笔者制定了全面的监控方案, 为取得必要的实测数据, 对系统进行了长达 3 个月的跟踪。跟踪时段除正常工作日以外, 对月末、结息日、年末以及其他特殊时段也进行了监控, 监控内容包括如下 4 个方面<sup>[1]</sup>:

(1) 并发性能, 如下所示:

1	onstat -k	输出 informix 锁信息
2	onstat -u	输出 informix 用户线程状态
3	onstat -g glo	输出每个 VP 的运行信息
4	onstat -g act	输出活动的线程信息
5	onstat -g wai	输出正在等待处理的线程

(2) 共享内存性能, 如下所示:

1	onstat -R	输出 LRU 信息
2	onstat -g mem	输出缓冲池统计
3	onstat -g seg	输出共享内存段的统计
4	onstat -b	输出当前 BUFFER 的使用统计

(3) I/O 性能, 如下所示:

1	onstat -D	输出 dbspace 和详细的 chunk 读写统计数据
2	onstat -l	输出日志信息
3	onstat -g ioq	输出按队列统计的磁盘 I/O
4	onstat -g iov	输出按 chunk 统计的磁盘 I/O
5	onstat -g ioF	输出按 CPU VP 统计的磁盘 I/O

(4) 操作系统性能, 如下所示:

1	Sar -u	CPU 资源使用统计
2	Sar -d	硬盘活动信息统计
4	Sar -w	输出内存交换信息

**作者简介:** 崔颖安(1975 -), 男, 讲师, 主研方向: 面向服务的分布式智能计算, 金融电子化

**收稿日期:** 2007-05-25 **E-mail:** feixiang@tom.com

通过对监控数据的整理，发现系统存在的主要问题是：在业务高峰期，系统 Cache 命中率低，线程上下文切换频繁，等待线程与活动线程数居高不下；在数据抽取过程中，存在大量日志信息的频繁 I/O 并伴有锁资源不足；与此同时，对处理器、内存、硬盘等资源也进行了监控，观察到 CPU 空闲率几乎为 0，处理器资源长期被应用进程占据，内存开销明显；业务负荷较低时，共享内存仍然出现 LRU 队列失效，频繁出现缓冲区数据溢出，Cache 命中率低于 95%，AIO 队列与高峰期时几乎没有差异，基本都为 2；硬盘使用正常，未见 I/O 异常。

## 2.2 诊断分析

经过分析，可以确定系统存在问题的主要根源是虚拟处理器(Virtual Process, VP)划分不合理和共享内存不足。由于共享内存资源不足，导致频繁的 I/O 交换；由于 VP 划分不合理，CPU VP 数量不能满足并行处理的要求，这样就形成了 VP 与 SHM 互相拖累的局面，并且影响外部 SAN 存储性能发挥。此外根据对测试数据的分析，发现系统还存在以下问题：(1)数据库建库方式不当，频繁的日志 I/O 使数据抽取性能下降；(2)数据装载策略欠佳，导致装载效率低下；(3)查询机制欠优，影响系统性能的提高。

## 3 系统调优方案

本系统是一个典型的 DSS 系统，根据业务特点和硬件资源状况，主要从以下几个方面进行调优：

### (1)并行处理能力优化

INFORMIX 数据库采用多进程多线程服务机制，它将每一个进程根据用户定义分解成若干线程，用线程响应用户的访问请求。这种服务方式与传统的多进程数据库相比有着明显的优势，一方面它能够在操作系统之上建立起更加符合数据库特性的任务调度方法，实现 CPU 的负载均衡；另外还能够用有限的进程来满足系统并发处理的要求，可以有效地降低系统负荷，提高系统的可伸缩性。

在INFORMIX中每一个oninit服务进程都被称作一个VP。若干个相同功能的虚拟处理器组成一个虚拟处理器类(Virtual Process Class, VP Class)，每一个VP Class都表示一种功能的虚拟处理器，这些进程长驻内存，主要作用如表 1 所示<sup>[1]</sup>。

表 1 VP Class 功能说明

VP 名	类别	说明	备注
CPU	Central Processing	负责处理所有会话线程及 KAIO 管理	数量可配置
PIO	Disk/Io	负责写物理日志	数量不可配
LIO	Disk/Io	负责写逻辑日志	数量不可配
AIO	Disk/Io	负责写业务数据	数量可配置
SHM	NetWork	负责共享内存通信	数量可配置
TLI	NetWork	负责传输层接口方式通信	数量可配置
SOC	NetWork	负责通过套接字通信	数量可配置
OPT	Optical	负责到光盘库的 I/O 调度	数量不可配
ADM	Administrative	负责管理功能	数量不可配
ADT	AUDITING	负责审计功能	数量不可配
MSC	Miscellaneous	负责系统调用服务	数量不可配

显然其中部分VP可以由用户根据具体业务进行合理的配置。该系统服务器共有 8 颗CPU，是一个SMP系统。原配置文件中对NUMCPUVPS和NUMAIOVP平均分配，现根据诊断情况进行调整，从而使承担并行处理的CPU资源更为宽裕：

```
SET NUMCPUVPS=6
```

```
SET SINGLE_CPU_VP=0
SET MULTIPROCESSOR=1
SET NUMAIOVP =2
SET NOAGE=0
```

### (2)共享内存优化

共享内存是INFORMIX的重要组成部分，包括驻留部分、虚拟部分、消息部分。驻留部分用于系统数据、磁盘数据在内存中的缓冲，虚拟部分用于VP和缓冲区的管理，消息部分用于应用进程与ONLINE系统的通信<sup>[2]</sup>。使用共享内存 在提升系统性能方面有很多好处：可使所有的数据库服务进程均共享同一内存区域，通过共享内存实现高效的进程间通信，提高系统的并发性；另外还能提高内存的使用效率，减少操作系统内存的调度压力。调整共享内存需要深刻理解驻留部分与消息部分的静态特性，另外还要按照一定规则与经验计算共享内存的各部分容量：

$$BUFFER\_VALUE=(Buffers \times Pagesize)+Buffers)/\alpha;$$

$$LOCKS\_VALUE=LOCK \times 64/\alpha$$

$$LOGBUFF\_VALUE=Logbuff \times 1024 \times 6$$

$$PHYSBUFF\_VALUE=Physbuff \times 1024 \times 3$$

$$REGSIZE=(BUFFER\_VALUE+LOCKS\_VALUE+$$

$$LOGBUFF\_VALUE+PHYSBUFF\_VALUE+51200)/\alpha$$

其中， $\alpha$  是一个功率参数，其数值主要由 CPU 的特点决定，取值范围在[0-1]，不同系统对  $\alpha$  取值的方法不同(X86 芯片与 RISC 芯片  $\alpha$  取值不同)，在本系统中经过多次试验调整，确定  $\alpha$  为 0.7。

根据需要，将共享内存参数的设置修改如下：

```
SET LOCKS=800000
SET BUFFERS=6120000
SET SHMADD=81920
SET SHMVIRTSIZE=12800
SET LRUS=60
```

### (3)数据抽取方法优化

从实时监测的分析结果来看，数据抽取效率低下的主要原因是日志I/O和索引排序耗时过高，经过多次对实际数据的试验，笔者对数据库建库方式和数据装载策略进行了调整。首先将数据库由缓冲日志建库模式改成无日志模式<sup>[3]</sup>，修改方式如下：

```
Create database bankdss withno log;
```

其次在进行数据装载以前，先删除表的各项索引，而后装载数据；在数据装载完成以后根据应用需要建立簇类索引。从而实现数据的最优存储，提高系统使用效率。方式如下：

```
Drop index safmx: depositmxidx;
Create cluster index depositmxidx on safmx (zh, jysj, jym)
Update statics low //每日抽取完成后进行
Update statics high //每个季度最后一日进行
```

### (4)查询优化器的使用

通常的查询一次是将所有查询结果都输出到共享内存缓冲区中，系统开销很大，资源消耗严重，然而实践表明，大部分用户往往关注的只是最初几屏输出的内容，有时甚至只关注最初几条记录(如本月的经营业绩或者昨日的业务统计等)。因此，在进行查询数据输出时，可以利用优化器选择一条查询路径，使其只输出填满一个缓冲区的记录数，如果用户继续查询则继续执行，这样就避免了不必要的查询和输出，极大地提高查询速度<sup>[4]</sup>。其参数设置如下：

```
Set Optimization= First_Rows; (下转第 110 页)
```