

Oracle Life DBA的一天

Oracle Life DBA的一天

1. 活动状态检查

通过查询基本视图，确认数据库和实例处于正常运行状态，可以对外提供数据服务。

1) 连接用户

```
select inst_id, username, count(*) from gv$session group by inst_id, username;
```

查询数据库以用户分组连接数

2) 实例状态

```
select instance_name, status from v$instance;
```

查询实例名称，状态，正常状态应为Open.

3) 会话信息

```
select sessions_current, sessions_highwater from v$license;
```

辅助查询，实例当前会话数和启动最高连接会话数量

```
select inst_id, username, count(*) from gv$session group by inst_id, username;
```

查询数据库连接数以实例和用户分组

4) 参数检查

```
select value from v$parameter where name='open_cursors';
```

查询给你参数的值

5) 参数修改

```
alter system set undo_retention=3600 comment= 'default 900' sid='*' scope=both;
```

修改给定的初始化参数，RAC环境需要注意SID参数

6) 隐含参数

```
alter system set "_optimizer_use_feedback"=false scope=spfile;
```

为了解决特殊问题，又是需要设置以下划线开头的隐含参数。

7) 实例异常

当连接数据库实例出现缓慢，挂起等现象，需要进行诊断和分析，甚至可能需要重新启动数据库实例

(1) 信息采集

```
sqlplus -prelim / as sysdba
```

```
oradebug systempid
```

```
oradebug unlimit
```

```
oradebug hanganalyze 3
```

```
oradebug dump systemstate 266
```

间隔一定时间（20秒），进行下一次数据采样。

```
oradebug hanalyze 3
```

```
oradebug dump systemstate 266
```

通过采集系统的Hang信息，系统状态信息等，可以分析系统挂起的原因，间隔采样，可以用于对比变化，辅助分析。

(2) 跟踪

```
alter session set events '10046 trace name context forever, level 12';
```

```
shutdown immediate;
```

```
startup mount
```

```
alter session set events '10046 trace name context forever, level 12';
```

```
alter database open;
```

如果在数据库关闭，启动时遇到阻塞，挂起，可以通过师范命令进行跟踪，获取跟踪文件进行分析。

(3) 安全停库

```
alter system checkpoint;
```

```
alter system archive log current;
```

```
shutdown immediate
```

如果数据库出现异常需要重新启动，可以执行检查点，归档命令，然后尝试正常关闭数据库

(4) 强制停库

```
shutdown abort
```

```
startup nomount
```

```
alter database mount;
```

```
alter database open
```

如果不能正常关闭数据库，强制关闭

8) 连接异常

当连接数据库出现异常，需要监测包括网络连通性，监听器状态的等信息。

(1) 连通性

tnsping tns_name

检查网络是否连通以及响应时间

(2) 监听器

lsnrctl status LISTENER

lsnrctl status LISTENER_SCAN1

lsnrctl service

在数据库服务器上，通过lsnrctl工具检查监听状态和服务信息。

(3) 监听日志检查

showalertert

通过adrci工具，显示各类告警文件，检查监听器日志，诊断监听问题

2. 日志信息检查

检查数据库各类日志信息，确认数据库实例，集群等是否出现错误，告警，如存在问题，则需要进一步分析和应对。

1) 告警日志

\$ORACLE_BASE/diag/rdbms/db_name/\$ORACLE_SID/trace/alert_\$ORACLE_SID.log

show parameter background_dump_dest

找到告警日志，检查实例是否存在ORA-错误提示等。

2) 集群日志

\$GRID_HOME/log/node_name/alertnode_name.log

\$GRID_HOME/log/node_name/crsd, cssd, evmd, ohasd/

3) ASM日志

\$GRID_HOME/diag/asm/+asm/ASM_INSTANCE_NAME/trace/alert_instance_name.log

4) Trace文件

select value from v\$diag_info where name='Default Trace File';

show parameter user_dump_dest

5) 集群状态

crsctl status resource -t

确保资源状态显示在线

6) errorstack分析

当遇到ORA-错误，而数据库的输出信息不足时，可以采用errorstack进行跟踪，采集更详细的转储信息

alter system set events='600 trace name errorstack forever, level 10';

alter system set events='600 trace name errorstack off';

对ORA-600错误设置跟踪，并关闭。

3. 重做日志维护

Oracle REDO日志时数据库的核心组件，检查其状态，维护其成员，监控其归档，审核其性能，时DBA的重要工作。

1) REDO组

select group#, sequence#, archived, status from v\$log;

查询日志组号，序号，是否归档完成和状态信息，如多组日志显示ACTIVE状态，则可能说明数据库存在IO方面的性能问题。

2) REDO成员

select group#, member from v\$logfile;

查看日志组和成员信息。

3) 增加日志组或成员

alter database add logfile group 10 ('/oracle/dbs/log1c.rdo', '/oracle/dbs/log2c.rdo') size 500M;

alter database add logfile member '/oracle/dbs/log3c.rdo' to group 10;

在日志切换频繁时，可能需要增加日志组或者加大日志大小。

4) 切换日志

alter system switch logfile;

切换日志组，开始写入下一个日志组。

5) 执行归档

alter system archive log current;

对当前日志组执行归档，切换到下一个日志组，

在RAC环境下会对所有实例执行归档，Thread参数制定归档实例。

6) 删除日志组或成员

alter database drop logfile group 10;

alter database drop logfile member '/oracle/dbs/redo03.log';

删除制定日志组或日志成员，注意只能对INACTIVE状态的日志执行删除操作

7) 归档检查

```
archive log list;
```

8) 归档状态变更

```
alter database archive log | noarchive log;
```

在MOUNT状态改变归档模式，注意启动归档模式之后务必制定备份归档的日常策略，防止磁盘空间被耗尽。

9) 调整归档路径

```
alter system set log_archive_dest_2='location=&path' sid='&sid';
```

如果数据库因归档耗尽空间，可以制定另外的归档路径，以尽快归档日志，回复数据库运行。

4. 空间信息检查

确保数据存储空间可用，定期检查表空间余量，进行表空间和文件维护。

1) 空间使用查询

```
select * from sys.sm$ts_used;
```

查看数据库表空间的使用信息

```
select * from sys.sm$ts_free;
```

查看数据库表空间的剩余空间

2) 文件信息

```
select tablespace_name, file_name from dba_data_files;
```

查看数据库表空间和数据文件信息

3) 文件维护

```
alter database datafile '&path' resize 900M;
```

```
alter database &tb_name add datafile '&path' size 900M;
```

对数据库的表空间容量进行扩容

5. 锁 门信息检查

Lock/Latch时数据库控制并发的核心手段，检查相关信息可以监控数据库的事物和运行状况

1) 锁信息

```
select sid, type, lmode, ctime, block from v$sqllock where type not in ('MR', 'AE');
```

查看锁回话ID，类型，持有时间等，如果block=1，可能意味着阻塞了其他会话。

2) 锁故障排查

在数据库出现锁竞争和阻塞时，需要排查和处理锁定，必要时通过kill阻塞进程消除锁定。

(1) 查询阻塞会话

```
select sid, sql_id, status, blocking_session from v$session where sid in (select session_id from v$locked_object);
```

查询当前锁事物中阻塞会话与被阻塞会话的sid, sql_id和状态信息

(2) 阻塞SQL文本

```
select sql_id, sql_text from v$sqltext where sql_id='&sql_id' order by piece;
```

通过sql_id查询得到SQL语句

(3) 锁阻塞对象信息

```
select owner, object_name, object_type from dba_objects where object_id in (select object_id from v$blocked_object);
```

通过sid查询阻塞对象的详细信息，如对象名称，所属用户等。

(4) 杀掉阻塞会话

```
alter system kill session '&sid, &serial#';
```

在Oracle实例内杀死阻塞的会话进程，其中sid, serial#为中止会话对应信息，来自v\$session

(5) 杀系统进程

```
select pro.spid, pro.program from v$session ses, v$process pro where ses.sid=&sid and ses.paddr=pro.addr;
```

```
kill -9 spid
```

找到进程号，并中止它

有时候，对于活动进程，在系统层面中止更为快速安全。

3) 门检查

```
select name, gets, misses, immediate_gets, spin_gets from v$latch order by 2;
```

检查数据库门的使用情况，misses, spin_gets统计高的，需要关注

```
select addr, gets from v$latch_children where name='cache buffers chains';
```

```
select hladdr, file#, dbablk from x$bh where hladdr in (select addr from v$latch_children where addr='&addr');
```

通过获得Latch的地址，找到该Latch守护的x\$BH中相关Buffer。

6. 等待 统计数据

Wait和Statistics数据分别代表了数据库的等待和运行数据，观察这些数据以了解数据库的

等待瓶颈和健康程度。

1) 等待时间查询

```
select sid, event, wait_time_micro from v$session_wait order by 3;
```

通过等待事件额等待时间，了解数据库当前连接会话的等待情况。

如果会话众多，需要限定查询输出行数。

2) TOP10 等待时间

```
select * from (select event, total_waits, average_wait, time_waited from
v$system_event where wait-class <> 'Idle' order by time_waited desc)
where rownum <= 10;
```

查询当前数据中TOP等待时间信息，需要分析和关注非空闲的显著等待。

3) 会话统计数据

```
select s.sid, s.statistic#, n.name, s.value from v$sesstat, v$statname n
where s.statistic# = n.statistic# and n.name = 'redo size' and sid = '&sid';
```

查询数据库会话的统计信息数据，

4) 系统级统计数据

```
select * from v$sysstat where name = 'redo size';
```

查询整个系统的统计数据，显示数据库实例启动以来的REDO日志生成量。

7. 对象检查

表、索引、分区、约束等是数据库的核心存储对象，其核心信息和对象维护时DBA重要的日常工作。

1) 表信息数据

```
select * from (select owner, table_name, num_rows from dba_tables order by num_rows
desc nulls last) where rownum < 11;
```

查看表的基本信息数据： 属主，表名，记录行数等。

2) 表结构查询

```
set long 12000
```

```
select dbms_metadata.get_ddl('TABLE', '&table_name', '&user') from dual;
```

根据提供的表名和用户（需大写），查询表的表结构信息（建表语句）

3) 表统计信息

```
select owner, table_name, last_analyzed from dba_tab_statistics where owner
= '&owner' and table_name = '&table_name';
```

查询给定用户，给定表（需大写），查询最后的统计信息分析收集时间。

统计信息影响执行计划，当SQL执行异常时，需要重点分析统计信息。

4) 表统计信息收集

```
exec dbms_stats.gather_table_stats(ownname=>'&owner', tabname=>'&table_name');
```

收集特定用户下的表对象的统计信息。

5) 索引信息数据

```
select * from (select index_name, table_name, num_rows, leaf_blocks, clustering_factor
from dba_indexes order by 5 desc nulls last) where rownum < 11;
```

查询索引的基本信息，包括叶块数，聚簇因子。

如聚簇因子过高接近行数可能代表索引效率不高。

6) 索引定义查询

```
set long 12000
```

```
select dbms_metadata.get_ddl('INDEX', '&table_name', '&user') from dual;
```

7) 索引统计信息及收集方法

```
select owner, index_name, last_analyzed from dba_ind_statistics where owner
= '&owner' and table_name = '&table_name';
```

```
exec dbms_stats.gather_index_stats(ownname=>'&owner', indname=>'&index_name');
```

8) 分区对象检查

```
select table_name, partitioning_type, partition_count, status from dba_part_tables;
```

```
select table_name, partition_name, high_value from dba_tab_partitions where rownum < 11;
```

查看分区表的基本信息： 分区类型，数量，边界值等。

9) 分区定义查询

```
select dbms_metadata.get_ddl('TABLE', '&part_table_name', 'user') from dual;
```

10) 分区统计信息相关

```
select owner, table_name, partition_name, last_analyzed from dba_tab_statistics
where owner='&owner' and table_name='&table_name';
```

查询分区表的统计信息收集时间

```
exec dbms_stats.gather_table_stats(ownname=>'&owner', tabname=>'&table_name');
```

对分区表进行手工收集统计信息。

11) 约束信息

```
select constraint_name, constraint_type from dba_constraints where table_name='&table_name';
```

查询指定数据表的约束信息，包括名称和类型。

12) 失效对象检查

```
select owner, object_name, object_type, status from dba_objects  
where status <> 'VALID' order by owner, object_name;
```

检查数据库中的失效对象信息，通常运行健康的数据库中不应有失效的对象。

13) 闪回查询

闪回查询对于恢复DML及部分误操作非常便利

(1) 时间闪回

```
select * from &table_name as of timestamp to_timestamp('2015-02-04 00:02:09', 'yyyy-mm-dd hh24:mi:ss');
```

基于时间点的表数据闪回查询

(2) SCN闪回

```
select * from &table_name as of scn &scn;
```

基于SCN的表数据闪回查询。

(3) 闪回删除

```
flashback table &old_table to before drop rename to &new_table;
```

对已删除的表进行闪回恢复并重命名。

8. AWR报告检查

通过AWR报告了解日常高峰时段数据库各项指标和运行状况，通过对比报告观察和基线的变化，

通过趋势分析持续关注数据库日常运行状态。

1) 本地AWR

```
@?/rdbms/admin/awrrpt
```

生产本地AWR报告信息，根据提示输入相关信息

2) 指定实例AWR

```
@?/rdbms/admin/awrrpti
```

生产指定实例的AWR报告

3) AWR对比报告

```
@?/rdbms/admin/awrrdrpt
```

生成本地AWR时间段对比报告

4) 指定实例对比报告

```
@?/rdbms/admin/awrrdrpi
```

5) AWR信息提取

```
@?/rdbms/admin/awrextr
```

使用awrextr脚本将AWR性能数据导出，可拥有留错或异地分析

6) AWR信息加载

```
@?/rdbms/admin/awrload
```

通过awrload，可将导出的AWR性能数据导入其他数据库，便于集中分析。

9. SQL报告检查

对TOP SQL进行持续关注和分析，通过SQL报告分析SQL的效率，性能，并做出报告和优化建议等。

1) 本地SQL报告

```
@?/rdbms/admin/awrsqrpt
```

2) 指定实例SQL报告

```
@?/rdbms/admin/awrsqrpi
```

3) 会话ID SQL Monitor报告

```
select dbms_sqltune.report_sql_monitor(session_id=>'&sid', report_level=>'ALL',  
type=>'&type') as report from dual;
```

4) 指定SQLID SQL Monitor 报告

```
select dbms_sqltune.report_sql_monitor(sql_id=>'&sql_id', report_level=>'ALL',  
type=>'&type') as report from dual;
```

5) Explain SQL 执行计划

```
explain plan for select count(*) from user_objects;  
select * from table(dbms_xplan.display);
```

通过explain plan for select语句的方法获取SQL执行计划

6) Autotrace SQL 执行计划

```
set autotrace traceonly explain;
select count(*) from user_objects;
set autotrace off;
```

7) DBMS_XPLAN SQL执行计划

```
select * from table (dbms_xplan.display_cursor('&sql_id', null, 'advanced'));
通过DBMS_XPLAN包获取SQL执行计划，需要提供sql_id.
```

8) 10053 跟踪事件

```
alter session set tracefile_identifier='10053';
alter session set events '10053 trace name context forever, level 1';
执行select 语句
```

```
alter session set events '10053 trace name context off';
```

通过10053事件来查看执行计划和详细的SQL解析过程，

trace文件提供了Oracle如何选择执行计划的原因

9) 绑定变量

```
select dbms_sqltune.extracbind(bind_data, 1).value_string from wrh$_sqlstat
where sql_id = '&sql_id';
```

```
select snap_id, name, position, value_string from dba_hist_sqlbind
where sql_id = '&sql_id';
```

查询SQL语句的绑定变量以及历史绑定变量值信息，需要提供sql_id.

10. 定时任务检查

检查数据库定时任务执行情况，确保后台任务正确执行，尤其关注统计信息收集等核心任务

1) 用户定时任务

```
select job, log_user, last_date, next_date, interval, broken, failures
from dba_jobs;
```

查询用户的定时任务（job）信息，确保任务在期望的时间成功执行。

2) 系统定时任务

```
select job_name, start_date, repeat_interval from dba_scheduler_jobs;
```

查询系统定时调度信息，

3) 系统定时任务-11g

```
select client_name, mean_job_duration from dba_autotask_client;
```

4) 启停统计信息任务-11g

```
exec dbms_scheduler.disable('SYS.GATHER_STATS_JOB');
```

```
exec dbms_scheduler.enable('SYS.GATHER_STATS_JOB');
```

5) 启停统计信息任务-11g

```
exec dbms_auto_task_admin.disable(client_name=>'auto optimizer stats collection',
operation=>null, window_name=>null);
```

```
exec dbms_auto_task_admin.enable(client_name=>'auto optimizer stats collection',
operation=>null, window_name=>null);
```

11. 备份

数据备份终于一切，日常应检查备份执行情况，并检查备份的有效性

确保备份能够保障数据安全，备份安全加密也应兼顾

1) 全库exp/imp

```
exp system/manager file=/full.dmp log=/full.log full=y
```

```
imp system/manager file=/full.dmp log=/full.log full=y
```

将数据库全库导出导入

2) 用户模式exp/imp

```
exp hr/hr file=/hr.dmp log=/hr.log owner=hr
```

```
imp hr/hr file=/hr.dmp log=/hr.log fromuser=hr touser=hr
```

导出导入指定模式

3) 表模式exp/imp

```
exp hr/hr file=/tables.dmp log=/tables.log tables=table_name
```

```
imp hr/hr file=/tables.dmp log=/tables.log tables=table_name
```

导出导入指定表

4) 全库模式expdp/impdp

```
expdp system/manager directory=exp_dir dumpfile=full.dmp full=y;
```

```
impdp system/manager directory=imp_dir dumpfile=full.dmp full=y;
```

用expdp/impdp讲数据库全库导出导入

expdp和impdp的目录指定只认directory参数的值，不认系统的绝对路径

directory对象需要在数据库中创建

5) 用户模式expdp/impdp

```
expdp system/manager directory=exp_dir schemas=scott dumpfile=expdp.dmp
```

```
impdp system/manager directory=imp_dir schemas=scott dumpfile=expdp.dmp
```

```
remap_scheme=scott:hr remap_tablespace=users:testtbs;
```

6) 表模式expdp/impdp

```
expdp scott/tiger directory=exp_dir tables=emp, dept dumpfile=tables.dmp;
```

```
impdp scott/tiger directory=imp_dir dumpfile=tables.dmp tables=emp, dept;
```

7) 物理备份检查

```
select backup_type, start_time, completion_time, block_size from v$backup_set;
```

8) 自动备份控制文件

```
in RMAN :
```

```
show all;
```

```
configure controlfile autobackup on;
```

9) 手动备份控制文件

```
in RMAN :
```

```
backup current controlfile;
```

```
in SQL :
```

```
alter database backup controlfile to '/back/control.bak';
```

10) 转储控制文件

```
alter session set events 'immediate trace name controlf level 8';
```

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

11) RMAN备份数据库

```
backup format '/data/backup/%U' database plus archivelog;
```

12. 基本信息检查

基本信息包括版本，组件补丁集等信息。定期检查数据库信息并登记在案

1) 版本检查

```
select * from v$version;
```

2) 组件检查

```
select * from v$option;
```

3) 容量检查

```
in asmcmd
```

```
lsdg
```

```
select group_number, disk_number, mount_status, total_mb, free_mb from v$asm_disk;
```

```
select group_number, name, state, total_mb, free_mb from v$asm_diskgroup;
```

4) PSU检查

```
select * from dba_registry_history
```

查询数据库的版本升级历史信息

```
$ORACLE_HOME/OPatch/opatch lsinventory
```

查询数据库补丁历史信息，是系统级命令

```
select group_number, disk_number, mount_status, total_mb, free_mb from v$asm_disk order by group_number, disk_number;
```

```
SELECT dg.name AS diskgroup, SUBSTR(a.name, 1, 18) AS name, SUBSTR(a.value, 1, 24) AS value, read_only FROM V$ASM_DISKGROUP_STAT dg, V$ASM_ATTRIBUTE a WHERE dg.name = 'DATA' AND dg.group_number = a.group_number;
```

```
SELECT dg.name AS diskgroup, dg.group_number as groupno, SUBSTR(a.name, 1, 18) AS name, SUBSTR(a.value, 1, 24) AS value, read_only FROM V$ASM_DISKGROUP_STAT dg, V$ASM_ATTRIBUTE a WHERE dg.group_number = a.group_number;
```

```
SELECT dg.name AS diskgroup, max(dg.group_number) as groupno FROM V$ASM_DISKGROUP_STAT dg group by dg.name;
```