

# 性能调整

## 1 内存调优

Oracle 实例启动的过程，其实就是将 Oracle 内存参数设置的值加载到内存中，并启动相应的后台进程进行相关的服务过程。Oracle 进程由服务器进程和用户进程组成，这些重要的后台进程在前面已经讲过。而内存结构由系统全局区 SGA 和程序全局区 PGA 组成。一般情况下，需要保证 SGA、PGA、OS 使用的内存，这三者的总和小于总物理 RAM 值。

SGA 包括以下五个区：数据缓冲区、共享池、日志缓冲区、JAVA 池和大池。

当数据库中的数据发生变化时，数据会被保存在数据缓冲区、日志缓冲区中。而数据缓冲区的大小是由初始化参数文件里的 DB\_CACHE\_SIZE 参数所设置的。

### (1) 数据缓冲区性能调优

数据缓冲区的调优目标就是尽可能让服务器进程要寻找的数据都可以在数据缓冲区里找到，这样可以减少磁盘 I/O，也代表数据缓冲区的命中率很高；同时，增加数据缓冲区的大小。一般情况下，将数据缓冲区的大小设置为可用内存的 40%。

```
SQL> conn sys/ora505 as sysdba
已连接。
SQL> select 1-(phy.value/(cur.value+con.value)) "CachHitRadio"
  2   from v$sysstat cur,v$sysstat con,v$sysstat phy
  3   where cur.name = 'db block gets'
  4     and con.name = 'consistent gets'
  5     and phy.name = 'physical reads';

CachHitRadio
-----
.978865607
```

db block gets 表示从内存读数据;

consistent gets 表示读一致性数据;

physical reads 表示从磁盘读数据;

```
SQL> select name,value from v$sysstat where name in('db block gets','consistent
gets','physical reads');

NAME                                                    VALUE
-----
db block gets                                           51534
consistent gets                                         566909
physical reads                                           17967
```

注意：数据缓冲区性能调优实现了对数据库缓冲区命中率的查询，一般情况下要保证其大于 90%，否则就需要调整数据库缓冲区的大小，使命中率达到该要求。

### (2) 共享池性能调整

共享池主要用来存放最近执行过的 SQL 指令、数据字典、PL/SQL 语法分析以及表的定义与权限等。共享池的大小由初始化参数文件中的 share\_pool\_size 参数决定，一般大小可用内存的 10%。

通过查询 v\$sgastst，确认了目前 SGA 的使用情况。

```
SQL> select pool,sum(bytes) from v$sgastat
2 where pool='shared pool' group by pool;
```

POOL	SUM(BYTES)
shared pool	163582616

```
SQL> select name,pool, bytes from v$sgastat
2 where name = 'free memory';
```

NAME	POOL	BYTES
free memory	shared pool	16846320
free memory	large pool	3163264
free memory	java pool	2821440

```
SQL> select to_char(round((1-sum(getmisses)/sum(gets))*100,1))||'%' "hit radia"
2 from v$rowcache;
```

hit radia
97.5%

注意：round 函数是按指定位数进行四舍五入。

上述语句查询到数据字典的命中率为 97.5%，如果该值小于 90%就需要加大共享池，以减少磁盘 I/O。

### (3) 其他

一般情况下，JAVA 池主要用于 JAVA 语言的开发，如果不使用 JAVA，原则上其值不能小于 20MB，一般情况下 30MB 就够了。

如果不设置 MTS，主要用于数据库备份恢复管理器 RMAN，大池在 5MB 到 10MB 之间就可以了。

```
SQL> select sum(gets),sum(waits),sum(waits)/sum(gets)
2 from v$rollstat;
```

SUM(GETS)	SUM(WAITS)	SUM(WAITS)/SUM(GETS)
15238	0	0

上述语句检查回滚段的占用，当 sum(waits)小于 sum(gets)值的 1%时就不需要调整。

### (4) 重做日志缓冲区

重做日志缓冲区的调优目标就是当服务进程每次去索取空间时都可以成功。但是如果分配给重做日志缓冲区的空间太大，就会造成其他 SGA 里的空间变小。

```
SQL> select event,total_waits,time_waited,average_wait
2 from v$system_event
3 where event like 'LOG FILE SWITCH COMPLETION%';
```

未选定行

上述语句检查了在动态视图中发生日志文件切换的事件，如果有则说明在线重做日志文件太小，需要加大或者多加几组在线重做日志文件。

## 2 物理调优

数据库中，如果将数据文件存放在不同位置，有助于降低磁盘 I/O 的等待时间，从而提高系统性能。

移动数据文件如下所示。

```
SQL> column tablespace_name format a20
SQL> column file_name format a40
SQL> select tablespace_name,file_name from dba_data_files;
```

TABLESPACE_NAME	FILE_NAME
USERS	D:\ORACLE\JIANGXUE\ORADATA\ORCL\USERS01.DBF
UNDOTBS1	D:\ORACLE\JIANGXUE\ORADATA\ORCL\UNDOTBS01.DBF
SYSAUX	D:\ORACLE\JIANGXUE\ORADATA\ORCL\SYSAUX01.DBF
SYSTEM	D:\ORACLE\JIANGXUE\ORADATA\ORCL\SYSTEM01.DBF

  

TABLESPACE_NAME	FILE_NAME
UNDOTBS01	F:\UNDOTBS01_1.DBF
UNDOTBS01	F:\UNDOTBS01_2.DBF

已选择6行。

在 undotbs01 中有两个数据文件，他们位于同一个磁盘中，按照原则来讲这是不合适的，因此通过以下操作将数据文件 undotbs01\_1.dbf 转移到其他磁盘上。

```
SQL> alter tablespace undotbs01 offline;

表空间已更改。

SQL> host move f:\undotbs01_1.dbf d:\undotbs01_2.dbf
移动了          1 个文件。

SQL> alter tablespace undotbs01 rename datafile
  2  'f:\undotbs01_1.dbf'
  3  to
  4  'd:\undotbs01_2.dbf';

表空间已更改。

SQL> alter tablespace undotbs01 online;

表空间已更改。
```

上述操作实现了将一个表空间 undotbs01 中的数据文件 undotbs01\_1.dbf 移动到 D 盘，并重新命名为 undotbs01\_2.dbf。

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

TABLESPACE_NAME	FILE_NAME
USERS	D:\ORACLE\JIANGXUE\ORADATA\ORCL\USERS01.DBF
UNDOTBS1	D:\ORACLE\JIANGXUE\ORADATA\ORCL\UNDOTBS01.DBF
SYSAUX	D:\ORACLE\JIANGXUE\ORADATA\ORCL\SYSAUX01.DBF
SYSTEM	D:\ORACLE\JIANGXUE\ORADATA\ORCL\SYSTEM01.DBF
UNDOTBS01	D:\UNDOTBS01_2.DBF
UNDOTBS01	F:\UNDOTBS01_2.DBF

已选择6行。

### 3 表空间改进

- (1) 控制用户占用的表空间值
- (2) 不同类型的数据放置到不同的表空间

在 Oracle 数据库中，临时数据被存放在临时表空间中，用户数据被存放在 users 表空间中，系统数据被存放在 system 及 sysaux 中。

(3) 调整表空间大小

```
SQL> select a.tablespace_name,a.bytes "bytes",
2 a.bytes-sum(nvl(b.bytes,0)) "used",
3 sum(nvl(b.bytes,0)) "free",
4 sum(nvl(b.bytes,0))/a.bytes*100 "%free"
5 from dba_data_files a, dba_free_space b
6 where a.file_id = b.file_id(+)
7 group by a.tablespace_name,a.bytes
8 order by a.tablespace_name;
```

TABLESPACE_NAME	bytes	used	free	%free
SYSAUX	651493376	619315200	32178176	4.93914093
SYSTEM	754974720	744554496	10420224	1.38020833
UNDOTBS01	2097152	1048576	1048576	50
UNDOTBS01	4194304	2359296	1835008	43.75
UNDOTBS1	78643200	23855104	54788096	69.6666667
USERS	5242880	1572864	3670016	70

已选择6行。

当表空间容量剩余百分比小于 20%时就需要对其进行调整。

(4) 当数据库中表空间被使用一段时间后，表空间会产生碎片。

首先查看数据库中某个表空间的自由空间：

```
SQL> select * from dba_free_space where tablespace_name = 'USERS';
```

TABLESPACE_NAME	FILE_ID	BLOCK_ID	BYTES	BLOCKS	RELATIVE_FNO
USERS	4	208	3538944	432	4
USERS	4	184	65536	8	4
USERS	4	176	65536	8	4

估计相邻自由空间的真正数量：对每一行，用起始块的 id(block\_id)加上自由块(blocks)的数量，如果它和下一行的块 id(block\_id)相等，则此两行是连续的。如第一行和第二行，第二行  $184+8=192$  不等于第一行的 208，所以这两行是不连续的。当数据库表空间中的某些表不在起作用时，可以将其删除，该操作完成后，会产生很多自由空间，此时需要将其合并。

```
SQL> alter tablespace USERS coalesce;
```

表空间已更改。