**pga\_aggregate\_target和\_pga\_max\_size都不能绝对限制实际PGA的使用**

By: [Andy Zhang](https://blogs.oracle.com/author/andy-zhang)

和sga\_target不同，pga\_aggregate\_target并不能限制PGA的大小。 另一个差别是，SGA是数据库启动时按照sga\_max\_size预先分配的，而PGA则是“按需分配”的。

       以下详细解释下，为什么pga\_aggregate\_target并不能限制PGA的大小。一般的文档会把PGA分为stack space和UGA两个部分，但这种提法太笼统，不利于我们理解这个问题。我们把它分为work area和work area之外的部分。

       所谓work area，就是session要执行SQL，需要在数据库内部分配的，为了存放中间结果的内存。 比如sort area，为了排序用的内存，比如hash area，为了hash join用的内存，这一部分直接和SQL执行相关，影响SQL执行的效率，比如更大hash area会让hash join更快。pga\_aggregate\_target实际上只限制work area的大小。当work area达到pga\_aggregate\_target的限制，则会产生4030错误。隐含参数\_pga\_max\_size只是限制单个进程使用的work area大小，也是在pga\_aggregate\_target的限制之内的更小限制。而work area之外的内存，不被pga\_aggregate\_target和\_pga\_max\_size所限制。所以你经常会看到PGA的大小超过了pga\_aggregate\_target。这通常是因为PL/SQL中的变量和数组中装入了巨大的数据造成的。通常的例子是bulk collect，imp/exp，sql loader等工具也会产生类似的问题。

      我下面给出一个具体的示例：

alter system set "\_pga\_max\_size"=10m scope=both;

alter system set pga\_aggregate\_target=100m scope=both;

SQL> shutdown immediate

startup

SQL> select ksppinm as "hidden parameter", ksppstvl as "value" from x$ksppi join x$ksppcv  using (indx) where ksppinm like '%\_pga\_max\_size%' order by ksppinm;

hidden parameter

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

value

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

\_pga\_max\_size

10485760

<=========限制单个进程使用PGA 100M(其实是work area)

执行一个PL/SQL，使用bulk collect使其内存使用大大超过pga\_aggregate\_target和\_pga\_max\_size：

DECLARE

  CURSOR c1 IS

    SELECT \*

    FROM oracle.employees;

  TYPE emp\_row IS TABLE OF c1%ROWTYPE;

  table\_set  emp\_row;

BEGIN

  -- Assign values to nested table of records:

  SELECT \*

    BULK COLLECT INTO table\_set

    FROM oracle.employees;

  -- Print nested table of records:

    FOR i IN table\_set.FIRST .. table\_set.LAST LOOP

      DBMS\_OUTPUT.PUT\_LINE (

        table\_set(i).email || ' ' ||

        table\_set(i).last\_name  || ', ' ||

        table\_set(i).first\_name

      );

    END LOOP;END;

/

select pid,spid,program,pga\_used\_mem/1024/1024,pga\_alloc\_mem/1024/1024 from v$process

where spid=3735758

PID SPID         PROGRAM                                          PGA\_USED\_MEM/1024/1024 PGA\_ALLOC\_MEM/1024/1024

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

19    3735758    oracle@nascds5 (TNS V1-V3)    616.22734165191650390625    616.63962650299072265625

《=========PGA使用了600M

select a.name,to\_char(b.value,'999,999,999') value

from v$statname a,v$mystat b

where a.statistic# = b.statistic#

and a.name like '%ga memory%';

NAME                                                             VALUE

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

session uga memory                                                  1,896,024

session uga memory max                                              1,896,024

session pga memory                                                  2,330,120

session pga memory max                                            647,400,968  <==========Peak PGA size for the session. 647,400,968

12c新追加了一个参数PGA\_AGGREGATE\_LIMIT来限制PGA的大小：

Limiting process size with database parameter PGA\_AGGREGATE\_LIMIT (Doc ID 1520324.1)

<https://blogs.oracle.com/database4cn/pgaaggregatetargetpgamaxsizepga>