Tom大师关于Execute to Parse的描述：

the only way to influence that number is to either change

a) the number of times you parse. b) the number of times you execute.

The formula used:

Execute to Parse %: dscr , round(100\*(1-:prse/:exe),2) pctval

Execute to Parse %：

一个语句执行和分析了多少次的度量。计算公式为：Execute to Parse =100 \* (1 - Parses/Executions)。如果系统Parses > Executions，就可能出现该比率小于 0 的情况。该值<0 通常说明 shared pool 设置或者语 句效率存在问题，造成反复解析，reparse 可能较严重,或者是可能同 snapshot 有关，通常说明数据库性能存在一定问题。

If the number of parse calls is near the number of execute calls, then this ratio drifts towards zero (as yours is). As the number of execute calls increases (while holding parse calls constant), this number drifts towards 100%. That means you have parsed a statement ONCE and executed it MANY TIMES (that is good, that is best)

cursor sharing = similar MIGHT change a hard parse into a soft parse (take a very very very bad thing and make it simply very very bad). cursor sharing similar CANNOT change the number of times parse is invoked however.

There is precisely, exactly and only ONE person that can do that. That is the application developer.

When they say “parse this”, we parse it - it matters not what the value of cursor sharing is (if you have a hard parse problem, if your soft parse percent is below 99%, you need to have the coders FIX that, you have (in addition to performance, memory, scalability issues) a HUGE security risk if you are not using binds).

The developers must cache open cursors they know will be used over and over. The easiest way (to me) to accomplish this is to move all SQL into plsql, plsql automagically caches statements for us, it is the most efficient method to interface with the database.

Alternatively, they can program it, or they can see if the API they are using can do it magically for them (search for jdbc statement caching on google for example if you are using jdbc)

But it will have to be done in the application, there is nothing we can do outside of the application to influence how often it parses.

<https://asktom.oracle.com/pls/asktom/f?p=100:11:0::::P11_QUESTION_ID:1594740500346667363>

a、Execute to Parse%是执行到解析的度量，最佳情况下，是一次解析多次执行，最佳的是使用软软解析；

b、涉及到的参数主要是OPEN\_CURSORS与session\_cached\_cursors，前者定义单个session可以打开游标数，后者定义游标可缓存长度

c、通常情况下上述两个参数的使用率应尽可能偏离80%，以确保性能及资源充足，注意，这2个参数增大应考虑是否pga以及sga需要进一步调整

查看当前系统session\_cache\_cursor配置

SELECT

    'session\_cached\_cursors' parameter,

    lpad(value,5) value,

    DECODE(value,0,' n/a',TO\_CHAR(100 \* used / value,'990')

                            || '%') usage

FROM

    (

        SELECT

            MAX(s.value) used

        FROM

            v$statname n,

            v$sesstat s

        WHERE

            n.name = 'session cursor cache count'

            AND s.statistic# = n.statistic#

    ),

    (

        SELECT

            value

        FROM

            v$parameter

        WHERE

            name = 'session\_cached\_cursors'

    )

UNION ALL

SELECT

    'open\_cursors',

    lpad(value,5),

    TO\_CHAR(100 \* used / value,'990')

    || '%'

FROM

    (

        SELECT

            MAX(SUM(s.value) ) used

        FROM

            v$statname n,

            v$sesstat s

        WHERE

            n.name IN (

                'opened cursors current',

                'session cursor cache count'

            )

            AND s.statistic# = n.statistic#

        GROUP BY

            s.sid

    ),

    (

        SELECT

            value

        FROM

            v$parameter

        WHERE

            name = 'open\_cursors'

    );