

クライアント／サーバ・システムの性能設計

大前 義次、鄒 林、堀 義人、筒井 麻喜
神奈川工科大学 情報工学科

あらまし

ワークステーションの処理能力は年々向上し、またそのグラフィカル・ユーザ・インタフェースは益々便利になっている。このため、これまでのメインフレームによる集中処理形態からワークステーションをネットワークでつないで分散処理を行なう形態へと急速に移行が進んでいる。これがクライアント／サーバ・アーキテクチャといわれるものである。これによってシステムの柔軟性が増えるとともに、情報化のコスト・パフォーマンスが向上すると期待されている。しかしながら、クライアント／サーバ・コンピューティングは新しく出現したテクノロジーであり、その設計上の諸問題は必ずしも明らかにされているとはいえない。

本研究は、OLTPを対象に、クライアント／サーバ・システムの性能特性を解明し、性能の最適化について明らかにしようとするものである。

Optimization of Performance for Client/Server Systems

Yoshitugu OHMAE, Lin ZOU, Yoshihito HORI, Maki TSUTSUI
Dept. of Information and Computer Sciences, Kanagawa Institute of Technology

Abstract

Workstations are becoming more powerful on computational capabilities, and more useful on graphical user interfaces. Networking capabilities allow for several workstations to be interconnected. The flexibility and cost-performance advantage are expected by this new paradigm for distributed computation, referred to as the client/server architectures.

Many companies have led to move considerable portions of their operations from the traditional mainframe - centered processing to the client/server computing. However, the study in this field is still young.

It is hard to say that a knowledge on the design of client/server architecture is sufficient.

We would like to explain the properties on optimization of performance for client/server systems from the point of view of an OLTP application.

1. まえがき

情報システムは、これまでのメインフレーム中心の集中処理形態から分散処理形態へと急速に移行が進んでいる。このような流れの中でクライアント/サーバ・アーキテクチャが新たに出現した。これによって、システムの巨大化に伴って急増する情報化投資を抑制し、またユーザ環境に、きめこまかに適合した情報処理機能の提供が期待されている。しかしシステム開発上、あるいは運用上、必ずしも明らかになっていない部分があり、設計ノウハウの蓄積が十分とはいえない。

クライアント/サーバ・システムは、フロントエンド処理を行なうクライアント側とバックエンド処理を行なうサーバ側とで負荷分担が行なわれる。さらに両者を結ぶ LAN 等のネットワークからなる。このようなシステムの性能上の最適化問題が新たな課題として提起されている。

本研究は、分散型 OLTP を対象に、ソフトウェア性能、並びにシステム性能に関して、その評価と最適化について検討を行なうものである。

2. クライアント/サーバ・システムの性能の最適化

クライアント/サーバ・システムの性能の最適化に当たって、最初に実施すべきことは、システムの性能を評価して、アプリケーション実行上のボトルネックを明らかにすることである。ボトルネックは、各リソースが容量一杯まで使われるときに発生する。それはソフトウェアであったり、CPU であったり、DISK であったり、あるいは LAN であったりする。これらの改善対策としては、ボトルネックを形成しているリソースのワークロードを減らしたり、リソースの処理能力を高めたりすることである。例えばサーバの処理方式を変えたり、新しいサーバを追加したり、アプリケーション処理の負荷をクライアントやサーバ間で再分配したりすることが考えられる。

3. クライアント/サーバ・システムの基本的機能

クライアント/サーバ・システムの特徴は、クライアント・プロセスとサーバ・プロセスをネッ

トワーク経由で別個のマシン上で実行することである。これを図 3.1 に示す。

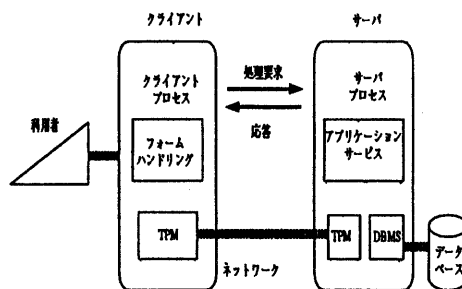


図 3.1 クライアント/サーバ・システム [2][4]

ここでクライアントは、入力データのチェックを行ない、処理しやすいデータ形式に編集し、サーバへ転送する。サーバからの処理結果の応答は再び見やすい形に処理した上で、ディスプレイに表示する。これは、フォームハンドリングサービスである。サーバは、クライアントから送られてきた処理要求をアプリケーションサービスに渡し、処理を実行し、その応答をクライアントに返すためにデータベース管理システムや、各種のリソースマネージャと組み合わせて構成されるモジュールである。さらにサーバは、クライアントの要求を満たすために、それ自身がクライアントになり、要求を他のサーバに送るといったネスティング構造を形作ることも多い。

分散 OLTP アプリケーションを処理するうえで分散トランザクション処理モニタ (TPM)[1][4] が重要である。TPM の基本的機能として、ルーティング機能とチューニング機能がある。

前者はクライアントからの処理要求を、メッセージ内容に基づいて、適切なそれぞれのサーバに割り当てる機能である。これはデータ依存型ルーティング [1] と呼ばれるものである。いったん、このルーティングが実行されると、以後の実行制御はタスク単位で行なわれる。この機能はこれまでのメインフレーム OS のディスパッチャの役割に相当している。

チューニングはアプリケーションをより効率的に処理するために行なうものである。クライアントからの処理要求は、それぞれのサーバの待ち行

表 3.1 ディスバッチ・タスク数を増やした場合

タスク数	1	2	3	4	5	6	7	8	9	10
レスポンスタイム [秒]	46.1	17.9	11.3	9.35	8.69	8.30	8.23	8.26	8.28	8.28
スループット [件/秒]	0.18	0.36	0.47	0.52	0.53	0.54	0.55	0.55	0.55	0.55
タスク平均待ち時間 [秒]	39.9	12.3	5.52	2.43	1.08	0.37	0.09	0.02	0	0
タスク利用率	1	1	0.96	0.90	0.81	0.72	0.64	0.56	0.50	0.45
サーバ平均利用率	0.23	0.44	0.57	0.62	0.65	0.66	0.66	0.66	0.66	0.66

表 3.2 高速サーバに改善した場合

サービス時間 [秒] (1/性能倍率)	5 (1/1)	2.5 (1/2)	1.67 (1/3)	1.25 (1/4)	1 (1/5)	0.625 (1/8)	0.5 (1/10)
レスポンスタイム [秒]	46.1	19.3	11.1	7.24	5.1	2.41	1.75
スループット [件/秒]	0.18	0.34	0.46	0.58	0.66	0.81	0.85
タスク平均待ち時間 [秒]	39.9	16.34	9.01	5.53	3.62	1.29	0.75
タスク利用率	1	1	1	0.99	0.97	0.90	0.85
サーバ平均利用率	0.23	0.21	0.19	0.17	0.16	0.13	0.11

列（キュー）に入れられる。サービスを提供するサーバが複数ある場合、TPM は自ら保持するシステム統計（CPU 利用率、待ち行列長等各サーバ情報）に基づいて、最も負荷の軽いサーバを選ぶことによって効率化を図る。これをロードバランシング（負荷均衡制御）という。[1]

このような機能は、これまでのメインフレーム OS のスケジューラの役割に相当している。

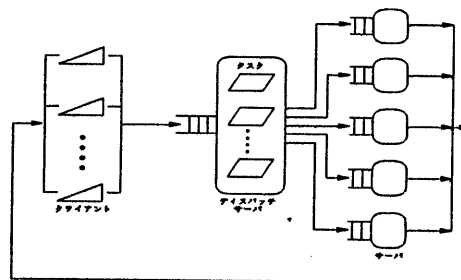


図 4.1 ルーティング機能の評価モデル

4. ソフトウェア性能設計

4.1 ソフトウェア・ボトルネック [3]

4.1.1 モデルの記述

システムの特性をルーティング機能に関して評価するモデルについて、以下考察する。クライアントからの処理要求が、サーバに到着すると、まずタスクキューに並ぶ。タスクが空きになると、タスクを取得し、それぞれの処理要求に対応したサーバプロセスが割り当てられる。もしサーバプロセスが処理中であれば、それぞれのサーバキューで待ち合わせた後に、そのサーバプロセスで処理が実行される。処理が終了し、応答が返されると、タスクは解放される。なお、それぞれのタスクは各処理要求を並行的に走らせるものである。このモデルでは、ディスバッチタスクを走らせるタスクサーバを仮想している。

4.1.2 性能特性

システムモデルに基づいて、シミュレーションを行なった結果を表 3.1 に示す。なお、入力パラメータは、クライアント 10 台、思考時間 10 秒、ディスバッチ時間 0.5 秒、サーバサービス時間 5 秒（指数分布）である。これから明らかなようにタスクを増やすにつれて、レスポンスタイム・スループット共に急速に改善される。このことからこの場合のシステム性能のボトルネックは、ソフトウェア・ボトルネックであることが明らかである。さらに、サーバのハードウェア性能を改善したときの効果をシミュレーションで検証する。タスク数 1 個の場合についてサーバの処理能力を 2 倍～10 倍まで変化させて見る。その結果は表 3.2 のようになる。表 3.1 と表 3.2 を比較してみると、サーバの処理速度を変えないで、タスク数を 6 個以上にした場合と、タスク数 1 個でサーバ処理速

度を4倍にした場合とレスポンスタイムもスループットもほぼ同等の値になることが分かる。以上の検討を通してソフトウェア・ボトルネックの発見とその解消による効果が明らかになった。しかし、表3.1から明らかなようにタスク数改善の効果には自ら限界がある。それを越える大幅な改善はサーバ性能の改善に待たなければならない。

4.2 ロードバランシング

この場合のシステム性能の評価モデルについて以下述べる。いま同種サービスを提供する複数のサーバがあるものとする。クライアントからの処理要求がサーバに到着すると、スケジューリング・サーバ（仮想）が一括管理して各サーバに振り分けるものとする。次の各方式のモデルを考える。

- (1) 各サーバの前にサーバキューがあり、ラウンドロビン (RR) で処理を実行する方式
- (2) サーバキューの待ち行列長の最短であるサーバに処理要求を渡す方式
- (3) サーバ利用率の最低のサーバに処理要求を渡す方式
- (4) 優先権関数評価値 (EV) の最大のサーバに優先的に処理要求を渡す方式

$$EV = K_1 / \text{待ち行列長} + K_2 / \text{平均利用率}$$

ここで、 K_1, K_2 は基準化重み係数

- (5) 各サーバに個別にキューを設けなくて、複数サーバが単一のキューを共有する方式 (MSSQ)

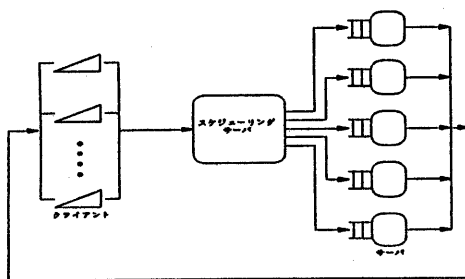


図 4.2 チューニング機能の評価モデル

4.2.1 性能特性

1. サーバキューのある場合

シミュレーションの結果は、図 4.3, 図 4.4 に示すようになる。なお入力パラメータは、サーバ台数 5 台、クライアント台数 50 台とし、思考時間 10 秒、サーバ・サービス時間 1.5 秒 (超指数分布) である。ここでレスポンスタイム並びにスループット共に (4) の方式が最もロードバランスがはかられ、性能が向上することが分かる。(2) の方式ではシステム負荷が増大するにつれて効果が顕著となるが、負荷が軽い場合にはあまり効果がない。これに対して (3) の方式では、反対に負荷の軽い場合効果があり、高負荷の場合、それほど効果的ではないことが分かる。これら両者の長所を取り入れた (4) が優れていることが分かる。なお、この場合、重み係数が等しいとき、最適化がはかれることが分かった。

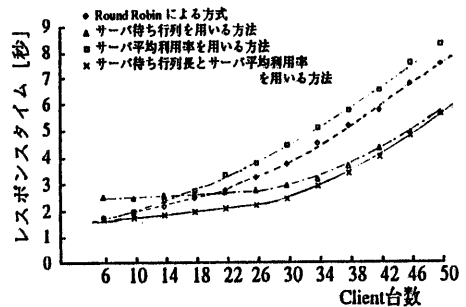


図 4.3 レスポンスタイムの比較

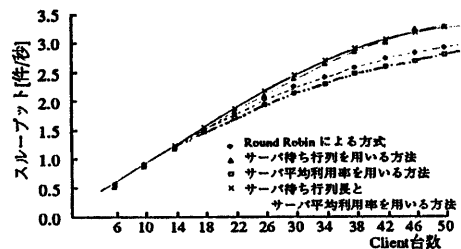


図 4.4 スループットの比較

2. MSSQ の場合

入力パラメータとして、サーバキューのある場合と同じ値で行った結果を、図 4.5、図 4.6 に示す。サーバキューのあるいずれの方式と比較しても、この場合の方が性能の向上がみられる。

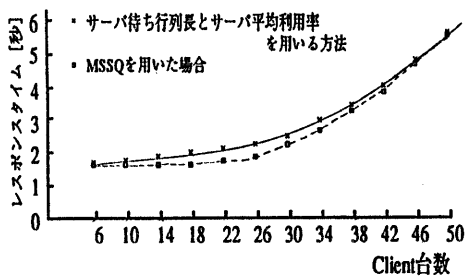


図 4.5 レスポンスタイムの比較

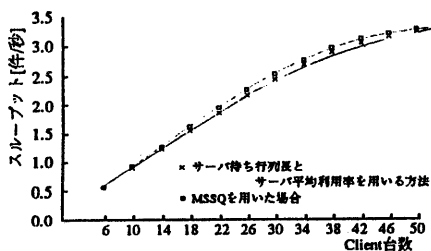


図 4.6 スループット

5. システム性能設計

5.1 クライアント／サーバ・システムのネスティング構造

クライアントとサーバは必ずしも固定のものではなく、必要に応じてその役割を演ずるという特徴を持っている。サーバは時としてクライアントにもなりうる。処理の必要に応じて、ネットワーク経由で次々にネスティングが行なわれる。このことから、クライアント／サーバ・システムにおいては、これまでの集中型と異なる新たな性能評価問題を提起している。

5.2 クライアント／サーバ・システムの性能設計

5.2.1 モデルの記述

具体的なクライアント／サーバ・システムについて考える。ここでは大規模な特許事務所のシステム例 [2] を取り上げる。客が特許申請書を事務所

に持参すると、クライアントである Scan Station によってスキャンされ、サーバである Scan Server の光ディスクに蓄積される (Scanning LAN Subsystem)。この情報は、さらに Central LAN 経由でイメージファイルに格納され、またデータベースにイメージのインデックスが登録される。特許審査者は、申請書をイメージファイルから取り出し、処理し、審査用ファイルに格納する (Examination LAN Subsystem)。処理済みの情報は印刷係に廻され、特許申請受付一覧として印刷される。 (Publishing LAN Subsystem)。これら各サブシステムは、相互に関連していて、いずれからでも中央のイメージディスクやデータベースとやり取り出来るようになっている。

以下の検討では、いま述べたうちの Scanning LAN サブシステムのみを取り上げて性能設計を行なう。この場合のシステムモデルを図 5.1 に示す。ここでの作業は、客が持参した特許申請書をワークステーション (Scan Station) のスキャナーで入力して TCP/IP によるイーサネット LAN 経由で Scan Server に送る。Scan Server は情報を圧縮処理して、イメージ情報をいったん Scan Server の光ディスクに蓄積する。このモデルでは、クライアントは Scan Station であり、サーバは Scan Server である。しかし Scan Server は、Image Server や Database Server にとってはクライアントの役割を演じるものである。

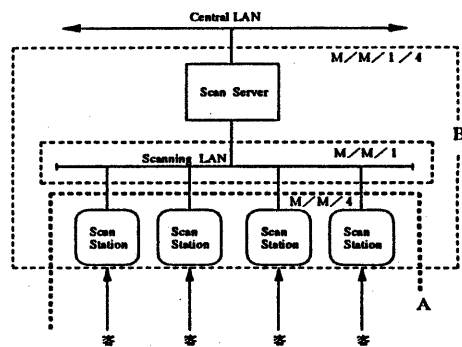


図 5.1 Scanning LAN サブシステム・モデル

5.2.2 システム構成

図 5.1 に示すようにこのシステムは、次の二つの部分から構成されると考えられる。

- (1) 客と Scan Station: これは待ち行列理論の OQN でモデル化される。
- (2) Scan Station と Scan Server、Scan Station と Scan Server 間: これは CQN でモデル化される。

5.2.3 性能特性

シミュレーション結果を図 5.2, 図 5.3 に示す。なお入力パラメータは Scan Station の処理率は 0.2 ページ/秒、Scanning LAN のサービス率は 3.3 ページ/秒、Scan Server のサービス率は 2.5 ページ/秒、思考時間は 5 秒とする。図 5.2 は構成要素別の利用率を示している。ここで Scan station のみが急速に利用率 90% に達していることが分かる。しかし他の Scan Server や Scanning LAN はまだ十分余裕がある。この場合、Scan Station がボトルネックを形成している。申請受付のレスポンスタイムも急激に増大している。この場合、Scan Station の台数をさらに増やすことによってシステム性能向上がさらに改善できることが考えられる。これをレスポンスタイムで示したものが図 5.3 である。図 5.3 の中で・印で表示したものは、待ち行列理論 (図 5.1 の A の部分は M/M/4、B の部分は CQN の MVA で計算) による。

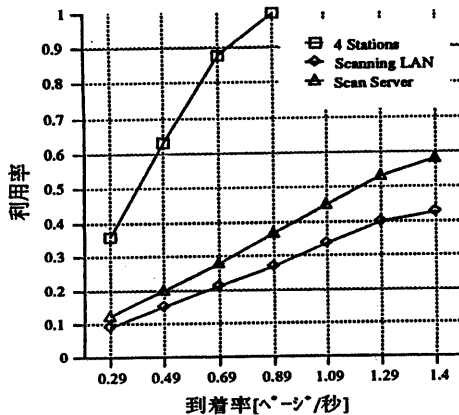


図 5.2 構成要素別利用率

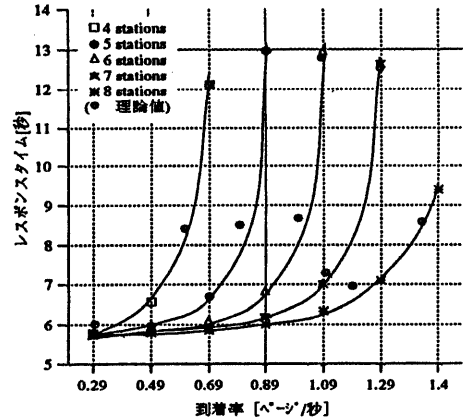


図 5.3 ステーション数とレスポンスタイム

6. むすび

クライアント/サーバ・システムの性能評価として、4章の検討を通してソフトウェア・ボトルネックやシステム制御アルゴリズムの性能に及ぼす影響等が明らかになった。

クライアント/サーバ・システムでは、クライアントとサーバは固定のものではなく、処理環境によって相互にその役割を演ずるものである。このことから、ネットワークを通して多くのネスティングがみられる。このような場合の性能設計のあり方についても5章のシステム性能設計の研究を通してそれを明らかにすることができた。

参考文献

- [1] 岩宮 好宏 他. “分散トランザクション処理”. 株式会社 リックテレコム (1994).
- [2] Paul E. Renaud. “INTRODUCTION TO CLIENT/SERVER SYSTEMS”. John Wiley & Sons Inc.(1993).
- [3] C. Murray Woodside. “Performance Engineering of Client-Server System” Performance Evaluation of Computer and Communication Systems. Joint Thrtorial Paper of Performance '93 and Sigmetrics '93, Springer-Verlag, p.221-231 (1994).
- [4] 渡辺 栄一 訳, J. グレイ著. “OLTP システム [増補版]”. マグロウヒル, p.221-231 (1994).