

はじめに

■コース概要と目的

本コースでは、Oracle の内部構造や各処理における動作など、Oracle データベースの運用管理や開発に必要な内容を解説します。

Oracle データベースに関わる方にとって、Oracle アーキテクチャの理解は非常に重要です。

特に運用管理や開発を担当される方は、非常に多機能な Oracle データベースを使いこなすために、アーキテクチャの深い理解が非常に重要です。例えば、何かのトラブルが発生した場合でも、発生状況を鑑みた上で多くの機能の中からどれを選択すれば良いのか、論理的に検討した上で正しい判断ができるようになります。

また、直接 Oracle データベースに触れることがない方でも、用語や構造の理解によってサポートセンターと円滑なコミュニケーションが図れるなどの利点があります。

■受講対象者





データベース管理者、Oracle を使用したアプリケーション開発を行う方。

■前提条件

リレーショナル・データベースに関する基本的な知識を有する方、もしくは「Oracle 概要」コースを受講された方。

■テキスト内の記述について

▼マーク

	指定バージョンからの新機能 (左記の場合、Oracle 12cR1 からの新機能)
	Enterprise Edition で使用できる機能
	知っておいた方が良いテクニック、もしくは注意事項
	参照ページ

第2章

SQL 処理と Oracle インスタンス

この章では、SQL の処理ステップを追いながら、Oracle インスタンスの役割について説明します。

1. SQL 処理と Oracle インスタンス (概要)
2. データベースへの接続
3. SQL の解析と共有プール
4. SQL の実行とデータベース・バッファ・キャッシュ
5. データの変更処理

1. SQL 処理と Oracle インスタンス (概要)

2章では SQL の処理ステップを追いながら、Oracle インスタンスがどのように使用されるかを解説します。
クライアントから発行された SQL は以下の流れで処理されます。

(1) データベースへの接続(ユーザー・プロセスとサーバー・プロセス)

クライアント側のアプリケーション・プログラム (SQL*Plus など) をユーザー・プロセス (クライアント・プロセス) といいます。ユーザー・プロセスはサーバー・プロセスと接続を確立し、SQL を渡します。
サーバー・プロセスは受け取った SQL を処理し、処理結果をユーザー・プロセスに返します。

(2) SQL の解析と共有プール

SQL を受け取ったサーバー・プロセスは、まずその SQL を解析 (PARSE) します。解析とは、SQL の構文にミスがないか、指定された表や列が存在するかなどのチェックのことです。
解析後、その結果が SGA 内の共有プールに保持されます。これにより、その後同じ SQL が発行された際、保持されている解析結果を再利用し、解析の負荷を軽減できます。

(3) SQL の実行とデータベース・バッファ・キャッシュ

解析後、サーバー・プロセスが SQL 処理に必要なデータをデータ・ファイルからデータベース・バッファ・キャッシュに読み込み、実行 (EXECUTE) します。

また、データベース・バッファ・キャッシュに読み込まれたデータは、SQL 実行後もメモリー上に残されたままになります。これにより、同じデータが別の SQL で必要になった際に再度データファイルから読み込む必要がなく、ディスク I/O の負荷を削減できます。

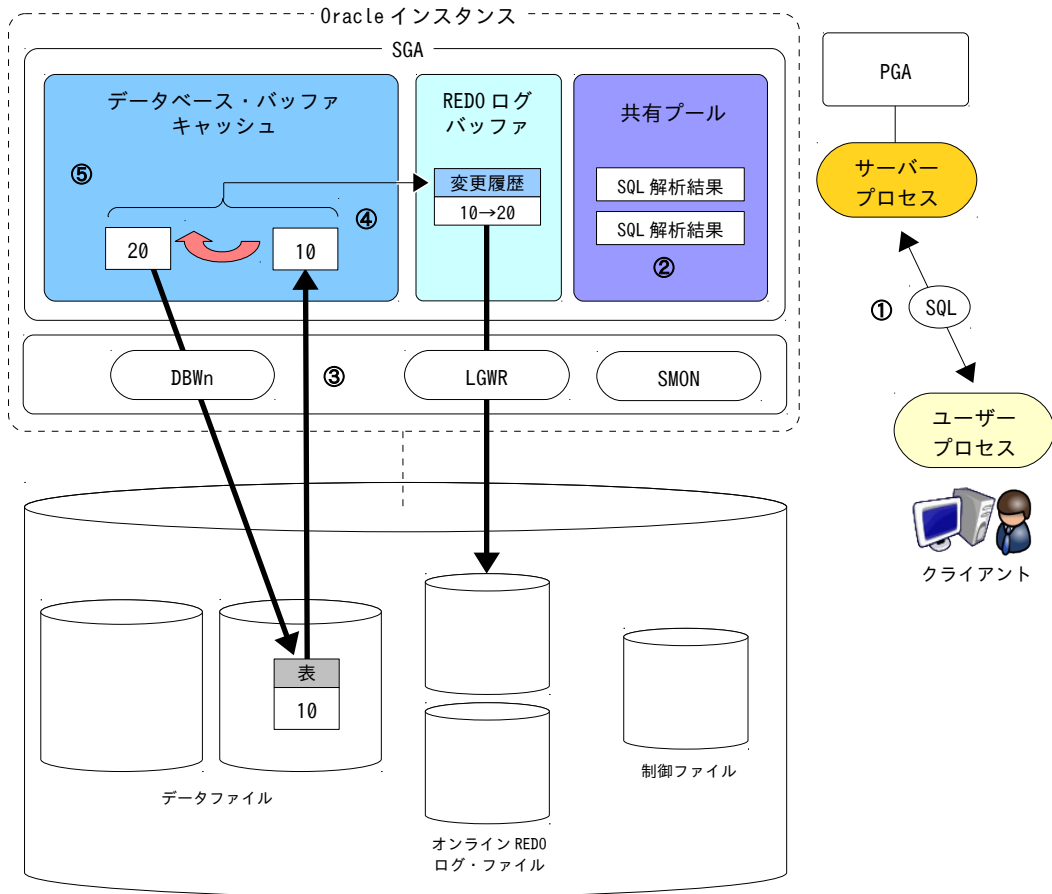
※データベース・バッファ・キャッシュで変更されたデータは、最終的にデータベース・ライター・プロセス (DBWn) によってデータファイルに書込まれます。

(4) データの変更と REDO ログ・バッファ

SQL が DML 処理や DDL 処理だった場合、障害に備えて変更履歴 (REDO レコード) を記録します。REDO レコードは REDO ログ・バッファに一時的に格納されます。

※REDO ログ・バッファに格納された変更履歴は、最終的にログ・ライター・プロセス (LGWR) によってオンライン REDO ログ・ファイルに書込まれます。

※REDO レコードは、REDO エントリとも呼ばれます。



- ① ユーザー・プロセスとサーバー・プロセスの間で接続を確立した後、サーバー・プロセスに SQL が渡される。
- ② サーバー・プロセスは、受け取った SQL を解析し、その結果を共有プールに保持。
- ③ サーバー・プロセスが SQL の実行に必要なデータをデータファイルからデータベース・バッファ・キャッシュに読み込む。
- ④ SELECT の場合、必要なデータを取り出してユーザーに返す。
- ④ 変更処理の場合、変更履歴を REDO ログ・バッファに保持し、データベース・バッファ・キャッシュ上のデータを変更する。
- ⑤ データベース・バッファ・キャッシュで使ったデータは、その後の再利用のために保持される。

※変更処理では、データベース・バッファ・キャッシュ上の変更済みデータは、DBWn がデータファイルに書出す。また、REDO ログ・バッファ上の変更履歴は LGWR がオンライン REDO ログ・ファイルに書出す。

2. データベースへの接続

クライアントと Oracle 間で接続（セッション）を確立し、SQL などをやり取りします。

（1）ローカル接続とリモート接続

データベースへの接続方法として、ローカル接続とリモート接続の2つがあります。

1) ローカル接続によるセッションの確立

クライアントとデータベース・サーバーが同一マシン上に存在する状態で、ネットワークを介さずに接続するものです。

ユーザー・プロセスがデータベースに接続を要求すると、データベース側にサーバー・プロセスが起動し、1対1で対応します。この接続形態を専用サーバー構成といいます。

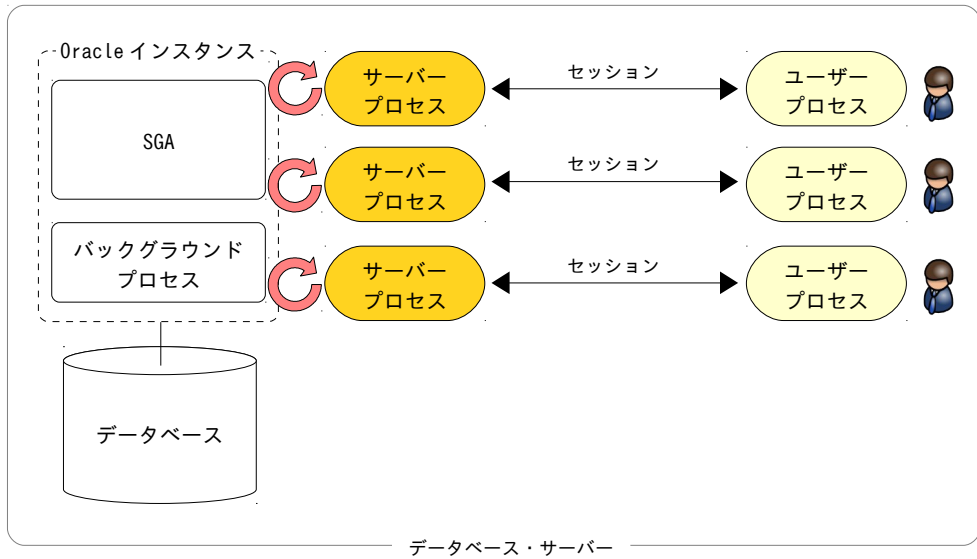
※telnet などリモート・サーバーにログインして、SQL*Plusなどを起動する方法もローカル接続になります。

2) セッションの終了

明示的に接続を切断するか、データベース・アプリケーションを終了するとセッションは終了します。

セッションを終了すると、サーバー・プロセスは停止し、リソースを解放します。

■ ローカル接続



3) リモート接続によるセッションの確立

クライアントとデータベース・サーバーが違うマシン上に存在し、クライアントからネットワークを介してリモートのサーバーに接続するものです。

リモート接続では、サーバー上にはクライアントからのネットワーク接続を受付けるために、リスナー・プロセスを起動する必要があり、また、クライアント側、データベース・サーバー側にはそれぞれ接続定義ファイルが必要です。具体的には、以下の構成・流れで接続を確立します。

※以下番号は右図内の番号に対応しています。

- ①接続先情報を定義したファイル (tnsnames.ora) をクライアントマシンに配置する。
- ②リスナー・プロセスの定義を記載したファイル (listener.ora) をサーバーマシンに配置。リスナー・プロセスを起動する際、このファイルが読み込まれる。
- ③ユーザー・プロセスは接続先情報を定義したファイル (tnsnames.ora) をもとにリスナー・プロセスに接続する。
- ④リスナー・プロセスは接続要求を受け入れ、対応するサーバー・プロセスを起動する。
- ⑤ユーザー・プロセスとサーバー・プロセスが1対1で直接通信し、SQL を実行する。

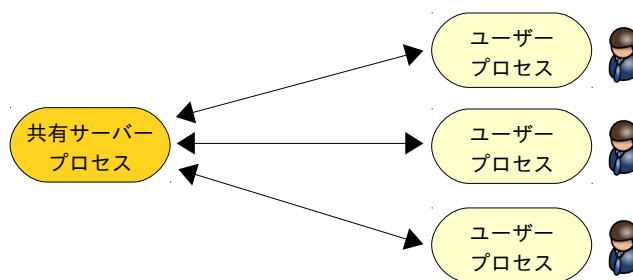
<参考>

クライアントからの接続要求ごとにサーバー・プロセスが起動するため、OLTPのような同時接続が多い環境では起動するプロセス数が多くなり、リソースの消費が問題になる場合があります。

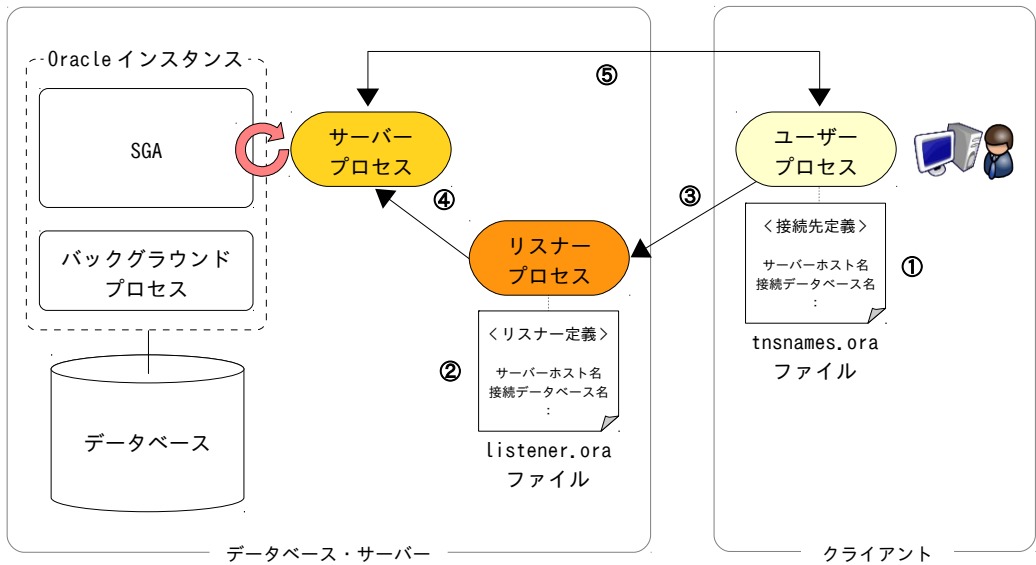
そのような場合、1つのサーバー・プロセスが複数のユーザー・プロセスの要求を処理する(1対nの関係)共有サーバー構成にすることも可能です。

※共有サーバー構成では、サーバー・プロセスは Oracle インスタンス起動時に起動します。その後、ユーザー・プロセスの接続要求数に応じて共有サーバー・プロセス数が自動的に調整されます。

④ 「共有サーバー構成」 (付-1)



■ リモート接続



tnsnames.ora ファイル	データベース・サーバーへの接続情報を記載した、クライアント側に配置するファイル ・接続先のデータベース名 ・データベース・サーバーのホスト名 ・リスナー・プロセスが使用するポート番号 など
listener.ora ファイル	リスナー・プロセスの定義情報を記載した、サーバー側に配置するファイル ・データベース・サーバーのホスト名 ・リスニングの対象とするデータベース名 ・リスナー・プロセスが使用するポート番号 など
リスナー・ プロセス	データベース・サーバー上でリモート接続を受付けるプロセス。listener.ora を読んで起動する ※リスナー・プロセスが停止していると、ネットワーク経由での接続が行えません。

※リスナー・プロセスや接続設定ファイルの作成、定義などの詳細は「データベース・マネジメント」研修で解説しています。

(2) PGA

サーバー・プロセスやバックグラウンド・プロセスは、自身が動作するための非共有のメモリー領域として PGA (プログラム・グローバル領域) を持ちます。ここでは、サーバー・プロセスの PGA について解説します。

1) PGA の構成

PGA は以下の要素で構成されます。

■スタック領域

プロセスに関するプライベートな情報 (プロセスが使用している OS リソースに関する情報など) を管理するメモリー領域です。

■UGA (ユーザー・グローバル領域)

セッション固有の情報を管理するメモリー領域です。主にプライベート SQL 領域と、SQL 作業領域から構成されます。

・プライベート SQL 領域

共有プールに格納されている解析結果と紐付いており、その SQL の問合せ状態やバインド変数の値などのデータが格納されています。

・SQL 作業領域

ソート処理やハッシュ処理で使用するメモリー領域です。このメモリー領域を使い切っても処理が完了しなかった場合、一時的にディスク領域に途中結果を書込みます (ディスクソートの発生)。

🕒 「一時セグメント」 (5-13)

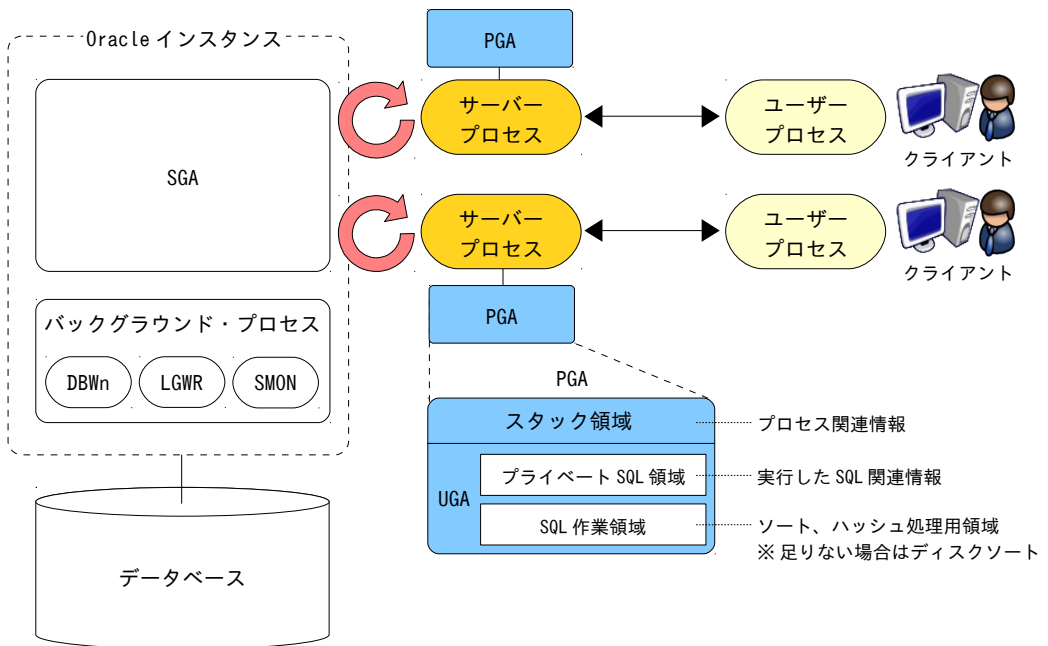
2) PGA の管理

運用中はディスクソート等によるパフォーマンス低下が発生しないように、PGA のサイズを適切に調整する必要があります。

デフォルトの設定では、PGA で使用できる総メモリーサイズを設定すれば、Oracle がその範囲内で状況に合わせて個々の PGA を自動調整します。

※PGA を個々に手動調整することも可能ですが、PGA は時間帯などで使用量が変動するため (同時接続数の増加やソート処理を伴う夜間バッチの実行など)、監視や調整がデータベース管理者の大きな負担となります。PGA のサイズ管理の詳細は「Oracle パフォーマンス・チューニング」研修で解説しています。

■PGA の構成



■PGA の管理

