一、名词解释

(1)SGA：SystemGlobal Area是OracleInstance的基本组成部分，在实例启动时分配;系统全局域SGA主要由三部分构成：共享池、数据缓冲区、日志缓冲区。

(2)共享池：Shared Pool用于缓存最近被执行的SQL语句和最近被使用的数据定义，主要包括：Librarycache（共享SQL区）和Datadictionarycache（数据字典缓冲区）。 共享SQL区是存放用户SQL命令的区域，数据字典缓冲区存放数据库运行的动态信息。

(3)缓冲区高速缓存：DatabaseBufferCache用于缓存从数据文件中检索出来的数据块，可以大大提高查询和更新数据的性能。

(4)大型池：Large Pool是SGA中一个可选的内存区域，它只用于shared server环境。

(5)Java池：Java Pool为Java命令的语法分析提供服务。

(6)PGA：ProcessGlobal Area是为每个连接到Oracledatabase的用户进程保留的内存。

二、分析与调整

(1)系统全局域：

SGA与操作系统、内存大小、cpu、同时登录的用户数有关。可占OS系统物理内存的1/3到1/2。

a.共享池Shared Pool：

查看共享池大小Sql代码

SQL>show parameter shared\_pool\_size

查看共享SQL区的使用率：

Sql代码

select(sum(pins-reloads))/sum(pins)"Library cache"from v$librarycache;

--动态性能表

LIBRARY命中率应该在90％以上，否则需要增加共享池的大小。

查看数据字典缓冲区的使用率：

Sql代码

select(sum(gets-getmisses-usage-fixed))/sum(gets)"Data dictionary cache"from v$rowcache;

--动态性能表

这个使用率也应该在90％以上，否则需要增加共享池的大小。

修改共享池的大小:

Sql代码

ALTERSYSTEMSET SHARED\_POOL\_SIZE =64M;

b.缓冲区高速缓存DatabaseBufferCache：

查看共享池大小Sql代码

SQL>show parameter db\_cache\_size

查看数据库数据缓冲区的使用情况：

Sql代码

SELECTname,valueFROM v$sysstat orderbynameWHEREnameIN(''DBBLOCK GETS'',''CONSISTENT GETS'',''PHYSICALREADS'');

SELECT \* FROM V$SYSSTAT WHERENAMEIN('parse\_time\_cpu','parse\_time\_elapsed','parse\_count\_ hard');

计算出来数据缓冲区的使用命中率＝1-(physicalreads/(dbblock gets+consistent gets)),这个命中率应该在90％以上，否则需要增加数据缓冲区的大小。

c.日志缓冲区

查看日志缓冲区的使用情况:

Sql代码

SELECTname,valueFROM v$sysstat WHEREnameIN('redo entries','redo log space requests')

查询出的结果可以计算出日志缓冲区的申请失败率：

申请失败率＝requests/entries，申请失败率应该接近于0，否则说明日志缓冲区开设太小，需要增加ORACLE数据库的日志缓冲区。

d.大型池：

可以减轻共享池的负担，可以为备份、恢复等操作来使用，不使用LRU算法来管理。其大小由数据库的'共享模式/db模式'如果是共享模式的话，要分配的大一些。

指定Large Pool的大小：

Sql代码

ALTERSYSTEMSET LARGE\_POOL\_SIZE=64M

e.Java池：

在安装和使用Java的情况下使用。

(2)PGA调整

a.PGA\_AGGREGATE\_TARGET初始化设置

PGA\_AGGREGATE\_TARGET的值应该基于Oracle实例可利用内存的总量来设置，这个参数可以被动态的修改。假设Oracle实例可分配4GB的物理内存，剩下的内存分配给操作系统和其它应用程序。你也许会分配80％的可用内存给Oracle实例，即3.2G。现在必须在内存中划分SGA和PGA区域。

在OLTP(联机事务处理)系统中，典型PGA内存设置应该是总内存的较小部分（例如20％），剩下80％分配给SGA。

OLTP：PGA\_AGGREGATE\_TARGET=(total\_mem \* 80%) \* 20%=2.5G

在DSS(数据集)系统中，由于会运行一些很大的查询，典型的PGA内存最多分配70％的内存。

DSS：PGA\_AGGREGATE\_TARGET=(total\_mem \* 80%) \* 50%

在这个例子中，总内存4GB，DSS系统，你可以设置PGA\_AGGREGATE\_TARGET为1600MB，OLTP则为655MB。

b.配置PGA自动管理

不用重启DB，直接在线修改。

SQL>altersystemset workarea\_size\_policy=autoscope=both;

System altered.

SQL>altersystemset pga\_aggregate\_target=512mscope=both;

System altered.

SQL>show parameter workarea

NAME                                 TYPE        VALUE   
------------------------------------ ----------- ------------------------------   
workarea\_size\_policy                 string      AUTO  --这个设置成AUTO

SQL>show parameter pga

NAME                                 TYPE        VALUE  
------------------------------------ ----------- ------------------------------  
pga\_aggregate\_target                 big integer 500M

c.监控自动PGA内存管理的性能

V$PGASTAT：这个视图给出了一个实例级别的PGA内存使用和自动分配的统计。

SQL>set lines 256

SQL>set pages 42

SQL>SELECT \* FROM V$PGASTAT;

NAMEVALUE UNIT

---------------------------------------------------------------- ---------- -----

aggregate PGA target parameter536870912 bytes

--当前PGA\_AGGREGATE\_TARGET的值

aggregate PGAauto target 477379584 bytes

--当前可用于自动分配了的PGA大小，应该比PGA\_AGGREGATE\_TARGET小

globalmemorybound26843136 bytes

--自动模式下工作区域的最大大小，Oracle根据工作负载自动调整。

total PGA inuse6448128 bytes

total PGA allocated11598848 bytes

--PGA的最大分配

maximum PGA allocated166175744 bytes

total freeable PGAmemory393216 bytes

--PGA的最大空闲大小

PGAmemory freed back to OS 69074944 bytes

total PGA usedforauto workareas 0 bytes

--PGA分配给auto workareas的大小

maximum PGA usedforauto workareas 1049600 bytes

total PGA usedformanual workareas 0 bytes

maximum PGA usedformanual workareas 530432 bytes

over allocation count1118

--实例启动后，发生的分配次数，如果这个值大于0，就要考虑增加pga的值

bytes processed114895872 bytes

extra bytesread/written4608000 bytes

cache hit percentage 96.14percent

--命中率

16rows selected.

--V$PGA\_TARGET\_ADVICE

SQL>SELECTround(PGA\_TARGET\_FOR\_ESTIMATE/1024/1024) target\_mb,

ESTD\_PGA\_CACHE\_HIT\_PERCENTAGE cache\_hit\_perc,

ESTD\_OVERALLOC\_COUNT

FROM v$pga\_target\_advice;

The output of this query might look likethefollowing:

TARGET\_MB CACHE\_HIT\_PERC ESTD\_OVERALLOC\_COUNT

---------- -------------- --------------------

6323367

1252430

250303

375390

500580

600590

700590

800600

900600

1000610

1500670

2000760

3000830

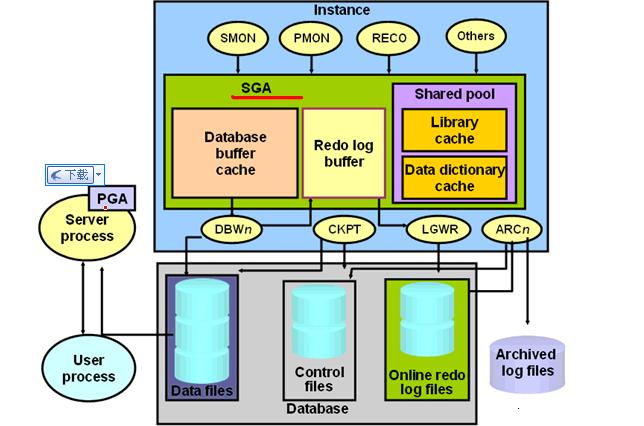
4000850

可以看出当TARGET\_MB为375M是ESTD\_OVERALLOC\_COUNT=0，所以可以将PGA\_AGGREGATE\_TARGET设置成375M。

附：oracle SGA与PGA区别：

SGA：是用于存储数据库信息的内存区，该信息为数据库进程所共享。它包含Oracle服务器的数据和控制信息,它是在Oracle服务器所驻留的计算机的实际内存中得以分配，如果实际内存不够再往虚拟内存中写。

PGA：包含单个服务器进程或单个后台进程的数据和控制信息，与几个进程共享的SGA正相反,PGA是只被一个进程使用的区域，PGA在创建进程时分配,在终止进程时回收。



另一篇文章中也这样写了相关的调整记录

如何估算PGA，SGA的大小，配置数据库服务器的内存

ORACLE给的建议是: OLTP系统PGA=(TotalMemory)\*80%\*20%。

DSS系统PGA=(TotalMemory)\*80%\*50%。

ORACLE建议一个数据库服务器，分80%的内存给数据库，20%的内存给操作系统，那怎么给一个数据库服务器配内存呢？

SQL>select \* from v$pgastat;

NAMEVALUE UNIT

---------------------------------------------------------------- ---------- ------------

aggregate PGA target parameter104857600 bytes

-----这个值等于参数PGA\_AGGREGATE\_TARGET的值，如果此值为0，表示禁用了PGA自动管理。

aggregate PGAauto target 75220992 bytes

-----表示PGA还能提供多少内存给自动运行模式，通常这个值接近pga\_aggregate\_target-total pga inuse.

globalmemorybound20971520 bytes

-----工作区执行的最大值，如果这个值小于1M，马上增加PGA大小

total PGA inuse30167040 bytes

-----当前分配PGA的总大小，这个值有可能大于PGA，如果PGA设置太小.这个值接近select sum(pga\_used\_mem) from v$process.

total PGA allocated52124672 bytes

-----工作区花费的总大小

maximum PGA allocated67066880 bytes

total freeable PGAmemory0 bytes ----没有了空闲的PGA

processcount23----当前一个有23个process

max processes count25

PGAmemory freed back to OS 0 bytes

total PGA usedforauto workareas 8891392 bytes

maximum PGA usedforauto workareas 22263808 bytes

total PGA usedformanual workareas 0 bytes ---为0自动管理

maximum PGA usedformanual workareas 0 bytes ---为0自动管理

over allocation count0

--如果PGA设置太小，导致PGA有时大于PGA\_AGGREGATE\_TARGET的值，此处为0，说明PGA没有扩展大于TARGET的值，如果此值出现过，那么增加PGA大小。

bytes processed124434432 bytes

extra bytesread/written0 bytes

cache hit percentage 100percent

---命中率为100%，如果太小增加PGA

recomputecount(total)6651

19rows selected

SQL>selectmax(pga\_used\_mem)/1024/1024Mfrom v$process;

----当前一个process消耗最大的内存

M

----------

9.12815189

SQL>selectmin(pga\_used\_mem)/1024/1024Mfrom v$process where pga\_used\_mem>0;---process消耗最少内存

M

----------

0.19186878

SQL>selectmax(pga\_used\_mem)/1024/1024Mfrom v$process ;

----process曾经消耗的最大内存

M

----------

9.12815189

SQL>selectsum(pga\_used\_mem)/1024/1024from v$process;----当前process一共消耗的PGA

SUM(PGA\_USED\_MEM)/1024/1024

---------------------------

28.8192501068115

如何设置PGA呢？我们可以在压力测试阶段，模拟一下系统的运行，然后运行

select(selectsum(pga\_used\_mem)/1024/1024from v$process)/(selectcount(\*)from v$process)fromdual;

得到一个process大约占用了多少的内存，然后估算系统一共会有多少连接，比如一共有500个连接，

那么sessions=1.1\*process+5=500，那么processes=450,

再乘以一个process需要消耗的内存，就能大约估算出PGA需要设置多大。

EG = 1.1 \* 450 = 495M估算的大一点550M就OK乐

最好将PGA设置的值比计算出的值大一点，PGA值设定好后，就可以根据系统的性质,如果系统为OLTOP，那么总的内存可以设置为PGA/0.16,最后也能估算出SGA的大小，建议还是多配点内存，反正便宜。

下面摘抄eygle的关于一个process能够分配的最大内存(串行操作)的规则:

10gR1之前，对于串行操作(非并行)一个process能够分配的最大的内存为min(5%pga\_aggregate\_target,100m)

10gR2之后，对于串行操作(非并行)一个process能够分配的最大内存有如下规则:

如果pga\_aggregate\_target<=500m,那么最大的内存为20%\*pga\_aggregate\_target.

如果500m<pga\_aggregate\_target<=1000m,那么最大内存为100m.

如果1000m<pga\_aggregate\_target<=2.5G,那么最大内存为10%\*pga\_aggregate\_target.

如果pga\_aggregate\_target>2.5G,那么最大内存为2.5G.

SQL>SELECT x.ksppinmNAME, y.ksppstvlVALUE, x.ksppdesc describ

FROM SYS.x$ksppi x, SYS.x$ksppcv y WHERE x.inst\_id=USERENV('Instance')

AND y.inst\_id=USERENV('Instance')AND x.indx= y.indxAND x.ksppinmLIKE'%&par%'

NAMEVALUE DESCRIB

--------------------------------------------------------------------------------

\_smm\_max\_size20480 maximum work area sizeinautomode(serial)

SQL>show parameter pga

NAMETYPEVALUE

------------------------------------ ----------- ------------------------------

pga\_aggregate\_target biginteger100M

此处我的一个process能够分配的最大内存为20M，因为我的PGA=100M，符合上面的规则。

隐含参数\_smm\_max\_size表示一个process能够分配最大的memory.

买了piner的《oracle高可用环境》一书，正好趁这段时间学习一下。

把看到的东西总结一下发表于此，今天先发第一章关于SGA与PGA的内容。

以后会陆续将总结在此发表，与大家共享。

SGA与PGA的结构如下图：

SGA:

查看SGA：

Sqlp>show sga

或select \* from v$sga;

TotalSystemGlobal Area 289406976 bytes

FixedSize1248600 bytes

VariableSize176161448 bytes

Database Buffers 109051904 bytes

Redo Buffers2945024 bytes

FixedSize：包括了数据库与实例的控制信息、状态信息、字典信息等，启动时就被固定在SGA中，不会改变。

VariableSize：包括了shard pool、large pool、java pool、stream pool、游标区和其他结构

Database Buffers：数据库中数据块缓冲的地方，是SGA中最大的地方，决定数据库性能

Redo Buffers：提供REDO缓冲的地方，在OLAP中不需要太大

V$sgastat记录了SGA的一些统计信息

V$sga\_dynamic\_components保存SGA中可以手动调整的区域的一些调整记录

Shard pool:

Shard\_pool\_size决定其大小，10g以后自动管理

Shard\_pool中数据字典和控制区结构用户无法直接控制，与用户有关的只有sql缓冲区（librarycache）。

将经常访问的过程或包用***DBMS\_SHARED\_POOL.KEEP***存储过程将该包pin在共享池中。

手工清除共享池的内容：altersystemflush shard\_pool;

共享池相关的几个常用的视图：

V$sqlarea记录了所有sql的统计信息，包括执行次数、物理读、逻辑读、耗费时间等

V$sqltext\_with\_newline完全显示sql语句，通过hash\_value来标示语句，piece排序

V$sql\_plan保存了sql的执行计划，通过工具查看

V$shared\_pool\_advice对共享池的预测，可以做调整SGA的参考

Databuffer：

在OLTP系统中要求databuffer的命中率在95%以上

selectsum(pins)"execution",sum(pinhits)"hits",

((sum(pinhits)/sum(pins))\*100)"pinhitration",

sum(reloads)"misses",((sum(pins)/(sum(pins)

+sum(reloads)))\*100)"relhitratio"

from V$librarycache

计算命中率的语句

select round((1 - (physical.value - direct.value - lobs.value)/logical.value)\*100,2) "Buffer Cache Hit Ratio"

from v$sysstat physical,v$sysstat direct,v$sysstat lobs,v$sysstat logical

where physical.name ='physical reads'

and direct.name ='physical reads direct'

and lobs.name ='physical reads direct (lob)'

and logical.name ='session logical reads';

PINSNUMBERNumberof times a PIN was requested for objects of this namespace

PINHITSNUMBERNumberof times allofthe metadata pieces ofthelibraryobject were foundinmemory

RELOADSNUMBERAny PIN of an object that isnotthefirst PIN performed since theobject handle was created,and which requires loadingtheobjectfromdisk

Oracle把从databuffer中获得的数据库叫cache hit，把从磁盘获得的脚cache miss

数据缓冲区中的数据块通过脏列表（dirtylist）和LRU列表（LRUlist）来管理。

Databuffer可细分为：default pool、keep pool、recycle pool对应的参数为db\_cache\_size、db\_keep\_cache\_size、db\_recycle\_size分别表示缓冲区大小

从9i开始oracle支持不同块大小的表空间，相应的可以为不同块大小的表空间指定不同块大小的数据缓冲区，不同块大小的数据缓冲区可以用相应的db\_nk\_cache\_size来指定，其中n可以是2、4、6、16或32

V$db\_cache\_advice对数据缓冲区的预测，可以做调整databuffer的参考

V$bh、x$bh记录了数据块在databuffer中缓冲的情况，通过这个视图可以找系统中的热点块。

通过下面语句找系统中top10热点快所在的热点对象:

**Select*/\*+ rule\*/* owner,object\_namefrom dba\_objects**

**Where data\_object\_id in**

**(select obj from**

**(select obj from x$bh orderby tch desc)**

**Whererownum<11)**;

PGA：

用来保存于用户进程相关的内存段。

从9i开始使用PGA自动管理，pga\_aggregate\_target参数指定session一共使用的最大PGA内存的上限。Workarea\_size\_policy参数用于开关PGA内存自动管理功能，auto/manual

在OLTP环境中，自动PGA管理只要设置到一定的值，如2G左右就能满足系统的要求。

自动内存管理：

从9i开始，sga\_max\_size参数设置SGA的内存大小，不能动态修改

从10g开始，指定了sga\_target参数后，所有的SGA组件如：shared pool、databuffer、large pool都不用手工指定了，Oracle会自动管理。这一特性就是自动共享内存管理ASMM。如果设置了sga\_target=0,就自动关闭自动共享内存管理功能。Sga\_target大小不能超过sga\_max\_size的大小。

手动管理SGA：

Altersystemset sga\_target=2000m;

Altersystemset db\_cache\_size=1000m;

Altersystemsetshared\_pool=200m;

Altersystemset sga\_target=0---------关闭自动共享内存管理ASMM

11G以后sga+pga整个内存可以自动管理AMM，相关参数memory\_max\_target memory\_target.设置好这两个参数后就不用关心SGA和PGA了

11g手动内存管理：

Altersystemset memory\_target=3000m;

Altersystemset sga\_target=2000m;

Altersystemset pga\_aggregate\_target=1000m;

Altersystemset memory\_target=0;---------关闭自动内存管理AMM

SGA+PGA最好不要超过总内存的70%

**在11g 中可以使用下面看各组件的值**

SQL> show parameter target

NAME                                 TYPE        VALUE

------------------------ ----------- ---------------------

archive\_lag\_target                   integer     0

db\_flashback\_retention\_target        integer     1440

fast\_start\_io\_target                 integer     0

fast\_start\_mttr\_target               integer     0

memory\_max\_target                    big integer 1024M

memory\_target                        big integer 1024M

pga\_aggregate\_target                 big integer 0

sga\_target                           big integer 0

如果需要监视 Memory\_target 的状况则可以使用下面三个动态试图：

   V$MEMORY\_DYNAMIC\_COMPONENTS

   V$MEMORY\_RESIZE\_OPS

   v$memory\_target\_advice

使用下面 Command 来调节大小：

SQL>ALTER SYSTEM SET MEMORY\_MAX\_TARGET = 1024M SCOPE = SPFILE;

SQL>ALTER SYSTEM SET MEMORY\_TARGET = 1024M SCOPE = SPFILE;

SQL>ALTER SYSTEM SET SGA\_TARGET =0 SCOPE = SPFILE;

SQL>ALTER SYSTEM SET PGA\_AGGREGATE\_TARGET = 0 SCOPE = SPFILE ;

注意：MEMORY\_MAX\_TARGET需要重启才能生效