

添中 redo日志组需求说明

目 录

[一、 文档概述](#)

[1. 需求背景](#)

[2. 需求目标](#)

[二、 脚本功能](#)

[1. 转移service脚本流程概述](#)

[2. 转移service逻辑处理](#)

[3. 转移service验收标准](#)

[三、 非功能需求](#)

[四、 风险分析](#)

[五、 其他说明](#)



文档概述

1. 需求背景

在Oracle数据库中，有一种日志文件叫做在线重做日志文件(online redo log)，它主要用于Oracle数据库所在服务器突然掉电、突然重启或者执行shutdown abort等命令使得在服务器重新启动之后，Oracle数据库没有办法正常的启动实例。此时，在线重做日志就派上了用场，Oracle会使用在线重做日志，把数据库恢复到服务器掉电前的那一个时刻，从而使得数据库能正常的启动起来。

在Oracle数据库中，默认情况下，至少会有两个重做日志组，而且每个组里面至少包含了一个重做日志文件。日志组不会自动增加，在一个写满之后，会自动去写下一个。在下一个被写满之后会又从第一个开始写起。在某些情况下，业务应用的提交事务与之前相比有所增加，此时，我们需要创建另外的redo日志组来缓减数据库的压力。

2. 需求目标

1) 从cmdb获取系统信息得到数据库ip和数据库名

2) 【架构检查】

a. 勾选需要操作的数据库

b. 校验勾选的数据库 是否属于同一套

a) 如果不是：程序退出，提示用户勾选的数据库不属于同一套

b) 如果是：将数据库的信息展示到页面上

3) 【场景检查】

a. 检查所选节点的数据库架构

a) 判断数据库角色是主库还是备库：

判断逻辑，

在oracle用户下执行SQL命令select database role from v\$database;如果返回PRIMARY, 然后查询下面的SQL看是否存在备库的**TNS名**, 如果存在, 则为DG架构中的****主库**, **否则不是DG架构;

```
select substr(value, instr(value, '=')+1, instr(value, ' ')-instr(value, '=')) value
```

```
from v$parameter
```

```
where name not like 'log_archive_dest_state_%'
```

```
and name like 'log_archive_dest_%'
```

```
and value is not null
```

```
and substr(lower(value), 1, 7) = 'service';
```

```
VALUE
```

yydsbdbg07 -- 有多个值返回, 说明该DG架构中存在多个备库;

yydsbdbg40

如果返回PHYSICAL STANDBY, 为备库;

如果是备库, 直接退出; 如果是主库, 进行下一步的判断;

b) 判断数据库是集群, 单机ASM, 还是单实例(集群通过数据库参数cluster_database来识别, 单机ASM和单实例通过数据库参数cluster_database和ps -ef | grep ASM来识别);

b. 判断勾选的实例个数是全部还是部分，根据用户勾选情况来确定；【强制按照选择一个节点来做】；

c. 展示所选节点的**实例名**和**数据库唯一名**信息：

```
set linesize 200
```

```
col name for a15
```

```
col value for a15
```

```
select name, value from v$parameter where name in ('instance_name', 'db_unique_name') order  
by name desc;
```

d. 展示所选节点的****redo/standby redo**日志组信息**【兼容11g/12c/19c】：

```
-- 单实例**/****单机A****SM/RA**C:
```

```
**查看****redo**日志组信息,
```

```
set linesize 200
```

```
set pagesize 100
```

```
column member format a95
```

```
select l.thread#,
```

```
l.group#,
```

```
listagg(lf.member, ', ' ) within group (order by lf.member) as member,
```

```
l.bytes/1024/1024 size_mb,
```

```
l.members,
```

```
l.status
```

```
from v$logl, vlogfile lf
```

```
where l.group# = lf.group#
```

```
group by l.thread#,
```

```
l.group#,
```

```
l.members,
```

l.bytes,

l.status

order by 1, 2;

-- **D**ata **G*uard*(****主: R****AC-多节点, 备: ****R****AC-多节点/****单机
A****SM/**单实例):

-- **D**ata **G*uard*(主: 单实例, 备: **单实例):

a) 在主库** (****R*AC任意一个实例*/****单实例) **上操作:

查看**redo**日志组信息,

set linesize 200

set pagesize 100

column member format a95

select l.thread#,

l.group#,

listagg(lf.member, ', ') within group (order by lf.member) as member,

l.bytes/1024/1024 size_mb,

l.members,

l.status

from vlogl, vlogfile lf

where l.group# = lf.group#

group by l.thread#,

l.group#,

l.members,

l.bytes,

l.status

order by 1, 2;

****查看****standby redo**日志组信息,**

```
set linesize 200
```

```
set pagesize 100
```

```
column member format a95
```

```
select l.thread#,
```

```
       l.group#,
```

```
       listagg(lf.member, ', ') within group (order by lf.member) as member,
```

```
       l.bytes/1024/1024 size_mb,
```

```
       l.status
```

```
from vstandby_l, vlogfile lf
```

```
where l.group# = lf.group#
```

```
group by l.thread#,
```

```
       l.group#,
```

```
       l.bytes,
```

```
       l.status
```

```
order by 1, 2;
```

b) 在备库** (从主库直接连到备库) **上操作:

注, 主库通过**备库的****T**NS名直接连到备库, 如:

```
[oracle@yyds01 ~]$
```

```
[oracle@yyds01 ~]$ sqlplus sys@yydsdbdg07 as sysdba
```

```
SQL*Plus: Release 19.0.0.0.0 - Production on Thu Feb 23 13:47:18 2023
```

```
Version 19.13.2.0.0
```

```
Copyright (c) 1982, 2021, Oracle. All rights reserved.
```

```
Enter password: ** (需要输入sys*用户的密码*) **
```

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

13:47:23 SYS@yydsdbdg07/yyds03>

****查看****redo**日志组信息,**

set linesize 200

set pagesize 100

column member format a95

select l.thread#,

l.group#,

listagg(lf.member, ', ') within group (order by lf.member) as member,

l.bytes/1024/1024 size_mb,

l.members,

l.status

from vlogl, vlogfile lf

where l.group# = lf.group#

group by l.thread#,

l.group#,

l.members,

l.bytes,

l.status

order by 1, 2;

****查看****standby redo**日志组信息,**

set linesize 200

```

set pagesize 100

column member format a95

select l.thread#,
       l.group#,
       listagg(lf.member, ', ') within group (order by lf.member) as member,
       l.bytes/1024/1024 size_mb,
       l.status
from vstandby_l, vlogfile lf
where l.group# = lf.group#
group by l.thread#,
         l.group#,
         l.bytes,
         l.status
order by 1, 2;

```

4) 【参数输入】

数据行数* (可根据具体的数据库架构情况添加多行***) **, 每一行的字段信息如下:

- a) ip: 不可编辑 (从上一步的展示信息中带入)
- b) 实例名: 不可编辑 (从上一步的展示信息中带入)
- c) 数据库唯一名: 不可编辑 (从上一步的展示信息中带入)
- d) 架构类型: 不可编辑 (从上一步的展示信息中带入, 可能存在的类型有: 单实例, 单机ASM, 集群, DG+单实例, DG+单机ASM, DG+集群)
- e) 操作: 下拉可选, 添加redo日志组
- f) thread号: 下拉可选 (redo日志组所在实例, 从上一步的展示信息中带入, **默认值为1**)
- g) redo日志组文件类型: 下拉可选 **** (如果是DG架构, 列出redo log或standby redo log; 如果是其他架构, 只列出redo log)****

h) redo日志组序号: 手动输入 (从上一步展示的日志组信息的最大日志组序号之后开始)

i) redo日志组的大小*(MB)*: 下拉可选 (从上一步展示的日志组信息中带入, 默认值为带入值的最大的大小, 有可能redo日志组有不同的大小, **强烈建议选择默认值**)

j) redo日志组的路径: 支持 **下拉可选** 和 **手动输入**【考虑标准化: 日志组的大小是否相等, 日志组的路径是否一致】

5) 【方案预览】

a. 参数检查:

a) 检查 多行之间的 ****r***edo**日志组序号** 和 ****redo**日志组的路径** 是否重复;

b) 检查 redo日志组的路径 是否已经存在;

c) 如果是DG架构, redo日志组文件类型必须包括redo log和standby redo log;

d) 检查 redo日志组所在磁盘的空间 是否足够;

b. 生成方案:

添加redo日志组(兼容11g/12c/19c)【每组添加一个成员】:

a. 单实例:

```
alter database add logfile thread 1 group 5 '/xxx/redo5a.log' size 1024m;
```

b. 单机ASM:

```
alter database add logfile thread 1 group 5 '+data/rac12c22/redo05.log' size 1024m;
```

c. ****RAC(多节点)【根据节点的个数N,thread号从1开始,直到N,group号从前面查出的最大的加1开始】: ****

****如果***R**AC是两节点, 为:**

```
alter database add logfile thread 1 group 5 '+data/rac12c22/redo05.log' size 1024m;
```

```
alter database add logfile thread 2 group 6 '+data/rac12c22/redo06.log' size 1024m;
```

****如果***R***AC是三节点, **为:**

```
alter database add logfile thread 1 group 5 '+data/rac12c22/redo05.log' size 1024m;
```

```
alter database add logfile thread 2 group 6 '+data/rac12c22/redo06.log' size 1024m;
```



```
alter database add logfile thread 3 group 7 '+data/rac12c22/redo07.log' size 1024m;
```

d. Data Guard(主: RAC**-****多*节点*, 备: RAC****-多节点/单机ASM/**单实例):

****如果R**AC是两节点, 为:**

```
alter database add logfile thread 1 group 5 '+data/rac12c22/redo05.log' size 1024m;
```

```
alter database add logfile thread 2 group 6 '+data/rac12c22/redo06.log' size 1024m;
```

```
alter database add standby logfile thread 1 group 15 '+data/rac12c22/redo15.log' size 1024m;
```

```
alter database add standby logfile thread 2 group 16 '+data/rac12c22/redo16.log' size 1024m;
```

直连到备库:

a. 取消日志实时应用(无论当前日志实时应用进程在哪个实例,在任意一个实例上都可以取消):

```
alter database recover managed standby database cancel;
```

b. 参数standby_file_management改为手动模式:

```
alter system set standby_file_management=manual;
```

c. 增加redo日志组:

```
alter database add logfile thread 1 group 5 '+data/rac12c22/redo05.log' size 1024m;
```

```
alter database add logfile thread 2 group 6 '+data/rac12c22/redo06.log' size 1024m;
```

```
alter database add standby logfile thread 1 group 15 '+data/rac12c22/redo15.log' size 1024m;
```

```
alter database add standby logfile thread 2 group 16 '+data/rac12c22/redo16.log' size 1024m;
```

d. 参数standby_file_management改为自动模式:

```
alter system set standby_file_management=auto;
```

e. 开启日志实时应用:

```
alter database recover managed standby database using current logfile disconnect from session;
```

注: 如果主备为3个节点的RAC, 那么,

主库的logfile和standby logfile需要增加3个, thread号分别为1,2,3;

备库的logfile和standby logfile也需要增加3个, thread号分别为1,2,3.****

e. Data Guard(主：单实例，备：单实例)：

```
alter database add logfile thread 1 group 5 '/xxx/redo5a.log' size 1024m;
```

```
alter database add standby logfile thread 1 group 15 '/xxx/redo15a.log' size 1024m;
```

直连到备库：

a. 取消日志实时应用(无论当前日志实时应用进程在哪个实例,在任意一个实例上都可以取消)：

```
alter database recover managed standby database cancel;
```

b. 参数standby_file_management改为手动模式：

```
alter system set standby_file_management=manual;
```

c. 增加redo日志组：

```
alter database add logfile thread 1 group 5 '/xxx/redo5a.log' size 1024m;
```

```
alter database add standby logfile thread 1 group 15 '/xxx/redo15a.log' size 1024m;
```

d. 参数standby_file_management改为自动模式：

```
alter system set standby_file_management=auto;
```

e. 开启日志实时应用：

```
alter database recover managed standby database using current logfile disconnect from session;
```

6) 【执行变更】

a. 执行方案预览生成的命令；

b. 查看redo和standby redo日志组信息,以验证添加新的redo和standby redo日志组是否成功；

-- 单实例/单机ASM/RAC:

查看redo日志组信息,以验证添加新的redo日志组是否成功：

```
set linesize 200
```

```
set pagesize 100
```

```
column member format a95
```

```
select l.thread#,
```

```
l.group#,
```

```
listagg(lf.member, ', ') within group (order by lf.member) as member,
```

```
l.bytes/1024/1024 size_mb,
```

```
l.members,
```

```
l.status
```

```
from v$logl, vlogfile lf
```

```
where l.group# = lf.group#
```

```
group by l.thread#,
```

```
l.group#,
```

```
l.members,
```

```
l.bytes,
```

```
l.status
```

```
order by 1, 2;
```

```
-- Data Guard(主: RAC-多节点, 备: RAC-多节点/单机ASM/单实例):
```

```
-- Data Guard(主: 单实例, 备: 单实例):
```

查看redo日志组信息,以验证添加新的redo日志组是否成功:

```
set linesize 200
```

```
set pagesize 100
```

```
column member format a95
```

```
select l.thread#,
```

```
l.group#,
```

```
listagg(lf.member, ', ') within group (order by lf.member) as member,
```

```
l.bytes/1024/1024 size_mb,
```

```
l.members,
```

```
l.status  
  
from vlogl, vlogfile lf  
  
where l.group# = lf.group#  
  
group by l.thread#,  
  
l.group#,  
  
l.members,  
  
l.bytes,  
  
l.status  
  
order by 1, 2;
```

查看standby redo日志组信息,以验证添加新的standby redo日志组是否成功:

```
set linesize 200  
  
set pagesize 100  
  
column member format a95  
  
select l.thread#,  
  
l.group#,  
  
listagg(lf.member, ', ') within group (order by lf.member) as member,  
  
l.bytes/1024/1024 size_mb,  
  
l.status  
  
from vstandbylogl, vlogfile lf  
  
where l.group# = lf.group#  
  
group by l.thread#,  
  
l.group#,  
  
l.bytes,  
  
l.status
```

order by 1, 2;

二、脚本功能

1. ****添加****redo**日志组脚本流程概述**

1) 【架构检查脚本】

根据勾选的数据库信息，判断是否所勾选的数据是否属于同一套数据库

2) 【场景检查脚本】

a. 判断勾选节点的数据库架构

b. 判断勾选的实例个数

c. 展示所选节点的实例名和数据库唯一名信息

d. 展示所选节点的redo/standby redo日志组信息

3) 【方案预览脚本】

根据前端页面的参数选择，生成添加redo日志组的操作命令

4) 【执行变更脚本】

a. 执行"方案预览"脚本生成的命令

b. 查看redo和standby redo日志组信息,以验证添加新的redo和standby redo日志组是否成功

2. ****添加****redo**日志组逻辑处理**

1) 【架构检查脚本】

a. 根据第一个勾选数据库ip下发脚本

b. 通过gv\$instance查出当前集群的数据库信息

c. 通过tnsname远程连接备库，查出备库的集群信息

d. 将得到的数据库信息与页面勾选的数据库信息比对判断

a) -- 如果勾选的数据库信息不属于同一套：页面报错，日志打印错误信息

b) -- 如果勾选的数据库信息属于同一套：勾选的数据库服务器下发场景检查脚本

2) 【场景检查脚本】

a. 判断勾选节点的数据库架构

-- 如果主备库同时勾选：页面报错，日志打印提示，添加redo日志组，主备库不能同时勾选

-- 如果勾选备库：页面报错，日志打印提示，添加redo日志组，需要直接在主库上操作，

b. 判断勾选的实例个数

-- 如果勾选的实例个数大于1，页面报错，日志打印提示，添加redo日志组，只能选择1个实例

c. 展示所选节点的实例名和数据库唯一名信息

d. 展示所选节点的redo/standby redo日志组信息

e. 参数输入

3) **【方案预览脚本】**

生成添加redo日志组的操作命令

4) **【执行变更脚本】**

a. 执行方案预览生成的命令

b. 查看redo和standby redo日志组信息

3. **添加****redo**日志组验收标准

通过架构检查，场景检查，方案预览，执行变更，能够成功添加新的redo日志组。

三、非功能需求

1) 脚本的执行不能影响系统业务

2) 脚本不能重复执行

四、风险分析

连接数据库后的操作无明显风险。

五、其他说明

无