服务器知识手册（二十）

狄登峰 主编

需要全本请访问 www.ertongbook.com

目　　录

数据库应用.....1
INTOramix 的监控和管理命令 1
快速装载 MYSQL 数据库 11
如何恢复系统数据库 12
数据仓库更新的新策略--工作流 14
自动备份 Oracle 数据库 33
用 Profiler 监测 Web 应用 38
Oeacle 数据库备份技术 46
Oracle 回滚段的概念、用法和规划及问题解决 55
Oracle 性能究极优化 63
优化 SQLServer 数据库服务器的内存配置 69
Foxpro 到 MySQL 的数据转换技术介绍 86
架设 freeradius+mysql 的 radius 服务器 97
SQLServer2000 简介 100
如何增加 Oracle 连接数 104
使用 Sybase 临时数据库提高性能 104
在 UnixWare7.1.1 上安装 Oracle8i 107
Unix 下数据库的通用接口函数 111
数据库体系结构：存储引擎 120

数据库应用

INTOramix 的监控和管理命令

监控 ONLINE 系统后动情况的工具主要有以下三类：

系统监控接口（SMI）

tbstat

tbcheck

一、系统监控接口（SMI）：

我们主要通过 SQL 命令操作 online 的内部数据库 sysmaster 中的内部表/结构，来获取有关的维护信息。Sysmaster 是在 online 初次初始化时，系统自动创建的。它实际主要存储了一些数据结构，而不是真正的表。使用 SMI 有如下限制：

不能对 SMI 中的表加锁或使用隔离级别。

不允许使用 insert，delete，update 等语句（只读）

不能使用 dbschema，dbexport 等命令

使用 selectrowid 语句将会产生不可预料的结果

主要的 SMI 表有：

sysdatabases：online 中的数据库信息

systabnames：某数据库中所有表的信息

syslogs：逻辑日志信息

sysdbspaces：数据库信息

syschunks，syslocks 等

例 1：显示处于脱机（offline）**状态的** chunk **的序号和所在数据库空间**

Selectchknum，dbsnumfromsyschunkswhereis_offline =1ormisline=!

例二：显示满 chunk **的信息**

Selectchknum，dbsnumfromsyschunkswherenfree=0

二、TBSTAT

列出当前时刻的信息（**实际也是读取** SMI **表**）

不需要磁盘 I/O

不需要锁等系统资源，因此不会影响系统性能

用法：tbstat[-abcdklmpstuzBDFPRX][-rseconds][-ofile][infile]

-aprintallinfo(options:bcdklmpstu)

-bprintbuffers（**缓冲区**）

-cprintconfigurationfile（**配置文件**）

-dprintdbspacesandchunks（dbspace **和** chunk）

-kprintlocks（**锁**）

-lprintlogging（**日志**）

-mprintmessagelog（**消息日志**）

-pprintprofile（profile **文件**）

-sprintlatches（**门闸**）

-tprinttblspaces（**表空间**）

-uprintusers（**用户**）

-zzeroprofilecounts

-Bprintallbuffers

-Dprintdbspacesanddetailedchunkstats

-Fprintpageflushers（**页刷新进程**）

-Pprintprofile,includingBIGreads

-RprintLRUqueues（LRU 队列）

-Xprintentirelistofsharersandwaitersforbuffers

-rrepeatoptionseverynseconds(default:5)

-oputsharedmemoryintospecifiedfile(default:tbstat.out)

infileuseinfiletoobtainsharedmemoryinformation

三、几个常用的 tbstat 选项：

tbstat-m：**显示消息日志的最后** 20 **行**.

消息日志的内容包括:

1)、**检查点信息**

2)、**读写错误信息**

3)、ONLINE **模式转换信息**

4)、**长事务**

5)、**日志文件满**(LOGFILEFULL)

若想显示完整信息,可直接编译消息日志文件.

Tbstat-d：**磁盘空间的使用情况，包括** DBSPACE **和** CHUNK **的信息**

例：RSAMVersion5.03.UC1--On-Line--Up09:45:41--816Kbytes

Dbspaces

addressnumberflagsfchunknchunksflagsownername

8040a2441111Ninformixrootdbs

1active,8total

Chunks

addresschk/dbsoffsetsizefreebpagesflagspathname

80409d84110300000231871PO-/dev/rdata

1active,8total

其中的 FREE **项，显示了该** CHUNK **的空闲空间大**

需要全本请访问 www.ertongbook.com

小(Kbytes).

Tbstat-l：**日志文件情况**

PhysicalLogging

Bufferbufusedbufsizenumpagesnumwritspages/io

P-2016000.00

phybeginphysizephyposphyused%used

1017821500096000.00

LogicalLogging

Bufferbufusedbufsizenumrecsnumpagesnumwritsrecs/pagespages/io

L-20161111.01.0

addressnumberflagsuniqidbeginsizeused%used

8042de941U---C-L110521a75006308.40

8042deb02F------0106f66750000.00

8042decc3F------0108cb2750000.00

8042dee84F------010a9fe750000.00

8042df045F------010c74a750000.00

8042df206F------010e496750000.00

其中：%USED：**使用百分比**

FLAGS **字段的含义**:

F:**空闲** B:**已备份** C:**正在接收事物记录**

U:**正在使用** A:**新增日志** L:**包含最后一个检查点**

Tbstat-u：ONLINE **的用户情况**

Users

addressflagspiduserttywaittoutlocksnreadsnwrites

804019f4------D329rootconsole0001792

80401a64------D0rootconsole00000

80401ad4------F330root00000

3active,20total

Transactions

addressflagsuserlockslogbeginisolationretryscoordinator

804022b4A----804019f400NOTRANS0

804028d8A----80401a6400NOTRANS0

2active,20total

其中：flages **字段的含义**:

第一列：（S：**等待** mutex；Y：**等待条件**；L：**等待锁**；B：**等待缓冲区**；

C：**等待检查点**；X：**长事务清理**；G：**等待长缓冲写**；T：**等待事务**）

第二列：（*：**事务执行时，发生** I/O **错误**）

第三列：（A：**正在备份**；B：**操作已被记录在日志中**；P：**分布处理已准备好**；

C：**正在提交**；R：**正在回滚**）

第四列：（P：**会话的主线索**）

第五列：（R：**在** readrsam **调用中**；X：**进程在关键分区**）

第七列：（M：**特殊监控**；D：**特殊线索**；C：**清理线索**；F：**特殊清页进程**；

B：**特殊** B+**树清页线索**）

Tbstat-k：**用户持有锁的情况**

锁按照粒度分为 6 **种**:**库锁、表锁、页锁、行锁、字节锁、键锁字节锁**：**更新包含有** VARCHAR **类型的行时，加在该行上的锁**。

键锁：用于索引树上的锁。一般在相应字节上加删除标志。

这几种锁的识别如下：

TYPETBLSNUMROWIDSIZE

库锁 HDR+X10000022050

表锁 100000e00

页锁 1000

行锁 101

字节锁 HDR+B909

键锁很大的 16 **进制数**

锁的生命期：

1、**不使用事务时，操作完成即会释放对应的锁**；

2、**若使用事务，锁将伴随整个事务**。(**即随这** COMMITWORK **或** ROLLBACKWORK **而释放**)

（**)**锁的几种类型：**共享锁**(SHARE)，**排它锁**(EXCLUSIVE)，**可升级锁****）

tbstat-p：**显示系统资源状况(是否充分)**

Profile

dskreadspagreadsbufreads%cacheddskwritspagwritsbufwrits%cached

18119935849.447820.00

isamtotopenstartreadwriterewritedeletecommitrollbk

311000010

ovtblsovlockovuserovbuffusercpusyscpunumckptsflushes

00000.000.002239

bufwaitslokwaitslockreqsdeadlksdltoutslchwaitsckpwaitscompress

00000000

OVTBLS：**指示** TBLSPACE **值不够**，TBLSPACES

值应增大

OVLOCK：锁溢出 LOCKS 应增大

OVUSER：用户表资源不够 USERS

OVBUF：缓冲区资源不够 BUFFERS

以上数字若持续增加，表明相应参数值应调整

DEADLOCKS：检测到的死锁数

DLTOUTS：网络超时死锁

DEADLOCK_TIMEOUT：一般为 60 秒，若增大此数字，可以减少死锁数，但会增加响应时间，用户看来性能更差；若减少此数字，死锁数可能会增大，但会减少响应时间，在用户看来，响应时间变快了。

四、ONCHECK：

定位并修复数据与索引的不一致

检查磁盘上的数据结构

显示不同数据结构的报告

某些选项可能回在它所涉及的表上家一个共享锁

用法：tbcheck[-clist][-plist][-qny]

[{database[:[owner.]table]|TBLSpacenumber|Chunknumber}

{rowid|pagenumber}]

-c-check（检查）

r-reservedpages（保留页）

e-TBLSpaceextentsandchunkextents

c-databasecatalogs（目录）

i-tableindexes（表的索引）

I-tableindexesandrowidsinindex（表的索引和索引的rowid）

d-TBLSpacedatarowsincludingbitmaps（表空间的数

据行包括位图）

D-TBLSpacedatarowsincludingbitmaps,remainderpagesandblobs

-p-print（显示）

r-reservedpages(-cr)（保留页）

e-extentsreport(-ce)（extent）

c-catalogreport(-cc)（目录）

k-keysinindex(-ci)（索引的键值）

K-keysandrowidsinindex(-cI)

l-leafnodekeysonly(-ci)

L-leafnodekeysandrowids(-cI)

d-TBLSpacedatarows(-cd)

D-TBLSpacedatarowsincludingbitmaps,remainderpagesandblobs(-cD)

t-TBLSpacereport

T-TBLSpacediskutilizationreport

p-dumppageforthegiven[tableandrowid|TBLSpaceandpagenumber]

P-dumppageforthegivenchunknumberandpagenumber

B-BLOBSpaceutilizationforgiventable(s)[database:[owner.]]table

-q-quietmode-printonlyerrormessages

-n-answerNOtoallquestions

-y-answerYEStoallquestions

五、几个主要的 tbcheck 选项：

TBCHECK-PTDATABASE：TABLE

产生有关 TBLSPACE **的报告。包括** EXTENT **和空间使用信息。信息的输出来自** TBLSPACETBLSPACE，

这是一个特殊的 TBLSPACE，用来跟踪 ONLINE 创建的数据库，是根 DBSPACE 中的第二个 TBLSPACE，其 TBLSNUM 为 1000002。这个 TBLSPACE 中的每行保存一个数据库的信息。

TBCHECK-CdDATABASE：TABLE

测试一个表空间的数据。它从表空间中读取所有页做一致性检查，并检测位图也以确认该页被正确映射。它所做的工作包括：确认数据的有效性，对特殊数据做相应标志(如 BLOB 页，剩余页等)。

TBCHECK-CiDATABASE：TABLE

测试表上的索引，比如是否所有的 ROWID 都指向表中的有效行，必要时，重建 SYSINDEXES 信息。它所做的工作包括：验证索引是否有效，重建被损坏的索引，并对索引树重新做相应调整。

索引策略：

应加索引的字段：用于连接(JOIN)的字段，用来过滤的字段(在 WHERE 条件中使用的字段，称为过滤器)，用于排序的字段(ORDERBY)。

不应加索引的字段：有高重复值的字段

另外，可以用主键限制索引个数，加复合索引来避免重复，用聚簇索引加快索引速度。

六、其它重要的工具：

UPDATESTATISTICS[LOW|MEDIUM|HIGH]

此语句的三个主要作用：更新内部表；产生数据分布；优化存储过程

应定期使用

tbinit

用于初始化数据库共享内存。每次开机后都必须执

需要全本请访问 www.ertongbook.com

行此命令。一般将它放在/etc/rc2.d 目录下的一个自引导文件中。

例如：当以应用软件注册时，系统提示：

DBERR–529/C-ISAM–123[OpenDatabase]

使用 finderr 查 529 错误，提示：

-529Cannotattachtosharedmemory.{不能使用共享内存}

此时，使用 tbinit 命令将数据库共享内存进行初始化后，系统恢复正常。

tbmode-sy|tbmode-ky

用于关闭数据库共享内存。为了数据的安全，每次关机前都必须执行此命令。为了简化操作，我们经常将它放在/etc/rc0.d 目录下的一个自引导文件中，让系统进入多用户时自动来执行此文件。

finderr

用来查询数据库的错误。当系统出现数据库错误时，系统处了给你简单的出错提示外，还给出一个错误号。你可以使用此命令对这种错误进行深入的分析，然后针对不同的错误进行相应的处理。

例如：当你在超级用户 root 下对储蓄数据库 bankstar 进行操作时，系统提示：

387:Noconnectpermission.

111:ISAMerror:norecordfound.

意思是：系统出现 387 错误，错误的原因是：没有 connect 权限。

原因分析：

操作数据库至少需要 CONNECT 权限。解决如下：informix 用户注册，执行如下 SQL 语句：grantconnectto

root

dbaccess

快速装载 MYSQL 数据库

MySQL 是一个小巧玲珑的数据库服务器软件，对于小型（当然也不一定很小）应用系统是非常理想的。除了支持标准的 ANSISQL 语句，它还支持多种平台。而在 Unix 系统上，该软件支持多线程运行方式，从而能获得相当好的性能。对于不使用 Unix 的用户，它可以在 WindowsNT 系统上以系统服务方式运行，或者在 Windows95/98 系统上以普通进程方式运行。

对于 Windows95/98 系统运行方式，实际创建出数据库，是在 DOS 环境里键入下列命令：mysqladmin-uroot createmydb,其中-u 选项告诉 MySQL 我们使用的是默认 root 用户，这样就创建了一个名为“mydb”的数据库。

从 MySQL 数据库中读取数据之前，我们得先往数据库里放一些数据。在现在这一阶段，还没有一个简便的方法来做这件事情。对 Win32 用户来说，很不幸，要在 DOS 下做些工作，就不得不使用 DOS 窗口：在 DOS 提示符下键入“MySQL”出现欢迎信息和“MySQL>”提示符，再键入“usemydb”,表示开始使用我们新建的数据库，然后再创建数据表（文件）的名称和数据结构，并为数据表加入数据。在这个过程中，任何错误输入都可能产生错误，导致前功尽弃，必须从头再来。

为了避免这种情况，我们需要采用其他方式。这里以大家都喜欢用的员工数据库为例，我们可以这样来添加数据：

把下面的文字复制到一个文件中，把该文件存在 MySQL 的 bin 目录下（我假定文件名是 mydb.dump）。

CREATETABLEemployees(idtinyint(4)DEFAULT0NOTNULLAUTO_INCREMENT,firstvarchar(20),lastvarchar(20),addressvarchar(255),positionvarchar(50),PRIMARYKEY(id),UNIQUEid(id));

INSERTINTOemployeesVALUES(1,Bob,Smith,128HereSt,Cityname,MarketingManager);

INSERTINTOemployeesVALUES(2,John,Roberts,45ThereSt,Townville,Telephonist);

INSERTINTOemployeesVALUES(3,Brad,Johnson,1/34NowhereBlvd,Snowston,Doorman);

如果文字是折行的，请确保每一个 INSERT 语句都另起一行。现在，我们要把数据加入到 mydb 数据库中了。在命令行下，键入下面的命令：

MySQL-urootmydb

此时你应该不会遇到什么错误。如果真的出错了，请仔细检查一下是否因上面的文字折行而引起错误。这样就完成了向数据库传送数据表的功能，操作起来方便了许多。

如何恢复系统数据库

在 SQLServer 数据库中，系统信息存储在系统数据库中，主要的系统数据库包括：

master-从整体上控制用户数据库和 SQLServer 操作，在创建了任何用户定义的对象后，都要备份它

model-为新数据库提供模版和原型

msdb-包含了有关作业、报警及操作员等信息如果包含系统数据库的介质变了，那么必须重建系统数据库，如果你仍然可以启动 SQLServer 服务，则可以通过 RESTORE 语句从系统数据库的备份中恢复数据库。

如果 master 坏了，不能启动系统,可以按照下面步骤进行恢复

1、重建系统数据库运行 c:\mssql7\binn\rebuildm.exe，按照提示进行即可，过程中需要系统数据库样本的路径，可在安装光盘中找到；

2、重建系统数据库后，启动 SQLServer 服务，用系统数据库的备份恢复数据库就行了通常恢复顺序为 master->msdb->model

在恢复 master 的备份时要注意：必须在 singleuser 模式下进行，有以下几种方法进入单用户模式：

1、可以在命令行模式下输入 sqlservr-c-f-m 或者输入 sqlservr-m 其中：-c 可以缩短启动时间，SQLServer 不作为 WindowsNT 的服务启动-f 用最小配置启动 SQLServer-m 单用户模式启动 SQLServer

2、可以在控制面板-服务-MSSQLServer 的启动参数中输入-c-f-m 或者输入-m,点击开始

3、还有一种更灵活的启动方法：用存在注册表里的启动参数启动在 MSSQLServer 项下添加项 SingleUser,具体内容如下所示：

HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer\SingleUser\Parameters

SQLArg0

REG_SZ:-dC:\MSSQL7\DATA\MASTER.DAT

SQLArg1

REG_SZ:-eC:\MSSQL7\LOG\ERRORLOG

SQLArg2:

REG_SZ:-lC:\MSSQL7\DATA\MASTLOG.DAT

SQLArg3

REG_SZ:-m 在命令行下输入 SQLServr-c-s SingleUser,注意：必须是在命令行下进入单用户模式后启动 QueryAnalyzer 执行语句：RESTOREDATABASEmasterformdisk=c:\(具体的备份文件名)

数据仓库更新的新策略--工作流

1．概述

数据仓库作为一种新技术，主要是为决策支持系统和 OLAP 应用提供软件架构。它从异构和分布式数据源中收集数据，这些数据首先被聚合，然后按照 OLAP 所定义的组织标准进行定制。数据仓库的结构能够通过一种分层存储的方式加以定义。这种方式涉及到的存储形式包括从底层的数据源到高度的聚合数据（数据集市）。在这两种存储形式之间，按照 OLAP 程序的要求，还存在一些其他不同的存储形式。其中之一就是对操作型数据的存储，操作型数据是以单一和干净的方式来表征数据源中的数据。企业级数据仓库（CDW）则包含高度聚合的数据，并且被组织成多维表的形式。从每个数据源中抽取的数据可以存储在中间数据容器中。显然，这种分层存储方式只是一种逻辑上的表示方式，它体现了从数据源到数据集市的数据流动过程。所有这些存储形式都不一定要具体实现，如果确实需要的话，他们也只能形成同一数据库的不同层面而已。

图 1 显示了一种典型的数据仓库结构。这只是一个逻辑视图，它的具体实现，不同厂家有自己不同的数据仓库产品解决方案。数据抽取和数据清洗的实现与每个数据源有关，对于不同的数据源提供有统一的或定制的工具。同样，数据的一致性（多数据源清洗）既可以与数据集成（多数据源操作）分开也可以合并到数据集成中进行。高级别的数据聚合工作可以看成一个计算技术的集合，这个集合的范围涵盖从简单的统计函数到高级的数据挖掘算法。对于不同的数据集市来说，数据定制技术是不同的。关键在于决策者想要看到的数据的详尽程度。

数据仓库更新是一个非常重要的过程，它决定了数据采集和数据聚合的实效性。确实，向决策者提供的数据的质量与以下因素有关。首先，与数据仓库系统在合理的时间内将数据从数据源转换到数据集市的能力有关。其次，与数据仓库对数据源中信息发生变化的敏感程度有关。大部分的设计考虑主要集中在对数据结构的选取和数据的更新技术上，这里的数据更新技术指的是对数据仓库更新的优化策略。

在对数据仓库更新的理解方面在相关的文献上存在着很大的误区。确实，这个过程经常被简化为视图维护问题或与数据导入混为一谈。本文的目的之一就是指出数据

需要全本请访问 www.ertongbook.com