

ネットワークが処理遅延原因になっているかの切り分ける手順

RAC 構成の Oracle 環境においては、自身や相手方のインスタンスのメモリ内のバッファキャッシュに読み込んだブロック情報や排他制御を使って確保しているレコードの状態を他のノードの Oracle インスタンスに伝えて、サーバー内のメモリに読み込まれたデータと他のノードの Oracle が更新したデータについて、制御情報を共有していく仕組みが備わっている

この RAC での制御情報を管理・統制しているのが、各インスタンスのそれぞれで構成されているノード間での**相互接続（インターコネクト）**である

The diagram illustrates a shared storage architecture. Three servers, labeled 'サーバーA', 'サーバーB', and 'サーバーC', are shown at the top. Each server consists of a monitor and a vertical server rack. A red line represents a network or data bus that runs horizontally across the top, connecting all three servers. Below this bus, a vertical line connects to a large cylinder representing '共有ストレージ' (Shared Storage). The entire setup is enclosed in a rectangular frame.

p091.docx

確認する待機イベントの種類

RAC 環境にて、相互接続（インターコネクト）に関してパフォーマンス問題が発生した場合には、以下に示す**ネットワークが原因**で発生する待機イベントが発生する

gc (global Resource) : Oracle RAC 環境でのバッファキャッシュ・ブロック

待機イベント名	内 容
gc [current/cr] multi block request	RAC 環境でのバッファキャッシュ・ブロックの受信待ち待機状態
gc [current/cr] block [2/3]-way	RAC 環境でノード間通信を行うインターコネクトに使われるネットワークスイッチで何らかの待機時間が発生した ボトルネックやハードウェア故障など
gc [current/cr] block busy	
gc [current/cr] grant 2-way	
gc current grant busy	
gc [current/cr] [block/grant] congested	
gc cr failure / gc current retry	ブロック受取りが出来ない理由から、再要求を発行して、返答待ち待機状態 ※ シングル Oracle 環境でも発生
cr request retry	

確認する対象資料：

~~V\$SESSION 動的ビュー~~ ←

AWR レポート or Statspack レポート

RAC 環境にて、相互接続（インターコネクト）に関してネットワーク遅延が発生する gc リソース関連の待機イベントは、時間的にはすべて短時間で解消する

問題となるのは、発生する回数である

よって、相互接続（インターコネクト）に関しての待機イベントは発生回数が多くても短時間に解消されるため、V\$SESSION の断片的な記録履歴からでは捉えることが出来ない

AWR レポートでの確認する内容

AWR レポートの確認場所 : Top10 Foreground Events セクション

ここに、gc リソース関連の待機イベントが記録されている場合、相互接続（インターコネクト）に関してネットワーク遅延が発生していると考えられる

待機イベント名	比較項目	正常値	異常値
gc [current/cr] multi block request	Waits	110	164,320
	Total Waits time(s)	0	32
cr request retry	Waits	4	502
	Total Waits time(s)	0	2.2

※ gc [current/cr] multi block request の Waits 回数値が、著しく上昇する

※ RAC 環境にて、ネットワーク遅延が発生している場合、gc リソース関連の待機イベントが多発する

しかし、この待機イベントは時間的には短時間なので、Top 5 Timed Events セクションには表示されない

注目しなければならないポイントは発生回数なので、Foreground Wait Events セクションに表示され、Waits 回数が多い

RAC 環境におけるネットワーク遅延発生の原因種類

gc (global Resource) 系待機イベントが多発した場合の原因分け

gc [current/cr] multi block request の Waits 回数と cr request retry の Waits 回数が増加した場合の原因推定は、以下の3つのどれかが疑わしい

- A) 受信側（サーバープロセス）が、すべてを受信できていない
- B) UDP バッファのオーバーフロー
- C) ネットワークカード (NIC)、もしくは、途中のネットワークスイッチでのパケット紛失

追加情報を取得して、A) or B) or C) のどれなのかを判別する

RAC 環境におけるネットワーク障害の**原因種類 (A、B、C)** の判別方法

「A) 受信側 (サーバープロセス) が、すべてを受信できていない」 への検証

A) が原因の場合、

初期化パラメータ : `GCS_SERVER_PROCESS` パラメータを増やす (1、8、32)
と、`gc` リソース関連の待機イベントの `Waits` 回数が低下する

`GCS_SERVER_PROCESS=1、8、32`

※ パラメータを変更しても ~~`Waits` 回数が、30 以下にならない場合には、~~
`AWR` レポートの `Top10 Foreground Events` セクションに `gc` リソース関連
の待機イベントが依然上がっているようであれば、他の原因である B) or C)
を疑う

「B) `UDP` バッファのオーバーフロー」 への検証

B) が原因の場合、

`netstat -s -p udp` コマンドを実行して、`socket overflows` が発生しないことを確認
する

発生していたら、`UDP` バッファの大きさに問題あるので、設定値を増やす

`UDP` バッファ・サイズの推奨値の計算

「`DB_BLOCK_SIZE × DB_FILE_MULTIBLOCK_READ`」

~~この値を初期化パラメータ `receive buffer size` に指定する~~

`UDP` バッファ・サイズの変更方法については、次ページを参照のこと

「C) ネットワークカード (NIC)、もしくは、途中のネットワークスイッチでの
パケット紛失」 への検証

C) が原因の場合、

`netstat -s -p udp` コマンドを実行して、`incomplete headers` パケットと `bad
checksums` パケットを確認する

発生していたら、パケットがドロップされている

※ `incomplet` : 不完全な

この場合の対処方法は、サーバーの `NIC` カードのハードウェア障害や経由するネ
ットワーク途中の機器の故障が考えられる

いずれの場合も `Oracle` 本体では対処のしようがないので、ネットワーク通信に関
する対応を実施する

UDP バッファ・サイズの設定値変更方法

UDP バッファ・サイズの設定は、OS に関するカスタマイズになる
よって、サーバーそれぞれの OS での設定変更を行う

<http://software.fujitsu.com/jp/manual/manualfiles/M070135/J2X12821/01Z200/sasus10/sasus167.html> より

【Solaris】の場合

1. スーパーユーザでのコマンド実行に変更する
 `> su`
2. 現在の UDP 受信バッファ・サイズを確認する
 `> ndd /dev/udp udp_recv_hiwat`
3. UDP 受信バッファ・サイズに関するカーネルパラメタを変更する
 `> ndd -set /dev/udp udp_recv_hiwat <バッファ・サイズ>`

※ 設定を常に有効にする場合には、RC プロシジャ (/etc/rc2.d など) への登録する

4. OS を再起動する

【Linux】の場合

1. スーパーユーザでのコマンド実行に変更する
 `> su`
2. 現在の UDP 受信バッファ・サイズを確認する
 `> /sbin/sysctl -n net.core.rmem_max`
3. UDP 受信バッファ・サイズに関するカーネルパラメタを変更する
 `> /sbin/sysctl -w net.core.rmem_max=<バッファ・サイズ>`

※ 永続的に設定を有効にする場合には、/etc/sysctl.conf に
 `/sbin/sysctl -w net.core.rmem_max=<バッファ・サイズ>`
を追加記述する

4. OS を再起動する

【Windows】の場合

1. 現在の UDP 受信バッファ・サイズを確認する

レジストリエディタで、以下のレジストリ情報を確認

レジストリキー：

HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControl
Set¥Services¥Afd¥Parameters

名前： DefaultReceiveWindow

種類： REG_DWORD

値： デフォルトの受信バッファ・サイズ（バイト）

2. UDP 受信バッファ・サイズに関するカーネルパラメタを変更する

レジストリエディタにて上記レジストリの値を変更する

3. OS を再起動する

【参考情報】

SQL> show parameter DB_BLOCK_SIZE

NAME	TYPE	VALUE
db_block_size	integer	8192

SQL> show parameter DB_FILE_MULTIBLOCK_READ

NAME	TYPE	VALUE
db_file_multiblock_read_count	integer	128

netstat -s -p udp コマンド実行例

IPv4 の UDP 統計

受信したデータグラム	= 22
ポートなし	= 0
受信エラー	= 0
送信したデータグラム	= 22

アクティブな接続

プロトコル	ローカル アドレス	外部アドレス	状態
-------	-----------	--------	----