

ディスク I/O レスポンス遅延の検索アプローチ

遅延 SQL 文を特定する方法 特定した SQL 文の I/O 回数減少化方法

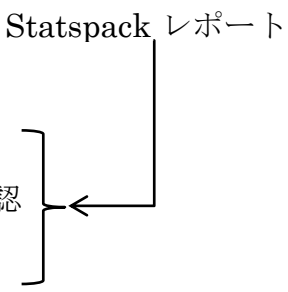
ディスク・アクセス負荷の過大からくるレスポンス遅延と、その対応方法

まず、調査することは本当に DiskI/O が多いのかの検証と、SQL 文によるテーブル・アクセス部分が多いのかの確認である

すなわち、テーブル・アクセス以外の Redo ログ書込みや PGA メモリ不足から来る一時表領域でのマージ操作やソート操作が DiskI/O の多い原因でないかを確認する

そのことがはっきり分かった後で、対象の SQL 文を確定させる

最後に、その SQL 文の DiskI/O が多い原因を調査し、原因に合わせた対応方法をとる

- どのディスクへのアクセスが多いかの確認 (iostat から確認)
 - OS のページング処理の発生確認 (vmstat から確認)
 - メモリ追加、OS 側メモリ・チューニング
 - バッファキャッシュのヒット率の過少からくる同一データの再読み込み確認 (Statspack レポートから確認)
 - SGA メモリの割当て配分の変更
 - 一時表領域の利用については、SQL トレースを採取して実行統計を確認する
~~もしくは、v\$sqlarea より、実行統計を確認する~~
 - Redo ログ書込みについては、Statspack による物理ファイル (Redo ログ) への書込み容量を確認する
また、待機イベントの発生状況を確認する (V\$SESSION、DBA_HIST_ACTIVE_SESS_HISTORY)
 - どの表領域へのアクセスが多いかを確認
 - どのテーブルへのアクセスによる Physical Reads が多いかを確認
 - どの SQL 文に対して、Physical Reads が多いかを確認
- 

Physical Reads が多い SQL 文が確定できてからは、その SQL 文に対しての対応を実施する

Physical Reads が多い SQL 文に対する対応手順

- ・ SQL 文による I/O 回数が多い

→ SQL 文による I/O 回数が多い場合の原因 と、その対応方法

A. SQL の実行回数が多い

- 回数を減らす手段として、実行計画を変更して、効率の良いアクセスにする
(ヒント句の指定を行い、効率良い実行計画へ **Statspack** レポート、**Statspack** キャッシュ・バッファをチューニングして効率 詳細レポートから確認)
- 回数を減らすことが、出来ない場合
効率化のために、テーブルのメモリ常駐化を行い、処理の効率化を図る
マテリアライズド・ビューの作成し、処理の効率化を図る

B. 1 SQL 文当たりの DiskI/O が多い

B.-1 表のレコード件数が多い

- 大きな表をフルスキャンしている場合は、インデックスを使うようにする

B.-2 テーブルの非効率なアクセスが行われている

- インデックスの作成
オブジェクトの統計情報の採取
- 実行計画の見直しを行う
ヒント句を記述して、実行計画のアクセス方法を明示的に指定する

B.-3 表のレコード検索方法である実行計画が変更された

- 途中で実行計画の変更があり遅延したのであれば、ヒント句を使って明示的にアクセス優先順位を示し実行計画を固定化させる

変更結果の反映確認には、

- 実行計画は `v$sql_plan`、
- 実行統計は `v$sqlarea` を使う