

WAN で実行する SQL Anywhere パフォーマンスの最適化

この文書では、WAN で実行している Adaptive Server Anywhere データベースのネットワーク・パフォーマンスのチューニングについて説明します。TCP/IP は WAN 実装用に選択されたプロトコルであり、この文書の主題です。

タイム値を適当な高いレベルに設定してください。ライブネスをテストする場合やクエリが完了したかどうかを判断する場合に、クライアントとサーバがパケットを交換することに注意してください。検討すべきスイッチは次のとおりです。

-tr これは、アクティブな要求の終了タイムアウトです。サーバは、クライアントから要求が再送信されるのをこの時間だけ待ってから、その要求と接続をタイムアウトします。使用中の設定でこの問題が発生した場合は、この値を大きくしてください。Adaptive Server Anywhere 6.0 では、このスイッチは不要になっています。

-ts これは、デフォルトのクライアント・リトライ・タイムアウトです。このスイッチを使用すると、サーバからの応答をどれだけ待ってから要求を再送信するのがクライアントに通知されます。この値を大きくすると、クライアントからの再送信が少なくなり、クライアントとサーバの間で交わされる情報の量が減少します。この値を大きくした場合は、サーバの要求タイムアウト期間を長くすることも検討してください (サーバの **-tr** スイッチを参照)。Adaptive Server Anywhere 6.0 では、このスイッチは不要になっています。

-tl これは、ライブネス・チェック用のスイッチです。この値を大きくしてもパフォーマンスは改善されませんが、ライブネスが原因で接続がタイムアウトし続ける場合は、この値を大きくしてください。このスイッチは、クライアントまたはサーバで設定できます。クライアントで設定すると、その特定のクライアントに対するサーバ設定は無効になります。

クライアントとサーバのパケット・サイズ (**-p**) を、WAN/プロトコルの組み合わせにとって適切と思われる最大値に設定します (正確な値を知っているのはネットワーク管理者だけです)。この目的は、実際にパケットに含まれている数だけローを返すフェッチを可能にすることです。ここでは、適切と思われる量のパケットだけを使用します。この値を大きくしすぎると、複数の IP パケットが作成されることがあるので注意してください。

結果セットに含まれる最初のいくつかのローだけを扱う場合は、**-r** パラメータを使用して複数ローのフェッチを禁止することを検討してください。このスイッチは、サーバまたは各クライアントで設定できます。このスイッチを使用すると、クライアントは複数のローではなく一度に 1 つのローだけをフェッチします。

多数のローを検索している場合は、**-b** スイッチを使用して、マルチレコード・フェッチで使われるパケットの数を制限することを検討してください。

-s スイッチを使用して、ネットワーク・バッファ用のバッファ領域の量を増やすことを検討してください。

-x tcpip{receivebuffersize=nnnnn} パラメータの使用を検討してください。このオプションを使用すると、SQL Anywhere クライアントまたはサーバが宛先となる TCP/IP パケットを受信するためのプロトコル・スタックで、メモリが事前に割り付けられます。プロトコル・スタックの内部でメモリを事前に割り付けると、ネットワーク型アプリケーションのクライアント/サーバ・パフォーマンスが向上することがあります。

-x tcpip{sendbuffersize=nnnnnn} パラメータの使用を検討してください。このオプションを使用すると、TCP/IP パケットを送信するためのプロトコル・スタックでメモリが事前に割り付けられます。サポートされるプラットフォームでは、パケットの送信が非同期に実行されます。また、メモリ・ブロックを事前に割り付けることにより、プロトコル・スタックとクライアント／サーバがより長い時間において同時に動作することが可能になるため、ネットワーク集約型アプリケーションのパフォーマンスが向上します。

host=x.y.z.w、timeout=X、dobroadcast=NO tcpip パラメータを使用して、接続速度を上げることを検討してください。

大きなデータベース・クエリに対してストアド・プロシージャを使用することを検討してください。これにより、小さな呼び出し文を送信してクエリを実行できるようになるため、ネットワーク全体にわたって大きな文をサーバに送信する必要はなくなります。

結果セットに多数のカラムが含まれている場合は、select 文の複数のカラム値を 1 つの値に結合することを検討してください (たとえば string() 関数を使用する)。

サード・パーティの開発ツール (PowerBuilder、Visual Basic、Delphi など) を使用するときには、パフォーマンスを高めることができるアプリケーション固有の設定がないかどうかを確認してください。たとえば、PowerBuilder では、BLOCK 接続プロパティを 1 に設定し、staticbind 接続プロパティを 1 に設定すると、多くの場合、WAN におけるアプリケーション・パフォーマンスが向上しました。

次に、チューニングを始めるときに使用できるコマンド・ライン・スイッチの推奨設定をいくつか示します。これらの設定から始めて、自分のテスト環境で各設定を調整し、どのようにパフォーマンスに影響するのかを確認してください。また、上記の各スイッチを追加／削除して、それがどのようにパフォーマンスに影響するのかを確認してください。特定のアプリケーションを使用している特定のネットワーク環境で最高のパフォーマンスを得るための方法はないため、試行錯誤が必要となります。LAN 環境でもアプリケーションを実行する場合は、LAN 環境に対する通信オプションの調整が必要になる場合があるので、必ずその環境でも設定をテストしてください。

SQL Anywhere Server Command Line switches

```
dbsrv50 -x TCPIP -p 1460 -tl 2000 -tr 2000
```

SQL Anywhere Client Command line switches

```
dbclient -b 32 -s 1000 -p 1460 -ts 300,3000 -tl 2000
```

```
-x
```

```
TCPIP{host=xxx.xxx.xxx.xxx;Timeout=20;DOBROADCAST=NO;receivebuffersize=100000;sendbuffersize=100000}
```

注意：ここに示したのは、通信のためのオプションです。ほかに検討すべきものとしては、すべての環境でサーバをより効率的にするオプションがあります (たとえば、キャッシュ・サイズとデータベース・ページ・サイズ)。詳細については、マニュアルを参照してください。

IPX/SPX プロトコルには、ルート (WAN) ネットワークにおける重大なパフォーマンス制限がいくつかあります。そのため、Novell は "パケット・バースト" と "SPX II" の変更によってそれらの制限を修正しています。

これらは、アプリケーション／ネットワーク環境の正しい組み合わせを決定するために調整をおこないます。