

数据库结构调整

1 参数文件管理

Oracle 使用初始化参数文件来保存 Oracle 例程启动时所要用的初始化参数信息。初始化参数文件分为文本参数文件 `pfile`、二进制参数文件 `spfile` 两种。`spfile` 是一个不可编辑的二进制文件，这两个参数文件之间是可以相互转换的。`spfile` 可以在线修改数据库参数，并且修改后不需要重新启动数据库就可以起作用。因此本实验重点讲述 `spfile` 的管理。

最初，服务器参数文件是根据传统的文本参数文件使用 `create spfile` 语句建立起来的。尽管可以用文本编辑器打开二进制的参数文件，并查看其中的文本，但是不能手动编辑，否则会破坏该文件，导致例程不能启动。

(1) 导出二进制参数文件

可以依靠二进制参数文件创建传统的文本初始化参数文件，这样就可以得到服务器参数文件的备份，并且列出例程当前使用的所有参数值，以便查找故障原因。

```
C:\Users\JiangXue>sqlplus/nolog

SQL*Plus: Release 11.1.0.7.0 - Production on 星期五 3月 30 10:44:45 2012

Copyright (c) 1982, 2008, Oracle. All rights reserved.

SQL> conn sys/ora505 as sysdba
已连接。
SQL> create pfile = 'd:\ORACLE\inittest.ora' from spfile;

文件已创建。
```

上述操作实现了通过 `spfile` 创建 `pfile`，创建成功的文件被保存在 `d:\ORACLE` 目录下，可以通过写字板打开该文件。

(2) 创建服务器参数文件

创建必须在使用 `startup` 命令之前完成才能起作用，实现了由 `pfile` 创建 `spfile` 具体如下：

```
SQL> shutdown immediate;
数据库已经关闭。
已经卸载数据库。
ORACLE 例程已经关闭。

SQL> create spfile from pfile='d:\ORACLE\inittest.ora';

文件已创建。
```

(3) 更改初始化参数

通过 `alter session` 或 `alter system` 语句可以更改初始化参数，当然执行该语句的用户也必须具有 `sysdba` 或 `sysoper` 系统权限。

```
SQL> startup
ORACLE 例程已经启动。

Total System Global Area  431038464 bytes
Fixed Size                  1347804 bytes
Variable Size               331353892 bytes
Database Buffers            92274688 bytes
Redo Buffers                 6062080 bytes
数据库装载完毕。
数据库已经打开。
```

首先查看参数 `job_queue_processes` 在修改前的值。

```
SQL> column name format a20
SQL> column value format a20
SQL> select name,value from v$parameter
      2  where name = 'job_queue_processes';
```

NAME	VALUE
-----	-----
job_queue_processes	1000

```
SQL> alter system set job_queue_processes=500 scope=both;
系统已更改。
```

注:scope =both 的目的:在修改初始化参数时,scope 可以有三种方式:spfile,memory,both。spfile 只更改服务器参数文件中的参数,只有当下一次启动时才会一直生效;而 memory 则只更改内存中的参数,不能永久保留;both 既可以更改服务器参数文件中的参数,也可以更改内存中的参数。采用 both 方式,既使更改永久有效,又可以及时查看。

更改后,查询 `v$parameter`,可以看到更改的内容已经生效了。

```
SQL> select name ,value from v$parameter
      2  where name='job_queue_processes';
```

NAME	VALUE
-----	-----
job_queue_processes	500

为了查看修改是否被永久地保留,因此关闭数据库并且在重新启动后查看。

```

SQL> shutdown immediate;
数据库已经关闭。
已经卸载数据库。
ORACLE 例程已经关闭。
SQL> startup
ORACLE 例程已经启动。

Total System Global Area  431038464 bytes
Fixed Size                  1347804 bytes
Variable Size              331353892 bytes
Database Buffers           92274688 bytes
Redo Buffers                6062080 bytes
数据库装载完毕。
数据库已经打开。
SQL> select name,value from v$parameter
      2  where name = 'job_queue_processes';

NAME                                VALUE
-----
job_queue_processes                500

```

2 控制文件管理

(1) 复用控制文件

控制文件非常重要，一般需要进行备份。

首先通过语句 `show parameter control_files` 查看控制文件所在的位置，才可以将控制文件多路复用。

NAME	TYPE	VALUE
control_files	string	D:\ORACLE\JIANGXUE\ORADATA\ORCL\CONTROL01.CTL, D:\ORACLE\JIANGXUE\ORADATA\ORCL\CONTROL02.CTL, D:\ORACLE\JIANGXUE\ORADATA\ORCL\CONTROL03.CTL

```

SQL> alter system set control_files =
      2  'd:\ORACLE\JIANGXUE\ORADATA\ORCL\CONTROL01.CTL',
      3  'd:\ORACLE\JIANGXUE\ORADATA\ORCL\CONTROL02.CTL',
      4  'd:\ORACLE\JIANGXUE\ORADATA\ORCL\CONTROL03.CTL',
      5  'F:\CONTROL04.CTL'
      6  SCOPE = SPFILE;

```

系统已更改。

```
SQL> shutdown immediate;
数据库已经关闭。
已经卸载数据库。
ORACLE 例程已经关闭。
```

执行完上述操作后，还需要复制一个现有的控制文件用来形成新的控制文件，其实现的具体命令如下：

```
host copy d:\ORACLE\JIANGXUE\ORADATA\ORCL\CONTROL01.CTL F:\CONTROL04.CTL
SQL> host copy d:\ORACLE\JIANGXUE\ORADATA\ORCL\CONTROL01.CTL F:\CONTROL04.CTL
已复制          1 个文件。
```

在 F 盘根目录下出现新的控制文件 control04.ctl

 CONTROL04 2012/3/31 11:04

```
SQL> startup
ORACLE 例程已经启动。

Total System Global Area  431038464 bytes
Fixed Size                 1347804 bytes
Variable Size             331353892 bytes
Database Buffers          92274688 bytes
Redo Buffers               6062080 bytes
数据库装载完毕。
数据库已经打开。
```

通过上述操作实现了控制文件的多路复用，并且在复用后可以成功打开数据库。

(2) 删除控制文件并打开数据库

当控制文件丢失或者损坏后，数据库将无法加载。下面讲述在某个控制文件被删除后，正常打开数据库的方法，具体如下所示。

```
SQL> shutdown immediate;
数据库已经关闭。
已经卸载数据库。
ORACLE 例程已经关闭。
SQL> host del f:\control04.ctl
```

```
SQL> startup
ORACLE 例程已经启动。

Total System Global Area  431038464 bytes
Fixed Size                 1347804 bytes
Variable Size             331353892 bytes
Database Buffers          92274688 bytes
Redo Buffers               6062080 bytes
ORA-00205: ??????????, ??????, ????????
```

删除控制文件 control04.ctl 可以看到数据库在加载过程中出现错误。以下讲述当控制文件被删除后，正常打开数据库的方法。

```
SQL> startup nomount;  
ORA-01081: ?????????? ORACLE - ???????
```

```
SQL> alter system set control_files =  
2 'd:\ORACLE\JIANGXUE\ORADATA\ORCL\CONTROL01.CTL',  
3 'D:\ORACLE\JIANGXUE\ORADATA\ORCL\CONTROL02.CTL',  
4 'D:\ORACLE\JIANGXUE\ORADATA\ORCL\CONTROL03.CTL'  
5 scope = spfile;
```

系统已更改。

上述操作，在关闭数据库后，打开数据库例程，同时更改 spfile，使得控制文件为系统中存在的完好无损的控制文件。

为了验证在修改 spfile 后，数据库是否可以成功打开，进行了以下操作。

```
SQL> shutdown immediate;  
ORA-01507: ???????
```

ORACLE 例程已经关闭。

```
SQL> startup  
ORACLE 例程已经启动。  
  
Total System Global Area 431038464 bytes  
Fixed Size 1347804 bytes  
Variable Size 331353892 bytes  
Database Buffers 92274688 bytes  
Redo Buffers 6062080 bytes  
数据库装载完毕。  
数据库已经打开。
```

(3) 备份控制文件

备份控制文件的目的是当某个控制文件丢失或损坏时，可以用这个备份的控制文件来替换。有两种方法可以进行控制文件的备份，具体如下所示。

```
SQL> alter database backup controlfile to 'f:\control.bkp';
```

数据库已更改。

上述操作产生了一个二进制的备份控制文件 control.bkp。当某个控制文件丢失或者损坏时，就可以用这个备份的控制文件来替换。

当然，也可以将控制文件备份为文本文件，也称为跟踪备份，被存放在由初始化参数 user_dump_dest 指定的目录中，备份语句如下所示。

```
SQL> alter database backup controlfile to trace;
```

数据库已更改。

下面查看跟踪备份文件的位置：

```
SQL> show parameter user_dump_dest
```

NAME	TYPE	VALUE
user_dump_dest	string	d:\oracle\jiangxue\diag\rdbms\orcl\orcl\trace

3 日志文件管理

重做日志文件在数据库的恢复中起着非常重要的作用，可以用来进行事务的撤销以及例程恢复。

(1) 添加重做日志文件组

```
SQL> alter database add logfile
2 group 4('f:\redo04_1.log')
3 size 100M;
```

数据库已更改。

以上语句为数据库添加了重做日志文件组 `group4`，但是它只是一个逻辑的概念，重做日志文件才是一个物理的、操作系统的文件。

(2) 添加重做日志文件组成员

```
SQL> alter database add logfile member
2 'f:\redo04_2.log'
3 to group 4;
```

数据库已更改。

上述操作向重做日志组 `group4` 添加成员 `redo04_2.log` 的操作。如果不知道组编号，也可以通过下述操作向组中其他成员的名称来添加新的成员，具体如下：

```
SQL> alter database add logfile member
2 'f:\redo04_3.log'
3 to
4 ('f:\redo04_1.log', 'f:\redo04_2.log');
```

数据库已更改。

通过查询 `v$log` 可以看到添加完成后的结果，如下所示。

```
SQL> select group#,members,status,first_time from v$log;
```

GROUP#	MEMBERS	STATUS	FIRST_TIME
1	1	CURRENT	31-3月 -12
2	1	INACTIVE	27-3月 -12
3	1	INACTIVE	30-3月 -12
4	3	UNUSED	

(3) 删除重做日志文件组

```
SQL> alter database drop logfile group 4;

数据库已更改。

SQL> select group#,members,status from v$log;
```

GROUP#	MEMBERS	STATUS
1	1	CURRENT
2	1	INACTIVE
3	1	INACTIVE

(4) 清除重做日志文件组

在数据库运行过程中，如果重做日志文件受到损坏，数据库就会因无法将损坏的重做日志文件归档而停止，因此通过清除重做日志文件组来避免数据库停止运行的情况，具体操作过程如下。

```
SQL> alter database clear logfile group 3;

数据库已更改。

SQL> select group#,members,status from v$log;
```

GROUP#	MEMBERS	STATUS
1	1	CURRENT
2	1	INACTIVE
3	1	UNUSED

在清除重做日志文件组时，当清除当前的以及活动的重做日志文件组时，会发生错误。从删除重做日志文件组示例的查询中可以看到，group3 的状态为 inactive，所以不会报错。清除后，group3 的状态改为 unused。group3 的组号仍然会被保存在控制文件中，因此还可以继续使用，只是在该组中的成员文件全部被清空了。

可以修改当前工作的重做日志文件组：

```
SQL> alter system switch logfile;

系统已更改。

SQL> select group#,members,status from v$log;
```

GROUP#	MEMBERS	STATUS
1	1	ACTIVE
2	1	INACTIVE
3	1	CURRENT

4 表空间调整

一般情况下，一个数据库有 system,sysaux,temp,undotbs,user 这几个表空间。其中系统数据就存放在表空间 system 以及 sysaux 中；而排序、汇总等产生的临时数据存放在 temp 表空间中；undotbs 表空间用来存放撤销数据，比如回滚操作产生的数据；user 表空间用来存放用户数据。在登录数据库后，用户可以创建自己想要创建的表空间类型。

(1) 创建表空间

```
SQL> conn sys/ora505 as sysdba
已连接。
SQL> create undo tablespace undotbs01
  2  datafile 'f:\undotbs01_1.dbf' size 4M
  3  autoextend on next 1M maxsize unlimited;

表空间已创建。
```

```
SQL> create bigfile tablespace bigtbs01
  2  datafile 'f:\bigtbs01_1.dbf' size 2M
  3  segment space management auto;

表空间已创建。
```

(2) 添加数据文件并使其自动扩展

```
SQL> alter tablespace undotbs01
  2  add datafile 'f:\undotbs01_2.dbf' size 2M;

SQL> alter database datafile 'f:\undotbs01_2.dbf'
  2  autoextend on next 1M maxsize 10M;

数据库已更改。
```


(3) 修改表空间状态

```
SQL> create table personinf
  2  (id int not null,
  3  name varchar2(20))
  4  tablespace bigtbs01;
```

表已创建。

```
SQL> alter tablespace bigtbs01 read only;
```

表空间已更改。

```
SQL> insert into personinf
  2  values(1,'Weihua');
```

```
insert into personinf
```

×

第 1 行出现错误:

ORA-00372: 此时无法修改文件 6

ORA-01110: 数据文件 6: 'F:\BIGTBS01_1.DBF'

```
SQL> alter tablespace bigtbs01 read write;
```

表空间已更改。

```
SQL> insert into personinf
  2  values(1,'Weihua');
```

已创建 1 行。

```
SQL> select * from personinf;
```

ID	NAME
1	Weihua

```
SQL> alter tablespace bigtbs01 offline;
```

表空间已更改。

```
SQL> insert into personinf  
2 values(2,'Binggan');  
insert into personinf
```

×

第 1 行出现错误:

ORA-00376: 此时无法读取文件 6

ORA-01110: 数据文件 6: 'F:\BIGTBS01_1.DBF'

(4) 删除表空间

```
SQL> drop tablespace bigtbs01 including contents and datafiles;
```

表空间已删除。

```
SQL> select tablespace_name from dba_tablespaces;
```

TABLESPACE_NAME

SYSTEM

SYSAUX

UNDOTBS1

TEMP

USERS

UNDOTBS01

已选择6行。