

V\$SYSSTAT

SQL 実行処理の進捗個数確認 (オラクル起動後の累計値) (Oracle 全体での統計値)

v\$sysstat の見方

v\$sysstat にて、SQL 文処理数やディスクとの I/O 数が増加しているかを確認して、Oracle システム **全体**での稼働状態を確認する

```
Select name , value from v$sysstat
```

value 値は、オラクル起動後の**累計値**

```
where name in ( 'execute count' , 'user commits' , 'physical read bytes' ,  
                'physical write bytes' , 'bytes sent via SQL*Net to client' ,  
                'bytes received via SQL*Net from client' , 'db block gets' ,  
                'db block gets from cache' );
```

name 列の値	Value 列が意味する値
execute count	SQL 実行数
user commits	commit 実行数 ≒スループット数と考える
physical read bytes	ディスクからの読み込みデータ量
physical write bytes	ディスクへの書き込みデータ量
bytes sent via SQL*Net to client	Oracle Net のデータ送信量
bytes received via SQL*Net from client	Oracle Net のデータ受信量
db block gets	logical read での読み込みデータ量
db block gets from cache	キャッシュからの読み込データ量

[確認ポイント]

- ・ value 列の値が、増加しているか
(直接の数値が大きいことではなく、2 回実行した時の数値の差分)
増加数が少なければ、Oracle システムが処理していないことを意味する

V\$SESSTAT

SQL 文に対する実行統計情報（セッション別）

CPU リソースを大量に使用しているセッションの確認

Redo ログを大量に更新しているセッションの確認

V\$SESSTAT ビュー

V\$SESSTAT ビューは、**セッション番号別**の ~~SQL 文の実行~~に対する実行統計の情報である

ただし、1 つの SQL 文に対する統計情報ではなく、**その接続セッション（サーバー・プロセス）**で処理を行ったすべての SQL 文の処理のすべてを合計した統計情報となる

なお、一度**セッションが切断**されると、**VALUE 値はクリア**される

【注 意】

~~サーバー・プロセスは、クライアントからのセッションの切断があった後も、接続待機状態でサーバーに残っている~~

~~次にクライアントからの接続があった場合には、同一のセッション ID (sid) で処理が行われる~~

~~よって、複数回の接続がクライアントから行われた場合には、そのすべての処理の合計の実行統計値が V\$SESSTAT ビューの値となってしまう~~

主な列項目

列 名	説 明
value	統計値（サーバー・プロセスへ接続してからの 累計値 ）
sid	セッション ID
statistic#	統計情報の番号 (V\$STATNAME の statistic#列に対応)

説明) V\$SESSTAT は、多数の統計情報の値が出力されてくるが、その統計項目の種類が列に分かれてくるのではなく、statistic#の値によって value 列の値が何を意味するのかが決定される

V\$STATNAME ビューの主な列値

列 名	説 明
statistic#	統計情報の番号 (V\$STATNAME の statistic#列に対応)
NAME	統計情報の番号が、何の統計情報かを示す説明

V\$STATNAME ビューの NAME 値

0 logons cumulative	7 recursive calls
1 logons current	8 recursive cpu usage
2 opened cursors cumulative	9 session logical reads
3 opened cursors current	10 session stored procedure space
4 user commits	•
5 user rollbacks	•
6 user calls	•

12 「CPU used by this session」は、該当セッションが使用した CPU 時間です
単位はセンチ秒であるため、値が 1053 の場合は 10.53 秒の CPU 使用を意味します

※ 各統計番号(STATISTIC#)に対応した統計名は、Oracle のバージョンごとにこと
なるので、毎回 V\$STATNAME ビューを Select して使用すること

```
Select  STATISTIC# , NAME  From  V$STATNAME  Where
        UPPER( NAME ) like  '%CPU%'  ;
```

使用例)

前調査 (V\$SESSTAT の出力レコードが、何に対する値なのか判別するための区分番号)

/* 待機イベントに対する statistic#コード番号の調査 */

```
SELECT name, statistic#
FROM V$STATNAME
WHERE name in ('redo size', 'CPU used by this session');
```

name	statistic#
CPU used by this session	12
redo size	134

CPU リソースを大量に使用しているセッションの調査方法

/* statistic#コード番号に対する統計情報の下限値との比較調査 */

```
SELECT SE.sid, SE.serial#, SE.username, ST.statistic#, ST.value
FROM V$SESSION SE, V$SESSTAT ST
WHERE SE.sid = ST.sid
      AND SE.username = '<ユーザー名>'
      AND SE.type = 'USER'
      AND ST.statistic# in (134) ←
      AND ST.value > 比較下限値 ;
```

sid	serial#	username	statistic#	value
141	26	SCOTT	134	269487

Redo ログを大量に使用 (更新多発) している処理の調査方法

/* statistic#コード番号に対する統計情報の下限値との比較調査 */

```
SELECT SE.sid, SE.serial#, SE.username, ST.statistic#, ST.value
FROM V$SESSION SE, V$SESSTAT ST
WHERE SE.sid = ST.sid
      AND SE.username = '<ユーザー名>'
      AND SE.type = 'USER'
      AND ST.statistic# in (12) ←
      AND ST.value > 比較下限値 ;
```

sid	serial#	username	statistic#	value
163	28	SCOTT	12	357812

クライアントとの接続が切れた時に、
そのセッションの V\$SESSTAT ビューの『実行統計値がクリアされる』
という確認テストの実行

テスト使用セッション番号：117

使用セッション番号の確認 SQL：

```
Select  sid ,  username  from  v$session  SE
      Where  username  =  'KOZUE' ;
```

統計情報の値確認用 SQL：

```
Col  username  Format  A10
```

```
Select  SN.name ,  ST.value ,  ST.sid ,  SE.serial# ,  ST.statistic# ,
      SE.username
      From  v$session  SE ,  v$sesstat  ST ,  v$statname  SN
      Where  ST.sid  =  117
      and  SE.sid  =  ST.sid
      and  ST.statistic#  in ( 11 , 12 )
      and  ST.statistic#  =  SN.statistic# ;
```

統計項目	接続直後	データ作成後	切断時	再接続後
CPU used when call started	0	8	レコードが選 択されません でした	2
CPU used by this session	0	389		2

統計情報の値確認用 SQL：

```
create  table  test_num(col1  number ,  col2  number );
```

```
begin
```

```
  dbms_random.SEED(1) ;
```

```
  for i in 1..100000 loop
```

```
    insert into test_num values(i, ceil(dbms_random.value( 1 , 1000000000 ))) ;
```

```
  end loop ;
```

```
end ;
```

```
/
```

```
select  count(*)  from  (
      select  col2 ,  count(*)  from  test_num
      group by  col2  having  count(*) >= 2 ) ;
```

```
drop  table  test_num;
```