

---

## Statspack レポートの視点

---

### 注意点

- 他のレポートと件数を比較する場合には、**対象時間間隔**の比率で件数を乗除し評価する
- Statspack レポートは、対象の時間間隔が長いので、一時的なレスポンス悪化では、現象に伴う評価数値が対象時間で平均化され、発見が難しい
- 悪化時間帯（短時間）だけに限った一時的なレスポンス障害の調査のためには、悪化時間帯の `v$sysstat`、`v$session`、`v$session_wait` を取得して、状況評価を行う

### 評価チェックすべき内容

- **正常時**採取の時と**異常時**採取の時の Statspack レポートを比較して、大きく違う**数値**の変化を洗い出す
- Load Profile セクションから、全体の総処理量の正常時と異常時の値が**著しく異なる因子**を特定する
- 「Top 5 Times Events」セクションの待機イベントについて、**リソース不足に関連しているイベント**が上位で発生していないかを確認する
- 「Top 5 Times Events」セクションにて、数値が高い待機イベント、CPU 実行時間を把握して、その項目に合った評価基準別・上位 SQL 文セクションの上位 SQL 文の理由を考える
- チェックする点は、待機イベントの総時間の長い SQL について「実行回数が多くなった」、「平均実行時間が長くなった」の2つについて判断する
  - ◆ 実行回数が多くなった → リクエスト数が多くなった場合は、正当である
  - ◆ 平均実行時間が長くなった

→ 待機イベントの時間が長くなった  
→ CPU 実行時間が長くなった

長くなった原因の追究  
が必要である

- ・ 実行回数が少なく、経過時間 (Elapsed Time) の長い SQL 文の評価について
  - ◆ CPU 時間が長い → 正常時と同じ値ならば、仕方がない  
もしくは、実行計画を改め、効率化を行い実行時間の短縮を図る  
正常時より長くなっていた場合は、実行計画が正常時と異常時で異なっていないか確認を行い、対応するデータ件数の増加が影響している場合には、合致した対応を行う
  - ◆ 待機イベント時間 ( Elapsed Time - CPU Time ) が長い  
→ 待機イベントの発生原因を追究して対応を行う
- ・ DB CPU が高い状態での注意点  
「Top 5 Times Events」セクション、もしくは「Top10 Foreground by Total Wait Time」セクションにおいて、DB CPU (CPU 使用中) が高い場合でも、resmgr : cpu quantum イベント (リソースマネージャーによる CPU 割当て待ち)、null event イベント (CPU 割当て待ち) が発生している場合は、CPU の能力不足が発生していることになる

出力される統計情報は、出力に指定した時の snap\_id の時間範囲内で実行された回数と、その実行時に確保したリソースの統計値が出力される

## Statspack レポートの出力セクション

### 指定順・待機イベント情報出力

- Wait Events (全待機イベント)  
ordered by Total Wait Time desc, Waits desc (idle events last)
- Background Wait Events (バックグラウンド・待機イベント)  
ordered by Total Wait Time desc, Waits desc (idle events last)
- Wait Event Histogram (待機イベントの待機時間分布グラフ)  
Ordered by Event (idle events last)

### 評価基準別・上位 SQL 文稼働情報セクション (1 SQL 文の実行統計値情報)

- SQL ordered by CPU (CPU 実行時間)
- SQL ordered by Elapsed (CPU 実行時間+待機イベント時間)
- SQL ordered by Gets (バッファ取得数)
- SQL ordered by Reads (物理ブロック読み込み数)
- SQL ordered by Executions (実行回数)
- SQL ordered by Parse Calls (SQL 文解析回数)

### データベース活動の記録情報

- Instance Activity Stats (データベース全体に対する統計サマリー値)  
(DB ブロック書き込み数、SQL 実行数、Redo ログへの書き込み量、コミットなどのカウント回数 )
- Tablespace IO Stats (表領域別アクセス処理統計情報)
- File IO Stats (表領域物理ファイル別アクセス処理統計情報)
- File Read Histogram Stats (表領域物理ファイル別アクセス処理の時間分布)
- Buffer Pool Statistics
- PGA Aggr Target Stats
- Process Memory Summary Stats
- Enqueue activity
- Latch Activity (ラッチの活動記録)  
**リソース不足から来る待機イベントの発生状況**
- Latch Sleep breakdown
- Latch Miss Sources
- Segments by Logical Reads
- Dictionary Cache Stats
- Rule Sets
- Shared Pool Advisory
- SGA Target Advisory
- SGA Memory Summary
- SQL Memory Statistics
- init.ora Parameters (初期化パラメータ情報)

Oracle 全体での実行統計情報の出力部分

STATPACK report for

Statspack 取得時の調査ポイント番号

Snapshot	Snap Id	Snap Time	Session
Begin Snap	25	15-Sep-05 10:15:20	43
End Snap	26	15-Sep-05 10:17:40	51
Elapsed		2.33 (mins) → 2 : 20 → 140 (seconds)	

STATPACK レポートの対象となった開始と終了の時刻  
 調査対象期間 (単位 : 分)  
 他のレポートと件数を比較する場合には、対象時間の比率で件数を計算し調整する

パフォーマンス問題を考える場合は、調査対象時間全体と待ち時間の割合いで考える

この例では、12(sec) / 140 (sec) %の稼働率なので、ほとんど D/B に負荷が掛っていないことがわかる

Top 5 Times Events

Event	回数 Waits	Time (S)	Avg Wait(ms)	% Total Call Time
CPU Time	----	12	242	77.78
db file sequential read	247	2	520	10.71
log file parallel write	5	1	133	7.94
db file scattered read	31	0	2017	1.57
control file parallel write	49	0	2640	1.35
		<b>総合計</b>	<b>平均</b>	<b>割合</b>
CPU TIME : 実行処理		実行 or 待ち	実行 or 待ち	時間構成
それ以外 : 待ちイベント		時間	時間	の比率

[確認ポイント]

待機イベントの種類により、発生している原因が推測できる

【v\$session、v\$session\_wait の見方】の「多発する events の待機イベントの種類による原因の推測」

データベース活動処理におけるデータ処理量 ~~リソース使用量~~  
 システム全体の総量を考える時の評価に使う

||

Load Profile	1 秒当たりの回数	トランザクション 1 回当たりの数	
	Per Second	Per Transaction	
Redo size:	REDO ログ書込み	15,625.58	832.88
Logical reads:	バッファ読み込み	781.29	41.64
Block changes:	バッファ書込み	103.80	5.53
Physical reads:	Disk I/O 読み込み	15.18	0.81
Physical writes:	Disk I/O 書込み	5.30	0.28
User calls:		262.88	14.01
Parses:		95.87	5.11
Hard parses:		1.59	0.08
Sorts:		6.08	0.32
Logons:		0.03	0.00
Executes:		109.86	5.86
Transactions:		18.76	
%Blocks changed per Read:		13.29	Recursive Call %: 58.85
Rollback per transaction %:		0.00	Rows per Sort: 8.66

### Instance Efficiency Percentage

メモリ・アクセス効率の統計値

Buffer Nowait	% : 99.96	Redo NoWait	% : 100.00
バッファメモリ取得待ち無率		redo ログ・メモリ取得待ち無率	
Buffer Hit	% : 98.06	In-memory Sort	% : 100.00
データキャッシュ・ヒット率		メモリー内ソート率	
Library Hit	% : 95.86	Sort Parse	% : 98.34
ライブラリキャッシュ・ヒット率		ソート解析率	
Execute to Parse	% : 12.74	Latch Hit	% : 99.87
実行計画解析率			
Parse CPU to Parse Elapsed	% : 0.45	Non-Parse CPU	% : 82.19

[確認ポイント]

- ・ヒット率が目標基準値 (90%) より低くないか確認する  
 低い場合には、キャッシュメモリの割当て量を追加するなどの処置を行う

**Background Wait Events** DB/Inst: ORCL/orcl Snaps: 343-353

-> %Timeouts: value of 0 indicates value was < .5%. Value of null is truly 0

-> Only events with Total Wait Time (s) >= .001 are shown

-> ordered by **Total Wait Time** desc , **Waits** desc ( **idle events last** )

Event	② Waits	%Time-outs	① Total Wait Time (s)	Avg wait (ms)	Waits /txn
log file parallel write	33,726	0	56	2	0.8
control file parallel write	117	0	1	9	0.0
os thread startup	6	0	0	58	0.0
db file sequential read	35	0	0	7	0.0
control file sequential read	60	0	0	1	0.0
events in waitclass Other	320	0	0	0	0.0
.	.			.	
.	.			.	
latch: cache buffers chains	1	0	0	4	0.0
rdbms ipc message	33,910	3	2,346	69	0.8
pmon timer	105	100	302	2875	0.0
Streams AQ: qmn slave idle wait	③ 11	0	301	27346	0.0
Streams AQ: qmn coordinator idle	22	50	301	13673	0.0
Streams AQ: waiting for time mana	10	50	294	29395	0.0
smon timer	14	0	161	11527	0.0

—— 1 SQL 文に対する実行統計情報の出力部分 ——

SQL ordered by Elapsed

実行時間（待機+CPU）が多い SQL の実行統計

Elapsed Time (s)	Executions	Elap per Exec (s)	%Total	CPU Time (s)	Physical Reads	Old Hash Value
1092.07 (待機+CPU)	5,645 ↑ 実行回数	0.19 ↑ 1 回当たり時間	17.5 ↑ 全体に占める割合	1.98 CPU 時間		32 4082140391

insert into logon values(:1,:2)

1041.38	1,658	0.63	16.7	1.04	2,036	1468979989
---------	-------	------	------	------	-------	------------

```

SELECT /*+ use_nl */ o.order_id, line_item_id, product_id, unit_price, quantity,
order_mode, order_status, order_total, sales_rep_id, promotion_id, c.customer_id,
cust_first_name, cust_last_name, credit_limit, cust_email
FROM orders o , order_items oi, c
ustomers c WHERE o.order_id = oi.order_id and o.customer_id =
    
```

## SQL ordered by Gets

読み込みブロック数が多い SQL の実行統計

読取りバッファ数が多いのは、「実行回数が多い」「データ・レコードが多い」「実行計画が悪い」の3つの要因が考えられます

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	Old	hash_value 値
全 SQL の Buffer Gets に対する割合			実行時間総計		
Module: SQL*Plus					
SELECT last_name FROM employ_table WHERE employ_id = 125					
125,211	33	10.1	0.74	0.83	2597802345
→					
SELECT 文 2 . . . . .					

### [確認ポイント]

- Buffer Gets 数が多い SQL 文に対しては、実行計画の変更などで回数の減少が図れないか考慮する
- 待機時間 ( Elapsed Time - CPU Time ) が長い SQL 文に対しては、原因を調査して時間短縮が図れるようにする
- CPU Time が長い SQL 文に対しては、時間短縮を図るために実行計画を見直す

## SQL ordered by Reads

物理 Disk I/O が多い SQL の実行統計

Physical Reads	Extentions	%Total	CPU Time(s)	Elapsed Time(s)	Old Hash Value
150,113	1,000	45.1	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	Old	hash_value 値
全 SQL の Physical Reads に対する割合			実行時間総計		
Module: SQL*Plus					
SELECT last_name FROM employ_table WHERE employ_id = 125					
100,256	33	30.3	0.74	0.83	2597802345
→					
SELECT 文 2 . . . . .					

### [確認ポイント]

- Physical Reads 回数が多い SQL 文に対しては、実行計画の変更などで Reads 回数の減少が図れないか考慮する
- 待機時間 ( Elapsed Time - CPU Time ) が長い SQL 文に対しては、原因を調査して時間短縮が図れるようにする
- CPU Time が長い SQL 文に対しては、時間短縮を図るために実行計画を見直す



## Instance Activity Stats セクション

Oracle 全体でのパフォーマンス統計値が出力される

・ Order by statistics name

Statistics	total	per Second	per Trans
パフォーマンス統計 の出力項目名	合計実行回数	1 秒間当り実行回数	トランザクション 当り実行回数
・			
・			
・			
parse count (total)	1,974	1.18	0.00
parse time CPU	401	0.24	0.00
parse time elapsed	9,174	5.47	0.01
physical read I/O requests	72,603	43.30	0.05
physical read bytes	3,756,736,5 12	2,240,591.19	2,365.46



出力されるパフォーマンス統計項目

統計名	内容
CPU used by this session	Oracle インスタンスによって使用された CPU 時間
consistent gets	読取り一貫性ブロックの読取りが行われた回数
db block gets	UPDATE などの時の CURRENT ブロックの読取り要求回数
session logical reads	db block gets と session logical reads の合計値
execute count	SQL 文が実行された回数
logons current	現在ログオン（継続+新規）しているセッション数
parse count (hard)	ハードパースが実行された回数
parse count (total)	ソフトパースも含めたパースの実行された回数
parse time elapsed	パースに要した時間の合計
parse time cpu	パースに要した CPU 時間の合計
physical read bytes	アプリとユーザーが行った SQL 文による読込みバイト数 単位：byte
physical read total bytes	すべてのインスタンスアクティビティ全体での物理読込み量 単位：byte
physical read IO requests	物理読込み要求回数
physical write bytes	アプリとユーザーが行った SQL 文による書込みバイト数 単位：byte
physical write total bytes	すべてのインスタンスアクティビティによる物理書込み量 単位：byte

physical write IO requests	書込み要求回数
redo size	REDO 情報の合計生成量 単位 : byte
table scan (long tables)	大規模表への Full Scan 回数 バッファキャッシュを経由しない
table scan (short tables)	バッファキャッシュを経由した小さい表への Full Scan 回数
table scan (direct tables)	direct path read で読みを行った回数
undo change vector size	UNDO 情報の合計生成量 単位 : byte
user commits	コミットの実行回数
user rollbacks	ロールバックの実行回数

全体でのディスク・アクセスに対する実行統計情報の出力例

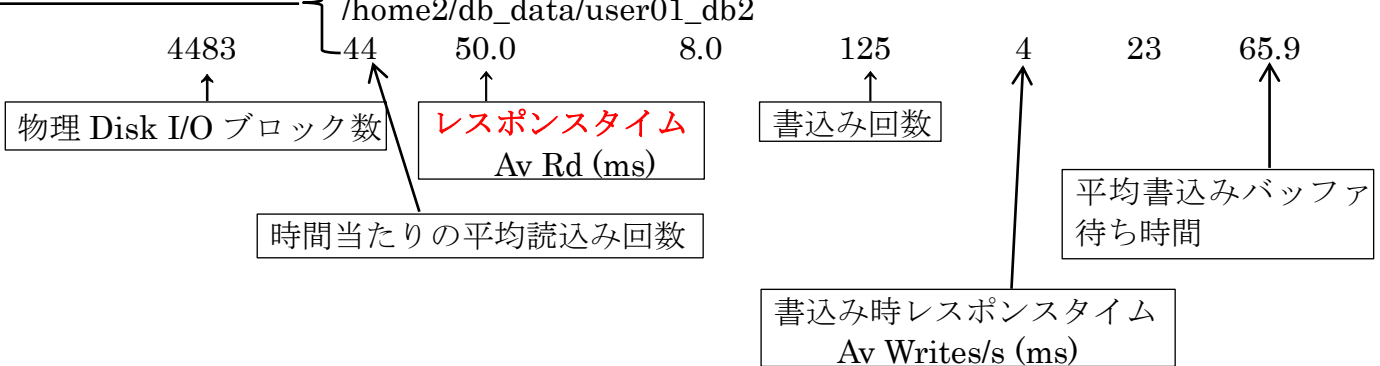
Tablespace

- ・ I/O アクセスが多いディスク・ボリュームが分かる
- ・ I/O アクセスが多い表領域名が分かる
- ・ 表領域の構成エクステンツのボリューム配置が分かる

Tablespace IO Stats DB/Inst: ORCL/orcl Snaps: 223-232  
 ->ordered by IOs (Reads + Writes) desc

Tablespace	File Name	AV Reads	AV Reads/s	Av Rd (ms)	Mx Rd Bkt	AV Blks/Rd	Writes	AV Writes/s	Buffer Waits	Av BufWt (ms)
------------	-----------	----------	------------	------------	-----------	------------	--------	-------------	--------------	---------------

User01	/home1/db_data/user01_db1	7771	77	50.5	##	7.9	74	5	10	54.1
	/home2/db_data/user01_db2	4483	44	50.0		8.0	125	4	23	65.9



[確認ポイント]

- ・ I/O 量が多く、**レスポンスタイムが悪い**表領域ファイルが、**ディスク能力不足**となっている可能性がある

→ Disk I/O に関する待機イベントが多発していないか確認の必要がある。

Statspack レポート (Top 5 Times Events セクション)

v\$session、v\$session\_wait

→ IOSTAT コマンドにて、ディスク稼働状況を調べる

「対応」ディスク負荷を下げるように、ファイル分散等を行う

- ・ I/O 量が少ないのに、**レスポンスタイムが悪い**表領域ファイルでは、**ディスク障害**が発生している可能性がある

「対応」障害しているディスクに対して、修理・交換を行う

→ 2つのディスクに分割してエクステンツ領域が確保されている

※ 各 SQL 文に対する実行計画や詳細な実行統計情報は、Statspack レポートには表示されない。

個別の SQL 文に対して実行計画情報や実行統計情報が必要な時は、別途に **Statspack 詳細レポート** を作成すれば情報が取得できる

なお、個別の SQL 文の実行統計情報を出力させるには、その SQL 文に対する OLD\_HASH\_VALUE 値を Statspack 詳細レポートに指示して出力する

Statspack 取得方法

Statspack レポートの作成

OLD\_HASH\_VALUE 値の求め方

Statspack 詳細レポートの作成

実行計画についての解説

実行統計情報についての解説