

快速脱! 休日出勤

.NETアプリケーション障害解析の極意

デバッグ技法

最終回 ボトルネック解析

株式会社NTTデータ
飯山 教史
IIYAMA, Takashi

システムのパフォーマンス

この連載ではデバッグの方法について説明してきた。最終回を迎えるにあたり、今回はシステムでパフォーマンスの問題が報告されたときにその原因を解析する方法を説明する。

このトピックを取り上げた理由は、上流工程で努力しても現場でパフォーマンスの問題が発生するのを多く見てきたからだ。それはこの問題が「コードの作り」というよりは「システム環境」に依存する度合いが多いことに関係し

ている。つまり運用段階になって初めて、パフォーマンスの問題は顕在化する傾向がある。

後工程になるほどバグの検出は難しいことはよく知られているが、それゆえパフォーマンスの問題も原因の特定が難しい。そこで今回は、パフォーマ

ンスの問題が発生したときにどのようにボトルネック解析を進めるかを手順に沿って説明することにした。解析手順をしっかりと理解して、ボトルネック解析作業を効率よく進めてもらいたい。

遅延の原因

まずはじめに、パフォーマンスの問題が発生したときに問題を起こしている可能性のあるシステムの構成要素と、その原因を表1にまとめておく。

以下が、各部位で遅延が起こる理由である。

●クライアント

クライアント端末に原因がある場合、その原因は大きく分けて

- ・コード
- ・リソース

の2つである。

コードがボトルネックの原因になる

表1: 遅延の原因

構成要素	理由
クライアント	コード(冗長なコード)
	コード(データ量)
	コード(ロック長)
	コード(デッドロック) ^[*1]
	リソース不足
ネットワーク	帯域不足
Webサーバー	コード(冗長なコード)
	コード(データ量)
	コード(ロック長)
	コード(デッドロック) ^[*2]
	設定(処理スレッド数など)
データベース	設計(テーブル配置など)
	設定(インデックスなど)
	リソース不足

* 1) C/Sシステムの場合にはクライアントでこのような問題が発生する。

* 2) Webシステムでもブラウザでオブジェクトを実行すればこのような問題が起こる。

レベル >>> Level

1 2 3 4 5

言語 >>> Language

Visual Basic

ツール >>> Tool

Visual Studio 2005 Team Edition for Software Developers

のは、「ロジックに問題があり膨大な処理を行なった結果として時間がかかる」場合や、「複数のソフトウェアを同時に実行してリソース不足を引き起こす」場合である。ロジックの問題でよく挙げられるのは、「頻繁なログの書き込み」や「クエリの結果セットとして受け取った膨大なデータの処理」などである。

リソースに問題がある場合は、「端末で同時に複数のサービスを実行した結果としてリソース不足になる」など、クライアント端末のリソース設計そのものに問題があることが多い。

●ネットワーク

ネットワークの問題は、クライアント⇄Webサーバー⇄各種サーバー間で送受信されるデータ量に対して帯域幅が不足している場合に起きる。帯域が不足する原因としては、「処理するリクエスト自体のデータ量が多い」「個々のリクエストで処理するデータ量は少ないが処理するリクエスト数が多いため帯域が不足する」などがある。

●Webサーバー (IIS)

Webサーバーがボトルネックになる場合、その原因はクライアントと同様に、

- ・コード
- ・リソース

そしてさらに、

- ・Webサーバーの設定

が考えられる。たとえばWebサーバー

のプロセスに割り当てるメモリ量、同時に実行するスレッド数や処理する場所 (IIS6.0であればコードの実行をinetinfo.exeで行なうかdllhost.exeで行なうかの違い) などである。

Webサーバーの設定は必要なメモリ量や発生するコンテキストスイッチに影響を与えるので、設定によってはボトルネックに直結する場合がある。

●データベース

データベースに原因がある場合、原因は大きく分けて、

- ・設計
- ・設定
- ・リソース不足

の3つがある。

設計に問題がある場合は、システムで要求されるクエリに対してテーブルの論理設計が適切ではない (例: テーブル結合が頻発する) など、データベースの論理設計が起因となり処理が高負荷になる。

設定に問題がある場合では、インデックスやトリガーなどがクエリ処理を高負荷にしてしまう。

設計や設定が適切であるにもかかわらずレスポンスタイムに問題がある場合は、データベースサーバーのリソースが不足している場合が多い。

以上、パフォーマンス問題の原因となる部位と、その理由を簡単に説明した。このようにパフォーマンス問題の解析とは、

①遅延部位の特定 (マシンまたはネットワーク)

②その理由の解明 (その部位のどのリソースが不足しているか)

になる。

以降では解析手順を示し、どのようにこれらを調査するのかを説明する。

解析作業の順番

はじめにボトルネックを切り分ける手順を説明する。一般的な解析プロセスを図1に示す。

それぞれの手順を説明していこう。

●処理の再現

レスポンスタイムに問題があり、そのシステムボトルネックを調査する場合は、必ず問題がある処理を再現する手順を調べる。再現手順として必要なのは以下の情報になる。

- ・前提条件 (同時に起動するプロセス、データベース内のデータ量など)
- ・処理の端点 (画面上のボタン、リンクなど)
- ・入力情報 (画面上の入力項目、メニューに渡すパラメータ値など)

図1：ボトルネックの解析プロセス

