

本文分析了数据库事务日志的运作原理,结合上海燃料总公司附属油品公司的信息系统综合论述了如何在大中型企业中进行以事务日志为基础的数据库的备份和还原,进而保持数据库的完整性,同时,指出了事务日志可能对数据库运行效率产生的影响,并从数据库开发的前台和后台阐述了如何避免事务日志对数据库产生的负效应。

# 用事务日志提高数据库的完整性及效率

同济大学经济与管理学院 凌晨凯 刘杰

## 引言

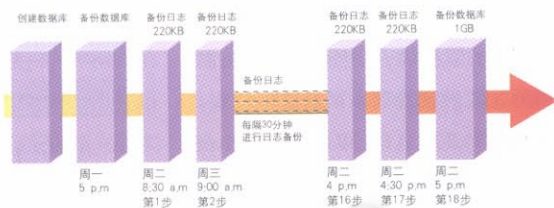
事务日志记录着对数据库的一切操作,因此在发生意外时,有步骤地操作事务日志可以重新恢复数据库的完整性。同时因为事务日志记录着用户对数据库的一举一动,如果用户对数据库的操作频繁,处理的数据量很大,或应用程序编写不当便会使数据库的事务日志频繁地溢出,这严重影响数据库的运行效率。本文以上海市燃料总公司油品信息管理系统为例,论述如何在大中型企业进行以事务日志为基础的数据库动态备份、还原策略,及如何改进前后台程序,减少事务日志的溢出次数,进而提高数据库运行效率。

## 事务日志与数据库的完整性

(1)维护数据库完整性方法。数据库管理系统提供了很多维持数据库完整性的工具,最常用的有镜像、备份数据库等。镜像就是服务器将两个设备互相影射,让他们始终保持完全同步。这样,当一个设备发生故障时,另外一个与其一模一样的设备可以马上接替。利用镜像来保持数据库的完整性是比较可靠的,但其缺点是要占用很大的磁盘空间,对硬件的要求高,并且若操作不当,会大大增加CPU的负荷。这对于那些资金有限的企业显然不太适合。备份数据库是应用最广的一种维护系统完整性的方法。它有众多优点,并被普遍接受。如今的备份数据库都是动态地备份(如:每隔2小时数据库自动备份一次)。这对于小型数据库是比较可行的,但若是一个大中型企业并且拥有一个含有大量数据的大型数据库,这种方法便不是很适合了。因为备份一个大型数据库需要一定量的时间,在这段时间里,如果有用户连接数据库,那么他对数据库的操作效率便会有很大影响。

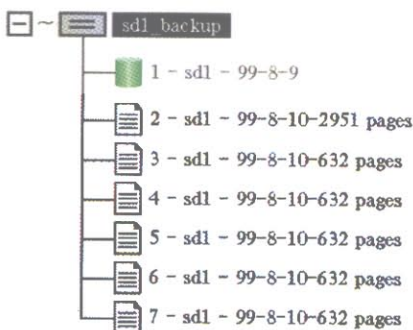
因此在这些大中型企业中,为了高效地保持数据库的完整性,可以使用备份事务日志与备份数据库结合的方式。事务日志记录了用户和管理员对数据库的所有操作,根据上述事务日志的理论可知,当数据库发生意外时,只需将事务日志恢复进数据库,那么在服务器掉电时,系统便会自动地执行事务日志记录下的那些操作,这样数据库的数据区便可恢复原状。

(2)基于事务日志的数据库备份及还原策略步骤。上海燃料总公司是一个国有大中型企业,其下属企业油品公司MIS系统—油品公司管理信息系统是一个以SQL SERVER 6.5为后台,POWERBUILDER为前台的分布式系统。由于油品公司的业务往来较为频繁,数据处理较多,因此必须经常性动态地将这些频繁更改或添加的数据进行备份。显然这里使用备份数据库便不合适。结合油品公司业务数据的重要性及被操作的频率,将备份日志的间隔定为30分钟。如下图:



油品公司上午8:00开始正式上班,进行业务往来,下午4:45下班。因此上午8:30时,先进行一次日志备份(不在8:00备份,因为8:00时还未进行业务,因而日志为空),之后每隔30分钟备份一次,直至下班前4:30做最后一次日志备份。在下班后5:00时,服务器开始空闲,这时在进行数据库备份(注:SQL SERVER进行数据库备份时,不仅将数据备份,而且还将日志一同备份),同时可以将先前备份的日志删去,以腾出磁盘空间。

下图是在99-8-10日上午11:02时数据库的备份情况。此时如果数据库发生意外,那么很容易将其恢复到11:00时的状态。步骤为:首先将99-8-9日下午5:00备份的数据库恢复,然后将99-8-10日上午11:00时备份的日志恢复进刚刚恢复好的数据库,这样保证了99-8-9日下午5:00以后至99-8-10日上午11:00之间的所有对数据库的操作都以日志的形式恢复进了数据库。然后重新启动数据库服务器,这样数据库服务器根据恢复进数据库的日志,将99-8-9日下午5:00至99-8-10日上午11:00的数据恢复进数据库,保证了数据的完整性。经测试,备份一个1GB的数据库平均需1分25秒,而备份一个日志平均只需12秒。由此可看出,基于事务日志的备份的效率要远远高于备份数据库。



实现上述操作,只需在SQL SERVER中编与两个存储过程,并用SQL SERVER中的任务管理器设定运行时间即可。

### 事务日志与数据库的效率

诚然事务日志对于维护数据库的完整性起着重要的作用,然而如果应用不当反而会大大降低数据库的运行效率。以下讨论事务日志是如何对数据库的效率产生影响及相应的解决方法。

#### ①事务日志影响数据库效率的原因

①事务日志与锁定。为了保证数据库的完整性,数据库一般在一个事务期间对涉及到修改的页面上放置了互斥锁。如果一个表中的页面涉及了修改,那么被事务中的操作语句修改的页面或表被上锁,而且一直锁到数据库遇到一个commit tran(或是rollback tran)。当数据被锁定时,只有一个用户可以操作它,其他的用户将进行等待。只要编程不出现错误,那么一个用户执行完自己的事务后,便会释放出锁定的数据,其他用户便可使用,其他用户等待的时间并不会很长。然而如果事务日志的空间已将近用完,这时如果再运行一个事务,很可能在该事务运行至一半时事务日志空间用完。这时,数据库一般不会处理日志的溢出,那么该事务就会被挂起,系统等待人为的干涉和处理。该事务运行所牵涉的数据也会被锁定至人为地切断该事务为止。若是事务日志溢出在短时间内并未被系统管理员发现,那么其他用户将因数据被锁定而阻塞,这大大降低了数据库的并发性和效率。

②事务日志与存储过程。建立存储过程是数据库开发人员必不可少的开发手段。一般程序开发人员喜欢用一个存储过程建立一个事务。这当然是一种行之有效的方法,它可以确保数据库的完整性。当然,当存储过程所执行的操作不是很多的时候,不会发生什么问题。而一旦一个事务所执行的操作较大时,如update一个大型的表(这个表有可能包含上万甚至上百万条记录),那么这时可能会造成事务日志空间还来不及被清空便已经被耗尽。这种情况经常在大数据库的应用中出现,而且比较不容易找出原因。碰到这种情况时,有的程序员会将一个存储过程分为几个较小的事务来处理,这虽然能够解决日志溢出的问题,但会破坏数据库的完整性。长时间运行的事务也会发生同存储过程一样的情况,它可能是因存储过程引发的,也有可能是由前台的程序所使用的嵌入式SQL语句

引发的。下面将结合油品MIS的开发过程,探讨一下解决上述问题的一些方法。

#### (2)关于事务日志的解决方法

①前台编程与事务日志。油品MIS的前端开发工具是power builder6.0(简称PB),该工具能够方便的操作和控制数据库。在PB的数据库连接选项中,有一项auto commit选项,该选项决定PB中有关数据库的操作。若选true则表示PB每执行一句有关数据库的语句就当作一个事务提交。若选false则表示事务的大小由开发者控制(即由开发者来决定什么地方begin tran,什么地方commit tran)。在编程时若碰到大型数据操作时,可在该操作前先将auto commit设为true,运行该操作后再将其改为false。如:

```
SOLCA_AutoCommit = True
insert sm4
values(srxh_m.getdate(),:khdm,:spdm,:ygdh,:ykdh,0 - :sl,:zsjh,:zsfph,:
:tdh,:jbr,:dycc,:bz + "红冲" + :jh,:bz1,:bz2);
//加入数据库语句清空日志
update sm4
set bz = isnull(bz,?) + "已被" + :srxh_m + "红冲"
where sm4.jh=:jh;
SOLCA.AutoCommit = false
```

在上述语句中,insert和update都要做大量的数据处理,这时可在insert使auto commit为true,这时,PB在进行完insert时自动提交,然后将日志清空,再执行update。这样即使insert占用了大量的日志空间,日志也会在执行update前被清空。但这种做法一定要在前后两个操作没有互相依赖的前提下才能进行,否则将破坏数据库的完整性。这时执行update时便不会因日志溢出而出现错误了。

②其他一些监控方法。油品MIS的操作平台是WINDOWS NT SERVER 4.0(简称NT),这使SQL SERVER能更好地与NT结合,来监控日志大小。运用NT的性能监视器,可以动态地监视日志的大小。只需在监控的选项中指明当日志已用空间的大小超过某一百分比(如80%)时,执行某一特定的任务便可。如油品MIS中,设定当日志已用空间的大小超过70%时,便自动备份日志,随后清空日志。

上述所讨论的问题和解决方法都是经常用到的。当然还有其他一些监视和控制事务或事务日志的方法,如MS TRANS ACTON COORDINATE等也是相当不错的工具。现在还有一些专门的第三方编制的读取事务日志的软件,它也能很好的控制事务日志。■

#### 参考文献

- 1 熊桂喜,高峰,MS SQL SERVER6.5开发指南(第二版),北京清华大学出版社,1999
- 2 陈向群,Powerbuilder5开发人员指南,北京机械工业出版社,1998,3
- 3 陶辅周,杨代伦,实用ORACLE系统开发和设计详解,北京电子工业出版社,1995