# SQL パフォーマンス・チューニング
## ～研修受講後のスキルアップサポート～

対応バージョン：Oracle 10gR1 ～ 12cR1

本資料は、アシスト Oracle 研修をご受講いただいたお客様からのご質問や、研修ではご案内できなかった情報などを FAQ にまとめたものです。研修受講後のスキルアップの一助として、是非お役立てください。
※ご利用上の注意事項は最後のページにまとめられております。ご確認のうえ、ご利用ください。

## 第 1 章　SQL パフォーマンスチューニングの基礎知識

1 Q. セグメント内での使用済みブロック数（HWM までのブロック数）の確認方法

A. 全表スキャンで読み込まれるブロック数は、HWM までのブロックです。
この HWM までのブロック（使用済みブロック）数は、
2 つの方法いずれかで確認することができます。

1. 統計情報の取得後、DBA_TABLES ビューの BLOCKS 列を参照。
2. DBMS_SPACE.UNUSED_SPACE プロシージャを使用する。

1 の場合はコストベースオプティマイザに影響を与え、
2 の場合は影響を与えません。

【2 の確認方法】

```
SQL> SET SERVEROUTPUT ON
DECLARE
        sum_blocks number;
        sum_bytes number;
        empty_blocks number;
        empty_bytes number;
        hwm_file_id number;
        hwm_extent_id number;
        hwm_extent_block number;
        blocks number;
BEGIN
 DBMS_SPACE.UNUSED_SPACE(
        segment_owner=>'SCOTT',
        segment_name=>'LOG_MASTER',
        segment_type=>'TABLE',
        total_blocks=>sum_blocks,
        total_bytes=>sum_bytes,
        unused_blocks=>empty_blocks,
        unused_bytes =>empty_bytes,
        last_used_extent_file_id=>hwm_file_id,
        last_used_extent_block_id=>hwm_extent_id,
        last_used_block=>hwm_extent_block
        );
 blocks:=sum_blocks-empty_blocks-1;
 DBMS_OUTPUT.PUT_LINE('使用ブロック数:'||blocks);
END;
/
使用ブロック数:18158

PL/SQL プロシージャが正常に完了しました。
```

…次ページへ続きます…


Copyright (C) K.K.Ashisuto All Rights Reserved.

| | | |
|---|---|---|
| | A. | …前ページからの続きです…<br><br>※DBMS_SPACE.UNUSED_SPACE の各パラメータの意味<br>segment_owner =オブジェクトの所有者<br>segment_name =オブジェクト名<br>segment_type =オブジェクトタイプ<br>total_blocks =セグメント合計ブロック数<br>total_bytes =セグメントサイズ(バイト)<br>unused_blocks =未使用ブロック数<br>unused_bytes =未使用サイズ(バイト)<br>last_used_extent_file_id =HWM を含むファイル ID<br>last_used_extent_block_id =HWM を含むエクステントの第 1 ブロック ID<br>last_used_block = HWM を含むエクステント内でのブロック番号 |


Copyright (C) K.K.Ashisuto All Rights Reserved.

| 第2章　SQLの診断ツール | | |
|---|---|---|
| 2 | Q. | SQLトレースのレポートで、diskの値がcurrent + queryの値を超えている場合があります。<br>なぜですか。 |
| | A. | diskの値がcurrent + queryの値を超える原因は、バッファ・キャッシュを経由せずに<br>データ・ファイルから直接読み込み(ダイレクト・パス読取り)が行われているためです。<br>例えば、ソートを必要とした処理で、このような結果になる場合があります。 |
| 3 | Q. | SQLトレースによってトレースファイルを出力すると、トレースファイルと同じファイル名で<br>拡張子が「.trm」のファイルが出力される場合があります。どのようなファイルなのでしょうか。 |
| | A. | 「.trm」拡張子のファイルは、トレース・マップ・ファイルと呼ばれるものです。<br>このファイルには、トレースファイルの構造情報が格納されており、内部的な検索で<br>使用されます。 |
| 4 | Q. | あるユーザーがSQLトレースを実行できるように、DBMS_SESSION.SET_SQL_TRACEプロシージャの<br>実行権限（EXECUTE権限）を付与しました。<br>しかし、そのユーザーがSET_SQL_TRACEプロシージャを実行すると「権限が不足しています」との<br>エラーメッセージが返りました。なぜでしょうか。 |
| | A. | DBMS_SESSION.SET_SQL_TRACEプロシージャを実行するには、プロシージャの<br>EXECUTE権限ではなく、「ALTER SESSIONシステム権限」が必要です。 |
| 5 | Q. | 実行したSQLがソフト解析だったかハード解析だったかをSQLトレースで確認できますか。 |
| | A. | TKPROFで出力した結果の「Misses in library cache during parse: 数値」で確認できます。<br>数値が0の場合はライブラリ・キャッシュ・ミス（ハード解析）がなかった、つまりソフト解析で<br>あったことがわかります。<br><br>※フォーマット時に「aggregate = no」と指定した場合は、同一SQLであっても個別に結果が<br>　出力されます（デフォルトは「aggregate = yes」です）。 |
| 6 | Q. | TKPROFやAUTOTRACEで表示される「current」は何を意味していますか。 |
| | A. | アクセスしたブロックが現行モード（DMLやSELECT FOR UPDATEでアクセス中）の場合、currentにそのブロック数が表示されます。 |


Copyright (C) K.K.Ashisuto All Rights Reserved.

| 第３章　索引の作成と管理 | | |
|---|---|---|
| 7 | Q. | 索引をオンライン作成している時に通常作成時よりも時間がかかる場合があるのはなぜでしょうか。 |
| | A. | ONLINE 句を使用して索引を作成した場合、索引作成中に発行された DML 文も索引に反映させるため、<br>DML 文を発行したトランザクションが終了するまで索引の作成は終了しません。<br>そのため、通常よりも作成時間がかかってしまうことがあります。 |
| 8 | Q. | 制約によって作成された索引を REBUILD できますか。 |
| | A. | 可能です。ALTER INDEX ～ REBUILD を実行してください。 |
| 9 | Q. | コンポジット索引に指定できる列数の上限 |
| | A. | コンポジット索引には最大 32 列まで指定できます。 |
| 10 | Q. | リーフ・ブロックの断片化解消方法（COALESCE 以外） |
| | A. | 断片化したリーフ・ブロックを結合する方法として、COALESCE 以外にセグメント縮小機能<br>(SHRINK)を使用できます。<br><br>本機能は、通常表に対して行う操作ですが、CASCADE オプションを指定することで、<br>表に定義されている索引に対しても同時に縮小処理を行います。<br><br>① ALTER TABLE 表名 ENABLE ROW MOVEMENT;　/*行の移動を許可*/<br>② ALTER TABLE 表名 SHRINK SPACE CASCADE;　/*CASCADE オプションで関連する索引も縮小*/<br>~~~~~~~<br><br>※セグメント縮小機能では、索引の高さは変わりません。 |
| 11 | Q. | DBA_INDEXES ビューの BLEVEL 列と INDEX_STATS ビューの HEIGHT 列の値が違うことについて |
| | A. | 階層の深さ(ツリーの高さ)は DBA_INDEXES ビューの BLEVEL 列や<br>INDEX_STATS ビューの HEIGHT 列で確認することができます。<br><br>これらの列は「高さ(height)」「blevel」などと呼ばれ、<br>「高さ(height)」は root ブロックから leaf ブロックまでに至るまでに通る<br>root,branch,leaf ブロックの数を示します。<br><br>「blevel」は、root ブロックから leaf ブロックに至るまでに通る root、branch ブロックの数<br>を示すので、「高さ(height) － 1」 になります。 |
| 12 | Q. | TRUNCATE 文で表を切り捨てた場合も自動メンテナンスが行われますか。 |
| | A. | TRUNCATE 文で表を切り捨てると、表に定義されている索引のデータも全て自動的に削除されます。 |


Copyright (C) K.K.Ashisuto All Rights Reserved.

| 第4章　効率的なSQLの記述方法 | | |
|---|---|---|
| 13 | Q. | SQLを変更できない環境でのチューニング方法 |
| | A. | アプリケーションがハードコーディングされている等で、SQLが修正できない場合、<br>以下のようなチューニングを検討します。<br><br>・索引の追加/削除<br>・初期化パラメータの変更(CBOに影響を与えるパラメータに対して)<br>・ヒント文を含めたビューを作成<br>・パラレル実行<br>(ALTER SESSION FORCE PARALLEL文、または表のPARALLELオプションで設定)<br>・プランスタビリティ(Oracle 10gまで)<br>・SQL実行計画管理(Oracle 11gから)<br>・不可視化索引の使用(Oracle 11gから) |
| 14 | Q. | ビューを定義する際のSELECT文にヒントを記述することはできますか。 |
| | A. | 可能です。ビューを問い合わせると、そのビューに定義されたSQLが実行され、<br>ヒントが有効になります。 |
| 15 | Q. | ヒントの記述を一部間違えると、全てのヒントが無効になるのでしょうか。 |
| | A. | ヒントの記述方法が間違っている場合、そのヒントだけが無効になります。<br>ヒントが複数指定してあった場合は適切に記述されているヒントは有効です。 |
| 16 | Q. | 索引フル・スキャンと索引スキップ・スキャンの動作の違いを教えてください。 |
| | A. | 索引フル・スキャンは、すべてのリーフ・ブロックを読込みます。索引スキップ・スキャンは、<br>すべてのリーフ・ブロックではなく、条件値に合致するエントリを含んだリーフ・ブロックを<br>読込みます。 |
| 17 | Q. | INDEX FAST FULL SCANが選択される前提条件 |
| | A. | オプティマイザが「INDEX FAST FULL SCAN」を選択するには、以下の条件のいずれかを<br>満たしている必要があります。下記条件を満たしているか、確認してください。<br><br>1.SELECT文で使用するすべての列が索引に含まれている。<br>2.ヒントINDEX_FFSを使用している。<br>3.NOT NULL制約が存在する。もしくはNULLを除く処理を行っている。 |


Copyright (C) K.K.Ashisuto All Rights Reserved.

| | | |
|---|---|---|
| 18 | Q. | 最初の解析時（ハードパース時）にバインド変数に割り当てられた値を確認する方法 |
| | A. | Oracle 10gまでの環境でバインド変数を使用している場合、はじめに割り当てられた値をもとに実行計画を決定します(bind peek機能)。<br><br>はじめに割り当てられた値を確認するには、DBMS_XPLAN.DISPLAY_CURSORのformatパラメータに「ADVANCED」もしくは「PEEKED_BINDS」を指定します。<br>※上記を実行すると実行計画が表示され、その下にある「Peeked Binds」セクションにバインドした値が表示されます。<br><br>■実行例 |

```
SQL> SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY_CURSOR(NULL,NULL,'PEEKED_BINDS'));

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------------------
SQL_ID  92y8n9h5vygp0, child number 0
-------------------------------------
SELECT * FROM customer.emp WHERE empno = :bind1

Plan hash value: 4025414177

--------------------------------------------------------------------------
| Id  | Operation         | Name | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |      |       |       |     7 (100)|          |
|*  1 |  TABLE ACCESS FULL| EMP  |     1 |    38 |     7   (0)| 00:00:01 |
--------------------------------------------------------------------------

Peeked Binds (identified by position):  /* このセクションにバインドした値が表示されます*/
--------------------------------------

   1 - :BIND1 (NUMBER): 7900              /* 値7900がバインドされたことが分かります */
```

※テキスト2-13のDBMS_MONITOR.SESSION_TRACE_ENABLEプロシージャでも確認できます。


Copyright (C) K.K.Ashisuto All Rights Reserved.

| 第6章　コストベース・オプティマイザと実行計画の決定 | | |
|---|---|---|
| 19 | Q. | オプティマイザ統計を収集するDBMS_STATSパッケージとANALYZEコマンドの違い |
| | A. | オプティマイザ統計はDBMS_STATSパッケージだけでなく、ANALYZEコマンドでも収集できます。<br>この2つは以下のような違いがあります。<br><br>・収集情報<br>DBMS_STATSは、コストベース・オプティマイザが必要とする情報のみを取得します。<br>それに対して、ANALYZEは行連鎖・行移行の数、未使用ブロックの数などの追加情報も<br>収集します。<br><br>・DBMS_STATSでは、統計情報が古くなったものだけを再取得し直すことができます。<br>※ANALYZEはできません。<br><br>・統計情報の保存<br>DBMS_STATSで収集した統計は、31日間保存されます(31日以内であれば以前の統計に<br>リストアできます)。ANALYZEでは、以前の統計は保存されないため、リストアはできません。<br><br>上記の特徴から、コストベース・オプティマイザのための統計取得が目的であれば、<br>DBMS_STATSパッケージを使用することをお勧めします。 |
| 20 | Q. | 特定の表や索引の統計情報が収集されているか（いつ収集されたか）を確認する方法 |
| | A. | DBA_TABLES、DBA_INDEXESビューのLAST_ANALYZED列には、前回統計収集を行った際の<br>日時が記録されています。この列に値がなければ、統計情報は収集されていません。<br><br>下記例では、EMP表の統計収集日時を確認しています。<br><br>SQL> SELECT table_name,last_analyzed FROM dba_tables<br>  2  WHERE  table_name = 'EMP';<br><br>TABLE_NAME  LAST_ANA<br>----------  --------<br>EMP  14-12-05 |
| 21 | Q. | もともとコストベース・オプティマイザを使用しているデータベースをアップグレードしたら、<br>実行計画が変わったSQLがありました。なぜでしょうか。 |
| | A. | コストベース・オプティマイザのコスト計算アルゴリズムは、バージョン毎に異なります。<br>そのため、変更を全く加えていなくてもOracleをアップグレードすると実行計画が変わる<br>可能性があります。 |
| 22 | Q. | 自動統計収集を無効化し（DBMS_AUTO_TASK_ADMIN.DISABLEを実行）、手動での統計収集を<br>一切行わなければ、オプティマイザ統計情報は変化しないと考えていいでしょうか。 |
| | A. | 索引のREBUILDのみ、内部的にオプティマイザ統計が再取得されます。<br>それ以外のケースで統計情報が変化することはありません。 |


Copyright (C) K.K.Ashisuto All Rights Reserved.

| | | |
|---|---|---|
| 23 | Q. | オプティマイザ統計を任意の値に変更してテストを行う方法 |
| | A. | DBMS_STATS.SET_TABLE_STATS プロシージャで、統計を任意の値に設定し、<br>それによる効果をシミュレーションできます。 |

```
/* EMP1 表の現在の統計を確認 */
SQL> SELECT table_name, num_rows,blocks, avg_row_len, user_stats
  2  FROM   user_tables
  3  WHERE  table_name='EMP1';

TABLE_NAME              NUM_ROWS     BLOCKS AVG_ROW_LEN USER_STAT
-------------------- ---------- ---------- ----------- ---------
EMP1                       1500         16          38 NO


/* 任意の統計値に設定 */
SQL> EXECUTE SYS.DBMS_STATS.SET_TABLE_STATS(-
>    ownname => 'CUSTOMER',-
>    tabname => 'EMP1',-
>    numrows => 1000,-        ←NUM_ROWS(行数)
>    numblks => 10,-          ←BLOCKS(HWM までのブロック数)
>    avgrlen => 30);          ←AVG_ROW_LEN(平均行サイズ)

※上記パラメータ以外に以下も設定できます。
  ・cachedblk:キャッシュ済みオブジェクトの平均ブロック数
  ・cachehit :オブジェクトに対する平均キャッシュ・ヒット率

/* 変更後の統計値を確認 */
SQL> SELECT table_name, num_rows,blocks, avg_row_len, user_stats
  2  FROM   user_tables
  3  WHERE  table_name='EMP1';

TABLE_NAME              NUM_ROWS     BLOCKS AVG_ROW_LEN USER_STAT
-------------------- ---------- ---------- ----------- ---------
EMP1                       1000         10          30 YES
                                                       ~~~

※USER_STATS 列で、ユーザーが任意に設定した統計かどうかが分かります。
※変更した統計は、統計を再取得するまで有効です。
```


Copyright (C) K.K.Ashisuto All Rights Reserved.

| 付録 | | |
|---|---|---|
| 24 | Q. | ANALYZE INDEX VALIDATE STRUCTURE 実行後、他のセッションで INDEX_STATS を参照しても結果が表示されません。なぜでしょうか。 |
| | A. | INDEX_STATS の情報はセッション間でのみ有効です。ANALYZE INDEX VALIDATE STRUCTURE を実行したセッションとは異なるセッションからは、分析結果は参照できません。 |

※ ご利用上の注意事項※
・本書の著作権は株式会社アシストに帰属します。
・本書は参考資料であり、掲載されている情報は予告なしに変更されることがあります。
・本書で使用している製品の名称は、各社の商標または登録商標です。
・本資料の内容に関するご質問はご遠慮ください。
・本資料はお客様の責任のもとでご利用ください。これらの使用によりいかなる損害が生じたとしても、株式会社アシストは一切保証致しかねますので、ご了承ください。


Copyright (C) K.K.Ashisuto All Rights Reserved.