

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。

| 第 1 章 動的パフォーマンス・ビュー | | |
|-----------------------|----|--|
| 1 | Q. | マルチブロック読み取りが行われる際には、データベース・バッファ・キャッシュに連続した空き領域が必要になるのでしょうか。 |
| | A. | マルチブロック読み取りでは、DB_FILE_MULTIBLOCK_READ_COUNT パラメータで設定されたブロック数がまとめて読み取られます。しかし、読み取り先となるメモリー領域がデータベース・バッファ・キャッシュ上で連続している必要はありません。 |
| 第 2 章 パフォーマンス情報の収集ツール | | |
| 2 | Q. | 解析をせずに SQL が実行されることがありますか。 (Instance Efficiency Indicators セクションの Execute to Parse について) |
| | A. | 同一セッションで 3 回以上実行された SQL は、解析処理が回避されます。 このような SQL に対しては、同一セッションでのみ使用可能な解析結果へのリンク (セッション・カーソル) が PGA 内に保持され、その後の同一 SQL 処理で解析処理が回避されます (ソフト解析も行われません)。 ※詳細は付録「カーソルに関する初期化パラメータ」の項目にある「SESSION_CACHED_CURSORS パラメータ」の項目を参照してください。 |
| 3 | Q. | SQL ordered by Parse Calls セクションでは、ソフト解析とハード解析のどちらのコール数が多い SQL の情報が出力されるのでしょうか。 |
| | A. | 「ソフト解析」のコール数が多い SQL の情報が出力されます。 |
| 4 | Q. | SQL ordered by Version Count セクションでは、どのような SQL の情報が出力されるのでしょうか。 |
| | A. | 解析結果のバージョン数が多く存在する SQL の情報が出力されます。 解析結果は、共有プール内ではソースコードなどの概要情報を持つ「親情報」と、参照オブジェクト情報や実行計画などの詳細情報を持つ「子情報」にわけて保存されています。解析結果のバージョンとは、子情報の数を表しています。例えば、以下のような場合に解析結果のバージョン数が増えることとなります。 ・ 同じソースコードで異なるオブジェクトにアクセスする場合 ※SCOTT ユーザーと ALLEN ユーザーが、それぞれ自身の作成した EMP 表にアクセスする など ・ 異なる初期化パラメータ値が設定されている複数のセッションで、同じソースコードの SQL が実行された場合 バージョンが異なると、都度再解析が行われていることを示すため、バージョン数の多い SQL は共有プールに負荷をかけている可能性が考えられます。 |

| | | |
|-----------------------------|----|--|
| 5 | Q. | セッション・レベルのスナップショットを取得することは可能ですか。 |
| | A. | <p>i_session_id パラメータを指定して STATSPACK.SNAP パッケージを実行することで、セッション・レベルのスナップショットを取得できます。</p> <p>■実行例</p> <p>1. セッション ID を調べます。</p> <pre>SQL> SELECT sid,username FROM v\$session 2 WHERE username IS NOT NULL;</pre> <pre>SID USERNAME ----- 9 SYS 10 SCOTT</pre> <p>2. SCOTT ユーザーのセッション統計を取得します。</p> <pre>SQL> EXECUTE statspack.snap(i_session_id=>10)</pre> <p>PL/SQL プロシージャが正常に完了しました。</p> |
| 6 | Q. | PERFSTAT ユーザーの使用する表領域の指定に制限はありますか。 |
| | A. | PERFSTAT ユーザーのデフォルト表領域と一時表領域には、SYSTEM 表領域を指定することはできません（エラーが発生しインストールに失敗します）。 |
| 7 | Q. | AWR が使用しているディスク領域サイズの確認方法 |
| | A. | <p>AWR は SYSAUX 表領域に格納されます。AWR が使用しているディスク領域は以下の SQL 文で確認できます。</p> <pre>SQL> SELECT space_usage_kbytes 2 FROM v\$sysaux_occupants 3 WHERE occupant_name = 'SM/AWR';</pre> <pre>SPACE_USAGE_KBYTES ----- 35712</pre> |
| 8 | Q. | AWR スナップショットの情報をビューで確認することはできますか。 |
| | A. | <p>DBA_HIST_ で始まるビューを使用すると、AWR スナップショットの情報を確認できます。例えば以下のようなビューがあります。</p> <ul style="list-style-type: none"> ・ DBA_HIST_SNAPSHOT ビュー：管理しているスナップショットの ID や取得期間を確認できます。 ・ DBA_HIST_BASELINE ビュー：システムで取得されたベースラインに関する情報を確認できます。 <p>※詳細は「Database データベース・リファレンス」マニュアルにてご確認ください。</p> |
| 第 3 章 代表的なチューニングポイント | | |
| 9 | Q. | ストアド・プロシージャを使用すると、SQL の解析負荷を低減させられますか。 |
| | A. | <p>ストアド・プロシージャを使用することで、SQL の解析負荷を低減させることができます。</p> <p>これは、ストアド・プロシージャは、解析済みの状態でデータベースに格納されるためです。また、複数のユーザーが同じストアド・プロシージャを実行する場合、SQL と同様に解析結果を共有することができます。</p> <p>※無名の PL/SQL ブロックは、実行するたびに解析が行なわれるため、CPU 負荷の高騰を招く可能性があります。そのため、大規模な PL/SQL ブロックはストアド・サブプログラムとしてデータベースに格納しておくことが推奨されます。</p> |

| | | |
|----|----|--|
| 10 | Q. | 無名の PL/SQL ブロックが実行されているかどうかは、どのように確認すればよいですか。 |
| | A. | <p>現行インスタンスで実行されたことがある無名の PL/SQL ブロックは、V\$SQLAREA ビューで特定できます。</p> <p>■実行例</p> <pre>SQL> SELECT SUBSTRB(sql_text, 1, 80) 2 FROM v\$sqlarea 3 WHERE command_type = 47 4 AND LENGTH(sql_text) > 100 ←サイズを指定して下さい。</pre> |
| 11 | Q. | SQL が解析された後で、オブジェクトの定義が変更されても、解析結果は再利用されるのでしょうか。 |
| | A. | SQL で参照されているオブジェクトの定義が変更される（例えば、列が削除されるなど）と、関連する SQL の解析結果は無効化され、次回実行時に再解析が行われます。 |
| 12 | Q. | ライブラリ・キャッシュ上にある特定のストアド・プロシージャのみを排除することはできますか。 |
| | A. | DBMS_SHARED_POOL パッケージの PURGE プロシージャで可能です。 |
| 13 | Q. | スキーマ名をつけている場合と、つけていない場合でも SQL の解析結果は共有されないのでしょうか。 |
| | A. | アクセスするオブジェクトが同一であったとしても、記述が同一ではない SQL となるため、解析結果は共有されません。 |
| 14 | Q. | 共有プールをフラッシュした場合、DBMS_SHARED_POOL パッケージで固定したオブジェクトも消去されますか。 |
| | A. | <p>消去されません。</p> <p>共有プールをフラッシュした場合、現在実行中のものや、固定しているオブジェクトは保持されます。</p> |
| 15 | Q. | ライブラリ・キャッシュ・ヒット率（Instance Efficiency Indicators の Library Hit）が 100%を超えてしまうのですが、なぜでしょうか。 |
| | A. | <p>ヒット率情報のもととなる V\$LIBRARYCACHE ビューの統計情報の更新時にロックが取得されていないことが原因です。</p> <p>V\$LIBRARYCAHCE ビューの PINHITRATIO 列の値は、「解析結果を見つけた回数（PINHITS 列）」を「解析結果を探した回数（PINS 列）」で割って求めています。PINS 列、PINHITS 列ともロックを掛けずに統計を更新しているため、多くのセッションで同じ SQL を同時実行すると、統計情報に不整合が発生し、ライブラリ・キャッシュ・ヒット率が 100%を超えることがあります。</p> <p>ただし、誤差の範囲は 1%未満となるため、チューニングを行うにあたっては気にする必要はありません。</p> |
| 16 | Q. | データベース・バッファ・キャッシュをフラッシュすることはできますか。 |
| | A. | <p>共有プールのフラッシュと同様にデータベース・バッファ・キャッシュのフラッシュも可能です。</p> <p>以下のコマンドで行えます。</p> <pre>SQL> ALTER SYSTEM FLUSH BUFFER_CACHE;</pre> <p>※性能検証などで何度もメモリーを空にする場合に有効です。</p> |
| 17 | Q. | UNDO 表領域でブロック競合が問題となっている場合は、どのように対処すればよいでしょうか。 |
| | A. | <p>UNDO 表領域の拡張を検査します。UNDO 表領域ではトランザクション数に応じて UNDO セグメントのサイズや数が自動調整されます。そのため、UNDO セグメントでブロック競合が問題となっている場合は、UNDO 表領域のサイズが不足していることが想定されます。</p> <p>※詳細は「データベース・マネジメント」研修でご紹介しています。</p> |
| 18 | Q. | log file switch completion は、どのようなときに発生する待機イベントでしょうか。 |
| | A. | <p>log file switch completion はログ・スイッチの完了を待機する際に発生する待機イベントです。</p> <p>log file switch(checkpoint incomplete)や log file switch(archiving needed)待機イベントなどに該当しない「その他」としてカウントされます。そのため、明確な待機発生原因を特定することはできませんが、一般的にはオンライン REDO ログ・ファイルへの I/O 負荷が高い場合に発生します。</p> |

第4章 自動メモリー管理機能

| | | |
|----|----|---|
| 19 | Q. | プロセスごとに PGA の使用量を調査することはできますか。 |
| | A. | <p>V\$PROCESS ビューの次の列を参照することで、プロセスごとに PGA の使用量を調査できます。</p> <ul style="list-style-type: none"> ・ pga_used_mem : プロセスが現在使用している PGA メモリー使用量 ・ pga_alloc_mem : プロセスに現在割当てられている PGA メモリー使用量 ・ pga_max_mem : 今までにプロセスへ割当てられた最大の PGA メモリー使用量 |
| 20 | Q. | LOG_BUFFER パラメータの設定値と、STARTUP 時に表示される REDO ログ・バッファのサイズが異なるのはなぜでしょうか。 |
| | A. | <p>REDO ログ・バッファのサイズは、決まったサイズで割り切れないとサイズが切り上げられるためです。</p> <p>Oracle ではメモリー領域を特定の単位（グラニユル）で割り当てます。メモリー領域のサイズがグラニユル単位で割り切れない場合は、グラニユルの倍数のサイズまで切り上げて割り当てられるため、初期化パラメータで指定したサイズと実際のサイズが異なることがあります。</p> <p>※厳密には、REDO ログ・バッファのサイズと Fixed Size との合計値がグラニユルの倍数になるように切り上げられます。</p> <p>※グラニユルのサイズはバージョンやプラットフォーム、SGA_MAX_SIZE の設定によって異なります。</p> |

付録

| 21 | Q. | NOLOGGING モードで表を作成したにも関わらず、LOGGING モードになってしまうのは、なぜでしょうか。 | | | | | | | | |
|------------|---------|---|------------|---------|--|-------|-----|--|-------|----|
| | A. | <p>NOLOGGING モードの表を作成するには、副問い合わせの前に「NOLOGGING」を指定する必要があります。</p> <p>■実行例</p> <pre>/*副問い合わせの後に NOLOGGING を指定*/ SQL> CREATE TABLE dept2 2 AS SELECT * FROM dept 3 NOLOGGING;</pre> <p>表が作成されました。</p> <pre>/*副問い合わせの前に NOLOGGING の指定*/ SQL> CREATE TABLE dept3 NOLOGGING 2 AS SELECT * FROM dept;</pre> <p>表が作成されました。</p> <pre>SQL> SELECT table_name, logging FROM user_tables 2 WHERE table_name IN ('DEPT2', 'DEPT3');</pre> <table border="1"> <thead> <tr> <th>TABLE_NAME</th> <th>LOGGING</th> <th></th> </tr> </thead> <tbody> <tr> <td>DEPT2</td> <td>YES</td> <td>← 副問合せの後に NOLOGGING を指定すると、NOLOGGING は無効 (LOGGING モード)</td> </tr> <tr> <td>DEPT3</td> <td>NO</td> <td>← 副問合せの前に NOLOGGING を指定すると、NOLOGGING は有効 (NOLOGGING モード)</td> </tr> </tbody> </table> | TABLE_NAME | LOGGING | | DEPT2 | YES | ← 副問合せの後に NOLOGGING を指定すると、NOLOGGING は無効 (LOGGING モード) | DEPT3 | NO |
| TABLE_NAME | LOGGING | | | | | | | | | |
| DEPT2 | YES | ← 副問合せの後に NOLOGGING を指定すると、NOLOGGING は無効 (LOGGING モード) | | | | | | | | |
| DEPT3 | NO | ← 副問合せの前に NOLOGGING を指定すると、NOLOGGING は有効 (NOLOGGING モード) | | | | | | | | |

| 22 | Q. | インスタンスの起動に合わせて、共有プールへのオブジェクトの固定を自動化するためにはどのように対処すればよいでしょうか。 | | | | | | | | | |
|-------|------------|--|-------|------|------|-------|------------|-----|-------|-----------|-----|
| | A. | <p>インスタンス起動時に実行されるトリガーを使用することで、インスタンスの起動に合わせて共有プールへのオブジェクトの固定を自動化できます。</p> <p>■実行例</p> <ol style="list-style-type: none"> DBMS_SHARED_POOL パッケージで共有プールに固定するプロシージャのリストを記載した「keep_object_list」プロシージャを作成します。 <pre>SQL> CREATE OR REPLACE PROCEDURE keep_object_list 2 IS 3 BEGIN 4 DBMS_SHARED_POOL.KEEP(' SCOTT.TEST_PROC'); ← 今回の2つのプロシージャ 5 DBMS_SHARED_POOL.KEEP(' SCOTT.TEST2_PROC'); ← を固定する。 6 END; 7 /</pre> <p>プロシージャが作成されました。</p> <ol style="list-style-type: none"> 「keep_object_list」プロシージャをデータベース起動時に実行されるように動作させる「keep_object」トリガーを作成します。 <pre>SQL> CREATE OR REPLACE TRIGGER keep_object 2 AFTER STARTUP ON DATABASE 3 BEGIN 4 keep_object_list; 5 END; 6 /</pre> <p>トリガーが作成されました。</p> <ol style="list-style-type: none"> データベースを再起動します。 トリガーによって「keep_object_list」プロシージャが実行され、指定したプロシージャが共有プールに固定されていることを確認します。 <pre>SQL> SELECT owner, name, kept 2 FROM v\$db_object_cache 3 WHERE name like 'TEST%PROC';</pre> <table border="1"> <thead> <tr> <th>OWNER</th> <th>NAME</th> <th>KEPT</th> </tr> </thead> <tbody> <tr> <td>SCOTT</td> <td>TEST2_PROC</td> <td>YES</td> </tr> <tr> <td>SCOTT</td> <td>TEST_PROC</td> <td>YES</td> </tr> </tbody> </table> | OWNER | NAME | KEPT | SCOTT | TEST2_PROC | YES | SCOTT | TEST_PROC | YES |
| OWNER | NAME | KEPT | | | | | | | | | |
| SCOTT | TEST2_PROC | YES | | | | | | | | | |
| SCOTT | TEST_PROC | YES | | | | | | | | | |

※ 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。