

LANアナライザを用いた

5G-1

データベースの挙動推測における一方式

横里 純一* 芝 諭* 飛山 哲幸** 堀川 博史** 小川 文男**

三菱電機(株) 情報システム研究所* 情報システム製作所**

1. はじめに

市販のソフトウェアを使用する際、一般に内部的な処理方法が不明である場合が多い。今回、LANアナライザによってキャプチャしたクライアント/サーバ型アプリケーション間のパケットを処理することによりタイムチャートを作成し、市販されているクライアントサーバ型データベース管理システム(市販DBMS)に対してクライアントアプリケーションから同時に要求が起こった場合の挙動の様子を観察した。ここでクライアントアプリケーションはサーバに処理を依頼している最中は他の処理を行わず、サーバからの返答を待っているものとする。またネットワーク上のパケットは暗号化されていないものとする。

2. タイムチャートの作成

データベースサーバが送受信するパケットをLANアナライザによりキャプチャした。キャプチャしたパケットから、各クライアント要求へのサーバからの処理待ち時間の分布を示すタイムチャートを次の処理によって作成する。

i) 関連パケットの抽出

キャプチャしたパケットにはデータベースサーバが送受信する全てのパケットが含まれており、データベース処理とは無関係のものもあるため、それを取り除く。

ii) パケットラベル付け

キャプチャしたパケットを送受信しているサーバアプリケーションとクライアントアプリケーションの組毎にラベル付けを行なう。このラベル付けは送受信アドレス、ソケット番号で決定される。

iii) パケット間の時間間隔の計算

同一ラベルで、直前に発せられたパケットからのオフセット時間を計算する。キャプチャしたパケットの時間情報はその直前にキャプチャしたパケットからのオフセット時間が記録されているため、同一ラベルの直前に発したパケットとの間にキャプチャされたパケットの全オフセット時間を累計した。

サーバアプリケーションからクライアントアプリケーションへのパケットの場合には、累計したオフセット時間はクライアントがサーバを待っている時間(サーバ待ち時間)、クライアントからサーバへのパケットの場合には、クライアントが処理を行っている時間(クライアント処理時間)とした。図1にパケットのラベル付けと処理時間の計算について示す。

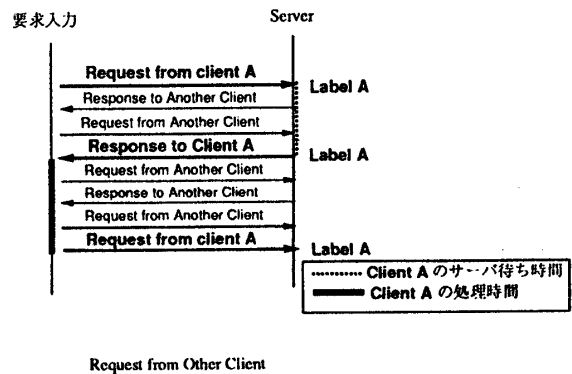


図1: パケットのラベル付と処理時間の計算

iv) タイムチャートの作成

横軸に時間をとり、各クライアント毎のサーバ待ち時間を帯グラフで表したタイムチャートを作成する。タイムチャートは同一ラベルのパケットから上記i)~iii)の処理により算出したサーバ待ち時間、クライアント処理時間を時間方向に時間の大きさに比例した長さのチャートを記入することにより作成する。タイムチャートには同一時刻に実行しているクライアントアプリケーションを並べて表示できるため、サーバ待ち時間の分布を見ることができ、それによりクライアントアプリケーションからの同時要求の様子を観察し、サーバアプリケーションの挙動の推測を行なうことができる。図2にタイムチャートを示す。

上記i)~iii)の処理を自動的に行なう処理時間計算ツールを作成し、その処理時間計算ツールから出力された処理時間情報を用いて、上記iv)のタイムチャートを表示するタイムチャート表示ツールの作成を行なった。

A method to infer the behavior of database management system using LAN analyzer

J.YOKOSATO, S.SHIBA, T.TOBIYAMA, H.HORIKAWA, H.OGAWA

MITSUBISHI Electric Corporation

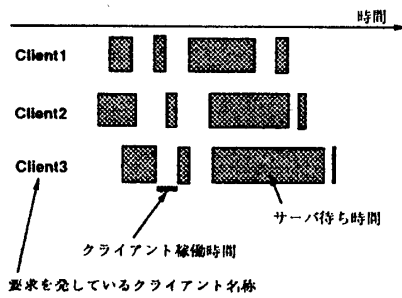


図 2: タイムチャートの説明

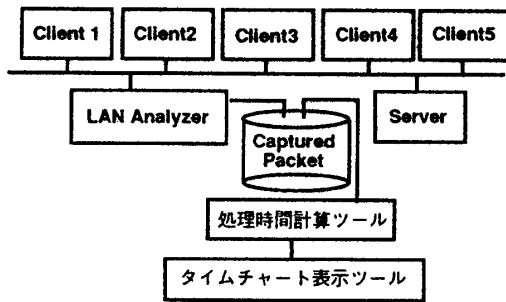


図 3: 実験システム

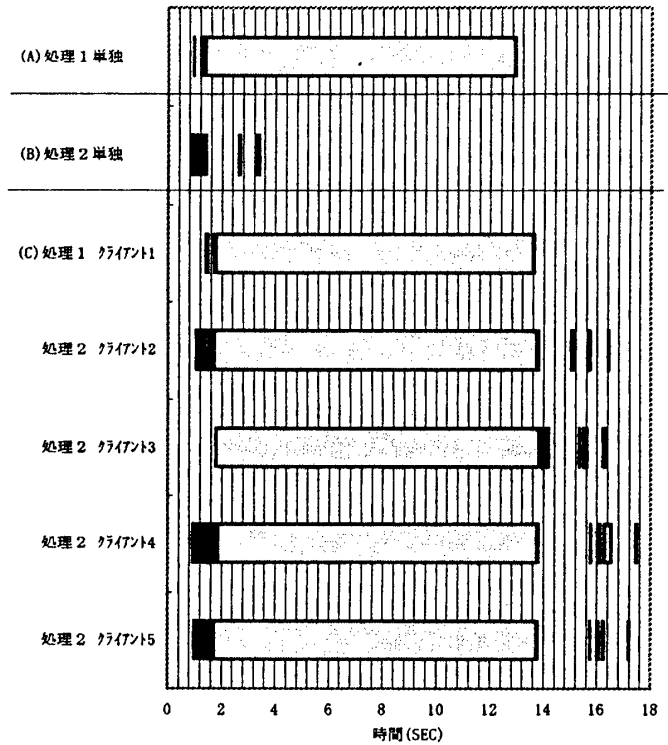


図 4: 実験によるタイムチャート

3. 測定実験

実際に市販 DBMS を使用して、実験を行なった。実験システムの構成図を図 3 に示す。実験ではデータベースサーバマシン 1 台に対して 5 台のクライアントマシンを用意し、クライアントアプリケーションはデータベース検索 (Select) を行なう 2 種類のアプリケーション (曖昧検索を行なうサーバ負荷の高いアプリケーション (処理 1) とサーバ負荷の低い検索を複数回行なうアプリケーション (処理 2)) で、それぞれのアプリケーションをクライアント台数 1~5 台と順次増加させ、同時実行を行なった場合のタイムチャートと、処理 1 を 1 台と処理 2 を 1~4 台と順次増加させ、同時に実行した時のタイムチャートを作成した。

(A) 処理 1 のクライアント 1 台単独実行、(B) 処理 2 のクライアント 1 台単独実行と、(C) 処理 1 をクライアント 1 台、処理 2 をクライアント 4 台で同時実行した時のタイムチャートを図 4 に示す。この図で網掛の部分はサーバ待ち時間で、白い部分はクライアント稼働時間である。

サーバ挙動推測の一例として、例えばこのタイムチャートから、図 4(B) で瞬時に終了していた処理 2 のサーバ間い合わせが、(C) では処理 1 の負荷の高いデータベース要求に影響されることによって待ち時間が非常に増加していることが観察される。この様子から実験を行なったデータベースはあるクライアントからの要求に対して、

その要求処理を終了するまで、他のクライアントからの処理を待たせる場合があるということが推測される。

4. 当方式の利点

サーバプログラムの応答時間をモニタする方式にはクライアントアプリケーションにタイムスタンプ関数を埋め込む方式や、サーバに管理用のプログラムを実行させる方式等がある。以下にこれら他の方式と比較した場合の本方式の利点をあげる。

- クライアント/サーバマシンに余分な測定負荷のオーバーヘッドがない。
- ネットワークに対して余分な測定負荷がない。
- アプリケーションプログラムの変更が必要ない。
- サーバへの要求を行なうために使用する API 関数が、その関数中でさらに複数の要求を発していた場合にも挙動をモニタすることができる。例えば 698 タイムスタンプを埋め込む方式では API 単位でしか情報を得ることができない。

5. おわりに

今回市販 DBMS に対するクライアントアプリケーションからの同時要求実行時の挙動推測を、クライアント/サーバ間の通信パケットを LAN アナライザによってキャプチャし、タイムチャートを作成することにより、サーバプログラムの挙動推測を行なうことができた。