



INFORMATION ANYWHERE

iAnywhere
A SYBASE COMPANY

Technical Guide

WAN上でのAdaptive Server Anywhereの 性能最適化



目次

はじめに	3
ネットワーク性能の測定	4
アプリケーションのチューニング	5
リクエストの数と伝送データ量を削減する対策	5
リクエスト数を削減する対策	5
ネットワーク上の伝送データ量を削減する対策	6
Adaptive Server Anywhereのチューニング	7
ネットワークのレイテンシーが大きい場合の対策	7
ネットワークのスループットが小さい場合の対策	8
設定例	9
LANとローカルマシンの性能への影響	10
法的注意	11

はじめに

ソフトウェアバージョン

本書の内容は、SQL Anywhere Studioのバージョン8以降を対象にしています。

本書はWANで稼動するAdaptive Server Anywhereネットワークサーバの性能をチューニングする方法を説明するもので、主にWAN実装で使用されるTCP/IPプロトコルについて論じています。ネットワーク性能のチューニングは、特定のアプリケーションとネットワークの性能を最大限に引き出すために繰り返す必要がある作業です。

チューニングは以下の手順で行うことをお勧めします。

1. アプリケーションが動作するネットワークの性能を測定する。
2. ネットワークの性能に基づいてアプリケーションをチューニングし、リクエストあるいは伝送データの量を削減する。
3. ネットワークの性能に基づいてAdaptive Server Anywhereのサーバオプションと接続パラメータをチューニングし、最適化する。
4. LANまたはマシンの動作についてチューニングの成果を再検討する。

多くの場合、アプリケーションのチューニングはAdaptive Server Anywhereのチューニング以上に性能に影響します。

次ページからは、上記の順に実際のチューニングの方法を見て行きましょう。

ネットワーク性能の測定

ネットワークの性能を示すものとして、レイテンシー（ネットワーク遅延）とスループットが使われます。レイテンシーは、あるマシンがデータをパケットとして送信してから別のマシンがそのデータを受け取るまでの時間を意味します。たとえば最初の実機がデータを送信した10ms後に第二の実機がデータを受け取ったとすると、レイテンシーは10msとなります。一方スループットとは、任意の時間内に伝送されるデータ量を意味します。たとえばあるマシンが1000KBのデータを送出し、別のあるマシンがそのすべてを受信するのに5秒を要したとすると、スループットは200KB/sとなります。LANのレイテンシーは通常は1ms未満、スループットは1MB/sを超えます。WANではレイテンシーは5ms～500ms程度といちじるしく大きくなり、スループットは4KB/s～200KB/s程度と小さくなるのが普通です。

二台の実機間のネットワークのレイテンシーは、pingユーティリティによるラウンドトリップ時間として測定できます。ラウンドトリップ時間とは、最初の実機から二台目の実機へとデータを送った際のレイテンシーに、二台目の実機から最初の実機へデータを送った際のレイテンシーを足したものです。ネットワークのスループットは、あるマシンから別のマシンへと200KB以上のサイズのファイルをコピーし、その時間を計ることで測定できます。コピーはFTPによる一般的なファイルコピーでも、インターネットブラウザによるファイルのダウンロードでもかまいません。

レイテンシーが大きい一方で妥当なスループットを持つネットワーク上でAdaptive Server Anywhereの性能を高めるには、クライアントが生成するリクエストの数を最小限に抑えなくてはなりません。またレイテンシーは妥当ながらスループットが小さいネットワークでは、クライアントとサーバ間で伝送するデータの量を最小限にする必要があります。

アプリケーションのチューニング

以下にアプリケーションを変更してWANの性能を高める対策を示します。

リクエストの数と伝送データ量を削減する対策

以下に示す対策を講じることで、一般にリクエストの数と伝送データ量の両方が削減できます。またこれらはスタンドアロン、LAN、WANのどの環境でもアプリケーションの性能向上に効果を発揮します。

- ◆ **多くのSQL文を必要とするロジックを、アプリケーションからストアドプロシージャまたは関数に移す**
- ◆ **同じベーシックなSQL文を二回以上使う場合は文を一つだけ用意し、それを異なるパラメータで繰り返し実行するようにする**
- ◆ **アプリケーションに不要なクエリまたはSQL文を実行させない**
特定のクエリが二回以上実行されるのであれば、アプリケーションを変更して最初のクエリ実行時に結果をキャッシュし、クエリを再実行するかわりにキャッシュされた値を使うようにしてください。
- ◆ **プロパティ、関数、変数値をそれぞれ取得するクエリがある場合は、それらを組み合わせて複数の列からなる一つのクエリにする**
たとえば `SELECT current user`、`SELECT @@servername`、`SELECT connection_property('BytesSent')` という三つのクエリを実行するかわりに以下のクエリを実行します。

```
SELECT current user, @@servername, connection_property('BytesSent')
```
- ◆ **サーバでjoinが実行可能な場合、複数のクエリでjoinを行わないようにする**
あるアプリケーションがクエリを実行し、さらにそのクエリの結果を使う第二のクエリを実行するのであれば、結果的に複数のクエリでjoinを実行しているのと同じことになります。複数ではなく一つのクエリでjoinを行えば、性能はいちじるしく向上します。たとえばアプリケーションが `SELECT T.x FROM T WHERE <conditions on T> ORDER BY <order>` というクエリを実行し、次に T.xの各値について `SELECT R.y FROM R WHERE R.z = <value of T.x>` を実行するとしましょう。これらは一つのクエリ `SELECT T.x, R.y FROM T, R WHERE <conditions on T> AND R.z = T.x ORDER BY <order>` に組み合わせることが可能です。
- ◆ **サードパーティの開発ツール (PowerBuilder、Visual Basic、Delphiなど) を使っている場合は、性能を改善するために実行できるアプリケーション独自の設定があるか確認する**
たとえばPowerBuilderでは、BLOCK接続のプロパティの変更がWAN上でのアプリケーションの性能を高める場合があります。

リクエスト数を削減する対策

以下の対策はリクエスト数を削減するもので、ネットワークのレイテンシーが大きい場合は特に効果的です。これらの対策はまた他のネットワークの性能向上にも適用できます。

- ◆ **get dataのかわりにbound列を使う**
これは特にカーソルから最初の行のfetchを実行する際にリクエストの数を減らします。
- ◆ **可能であればSQL文を結合して一つのバッチ処理（セミコロンで分かれた一連のSQL文で、一つの文のように実行される）にする**
- ◆ **余分なcommitリクエストを削除する必要がある場合に限り、autocommitを無効にして、明示的にcommitする**
ただし大半の場合は過度のブロックを回避するため、commitまたはrollbackはユーザー入力待ちの前に行います。
- ◆ **fetch操作では列の設定サイズを大きく取り、insert操作ではパラメータ値の配列を使用するようにする**
これにより各行に一つのリクエストを使う代わりに、一つのリクエストで複数の行のfetchまたはinsertを実行します。prefetchもまた各リクエストあたり複数の行のfetchを実行します。

ネットワーク上の伝送データ量を削減する対策

以下の対策は一般に伝送データの量を減らすので、ネットワークのスループットが小さい場合は特に有効です。これらの対策はどのネットワークでも性能低下の原因となることはないはずですが。

- ◆ **大きなデータベースクエリにストアドプロシージャを使うことを検討する**
これによりクエリ実行時に小さなCALL文を送ることができるので、ネットワークでサーバへ大きな文を送る必要がなくなります。
- ◆ **クエリの最初の行あるいは最初の二、三行のみをfetchするとわかっている場合は、クエリにFIRST またはTOP *n*節を加える（クエリの最初の二、三行をスキップしたい場合はSTART AT節を用いる）**
これらの節は特にfetchが可能な場合、使わない行の伝送を防止します。またこれらの節の追加により、クエリ・オプティマイザーに能率的なクエリ実行方法を学習させることができます。

Adaptive Server Anywhereのチューニング

以下は、Adaptive Server AnywhereをチューニングしてWAN上のアプリケーションの性能を高める方法です。ネットワークのレイテンシーやスループットとは関係しません。

- ◆ **クライアント接続パラメータ"CommLinks=TCPIP(host=w.x.y.z;port=p);DoBroadcast=none"を使う**
WANによる接続でこれらのオプションが必要となる理由は、サーバを探すのに用いられるUDPパケットに反応しないように設定されているサーバからはブロックされ、またファイアウォールの機能により配送されない可能性もあるためです。オプションなしで接続できる場合であっても、オプションを使うと接続時間が縮まる可能性があります。w.x.y.zの代わりにhostnameを使うこともできます。サーバがデフォルトのTCP/IPポート(2638)を使っている場合はポートオプションは必要ありません。
- ◆ **接続がLivenessのせいでドロップされる場合、Livenessタイムアウトを増やすことを検討する**
この値を増やしても性能は上がりませんが、接続がLivenessのためにタイムアウトされる状況が続くのであれば、この値を増やす必要があります。このオプションは、クライアントではLivenessTimeout接続パラメータ、サーバでは-tlで設定できます。LivenessTimeout接続パラメータは、WAN接続に限りLivenessTimeoutを変更できます。
- ◆ **サーバの-pオプションまたはクライアントのCommBufferSize接続パラメータにより、パケットサイズをWANとプロトコルの組み合わせが許す最大値まで増やすことを検討する**
デフォルト値の1460は、WANを挟んで共にイーサネットカードを備えたクライアントとサーバを含め、大部分のイーサネットハードウェア上のTCP/IPに適しています。さらに大きな値(たとえば5000や10,000)は、大きなfetch、複数行のfetch、大きなパケットによるデータ伝送をサポートするネットワークでBLOBオペレーションの性能を高める可能性があります。
大半のギガビットイーサネットハードウェアはジャンボフレームを使う8KBの伝送パケットをサポートできますが、大きなサイズを使うように設定する必要があるので注意が必要です。またこの対策はWANが大きなサイズのパケットを同様に伝送できる場合に限り有効です。CommBufferSize接続パラメータを使えば、大きなパケットサイズが有効な接続に限りパケットサイズを変えることができます。なお不適当なパケットサイズは性能の低下を招く場合があります。

ネットワークのレイテンシーが大きい場合の対策

以下の対策はほとんどの場合リクエストの数を減らします。これらはネットワークのレイテンシーが大きい場合に特に効果的です。

- ◆ **prefetch動作の変更を検討する**
アプリケーションが行をfetchを実行するとき、Adaptive Server Anywhereはカーソルタイプやその他の条件に応じて追加の行のprefetchを行います。prefetchはカーソルから多数の行がfetchされる際のリクエストを減らすので、特にレイテンシーが大きいネットワークの性能を大きく改善する場合があります。absolute fetch、relative negativeまたはrelative 0 fetch、rollback、ある種のget dataなどの操作は、prefetch行の廃棄または再fetchの原因となって性能低下をもたらすことがあるので注意してください。prefetchは、forward-onlyまたはread-onlyカーソルで最初の行とそれ以外の行をfetchするときにも最適しています。アプリケーションが各カーソルから多くの行をfetchし、しかもfetch next操作のみを行う場合は、PrefetchRowsおよびPrefetchBuffer接続パラメータが性能を改善する可能性があります。

アプリケーションが結果セットのごく一部をfetchする場合は、多くの行をprefetchすると性能を下げる可能性があります。またODBCあるいはOLE DBアプリケーションでは、多くのカーソルがオープンされる場合、PrefetchOnOpen接続パラメータが有効です。

◆ **特に多くのカーソルがオープンされクローズされる場合は、LazyClose接続パラメータの使用を検討する**

これはカーソルのクローズ時に余分のリクエストを消去するものです。ただしこのパラメータの使用はキャンセルリクエストがしばしば生じるアプリケーションにはお勧めできません。

ネットワークのスループットが小さい場合の対策

ネットワークのスループットが小さい場合は以下の対策が有効です。

◆ **サーバの-pcオプション、またはクライアントのCompression接続パラメータによる通信圧縮の使用を検討する**

通信圧縮は伝送データの総量と伝送パケット数を減らしますが、リクエストの数は減らしません。通信圧縮は、大きなfetch、複数行のfetch、BLOBオペレーションで特に有効です。ただし通信圧縮はクライアントとサーバで多くのCPUとメモリを消費し、結果的にLANの性能を下げてしまうことも多いので注意が必要です。クライアントのCompression接続パラメータを使えば、WAN接続に限り通信圧縮を実現できます。

◆ **クライアントとサーバの両方でTCP/IPプロトコルオプションReceiveBufferSizeとSendBufferSizeを使うことを検討する**

これらのオプションはTCP/IPパケットを送受信するプロトコル・スタックにメモリをあらかじめ割り当てるもので、ネットワークへの依存度が高いアプリケーションのクライアント／サーバの性能を高める可能性があります。デフォルト値はマシン依存ですが、試験に適切な値は65,536～262,144バイトの範囲です。

◆ **アプリケーションがカーソルの最初の行のみを使う場合、DisableMultiRowFetch接続パラメータでprefetchを無効にする**

これを実行することで、性能が改善される可能性があります。アプリケーションのチューニングでも述べたとおり、DisableMultiRowFetchの代わりにクエリでFIRST節を使って最初の行のみをfetchするようにアプリケーションを変更すると、より良い性能を得られる場合があります。ただしDisableMultiRowFetchは、高速ネットワークを含む多くの場合に性能の低下を引き起こすことがあります。

設定例

以下にチューニングの当初に利用できる設定をご紹介します。チューニングをこれらの設定から始め、試験環境で調整しながら性能に及ぼす影響を確認するとよいでしょう。前記の各種サーバオプションや接続パラメータを追加あるいは削除し、それらが性能に及ぼす影響を把握してください。任意のネットワーク環境中の任意のアプリケーションの性能を最適化するには、若干の試行錯誤は避けられません。

LAN性能への影響の可能性については[LANとローカルマシンの性能への影響](#)を参照してください。

データベースサーバのコマンドライン・オプション

```
dbsrv9 -x TCPIP(SendBufferSize=100000;ReceiveBufferSize=100000)
```

```
-n server_name ...
```

クライアント接続パラメータ

```
"ServerName=server_name; Compression=Yes;
```

```
CommLinks=TCPIP(Host=w.x.y.z; DoBroadcast=NONE;
```

```
SendBufferSize=100000; ReceiveBufferSize=100000);
```

```
PrefetchRows=50; PrefetchBuffer=1000; PrefetchOnOpen=Yes;
```

```
LazyClose=Yes; ..."
```

NOTE

ここで示すオプションは通信機能に関連するものです。キャッシュやデータベースのページサイズなど、サーバの効率改善のために検討すべきオプションは他にもあります。詳細についてはSQL Anywhere Studioに添付の文書をご覧ください。

LANとローカルマシンの性能への影響

アプリケーションをLAN環境で動かす可能性があるなら、そこでもAdaptive Server Anywhereの通信オプションを忘れずに試験してください。その環境向けにオプションをチューニングする必要があるかもしれません。

本書で述べたアプリケーションに対するチューニング対策は、通常はLANおよびローカルマシンの性能を改善することはあっても、性能を下げることはありません。

ただし以下に示すAdaptive Server Anywhereのチューニングについては、LANまたはローカルマシンの性能を下げる可能性があります。これらはWANクライアントに対してのみ設定するか、あるいは試験を通じてLANでも性能改善に効果があると確認した後に限り利用してください。

- ◆ [liveness timeoutの変更](#)
この変更により、切断された接続が長時間にわたって検知されない場合があります。
- ◆ [パケットサイズの変更](#)
不適切に設定された場合はスループットを下げる可能性があります。
- ◆ [通信圧縮の有効化](#)
メモリとCPUの消費の増加がスループットの利得を上回り、LANでの性能低下を招く場合がしばしばあります。
- ◆ [prefetch設定の変更](#)
prefetch設定（PrefetchRows、PrefetchBuffer、DisableMultiRowPrefetch）の変更は、どんなものであれ、アプリケーションと環境によっては性能の低下を引き起こす可能性があります。

法的注意

Copyright(C) 2006 iAnywhere Solutions, Inc. All rights reserved.

iAnywhere、iAnywhere Solutions、iAnywhere Solutions(ロゴ)、Adaptive Server、SQL Anywhere、Mobile LinkはiAnywhere Solutions, Inc.またはSybase, Inc.とその系列会社の米国または日本における登録商標または商標です。その他の商標はすべて各社に帰属します。

Mobile Linkの技術には、Certicom, Inc.より供給を受けたコンポーネントが含まれています。これらのコンポーネントは特許によって保護されています。

本書に記載された情報、助言、推奨、ソフトウェア、文書、データ、サービス、ロゴ、商標、図版、テキスト、写真、およびその他の資料(これらすべてを"資料"と総称する)は、iAnywhere Solutions, Inc.とその供給元に帰属し、著作権や商標の法律および国際条約によって保護されています。また、これらの資料はいずれも、iAnywhere Solutions, Inc.とその供給元の知的所有権の対象となるものであり、iAnywhere Solutions, Inc.とその供給元がこれらの権利のすべてを保有するものとします。

資料のいかなる部分も、iAnywhere Solutionsの知的所有権のライセンスを付与したり、既存のライセンス契約に修正を加えることを認めるものではないものとします。

資料は無保証で提供されるものであり、いかなる保証も行われません。iAnywhere Solutionsは、資料に関するすべての陳述と保証を明示的に拒否します。これには、商業性、特定の目的への整合性、非侵害性の黙示的な保証を無制限に含みます。

iAnywhere Solutionsは、資料自体の、または資料が依拠していると思われる内容、結果、正確性、適時性、完全性に関して、いかなる理由であろうと保証や陳述を行いません。Sybaseは、資料が途切れていないこと、誤りがないこと、いかなる欠陥も修正されていることに関して保証や陳述を行いません。ここでは、「iAnywhere Solutions」とは、iAnywhere Solutions, Inc.またはSybase, Inc.とその部門、子会社、継承者、および親会社と、その従業員、パートナー、社長、代理人、および代表者と、さらに資料を提供した第三者の情報元や提供者を表します。



アイエニウェア・ソリューションズ株式会社

<http://www.ianywhere.jp/>