

物理 I/O に対する評価

物理 I/O が多い現象に対する原因と減らすための対策

「物理 I/O」に問題があるかの調査経緯

① Report Summary セクションの Top 5 Timed Events より、時間が長かった待機イベントの種類を特定する。

特定した待機イベントが、db file sequential read などの Disk I/O に関する待機イベントが多く占めていれば、以下の調査を続行する

② SQL セクションの Elapsed Time より、elapsed 時間が長い SQL 文を特定する

③ v\$SQLAREA ビュー、もしくは、Statspack 詳細レポートから対象 SQL 文の「実行統計情報」の取得する

④ 実行統計情報について、以下のとおりの内容を確認する

Statspack 詳細レポートより

↓				バッファ読取りブロック			バッファ書込みブロック
実行統計情報				↓			↓
call	count	cpu	elapsed	disk	query	current	rows

↑	解析数	↑	CPU+待機時間	↑			↑
解析種		CPU 時間		ディスク読取りブロック			アクセスレコード数
Parse	1	0.02	0.06	0	0	0	0
Execute	1	0.00	0.00	0	0	0	0
Fetch	4	0.01	0.24	187	292	0	35

物理 I/O 待機イベントが多い SQL 文に関して、待機時間（ Elapsed - CPU ）を **disk** 列で割ると、およそその 1 ブロック当りの平均読込み時間が求められる

この「1 ブロック当りの平均読込み時間」の値が他の SQL 文（他の時間帯の負荷が低い時の SQL 文でも可能）と比較して、**大きい場合は I/O 読込み性能がボトルネック** になっている可能性あり

- disk 値（ディスク読取りブロック）が小さい場合には、
→ ディスク障害が発生して、処理能力が低下している
- disk 値（ディスク読取りブロック）が大きい場合には、
→ ディスク負荷が大きく、ディスクの処理能力オーバーが発生している

※ 厳密には、他のプロセスとの物理ディスクのアクセス競合が起こっていないか確認するため、対象物理ディスクを求め、その物理ディスクの全体アクセス量の多少を求める必要がある

ディスク負荷が大きく、ディスクの処理能力オーバーが発生している場合の対応
少しでも、ディスク・アクセスを少なくするために以下の内容を検討する



物理 I/O が増加する原因と対策

【原因】

- A) バッファ・キャッシュにデータブロックがなく、物理ディスクへ I/O を行う
- B) SQL 実行計画が不適切で、データベースへの読み込みの総量が多すぎる
- C) `direct path read` が発生して、バッファ・キャッシュの使用が回避されている

【対応】

「A) バッファ・キャッシュにデータブロックがない」への対応方法

対応としては、オブジェクトのデータをキャッシュに載せ、エージアウトされにくくする

《この方法は以下の通りです》

- ・初期化パラメータ `db_cache_size` を上げ、バッファ・キャッシュを拡張する
 - ・対象データを事前に参照（空回し）し、データブロックをバッファに載せておく
 - ・対象データが存在するオブジェクトに対し、**KEEP** バッファ・キャッシュを割当てて、エージアウトされ難くする
- もしくは、他の再使用の頻度が低いオブジェクトに対し、**RECYCLE** バッファ・キャッシュを割当てて、バッファ・キャッシュ全体のエージアウトを少なくする

「B) SQL 実行計画が不適切」への対応方法

実行計画が不適切で、アクセスする回数自体が不必要に多くなっているのを、これを少なくして、物理 I/O 回数を下げる。

《この方法は以下の通りです》

- ・絞り込み後のレコード数が少ない表の順で、**結合**されるように、SQL ヒント句を付け加える
- ・テーブル間の結合で、**直積 (cartesian)** が発生してれば、他の結合方法に変更して、アクセス数を減らす（直積でしか処理出来ない場合は、そのまま使用）
- ・インデックスを使用して、アクセスするレコード数を減らす
- ・不要なテーブルにアクセスしていれば、それを外す

「C) direct path read が発生している」への対応方法

大きなデータブロック (バッファ・キャッシュ・サイズの一定以上の割合の大きさ) に一度にアクセスしようとする、バッファ・キャッシュを使用せずに直接ディスクを読み込む動作に変更される (Oracle システム内部での自動判断)

このことを確認するには、実行統計情報に、「db file ~」ではなく「direct path read」が発生している

《この回避方法は以下の通りです》

- ・インデックスを使用して、アクセスするレコード数を減らす
- ・アクセスする列が少ない場合には、複合インデックスを作成して、テーブルへのアクセスを回避する