

数据库启停标准操作流程

文档控制

此文档仅供内蒙古移动与东方龙马支持服务部审阅，不得向与此无关的个人或机构传阅或复制。

修改记录

日期	作者	版本	修改记录
2016-01-05	杜诚文	1.0	

分发者

姓名	职位

审阅记录

姓名	职位

目录

文档控制.....	2
修改记录.....	2
分发者.....	2
审阅记录.....	2
1. 流程说明.....	4
1.1. 流程编写目的.....	4
1.2. 相关人员.....	4
1.3. 操作流程.....	6
1.4. 操作提示符.....	6
2. 停数据库操作 (oracle).....	8
2.1. 确认当前登录主机环境.....	8
2.2. 确认操作数据库 SID.....	8
2.3. 停数据库相关应用.....	8
2.4. 停数据库监听.....	8
2.5. 检查应用连接.....	10
2.6. 进行日志切换.....	11
2.7. 停数据库.....	12
2.8. 停数据库集群 (root).....	13
3. 启动数据库操作.....	16
3.1. 启动数据库集群.....	16
3.2. 启动数据库监听.....	17
3.3. 启动数据库.....	18
3.4. 启动数据库 service.....	19
3.5. 进行数据库连接测试.....	19
3.6. 进行应用连接测试.....	19
4. 其他待补充内容.....	20

1. 流程说明

1.1. 流程编写目的

为配合应用变更,主机维护,异常宕机重启等工作,需要进行重启数据库,重启 ORACLE 集群软件,主机。为保障数据库启停操作的规范性,特定制该操作规范。

该规范只针对数据库以及使用 ORACLE 集群软件的集群环境,主机集群、集群文件系统、主机重启等规范由其他应用维护厂家提供。

1.2. 相关人员

1.2.1. 数据库与应用负责人信息

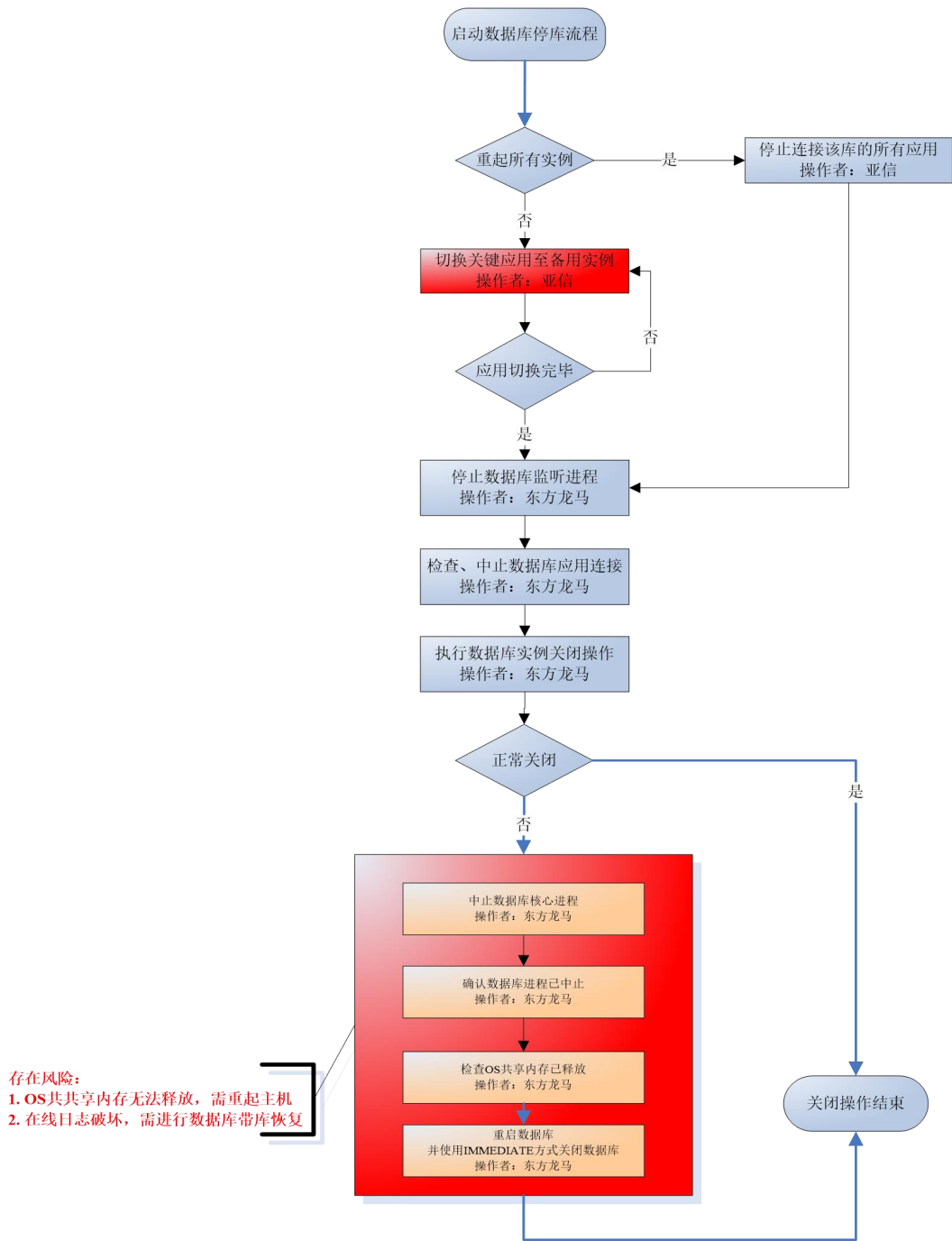
数据库名称	数据库负责人	联系方式	业务负责人	联系方式
ZWDB				
CRMDB				
JFDB				
KFDB				
NGKTDB				
CBOSSDB				
PBOSSDB				
ZWCXDB				

JFCXDB			刘永胜	18888888888
JSDB				
DSJSDB				

1.2.2. 流程相关人员信息

公司	联系人	联系方式	紧急联系人	联系方式
业务支撑室				
HP				
赛门铁克				
亚信				
其他应用厂家				
东方龙马				

1.3. 操作流程



1.4. 操作提示符

\$ 为操作系统命令行提示符，表明该操作命令为普通用户操作系统命令。

为操作用户根用户提示符，表明该操作命令使用 root 用户进行。

SQL> 为 SQL 命令提示符，表明该操作命令为 SQL 命令。

2. 停数据库操作 (oracle)

以下操作，如无特殊说明，均使用 **oracle** 用户进行操作。在进行停监听、停实例、kill 用户进程和数据库进程之前，一定要得到用户以及管理人员的确认，以防止误操作造成的故障。

2.1. 确认当前登录主机环境

确认当前登录主机与操作主机一致。

```
$ hostname
```

2.2. 确认操作数据库 SID

```
$ echo $ORACLE_SID
```

2.3. 停数据库相关应用

该操作由应用厂家完成，数据库中检查类型为 **USER** 的进程是否有减少。

```
SQL> SELECT INST_ID, COUNT(1) FROM GV$SESSION WHERE TYPE = 'USER'  
GROUP BY INST_ID;
```

如果仍然有大量的用户类型的进程存在，检查进程名及主机信息，告知应用厂家进行处理。

```
SQL> SET LINESIZE 120;  
SQL> COL MACHINE FOR A30;  
SQL> SELECT INST_ID, PROGRAM, MACHINE, COUNT(1) FROM GV$SESSION  
WHERE TYPE = 'USER' GROUP BY INST_ID, PROGRAM, MACHINE ORDER BY 4;
```

2.4. 停数据库监听

2.4.1. 非集群环境

非集群环境下，数据库默认的监听名称为 **LISTENER**，可以通过下面的命令查询当前运行的数据库监听的名称：


```
$ ps -ef | grep tns
```

```
/oracle/app/oracle/product/10.2.0/db/bin/tnslsnr LISTENER_RAC10G1 -inherit
```

蓝色部分，就是数据库监听的名称，非标准监听名称，可以通过下面命令进行关闭：

```
$ lsnrctl stop LISTENER_RAC10G1
```

通过

```
$ ps -ef | grep tns
```

命令检查监听是否正常关闭。

如果监听不能正常关闭，也可以使用 `kill -9` 的方式杀掉监听进程。

2.4.2. 集群环境

集群环境下，可以通过常规的 `lsnrctl` 命令关闭监听，也可以通过 `srvctl` 命令或 `crs` 命令的方式关闭监听。但是建议使用 `srvctl` 命令进行资源管理操作。

集群环境中的停监听操作，根据要求停库要求进行对应主机的操作。

a. 查询集群中监听运行状态

```
$ crs_stat -t | grep lsnr
```

ora....G1.lsnr application	ONLINE	ONLINE	rac10g1
ora....G2.lsnr application	ONLINE	ONLINE	rac10g2

b. 停止节点监听

根据需要，停止相应数据库主机的监听程序

```
$ srvctl stop listener -n rac10g1
```

c. 停止 SCAN_LISTENER (11g)

ORACLE 11G 中引入了 SCAN_LISTENER 的概念，应该如果需要停止 11g 集群的所有实例，需要停止 SCAN_LISTENER。

```
$ srvctl stop scan_listener
```

d. 查询集群中监听运行状态

确保操作节点的监听被停止。

```
$ crs_stat -t | grep lsnr
```

ora....G1.lsnr application	OFFLINE	OFFLINE	
ora....G2.lsnr application	ONLINE	ONLINE	rac10g2

➤ 其他停止监听的方式:

- a. 通过 `lsnrctl` 命令方式停监听，该操作需要在集群中所有节点进行

```
$ ps -ef | grep tns  
/oracle/app/oracle/product/10.2.0/db/bin/tnslsnr LISTENER_RAC10G1 -inherit
```

```
$ lsnrctl stop LISTENER_RAC10G1
```

- b. 通过 `crs_stop` 命令停监听

```
$ crs_stat | grep lsnr  
NAME=ora.rac10g1.LISTENER_RAC10G1.lsnr  
NAME=ora.rac10g2.LISTENER_RAC10G2.lsnr
```

```
$ crs_stop
```

```
Usage:  crs_stop resource_name [...] [-f] [-q] ["attrib=value ..."]  
        crs_stop -c cluster_member [...] [-q] ["attrib=value ..."]  
        crs_stop -all [-q]
```

该命令使用的参数为集群资源，因此参数为 `NAME` 说对应的整个值

```
$ crs_stop ora.rac10g1.LISTENER_RAC10G1.lsnr
```

2.5. 检查应用连接

检查数据库的非本地会话连接，该操作所有节点都需要进行，**需要注意进程运行的用户**:

```
$ ps -ef | grep "LOCAL=NO"
```

如果存在大量的进程，表明仍有大量应用进程没有停止，通过下面的命令进行检查:

```
SQL> SET LINESIZE 120;  
SQL> COL MACHINE FOR A30;  
SQL> SELECT INST_ID, PROGRAM, MACHINE, COUNT(1) FROM GV$SESSION  
WHERE TYPE = 'USER' GROUP BY INST_ID, PROGRAM, MACHINE ORDER BY 4;
```

可以通过 `kill -9` 的方式，杀掉非本地连接(**该操作需要应用厂家确认后才能进行**):

```
$ hostname  
$ id  
$ ps -ef | grep "LOCAL=NO" | awk '{print $2}' | xargs kill -9
```

注意：在 ORACLE11G 的版本，由于监听程序运行在 grid 用户下，因此 LOCAL=NO 的进程属组可能是 grid 用户。在需要之前需要进行确认。

2.6. 进行日志切换

为保证在数据库关闭过程中，所有的脏数据都写入磁盘，日志都完成归档操作。建议在执行关闭数据库操作前，进行日志切换和检查点操作。

将集群中所有节点的脏数据写入到数据文件：

```
SQL> alter system checkpoint global;
```

global 参数针对集群数据库，单实例数据库无需该参数。

对集群中所有实例，进行日志切换操作，日志切换操作建议执行多次：

```
SQL> alter system archive log current;
```

在非归档模式下，该命令无法正常执行，可以通过在各节点执行：

```
SQL> alter system switch logfile;
```

命令进行替代。

再次执行检查点操作：

```
SQL> alter system checkpoint global;
```

确保停库时，ACTIVE 状态的日志不大于 1 组：

```
SQL> SELECT INST_ID, GROUP#, STATUS, ARCHIVED FROM GV$LOG WHERE  
STATUS IN ('ACTIVE', 'CURRENT');
```

确保 MTTR 时间在可接受的范围内（数据库异常关闭，启动后恢复所需时间）：

```
SQL> SET LINESIZE 120;
```

```
SQL> SELECT  
RECOVERY_ESTIMATED_IOS,ACTUAL_REDO_BLKs,TARGET_REDO_BLKs,LOG_F  
ILE_SIZE_REDO_BLKs,TARGET_MTTR,ESTIMATED_MTTR FROM  
GV$INSTANCE_RECOVERY;
```

2.7. 停数据库

2.7.1. 非集群环境

通过 `immediate` 的方式，进行数据库关闭操作

```
SQL> shutdown immediate;
```

2.7.2. 集群环境

➤ 停实例

在停止部分实例的情况下，需要使用停实例的方式来停止指定的数据库实例。

```
$ hostname
$ id
$ ps -ef | grep ora_smon

$ srvctl stop instance -d <DBNAME> -i <INSTANCE_NAME>
```

➤ 停数据库

通过 `srvctl` 命令方式关闭数据库：

```
$ crs_stat | grep db$
NAME=ora.crmdb.db

$ srvctl stop database -d crmdb
```

通过 `crs_stat` 命令检查数据库是否正常关闭，`inst` 及 `db` 资源就是对应的实例和数据库资源：

```
$ crs_stat -t
```

Name	Type	Target	State	Host
ora....b1.inst	application	OFFLINE	OFFLINE	
ora....b2.inst	application	OFFLINE	OFFLINE	
ora.crmdb.db	application	OFFLINE	OFFLINE	

注意：ORACLE 11G 版本中，`crs` 资源中已经不在有实例的信息。

2.7.3. 执行 `abort` 操作

注意，该操作可能会造成数据库的损坏，非特色情况下不建议采用该操作。

如果上述步骤无法正常关闭数据库，可采用中止 **PMON** 进程或 **shutdown abort** 方式非正常中止数据库运行，数据库实例的核心进程全部关闭后，注意检查操作系统的共享内存段是否已释放。在共享内存段释放前不要重新启动数据库。

```
SQL> shutdown abort;
```

```
$ ipcs -m | grep oracle
```

```
0x61a4a848 3047431      oracle      640          132120576  14
```

如果返回结果中仍然有 **oracle** 的共享内存信息，表明数据库没有完全停止，检查是否仍然有遗留进程，或者等待共享内存资源的释放。

在执行 **abort** 操作后，一定要执行一次正常的数据库起停操作，防止异常操作带来的数据损坏

```
SQL> startup;
```

```
SQL> shutdown immediate;
```

2.8. 停数据库集群 (root)

停数据库集群需要使用 **root** 用户。该操作需要在集群中所有节点进行。

执行该操作前，需要检查数据库是否正常停止：

```
$ crs_stat -t
```

确保，资源中以 **inst/db** 结尾的资源状态已经为 **OFFLINE**。如果状态为 **ONLINE**，请参照前面的停数据库步骤进行操作。

集群的停止操作，需要在 **CRS_HOME(10g)/GRID_HOME(11g)**的 **bin** 目录下进程：

```
# cd /oracle/app/oracle/product/10.2.0/crs/bin
```

```
# ./crsctl stop crs
```

```
Stopping resources. This could take several minutes.
```

```
Successfully stopped CRS resources.
```

```
Stopping CSSD.
```

```
Shutting down CSS daemon.
```

```
Shutdown request successfully issued.
```

检查关闭结果：

```
# crsctl check crs
```

```
Failure 1 contacting CSS daemon
```

```
Cannot communicate with CRS
```

```
Cannot communicate with EVM
```

确保该命令没有返回结果：

```
# ps -ef | grep oracle | grep d.bin
```

2.9. 操作脚本示例

以 CRMDB 为例：

2.9.1. 确认数据库及主机信息

```
$ hostname
```

```
SQL> show parameter instance;
```

2.9.2. 检查应用进程信息

```
SQL> SELECT INST_ID, PROGRAM, MACHINE, COUNT(1) FROM GV$SESSION  
WHERE TYPE = 'USER' GROUP BY INST_ID, PROGRAM, MACHINE ORDER BY 4;
```

2.9.3. 停数据库监听

```
$ crs_stat | grep lsnr
```

```
$ srvctl stop listener -n crmdb1
```

```
$ srvctl stop listener -n crmdb2
```

```
$ srvctl stop listener -n crmdb3
```

2.9.4. 杀非本地连接进程

```
$ ps -ef | grep "LOCAL=NO" | awk '{print $2}' | xargs kill -9
```

2.9.5. 切换数据库日志

```
SQL> alter system checkpoint global;
```

```
SQL> alter system archive log current;
```

```
SQL> alter system archive log current;
```

```
SQL> alter system archive log current;
```

```
SQL> alter system archive log current;
```

```
SQL> alter system archive log current;
```

```
SQL> alter system archive log current;
```

```
SQL> alter system checkpoint global;
```

2.9.6. 停数据库

```
$ crs_stat | grep db$
```

```
$ srvctl stop database -d crmdb
```

2.9.7. 停集群

```
# crsctl stop crs
```

3. 启动数据库操作

以下操作，如无特殊说明，均使用 **oracle** 用户进行操作。正常情况下在集群启动会，会将所有相关资源自动启动。但可能存在特殊情况，如资源启动状态设置为 **disable**、手工停止集群未重启主机的情况，需要手工进行资源的启动操作。

3.1. 启动数据库集群

启动数据库集群需要使用 **root** 用户。该操作需要在集群中所有节点进行。

命令的执行，需要在 **bin** 目录下进行：

```
# cd /oracle/app/oracle/product/10.2.0/crs/bin
[root@rac10g1 bin]#
```

```
[root@rac10g1 bin]# crsctl start crs
Attempting to start CRS stack
The CRS stack will be started shortly
```

该过程可能时间较长，可以通过监控集群相关的进程以及 **crs** 日志进行跟踪：

```
[root@rac10g1 bin]# ps -ef | grep d.bin
```

```
[root@rac10g1 bin]# cd ../log/rac10g1/crsd/
[root@rac10g1 crsd]# tail -f crsd.log
```

集群启动成功后，可以通过 **crs_stat** 命令查看集群状态：

```
# crs_stat -t
```

Name	Type	Target	State	Host
ora....0g1.gsd application		ONLINE	ONLINE	rac10g1
ora....0g1.ons application		ONLINE	ONLINE	rac10g1
ora....0g1.vip application		ONLINE	ONLINE	rac10g1

集群启动成功后，要确保集群资源中的 **gsd/ons/vip** 进程状态为 **ONLINE**，并且运行在自己的节点上。

3.2. 启动数据库监听

3.2.1. 非集群环境

非集群环境下，数据库默认的监听名称为 LISTENER，可以通过 listener.ora 配置文件，查看监听的名称：

```
$ cd $ORACLE_HOME/network/admin
[oracle@rac10g1 admin]$

$ cat listener.ora

LISTENER_RAC10G1 =
  (DESCRIPTION_LIST = .....
  )
```

通过 lsnrctl start 命令进行监听的启动：

```
$ lsnrctl start LISTENER_RAC10G1
```

通过 lsnrctl status 命令，检查监听的运行状态：

```
$ lsnrctl status LISTENER_RAC10G1
```

3.2.2. 集群环境

集群环境下，可以通过 srvctl 命令进行监听的启动。

查询集群中监听的名称：

```
$ crs_stat | grep lsnr
NAME=ora.rac10g1.LISTENER_RAC10G1.lsnr
NAME=ora.rac10g2.LISTENER_RAC10G2.lsnr
```

执行 srvctl 命令

```
$ srvctl start listener -h
Usage: srvctl start listener -n <node_name> [-l <lsnr_name_list>]

  -n <node>           Node name
  -l "<lsnr,...>"      Comma separated listener names
  -h                  Print usage
```

```
$ srvctl stop listener -n rac10g1 -l LISTENER_RAC10G1
```

3.3. 启动数据库

3.3.1. 非集群环境

通过 open 的方式，该方式是 startup 的默认方式，进行数据库启动操作：

```
SQL> startup;
```

3.3.2. 集群环境

通过 srvctl 命令方式启动数据库：

```
$ srvctl start database -d crmdb
```

检查各节点的 alert 日志，跟踪数据库关闭过程：

```
$ cd $ORACLE_BASE/admin/${DB_NAME}/bdump
```

```
$ tail -f alert_crmdb1.log
```

`${DB_NAME}` 参数为数据库名。

通过 crs_stat 命令检查数据库是否正常启动，inst 及 db 资源就是对应的实例和数据库资源：

```
$ crs_stat -t
```

Name	Type	Target	State	Host
ora....b1.inst	application	ONLINE	ONLINE	rac10g1
ora....b2.inst	application	ONLINE	ONLINE	rac10g2
ora.crmdb.db	application	ONLINE	ONLINE	rac10g2

3.3.3. 运行状态检查

确保每个实例的运行状态都为 OPEN 状态：

```
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY-MM-DD HH24:MI:SS';
```

```
SQL> SELECT INST_ID, STARTUP_TIME, STATUS FROM GV$INSTANCE ORDER BY  
INST_ID;
```

3.4. 启动数据库 service

查看当时数据库配置的所有 service

```
$ srvctl config service -d crmdb  
srvcrmdb1 PREF: crmdb1 AVAIL: crmdb2  
srvcrmdb2 PREF: crmdb2 AVAIL: crmdb1
```

通过 srvctl 命令启动 service:

```
$ srvctl start service -d crmdb -s srvcrmdb1  
$ srvctl start service -d crmdb -s srvcrmdb2
```

查看 service 的运行状态:

```
$ srvctl status service -d crmdb  
Service srvcrmdb1 is running on instance(s) crmdb1  
Service srvcrmdb2 is running on instance(s) crmdb2
```

3.5. 进行数据库连接测试

通过 vip 地址和 scan(11g)地址, 进行数据库远程登录测试, 确保所有实例都能够登录成功。

```
$ sqlplus system/password@rac10g1-vip:1521/crmdb  
$ sqlplus system/password@rac10g2-vip:1521/crmdb  
  
$ sqlplus system/password@scan_ip:1521/crmdb
```

3.6. 进行应用连接测试

由亚信进行应用连接测试。

4. 其他待补充内容

4.1. 启动 OSW