

何月顺 汤彬

(江西抚州华东地质学院信息工程系 344000)



摘要: 本文主要针对 Oracle 数据库可能遇到的各种故障提供了相应的恢复方法和过程, 及恢复过程中的注意事项以便尽最大可能挽救数据。

关键词: Oracle 数据库 数据文件 灾难 恢复

随着信息技术的迅速发展和应用面的迅速扩大, 企业对信息系统的依赖性越来越高, 数据库作为信息系统的核心担当着重要的角色, 尤其在一些对数据可靠性要求很高的行业如银行、证券、电信等, 如果发生意外停机或数据丢失其损失会十分惨重。为此数据库管理员应针对具体的业务要求制定详细的数据库备份与灾难恢复策略, 并通过模拟故障对每种可能的情况进行严格测试, 只有这样才能保证数据的高可用性。数据库的备份是一个长期的过程, 而恢复只在事故发生后进行, 恢复可以看作是备份的逆过程, 恢复程度的好坏很大程度上依赖于备份的情况。此外, 数据库管理员在恢复时采取的步骤正确与否也直接影响最终的恢复结果。

1 数据库恢复的类型

要对 Oracle 数据库备份与恢复有清晰的认识, 首先有必要对数据库的几种运行状态有充分的了解。Oracle 数据库的运行状态主要分为三种, 依次为:

(1) Nomount (非安装) Oracle 只是读取 ini 文件中的配置信息, 并初始化 SGA 区。

(2) Mount (安装) Oracle 除了需要读取 ini 文件还要读取控制文件, 并从中获取有关数据库的物理结构等信息。

(3) Open (打开) 数据库要检查所有文件处于

同一时间点, 对错误进行恢复, 对未完成事务回滚, 并最终可以允许用户访问。

数据库的恢复可分为两大类: 完全恢复; 不完全恢复:

① 完全恢复指将数据库恢复到发生故障的时间点, 不丢失任何数据。不完全恢复指将数据库恢复到发生故障前的某一个时间点, 此时间点以后的所有改动将会丢失。如果没有特殊需求, 建议应尽量使用完全恢复。

② Oracle 数据库的恢复过程分两步进行, 首先将把存放在重做日志文件中的所有重做运用到数据文件, 之后对重做中所有未提交的事务进行回滚, 这样所有数据就恢复到发生灾难那一时刻了。数据库的恢复只能在发生故障之前的数据文件上运用重做, 将其恢复到故障时刻, 而不能将数据文件反向回滚到之前的某一个时刻。例如:

我们有一个 2002/02/01 的数据库备份, 在 2002/03/02 时发现数据库中数据发生混乱, 希望将数据库恢复到 2002/03/01 时的状态, 我们只能先恢复 2002/02/01 的数据库备份然后在其上运用重做记录使其前滚到 2002/03/02 时的状态, 而不能将 2002/03/02 的数据库向后回滚到 2002/03/01。

2 数据库恢复方式

为了系统地设计数据库的恢复方案, 可以先对可能遇到的错误进行分类, Oracle 数据库的错

误主要分为五大类: SQL 语句失败、线程失败、实例失败、用户操作失败、存储设备失败。

如果发生前三种失败, 不需要人为干涉, Oracle 系统会自动进行恢复。对于用户操作型的失败 (如误删除数据), 我们采取的补救措施主要有导入最新的逻辑备份或进行到某一时间点的不完全恢复。从 Oracle 8 之后的新版本中引入了基于表空间的时间点恢复 (TSPITR), 可以单独将包含错误操作的表空间恢复到指定时间, 而不必对整个数据库进行不完全恢复。当错误操作发现比较及时而且数据量不大的情况下, 也可以考虑使用 logminer 生成反向 SQL。

由于存储设备失败的情况比较复杂, 所以作为本文讨论的重点。存储设备的失败必然会使放置在其上的文件变为不可用, 先将 Oracle 数据库所涉及到的文件进行划分, 主要可分为:

- (1) Oracle 的系统文件, 指 Oracle 的运行文件;
- (2) 各种应用程序;
- (3) 数据库控制文件;
- (4) 数据库联机重做日志文件;
- (5) 数据文件;
- (6) 归档日志文件。

2.1 控制文件的恢复

控制文件中记录着整个数据库的结构, 每个数据文件的状况, 系统 SCN、检查点计数器等重要信息, 在创建数据库时会让用户指定三个位置来存放控制文件, 他们之间互为镜像, 当其中任何一个发生故障, 只需将其从 ini 文件中注释掉故障数据文件就可重新将数据库启动。当所有控制全部失效时, 可以在 Nomount 模式下执行 create controlfile 来重新生成控制文件, 但必须提供 redo log, data file, 文件名和地址以及 MAXLOGFILES, MAXDATAFILES

Disaster Recovery of Oracle Database

MAXINSTANCES 等信息。如果失败之前运行过 `alter database backup controlfile to trace` 或 `alter database backup controlfile to 'XXX'` 对控制文件作备份, 恢复时可使用生成的脚本来重建或用备份文件覆盖, 如果使用了旧的控制文件在恢复时要使用 `recover XXX using backup controlfile` 选项来进行恢复, 并使用 `resetlogs` 选项来打开数据库。

2.2 日志文件的恢复

如果丢失的是联机日志文件, 分两种情况处理:

- (1) 丢失的是非活动的日志文件;
- (2) 丢失的是当前激活的日志文件。

2.2.1 非活动日志文件

如果是第一种情况, 而发生故障的日志文件组又具有多个成员, 可以先将数据库 `shutdown`, 然后用操作系统命令将损坏日志文件中好的日志成员文件把损坏的成员文件覆盖(在同一个日志成员组中的所有日志文件的各为镜像的), 如果其物理位置不可用, 可将其拷贝到新的驱动器上, 使用 `alter database rename file 'XXX' to 'XXX'` 改变文件位置, 之后启动数据库, 如果正常马上进行一个冷备份。如果损坏的日志组中只有一个日志成员, 先 `mount` 上数据库, 将其转换为 `noarchivelog` 模式, 执行 `alter database add logfile member 'XXX' to group 'X'` 给相关组增加一个成员, 再执行 `alter database drop logfile member 'bad-file'` 将损坏的日志文件删除, 由于数据库的结构发生变动需要备份控制文件, 之后将数据库改回 `archivelog` 模式, 做一个冷备份。

2.2.2 当前激活日志文件

如果丢失的是当前激活的日志文件, 数据库又没有镜像而且当前日志组中所有成员均变为不可用。首先将数据库 `shutdown abort`, 从最近的一次全备份中恢复所有的数据文件, 将数据库启动到 `mount` 状态。如果原来的日志文件

物理位置不可用, 使用 `alter database rename file 'XXX' to 'XXX'` 改变文件的存放位置。然后, 使用 `recover database until cancel` 命令来恢复数据库, 直到提示最后一个归档日志运用完之后, 输入 `cancel`。之后用 `alter database open resetlogs` 打开数据库, 如果没有问题, 立即进行一个冷备份。注意! 所有包含在损坏的 redo log 中的信息将会丢失, 也就是说数据库崩溃前已经提交的数据有可能会丢失。这对于某些要求很高的应用将会损失惨重, 因此应尽量使每个日志组具有多个日志成员, 并且放置在不同的驱动器上一防止发生介质故障。

2.3 数据文件的恢复

数据文件发生故障的情况也分为多种情况:

- (1) 丢失包含在 `SYSTEM` 表空间的数据文件;
- (2) 丢失没有回滚段的非 `SYSTEM` 数据文件;
- (3) 丢失有回滚段的非 `SYSTEM` 数据文件。

2.3.1 系统表空间数据文件

如果损坏的是系统表空间的数据文件。唯一的办法是从上一次备份中恢复受损的数据文件, (如果原位置不可用, 则使用 `alter database rename` 命令改变新文件的位置), 之后在数据库 `mount` 的状态下执行 `recover database/datafile` 对数据库进行回复, 才能将数据库打开。注意: 当 `SYSTEM` 表空间或其中的数据文件脱机, 数据库是无法被打开的, 因此必须在 `mount` 状态下将所有的恢复工作完成。

2.3.2 没有回滚段非系统数据文件

当丢失的数据文件不属于系统表空间而且也不包含回滚段时, 有可选择在数据库的两种状态下进行恢复——在数据库 `open` 的状态或者在数据库 `mount` 的状态。如果用户急于访问数据库中未受损部分的数据或对损坏的数据文件进行恢复需要很长时间, 可以先使受损的数据文件脱机, 将数据库打开给用户访问, 再恢复受损的数据文件最后将其联机。步骤如下:

先在数据库 `mount` 时, 将相关的数据文件或表空间进行脱机 `alter database datafile 'XXX' offline`, 然后将数据库 `open`, 这样就能使数据库未受损的部分先供用户访问, 之后再进行 `recover datafile/tablespace`, 完成后用 `alter database datafile/tablespace 'XXX' online` 使其恢复联机就可被访问了。当然用户也可以选择是在数据库 `mount` 状态下, 用 `recover database/datafile` 将所有的恢复工作做完, 将所有数据文件一起打开供用户访问。

2.3.3 有回滚段非系统数据文件

如果丢失的数据文件是最后一种情况, 即包含有回滚段的非系统表空间数据文件, 也可以选择是在数据库先 `open` 的状态还是在 `mount` 状态下恢复。不过与上一种情况不同的是当包含回滚段

的数据文件损坏时,如果使其先offline将数据库打开,那么所有数据库崩溃前未提交的事务涉及到的表将无法访问,也就是说在回滚段恢复前其中涉及的对象都不允许被访问,而且当所有包含回滚段的数据文件都在offline状态时,数据库无法进行任何DML操作,因此在数据库open状态恢复包含回滚段的数据文件时,可以先创建几个临时回滚段供数据使用 create rollback segment temp1 tablespace system; alter rollback segment temp1 online; 当数据文件恢复后再将他们删除 alter rollback segment temp1 offline; drop rollback segment temp1. 注意: 当用这种方法使恢复的数据文件online之后,所有的原有回滚段将处于offline状态,必须手工使用alter rollback segment RBSxx online使他们恢复联机状态,这样才能被

数据库正常使用。如果在数据库mount状态下完成所有恢复,则不需要上述步骤。

如果丢失数据文件后,用户发现没有故障前的数据文件的备份,而且自从丢失的数据文件最早建立之后一直没有使用过resetlogs选项打开过数据库,也就是说用户的控制文件是在损坏的数据文件建立前创建的,归档日志中包括对损坏数据文件的所有重做记录。用户就还有一种恢复方法,用户可以先将损坏的数据文件或表空间脱机 alter database datafile / tablespace XXX offline,之后执行 alter database create datafile 'new/XXX.db' as 'old/XXX.dbf',数据库会根据保存在控制文件中的信息重建一个空的数据文件,之后再执行recover tablespace/datafile将所有重做记录运用到数据文件,使其完全恢复到当前状态,之后便可再将其恢复联机。

如果丢失的是最后一种文件——归档文件或归档文件所处的物理位置不可用,首先 shutdown数据库,立即作一个冷备份,然后修改ini文件中的归档日志文件目的路径,重新启动数据库。以后再发生灾难只需从最新的备份中将相关文件恢复,数据库作recover时就不需要备份之前丢失的归档文件了。在Oracle 8之后的新版本中提供了log-archive-duplex-dest和log-archive-dest-1...5等参数允许保留多份归档文件到不同位置,甚至到远端服务器从而保证归档文件的可靠性。

3 数据库恢复过程中的注意事项

本文讨论所有情况的默认前提是数据库运行在归档(ARCHIVELOG)方式下,并只涉及到一般常见的情况和最基本的恢复方法。

(1) 使用Oracle提供的恢复管理器RMAN也能完成上述任务,如果运行环境比较复杂,建议使用RMAN来做备份和恢复。

(2) 一旦数据库发生灾难,最好在恢复

之前做一次完全的冷备份,以便在进行恢复时产生差错还可以进行补救,很大一部分数据丢失是由于不正确的恢复操作所引起的。

(3) 当数据库完成恢复之后,尤其是使用resetlogs选项打开数据库之后,要马上关闭数据库进行一次完全的冷备份。因为,为防止放弃的重做日志被下次恢复时再次运用,resetlogs选项会重新创建redo log文件并将其计数清零,这将使之前做的所有备份将变为不可用(一般情况下)。

(4) 要特别注意当进行数据库完全恢复,从发生故障的时间点前的备份中恢复损坏文件时,一定不要使备份中的redo log文件覆盖了当前的redo log文件,否则就只能进行不完全恢复并且要丢失一部分数据了。

4 结论

数据库备份是组成数据库的所有文件的集合。如果毁坏或丢失了数据库组成部分的某个文件,可以从数据库备份中提取丢失文件的一个拷贝来修复数据库中的这个文件;数据库恢复操作恢复由于某种故障而丢失的工作。在正确配置的情况下,Oracle能够对所有故障恢复数据库,这些故障包括简单的系统崩溃或更为严重的磁盘故障,甚至还包括彻底的站点毁坏,而Recovery Manager能够使Oracle数据库的备份和恢复更为容易。 ■

参考文献

- 1 Steve Bobrowski. Oracle 8i for Windows NT Starter Kit. The McGraw-Hill Companies. 2000.
- 2 刘辰. 数据库系统管理与应用. 人民邮电出版社. 2000.