

## 「SQL 実行統計情報」の見方

### 【SQL トレースの「SQL 実行統計情報」】

#### 特 徴

セッションにて、トレース取得を取得するので、調査対象が限定できる

セッション中の**多数**の SQL 文に対する SQL 実行統計情報が出力されるので、調査したい SQL 文の部分、トレース結果を作成した後から選別することが出来る

```
select department_name from employees e, departments d
where e.department_id = d.department_id
and salary > 1000
```

call	count	cpu	elapsed	disk	query	current	rows
↑	<b>解析数</b>	↑	<b>CPU+待機時間</b>	↑			↑
<b>解析種</b>		<b>CPU 時間</b>		<b>ディスク読取りブロック</b>			<b>アクセスレコード数</b>
Parse	1	0.02	0.06	0	0	0	0
Execute	1	0.00	0.00	0	0	0	0
Fetch	4	0.01	0.24	187	292	0	35
total	6	0.03	0.30	187	292	0	35

query : バッファ上の読取りブロックにアクセスした数

current : バッファ上のブロックを更新した数

Rows	Rows	Source	Operation
-- ↑ --			
<b>アクセスレコード数</b>	↑		
		<b>Oracle が行う内部操作</b>	
0		SELECT	
100		TABLE ACCESS FULL	

実行計画に関する  
情報

## 【autotrace の「SQL 実行統計情報」】

### 特 徴

autotrace では、コマンド実行直後の SQL\*PLUS 画面に直接 SQL 実行統計情報が表示される

実際にアクセスしたブロック数が、正確に把握できる

### ・実行統計 情報

-----

0	recursive calls	
0	db block gets	
3	consistent gets	←アクセスしたブロック数 (バッファ分+ディスク I/O 分)
1	physical reads	←ディスク I/O したブロック数
0	redo size	
410	byte sent via SQL*NET to client	
512	bytes received via SQL*NET from client	
2	SQL*NET roundtrips to/from client	
0	sorts (memory)	
0	sorts (disk)	
1	rows processed	

この2つの項目の数値を小さくすれば、実行時間を短くできる

実行統計

### — 見 方 —

(1) consistent gets、physical reads の指標を重要視すること

~~(2) SQL 文の処理時間は、consistent gets の値に比例した時間となる。比例しない処理時間の SQL 文を見つけたら原因を調べる~~

## 【v\$SQLAREA ビューからの「実行統計情報」】

SQL 文、HASH\_VALUE 値から、その SQL 文がリソースをどれくらい使用したかの調査

特 徴

操作が面倒であるが、他の方法がとれない場合に使える

出力した実行統計情報：

SQL\_TEXT

```
-----  
EXECUTIONS  DISK_READ  BUFFER_GETS  HASH_VALUE  ADDRESS  
-----  
CPU_TIME  ELAPSED_TIME  
-----
```

以上 (↑) 列名

以下 (↓) Select の結果 (レコードの列の値)

SQL テキスト文

```
select * from big_test4 where no = :no
```

実行回数	DISK 読込回数	バッファ取得回数	hash_value 値	アドレス値
1	9	124	3188686040	0000000739111893
CPU 時間合計		実行時間総計		
6807		8576		

## 【Statspack レポートと Statspack 詳細レポートによる「SQL 実行統計情報」】

Statspack を代表とする Oracle の稼働状況統計は、対象 SQL 文を実行した時間が提供される

よって、得られる情報は、この SQL は時間が長時間であったかの判断資料にのみ使える  
すなわち、長時間の SQL 文がどれであったかを検出するには、とても有効なデータである

これに対して、SQL 文の実行時間のスピードアップのために改修を検討する場合には、実行計画に示されているオペレーション（操作）を視点にして、各オペレーションごとのデータアクセス量を調査対象とする

そして、そのデータアクセス量を判断基準にして、アクセス方法が妥当か、表の結合方法は適切かを検討して、改善していく必要がある

この部分については、Statspack 詳細レポート、SQL トレース・ファイルの SQL 実行統計情報もしくは、OEM 画面の「SQL ワークシート」の [プラン] タブ部分に出力される

なお、SQL 文の実行時間の短縮を考えて、SQL 文の改修を行った場合に、実行統計の比較に使う評価項目は、  
アクセス Byte より Cost を評価した方がよい

これは、データベースのブロック読み込みを行いまとめて読み込んだ時と、インデックス使用時の対象レコードのみ読み込んだ時の 1 バイト当たりの読み込み時間は、ブロック読み込みの方が短い（約 1 / 8）

読み込みデータ量にこのアクセス効率が考慮されて、実行時間が決まる

このことを踏まえているのが、Cost 出力である

## 【Statspack の「SQL 実行統計情報」】

Statspack のレポートでは、指定した 2 つの snap\_id の間<sup>間</sup>で発生した SQL 文の実行計画と実行統計が出力される。なお、入力する snap\_id は非連続でよい。

### 特 徴

- ・処理を実行してからの事後でも、調査が出来る
- ・アプリケーションなどのソースコードが公開されていない場合でも、SQL 文に対しては、実行計画や実行統計の情報が取得できる

## Statspack レポート

リスト名 : Statspack のレポート出力時に指定したレポート名 ( report\_name )

**SQL ordered by Gets** (バッファの読取り数が多い順)  
(Statspack レポートの時間範囲内での多い順)

Buffer Gets	Extentions	%Total	CPU Time(s)	Elapsed Time(s)	Old Hash Value
226,591	1,000	33.1	1.63	1.63	3103551479

バッファ読取数 実行回数 CPU 時間合計 待ち時間含む実行時間総計 old\_hash\_value  
全 SQL の Buffer Gets に対する割合

Module: SQL\*Plus  
 SELECT last\_name FROM employ\_table WHERE employ\_id = 125

上に表示されている統計情報と実行計画の対象となる SQL 文

多数の SQL 文の統計情報が並ぶ

(タイトル行は、先頭のみしか記述されない 見にくい  
 調査したい SQL 文と一致している記述を、探し出す

**SQL ordered by Reads** (ディスク I/O 数が多い順)  
(Statspack レポートの時間範囲内での多い順)

上記と同様にディスク I/O 数が多い順に、複数の SQL 文の統計情報が並ぶ

Physical Reads	Extentions	%Total	CPU Time	Elapsed Time	Old Hash Value
226,591	1,000	33.1	1.63	1.63	3103551479

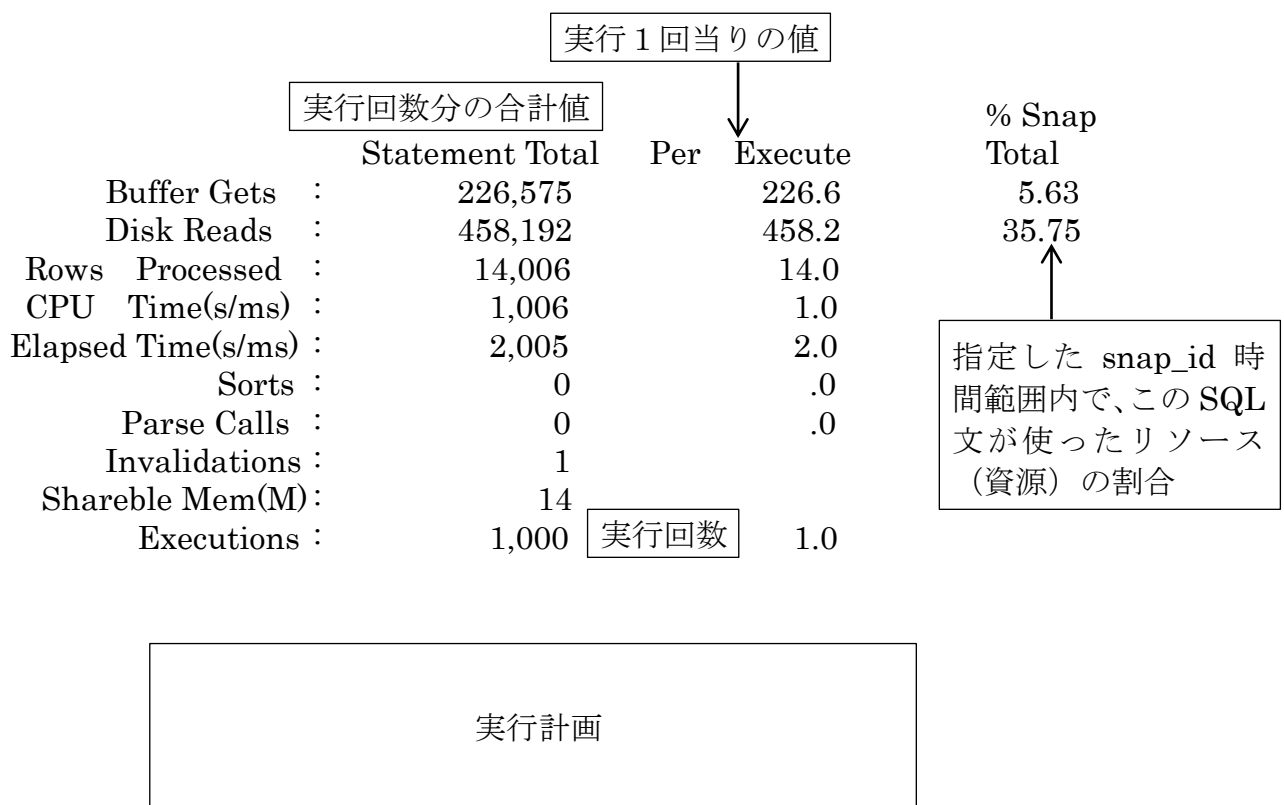
Module: SQL\*Plus  
 SELECT last\_name FROM employ\_table WHERE employ\_id = 125

## Statspack 詳細レポート

詳細な実行統計を表示するには、上記で調査した SQL 文の Old\_Hash\_Value 値を使って Statspack の詳細レポート (sprepsql) を起動させる  
 リスト名 : Statspack のレポート出力時に指定したレポート名 ( report\_name )  
 \_sqlrep.lst

P8. から出力した実行統計 :

~~なお、SQL ステートメントは、Statspack のレポート (spreport) 側にしか表示されていない~~



```
select department_name from employees e, departments d
where e.department_id = d.department_id
and salary > 1000
```

上に表示されている統計情報と実行計画の対象となる SQL 文