ACS, ECS [e Cardinality/Statistics feedback]

ITOUG Tech Day – Database Stream
Milano – 8 Giugno 2017

About Donatello Settembrino

- Performance Team di ICTeam dal 2013
- Oracle (da 8i a 12c)
- Assessment prestazionali, optimizer Oracle, PL/SQL

ITOUG
ITALIAN ORACLE USER GROUP

dosettembrino

donatello settembrino

donatello.settembrino@icteam.it
performance_team@icteam.it

AGENDA

- Introduzione
- Elenco hidden parameter
- Comportamento di ACS senza istogrammi
- Comportamento di ACS con istogrammi
- Principali evidenze

Bind e literal

```
SELECT COUNT(DISTINCT n2)
FROM t
WHERE n1 = 1000
AND     n2 >= 1

SELECT COUNT(DISTINCT n2)
FROM t
WHERE n1 = :b1
AND     n2 >= :b2
```

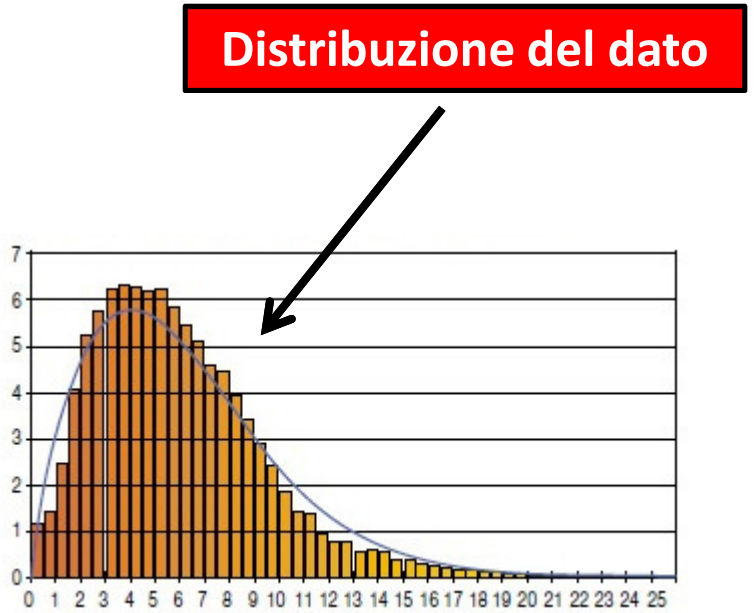
Cosa rende flessibile un piano di esecuzione

- Statistiche sugli oggetti
- Parametri CBO
- Dynamic Sampling
- Cardinality Feedback
- Cursor Sharing
- Adaptive Cursor Sharing
- Adaptive Plan

Bind peeking [1]

- Feature introdotta dalla versione 9i
- L'optimizer "sbircia" il valore della bind variable sin dalla prima esecuzione di un cursore
- L'optimizer calcola la selectivity come se il literal fosse sostituito alla bind variable
- Migliora quindi la qualità del piano di esecuzione

Bind peeking [2]



Quando l'optimizer Oracle sceglie il piano "sbirciando" la bind variable non è garantito che quel piano sia il migliore per tutti i possibili valori, ma lo è solo per quello che in letteratura è conosciuto come peeked value

Strategie differenti: Pro e contro

Uso di literal

→Hard parse per ogni cambio valore

→Uso intensivo di risorse (hard parse, child cursor - CPU, SGA)

→Piani "ottimali"

Uso di bind variable

→Evita hard parse

→Minore utilizzo di risorse

→La condivisione del piano non sempre è ottimale

Adaptive Cursor Sharing nasce con l'ambizione di ottenere il meglio dalle due strategie appena descritte

- **Riduzione dell'utilizzo di risorse**
- **Piani ottimali**

Adaptive Cursor Sharing (ACS)

- introdotto dalla versione Oracle 11g
- consente di generare ed utilizzare differenti piani ottimali in presenza di bind variable
- La caratteristica principale è quella di riconoscere se un cursore è bind sensitive (valore 'Y' nella colonna IS_BIND_SENSITIVE in V$SQL o v$SQLAREA)
- Un cursore è bind sensitive se l'optimizer Oracle ritiene che il piano ottimale può dipendere da differenti valori passati alla bind variable
- Prima di etichettare un cursore come "bind sensitive" vengono eseguiti una serie di controlli

Cursor Bind Sensitive - requisiti

Doc ID 740052.1

- query is executed with bind peeking
- binds using any of the following relational operators = < > <= >= != or a user defined bind operator e.g. contains(e.job,:job,1)>0,
  From 11.2.0.2 the "LIKE" operator is also supported
- A histogram exists on the column containing the bind value.

Cursor Bind Sensitive - check

Doc ID 740052.1

Quando ACS viene disabilitato:

- **Extended cursor sharing has been disabled**
- The query has no binds
- Parallel query is used
- Certain parameters like ("_optim_peek_user_binds"=false) are set
- You are using a /*+ NO_BIND_AWARE */ hint
- Outlines are being used
- Query is recursive
- The number of binds in a given sql statement is greater than 14. ** Could be less depending on version and setting of fix_control for Bug 10182051.

Extended Cursor Sharing (ECS)

- **Extended Cursor Sharing** (aka **Bind Aware**) è una delle condizioni necessarie affinché ACS non venga disabilitato
- Lo scopo di ECS è quello di segnare un cursore come **bind aware** se dopo la prima esecuzione viene riscontrata una significativa differenza della cardinality
- la colonna V$SQL.is_bind_aware è valorizzata con ‘Y'.

Cardinality/Statistics feedback

- **Cardinality Feedback** (Statistics Feedback in Oracle 12c) è una feature presente dalla versione 11gR2
- aiuta l'optimizer Oracle nella scelta del piano di accesso migliore quando un SQL statement viene utilizzato più volte e la stima della cardinalità non è corretta (statistiche sugli oggetti mancanti o inaccurate, predicati complessi ecc.)
- Dopo la prima esecuzione se la cardinality stimata è (significativamente) differente viene memorizzata per utilizzi futuri e lo statement viene segnato come **REOPTIMIZABLE** (nuova colonna in 12c V$SQL.is_reoptimizable.

Hidden parameter

NAME	VALUE	DEFLT	KSPPDESC
_optimizer_adaptive_cursor_sharing	TRUE	TRUE	optimizer adaptive cursor sharing
_optimizer_extended_cursor_sharing	UDO	TRUE	optimizer extended cursor sharing
_optimizer_extended_cursor_sharing_rel	SIMPLE	TRUE	optimizer extended cursor sharing for relational operators
_optimizer_use_feedback	TRUE	TRUE	optimizer use feedback

- **_optimizer_adaptive_cursor_sharing** (TRUE/FALSE)
- **_optimizer_use_feedback** (TRUE/FALSE)
- **_optimizer_extended_cursor_sharing** (NONE/UDO)
- **_optimizer_extended_cursor_sharing_rel** (NONE/SIMPLE)

Esecuzione dei test

select count(distinct n2) from feedback where n1 = :b1 and n2 >= :b2

- 1° esecuzione: accesso con indice; (11g phv 3994666447 – 12c phv 2767038796)
- 2° esecuzione: accesso in Full Table Scan; (11g e 12c phv 1141924756)
- 3° esecuzione: accesso in Full Table Scan;
- 4° esecuzione: accesso in Full Table Scan;
- 5° esecuzione: accesso con indice;

Esecuzione dei test - Execution plan

select count(distinct n2) from feedback where n1 = :b1 and n2 >= :b2

Differente PHV (ma ''identico'' execution plan, accesso con indice)

11g

```
Plan hash value: 3994666447

-----------------------------------------------------
| Id  | Operation                    | Name          |
-----------------------------------------------------
|   0 | SELECT STATEMENT             |               |
|   1 |  SORT AGGREGATE              |               |
|   2 |   VIEW                       | VW_DAG_0      |
|   3 |    HASH GROUP BY             |               |
|   4 |     TABLE ACCESS BY INDEX ROWID| FEEDBACK      |
|   5 |      INDEX RANGE SCAN        | FEEDBACK_N1_IDX |
-----------------------------------------------------
```

12c

```
Plan hash value: 2767038796

-------------------------------------------------------------------------
| Id  | Operation                             | Name          |
-------------------------------------------------------------------------
|   0 | SELECT STATEMENT                      |               |
|   1 |  SORT AGGREGATE                       |               |
|   2 |   VIEW                                | VW_DAG_0      |
|   3 |    HASH GROUP BY                      |               |
|   4 |     TABLE ACCESS BY INDEX ROWID BATCHED| FEEDBACK      |
|   5 |      INDEX RANGE SCAN                 | FEEDBACK_N1_IDX |
-------------------------------------------------------------------------
```

11g e 12c

```
Plan hash value: 1141924756

----------------------------------------
| Id  | Operation          | Name      |
----------------------------------------
|   0 | SELECT STATEMENT   |           |
|   1 |  SORT AGGREGATE    |           |
|   2 |   VIEW             | VW_DAG_0  |
|   3 |    HASH GROUP BY   |           |
|   4 |     TABLE ACCESS FULL| FEEDBACK  |
----------------------------------------
```

La differenza dell'execution plan (PHV) è dovuta a:
db file parallel read o db file scattered read oppure
db file sequential read

Prime evidenze

- Il parametro **_optimizer_extended_cursor_sharing** non sembra avere differenti effetti modificando il valore
- Imponendo i due differenti valori (UDO – NONE) non sono state rilevate differenze comportamentali
- UDO – User Defined Operation (or Operator), sembrerebbe essere utilizzato in presenza di operatori definiti dall'utente

In assenza di istogrammi

- con **_optimizer_adaptive_cursor_sharing = FALSE** (quindi ACS spento) e senza istogrammi il piano quindi non è mai **"adaptive"** , il primo piano calcolato resta valido per le successive esecuzioni anche al variare degli altri hidden parameter
- **Doc ID 740052.1**

11g e 12c

In assenza di istogrammi - piccole differenze

- con **_optimizer_adaptive_cursor_sharing = FALSE** (quindi ACS spento) e senza istogrammi il piano quindi non è mai **"adaptive"**
- Al variare di **_optimizer_extended_cursor_sharing_rel:** il piano resta invariato ma cambiano le caratteristiche del cursore

Legenda:

s – bind sensitive
B – bind aware
S - sharable

```
NAME                                     VALUE
---------------------------------------- ----------------
_optimizer_adaptive_cursor_sharing       FALSE
_optimizer_extended_cursor_sharing       UDO
_optimizer_extended_cursor_sharing_rel   SIMPLE
_optimizer_use_feedback                  TRUE

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 2767038796 Y Y Y          1

. . .

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 2767038796 Y Y Y          5
```

```
NAME                                     VALUE
---------------------------------------- ----------------
_optimizer_adaptive_cursor_sharing       FALSE
_optimizer_extended_cursor_sharing       UDO
_optimizer_extended_cursor_sharing_rel   NONE
_optimizer_use_feedback                  TRUE

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 2767038796 N N Y          1

. . .

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 2767038796 N N Y          5
```

E con ACS attivo?

- **_optimizer_adaptive_cursor_sharing = TRUE** (quindi ACS acceso) il piano non cambia mai, il piano calcolato durante la prima esecuzione resta invariato anche per le successive
- (**_optimizer_extended_cursor_sharing_rel = SIMPLE**) e con ECS attivo cambia qualcosa?
- variando il valore di **_optimizer_use_feedback** ho comportamenti differenti ?

In assenza di istogrammi

- _optimizer_adaptive_cursor_sharing = TRUE
- _optimizer_extended_cursor_sharing_rel = SIMPLE
- _optimizer_use_feedback (spento)

```
NAME                                     VALUE
---------------------------------------- ---------------
_optimizer_adaptive_cursor_sharing       TRUE
_optimizer_extended_cursor_sharing       UDO
_optimizer_extended_cursor_sharing_rel   SIMPLE
_optimizer_use_feedback                  FALSE

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 2767038796 Y N Y          1

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 2767038796 Y N Y          2

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 2767038796 Y N N          2
6avmk9kmqnx0t            1 2767038796 Y Y Y          1

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 2767038796 Y N N          2
6avmk9kmqnx0t            1 2767038796 Y Y Y          2

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 2767038796 Y N N          2
6avmk9kmqnx0t            1 2767038796 Y Y Y          3
```

11g e 12c - stesso comportamento

In assenza di istogrammi - cardinality feedback acceso

- _optimizer_adaptive_cursor_sharing = TRUE
- _optimizer_extended_cursor_sharing_rel = SIMPLE

11g

```
NAME                                     VALUE
---------------------------------------- ---------------
_optimizer_adaptive_cursor_sharing       TRUE
_optimizer_extended_cursor_sharing       UDO
_optimizer_extended_cursor_sharing_rel   SIMPLE
_optimizer_use_feedback                  TRUE

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 3994666447 Y N Y          1

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 3994666447 Y N Y          2

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 3994666447 Y N N          2
6avmk9kmqnx0t            1 3994666447 Y Y Y          1

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 3994666447 Y N N          2
6avmk9kmqnx0t            1 3994666447 Y Y Y          2

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 3994666447 Y N N          2
6avmk9kmqnx0t            1 3994666447 Y Y Y          3
```

CARDINALITY FEEDBACK

CARDINALITY FEEDBACK

CARDINALITY FEEDBACK

12c R1

```
NAME                                     VALUE
---------------------------------------- ---------------
_optimizer_adaptive_cursor_sharing       TRUE
_optimizer_extended_cursor_sharing       UDO
_optimizer_extended_cursor_sharing_rel   SIMPLE
_optimizer_use_feedback                  TRUE

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 2767038796 Y N Y          1

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 2767038796 Y N Y          2

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 2767038796 Y N N          2
6avmk9kmqnx0t            1 2767038796 Y Y Y          1

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 2767038796 Y N N          2
6avmk9kmqnx0t            1 2767038796 Y Y N          1
6avmk9kmqnx0t            2 1141924756 Y Y Y          1

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 2767038796 Y N N          2
6avmk9kmqnx0t            1 2767038796 Y Y N          1
6avmk9kmqnx0t            2 1141924756 Y Y N          1
6avmk9kmqnx0t            3 2767038796 Y Y Y          1
```

STATISTICS FEEDBACK

Legenda:

s – bind sensitive
B – bind aware
S - sharable

Si torna indietro in 12.2?

12c R2

```
NAME                                     VALUE
---------------------------------------- ---------------
_optimizer_adaptive_cursor_sharing       TRUE
_optimizer_extended_cursor_sharing       UDO
_optimizer_extended_cursor_sharing_rel   SIMPLE
_optimizer_use_feedback                  TRUE

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 2767038796 Y N Y          1

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 2767038796 Y N Y          2

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 2767038796 Y N N          2
6avmk9kmqnx0t            1 2767038796 Y Y Y          1

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 2767038796 Y N N          2
6avmk9kmqnx0t            1 2767038796 Y Y Y          2

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 2767038796 Y N N          2
6avmk9kmqnx0t            1 2767038796 Y Y Y          2
```

Legenda:

s – bind sensitive
B – bind aware
S - sharable

Cambio di strategia (Release 12.2)

Initialization Parameter Default Changes in Oracle Database 12c Release 2 (12.2)

Review this list of initialization parameter default setting changes for Oracle Database 12*c* release 2 (12.2).

OPTIMIZER_ADAPTIVE_PLANS and OPTIMIZER_ADAPTIVE_STATISTICS

OPTIMIZER_ADAPTIVE_FEATURE functions are replaced by two new parameters: OPTIMIZER_ADAPTIVE_PLANS, and OPTIMIZER_ADAPTIVE_STATISTICS.

OPTIMIZER_ADAPTIVE_PLANS controls adaptive plans. It is set by default to TRUE. When set to TRUE, this parameter determines alternate execution plans built with alternative choices that are based on statistics collected as a query executes.

OPTIMIZER_ADAPTIVE_STATISTICS controls adaptive statistics. It is set by default to FALSE. When set to TRUE, the optimizer augments the statistics gathered in the database with adaptive statistics gathered at SQL statement parse time to improve the quality of SQL execution plans. Some query shapes are too complex to rely upon base table statistics alone. The optimizer augments them with adaptive statistics to determine more accurately the best SQL execution plan.

ALTER SESSION SET optimizer_adaptive_statistics = true;

Statistics feedback per singola tabella (NO JOIN) è sempre abilitato

Utilizzo di Statistics/Cardinality Feedback

L'utilizzo di Statistics feedback 12c**R1** (ma anche Cardinality feedback in 11g) **in assenza di istogrammi** è stato rilevato solo in presenza della seguente combinazione di parametri:

_optimizer_adaptive_cursor_sharing = TRUE
_optimizer_extended_cursor_sharing_rel = SIMPLE
_optimizer_use_feedback = TRUE

In 12c**R2** nel test case Statistics feedback **non è mai stato utilizzato**

In assenza di istogrammi- Nota

- **In 11g** anche con l'intervento di Cardinality Feedback **il piano non cambia mai**

- **In 12cR1 l'intervento di Statistics Feedback in assenza di istogrammi e quindi con ACS spento** (l'assenza di istogrammi sulle colonne coinvolte nei predicati che contengono bind variable preclude l'intervento di ACS), **sembrerebbe quindi contribuire a rendere comunque il piano "adaptive"**

- **In 12cR2 si ha lo stesso comportamento 11g ma non è stato notato l'intervento di Statistics Feedback**

Cosa succede in presenza di istogrammi?

- una colonna referenziata nella WHERE clause i cui dati sono rappresentativi di una distribuzione asimmetrica probabilmente presenta un istogramma calcolato durante la raccolta delle statistiche.

- alla prima esecuzione l'optimizer Oracle determina che l'esecuzione del cursore dipende dal valore assegnato alla bind, il cursore (child_number #0) viene quindi "segnalato" come **bind sensitive**.

In presenza di istogrammi

- L'attribuzione di bind sensitive fa si che venga attivato un **monitoring** del cursore al variare del valore passato alla bind variable per stabilire se è necessario un cambio piano

- Quando un nuovo piano viene individuato (perché il precedente non è ritenuto ottimale per l'esecuzione corrente) viene creato un nuovo child_number (#1), il cursore viene segnato come bind aware ed il precedente child_number (#0) viene dichiarato non sharable (cioè non viene più utilizzato, successivamente oggetto di flush dallo shared pool).

Monitoring cursor - informazioni di dettaglio

V$SQL_CS_SELECTIVITY

V$SQL_CS_SELECTIVITY exposes the valid selectivity ranges for a child cursor in extended cursor sharing mode. A valid range consists of a low and high value for each predicate containing binds. Each predicate's selectivity (with the current bind value) must fall between the corresponding low and high values in order for the child cursor to be shared.

Column	Datatype	Description
ADDRESS	RAW(4)	Address of the handle to the parent for this cursor
HASH_VALUE	NUMBER	Hash value of the parent statement in the library cache
SQL_ID	VARCHAR2(13)	SQL identifier of the parent cursor in the library cache
CHILD_NUMBER	NUMBER	Number of the child cursor
PREDICATE	VARCHAR2(40)	Predicate whose selectivity must fall between low and high values
RANGE_ID	NUMBER	Identifier for the range used to match up the low and high values for multiple predicates
LOW	VARCHAR2(10)	Lower bound for allowable selectivity
HIGH	VARCHAR2(10)	Upper bound for allowable selectivity

V$SQL_CS_HISTOGRAM

V$SQL_CS_HISTOGRAM summarizes the monitoring information stored by adaptive cursor sharing. This information is used to decide whether to enable extended cursor sharing for a query. It is stored in a histogram, whose bucket's contents are exposed by this view.

Column	Datatype	Description
ADDRESS	RAW(4)	Address of the handle to the parent for this cursor
HASH_VALUE	NUMBER	Hash value of the parent statement in the library cache
SQL_ID	VARCHAR2(13)	SQL identifier of the parent cursor in the library cache
CHILD_NUMBER	NUMBER	Number of the child cursor being monitored
BUCKET_ID	NUMBER	Bucket number of the monitoring histogram
COUNT	NUMBER	Value in this bucket of the histogram

V$SQL_CS_STATISTICS

V$SQL_CS_STATISTICS contains the raw execution statistics used by the monitoring component of adaptive cursor sharing. A sample of the executions is monitored. This view exposes which executions were sampled, and what the statistics were for those executions. The statistics are cumulative for each distinct set of bind values.

Column	Datatype	Description
ADDRESS	RAW(4)	Address of the handle to the parent for this cursor
HASH_VALUE	NUMBER	Hash value of the parent statement in the library cache
SQL_ID	VARCHAR2(13)	SQL identifier of the parent cursor in the library cache
CHILD_NUMBER	NUMBER	Number of the child cursor being monitored
BIND_SET_HASH_VALUE	NUMBER	Hash of the values of the binds
PEEKED	VARCHAR2(1)	Indicates if this is the bind set used to build the cursor (Y) or not (N)
EXECUTIONS	NUMBER	Number of times this bind set has been executed and monitored
ROWS_PROCESSED	NUMBER	Cumulative number of rows processed by all row sources in the plan over all monitored executions with this bind set
BUFFER_GETS	NUMBER	Cumulative number of buffer gets over all monitored executions with this bind set
CPU_TIME	NUMBER	Cumulative CPU time (in microseconds) used by this cursor for monitored executions with this bind set

Obsoleta da 12c

Monitoring cursor - esempio

```
SQL> exec :b1 := 1000;
SQL> exec :b2 := 1;
SQL> select count(distinct n2) from feedback where n1 = :b1 and n2 >= :b2;

COUNT(DISTINCTN2)
-----------------
                1

Plan hash value: 2767038796
-------------------------------------------------------------------------
| Id  | Operation                                  | Name             |
-------------------------------------------------------------------------
|   0 | SELECT STATEMENT                           |                  |
|   1 |  SORT AGGREGATE                            |                  |
|   2 |   VIEW                                     | VW_DAG_0         |
|   3 |    HASH GROUP BY                           |                  |
|   4 |     TABLE ACCESS BY INDEX ROWID BATCHED| FEEDBACK         |
|   5 |      INDEX RANGE SCAN                      | FEEDBACK_N1_IDX  |
-------------------------------------------------------------------------

SQL> select child_number, executions, buffer_gets,is_bind_sensitive BS, is_bind_aware BA, plan_hash_value
  2  from v$sql
  3  where sql_id = '6avmk9kmqnx0t';

CHILD_NUMBER EXECUTIONS BUFFER_GETS B B PLAN_HASH_VALUE
------------ ---------- ----------- - - ---------------
           0          1          40 Y N      2767038796

SQL> select CHILD_NUMBER,PREDICATE,RANGE_ID,LOW,HIGH
  2  from v$sql_cs_selectivity
  3  where sql_id = '6avmk9kmqnx0t';

Nessuna riga selezionata
```

Monitoring cursor - esempio

```
SQL> exec :b1 := 1;
SQL> exec :b2 := 1;
SQL> select count(distinct n2) from feedback where n1 = :b1 and n2 >= :b2;

COUNT(DISTINCTN2)
-----------------
           102400

Plan hash value: 2767038796
-------------------------------------------------------------------------
| Id  | Operation                                | Name             |
-------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                  |
|   1 |  SORT AGGREGATE                          |                  |
|   2 |   VIEW                                   | VW_DAG_0         |
|   3 |    HASH GROUP BY                         |                  |
|   4 |     TABLE ACCESS BY INDEX ROWID BATCHED| FEEDBACK         |
|   5 |      INDEX RANGE SCAN                    | FEEDBACK_N1_IDX  |
-------------------------------------------------------------------------
```

```
SQL> select child_number, executions, buffer_gets,is_bind_sensitive BS, is_bind_aware BA,plan_hash_value
  2  from v$sql
  3  where sql_id = '6avmk9kmqnx0t';

CHILD_NUMBER EXECUTIONS BUFFER_GETS B B PLAN_HASH_VALUE
------------ ---------- ----------- - - ---------------
           0          2       51422 Y N      2767038796

SQL> select CHILD_NUMBER,PREDICATE,RANGE_ID,LOW,HIGH
  2  from v$sql_cs_selectivity
  3  where sql_id = '6avmk9kmqnx0t';

Nessuna riga selezionata
```

Monitoring cursor - esempio

```
SQL> select count(distinct n2) from feedback where n1 = :b1 and n2 >= :b2;

COUNT(DISTINCTN2)
-----------------
           102400

Plan hash value: 1141924756
-----------------------------------------
| Id  | Operation             | Name     |
-----------------------------------------
|   0 | SELECT STATEMENT      |          |
|   1 |  SORT AGGREGATE       |          |
|   2 |   VIEW                | VW_DAG_0 |
|   3 |    HASH GROUP BY      |          |
|   4 |     TABLE ACCESS FULL| FEEDBACK |
-----------------------------------------
```

```
SQL> select child_number, executions, buffer_gets,is_bind_sensitive BS, is_bind_aware BA,plan_hash_value
  2  from v$sql
  3  where sql_id = '6avmk9kmqnx0t';

CHILD_NUMBER EXECUTIONS BUFFER_GETS B B PLAN_HASH_VALUE
------------ ---------- ----------- - - ---------------
           0          2       51422 Y N      2767038796
           1          1       68287 Y Y      1141924756

SQL> select CHILD_NUMBER,PREDICATE,RANGE_ID,LOW,HIGH
  2  from v$sql_cs_selectivity
  3  where sql_id = '6avmk9kmqnx0t';

CHILD_NUMBER PREDICATE                                  RANGE_ID LOW       HIGH
------------ ---------------------------------------- ---------- --------- ---------
           1 =B1                                               0 0.446524  0.545751
           1 >=B2                                              0 0.900000  1.100000
```

Monitoring cursor - esempio

```
SQL> exec :b1 := 1000;
SQL> exec :b2 := 1;
SQL> select count(distinct n2) from feedback where n1 = :b1 and n2 >= :b2;

COUNT(DISTINCTN2)
-----------------
                1

Plan hash value: 2767038796
-------------------------------------------------------------------
| Id  | Operation                            | Name            |
-------------------------------------------------------------------
|   0 | SELECT STATEMENT                     |                 |
|   1 |  SORT AGGREGATE                      |                 |
|   2 |   VIEW                               | VW_DAG_0        |
|   3 |    HASH GROUP BY                     |                 |
|   4 |     TABLE ACCESS BY INDEX ROWID BATCHED| FEEDBACK        |
|   5 |      INDEX RANGE SCAN                | FEEDBACK_N1_IDX |
-------------------------------------------------------------------

SQL> select child_number, executions, buffer_gets,is_bind_sensitive BS, is_bind_aware BA,plan_hash_value
  2  from v$sql where sql_id = '6avmk9kmqnx0t';

CHILD_NUMBER EXECUTIONS BUFFER_GETS B B PLAN_HASH_VALUE
------------ ---------- ----------- - - ---------------
           0          2       51422 Y N      2767038796
           1          1       68287 Y Y      1141924756
           2          1           3 Y Y      2767038796

SQL> select CHILD_NUMBER,PREDICATE,RANGE_ID,LOW,HIGH
  2  from v$sql_cs_selectivity where sql_id = '6avmk9kmqnx0t';

CHILD_NUMBER PREDICATE                                  RANGE_ID LOW         HIGH
------------ ---------------------------------------- ---------- ---------- ----------
           2 =B1                                               0 0.000004   0.000005
           2 >=B2                                              0 0.900000   1.100000
           1 =B1                                               0 0.446524   0.545751
           1 >=B2                                              0 0.900000   1.100000
```

Monitoring cursor - esempio

```
SQL> select child_number, bucket_id , count from v$sql_cs_histogram
where sql_id = '6avmk9kmqnx0t';

CHILD_NUMBER  BUCKET_ID      COUNT
------------ ---------- ----------
           2          0  small    1
           2          1  medium   0
           2          2  large    0
           1          0           0
           1          1           1
           1          2           0
           0          0           1
           0          1           1
           0          2           0
```

In presenza di istogrammi

- in presenza di istogrammi il piano diventa “adaptive” solo quando **ECS** è acceso cioè quando il parametro **_optimizer_extended_cursor_sharing_rel** risulta essere impostato al valore **SIMPLE**, ciò avviene sia con ACS acceso (**_optimizer_adaptive_cursor_sharing = TRUE**) che con ACS spento (**_optimizer_adaptive_cursor_sharing = FALSE**).

- In realtà sarebbe quindi più corretto dire che con **ECS acceso e ACS spento** il piano diventa adaptive in virtù della presenza di istogrammi sulla colonna . Il piano (probabilmente) corretto viene individuato ad ogni esecuzione grazie proprio alla selettività imposta dagli istogrammi

In presenza di istogrammi

```
NAME                                     VALUE           DEFLT
---------------------------------------- --------------- ---------
_optimizer_adaptive_cursor_sharing       FALSE           TRUE
_optimizer_extended_cursor_sharing       UDO             TRUE
_optimizer_extended_cursor_sharing_rel   SIMPLE          TRUE
_optimizer_use_feedback                  FALSE           TRUE


SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 3994666447 Y Y Y          1

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 3994666447 Y Y Y          1
6avmk9kmqnx0t            1 1141924756 Y Y Y          1

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 3994666447 Y Y Y          1
6avmk9kmqnx0t            1 1141924756 Y Y Y          2

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 3994666447 Y Y Y          1
6avmk9kmqnx0t            1 1141924756 Y Y Y          3

SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 3994666447 Y Y Y          2
6avmk9kmqnx0t            1 1141924756 Y Y Y          3
```

Viene bypassato il controllo su bind sensitive

Già da subito alla seconda esecuzione viene individuato il piano corretto

Legenda:

s – bind sensitive
B – bind aware
S - sharable

In presenza di istogrammi

Quando **ECS e ACS sono invece entrambi accesi**, l'optimizer riceve questa volta il contributo di Adaptive Cursor Sharing, per accorgersi quando è necessario un nuovo piano occorrono alcune esecuzioni:

```
NAME                                     VALUE           DEFLT
---------------------------------------- --------------- ---------
_optimizer_adaptive_cursor_sharing       TRUE            TRUE
_optimizer_extended_cursor_sharing       UDO             TRUE
_optimizer_extended_cursor_sharing_rel   SIMPLE          TRUE
_optimizer_use_feedback                  TRUE            TRUE
```

```
SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 3994666447 Y N Y          1
```

```
SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 3994666447 Y N Y          2
```

```
SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 3994666447 Y N N          2
6avmk9kmqnx0t            1 1141924756 Y Y Y          1
```

```
SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 3994666447 Y N N          2
6avmk9kmqnx0t            1 1141924756 Y Y Y          2
```

```
SQL_ID        CHILD_NUMBER        PHV s B S EXECUTIONS
------------- ------------ ---------- - - - ----------
6avmk9kmqnx0t            0 3994666447 Y N N          2
6avmk9kmqnx0t            1 1141924756 Y Y Y          2
6avmk9kmqnx0t            2 3994666447 Y Y Y          1
```

Sono necessarie 3 esecuzioni prima che venga utilizzato un nuovo piano di esecuzione

In presenza di istogrammi (terza evidenza)

Quando invece il parametro _optimizer_extended_cursor_sharing_rel = NONE il piano non è mai adaptive, non cambia mai e resta sempre quello calcolato alla prima esecuzione.

SQL_ID	CHILD_NUMBER	PHV	s	B	S	EXECUTIONS
6avmk9kmqnx0t	0	3994666447	N	N	Y	1
6avmk9kmqnx0t	0	3994666447	N	N	Y	2
6avmk9kmqnx0t	0	3994666447	N	N	Y	3
6avmk9kmqnx0t	0	3994666447	N	N	Y	4
6avmk9kmqnx0t	0	3994666447	N	N	Y	5

11g

SQL_ID	CHILD_NUMBER	PHV	s	B	S	EXECUTIONS
6avmk9kmqnx0t	0	2767038796	N	N	Y	1
6avmk9kmqnx0t	0	2767038796	N	N	Y	2
6avmk9kmqnx0t	0	2767038796	N	N	Y	3
6avmk9kmqnx0t	0	2767038796	N	N	Y	4
6avmk9kmqnx0t	0	2767038796	N	N	Y	5

12c

Hint

- /*+ BIND_AWARE */

non esegue la fase di monitoraggio sul bind sensitive

- /*+ NO_BIND_AWARE */

spegne ACS

Impatto di statistics_level su ACS e Card feedback

"In any case, when STATISTICS_LEVEL is set to BASIC at the *system level* both features are disabled "

https://antognini.ch/2011/09/impact-of-statistics_level-on-cardinality-feedback-and-adaptive-cursor-sharing/

Possibili problemi utilizzando ECS

Doc ID 296377.1

- *BIND_EQUIV_FAILURE*

The bind value's selectivity does not match that used to optimize the existing child cursor. When adaptive cursor sharing is used and the cursor is bind aware, then if the selectivity is outside of the current ranges and a new plan is desirable then a new child is raised with this as the reason code for non-sharing of the previous plan. For an example, see Document 836256.1. After each execution in the example, run:

```
select sql_id, address, child_address, child_number, BIND_EQUIV_FAILURE from v$sql_shared_cursor where sql_id='19sxt3v07nzm4';
```

... once the cursor is marked as bind aware and a second plan is seen then the following will be the resultant output:

```
SQL_ID        ADDRESS          CHILD_ADDRESS    CHILD_NUMBER B
------------- ---------------- ---------------- ------------ -
19sxt3v07nzm4 000000007A1C0DE0 000000007A1BF980            0 N
19sxt3v07nzm4 000000007A1C0DE0 000000007A10DDB0            1 Y
```

As can be seen, the new version is created due to BIND_EQUIV_FAILURE

Questions and Answers