

巨大UNIXシステムでのネットワークパフォーマンス

山賀 正人

千葉大学総合情報処理センター

千葉県千葉市稲毛区弥生町 1-33

043-290-3544 (Fax 兼), yamaga@ipc.chiba-u.ac.jp

概要

授業のような同時に多数のクライアントが WWW サーバにアクセスする際に最も良いパフォーマンスを得るためのプロキシサーバの設定を調べる。

キーワード

プロキシサーバ、マルチプロセッサ、squid

1. はじめに

現在、インターネットを利用したサービスの中で World Wide Web(WWW) は電子メールに匹敵する、また場合によってはそれ以上に重要かつ普及度の高いものになった。そのような中で大学などのような利用者数の多い組織では、WWW へのアクセスがサーバやネットワークに多大な負担をかけてしまうという問題が生じる。この問題を解決する手段として、アクセス頻度の高い外部データを定期的に取得して保存するミラーリングの機構や、一度アクセスした外部データを自組織のサーバに短期間保存するキャッシングの機構が開発され、既に実用化されている。

しかし、このキャッシング用のサーバ(プロキシサーバ)の運用、特に各種パラメータの設定は試行錯誤に頼っている場合が多い。その最大の原因は、WWW の普及があまりに急速であったため運用の経験が蓄積されておらず、例えばこのようなシステム・ネットワーク構成の場合はどのような設定が最適であるかを系統的に示すものがないということである。

また本学では 100 人単位の情報リテラシの授業がほぼ全学必修の形で行なわれており、その授業において WWW を利用する際に 100 人同時の外部データへのアクセスが内部のネットワークやプロキシサーバに多大な負荷を与え、講義時間を大量にロスしてしまうという深刻な問題を生んでいる。

このような背景のもとで本発表では、上記教育システムを用いて授業中のような同時大量アクセスを行なう場合にどのような設定が最高のパフォーマンスを与えるかを調べた実験結果を発表する。特に注目すべき点は本システムのサーバが CPU を 28 個有するマルチプロセッサのマシンであるということ、また本来シングル CPU で動作させることを前提に作られているサーバプログラムをどのような設定で運用すれば、このマルチプロセッサの仕組みを有効に生かせるかを調べた点である。

2. 実験

今回の実験で用いた本学の教育システムは以下のような構成になっている

プロキシサーバ・ファイルサーバ CRAY スーパーサーバ CS6400 1 台

- ・ 60 MHz SuperSPARC × 28
- ・ 1.8 GByte メモリ
- ・ 270 GByte ハードディスク

X クライアントワークステーション

CRAY エントリーレベルスーパーコンピュータ EL92 8 台

- ・ 133 MFLOPS × 2
- ・ 512 MByte メモリ
- ・ 21 GByte ハードディスク

X 端末 230 台

これらが 10BaseT と FDDI によってネットワーク接続されている。図 1 に構成を示す。

実験方法として、ある時刻に EL92 上でブラウザ (lynx) が一斉に起動して、特定の URL にアクセスするように EL92 上の cron に設定し、プロキシサーバがクライアントから URL を受け取ってからデータを全てのクライアントに渡し終わるまでの実時間をログから調べるという方法を取った。もちろん実験の際には純粋に本実験のデータのみが取り出せるように利用者などの全くいない時間帯に行ない、全てのマシンは NTP (Network Time Protocol) を使って正確に時刻を合わせてある。

今回の実験ではプロキシサーバのプログラムに squid を用いた。以前、本システムでは DeleGate を利用していたのだが、DeleGate ではキャッシュデータやログをそれぞれ単一のディレクトリやファイルで管理するため、同時に大量のアクセスが行われるとディスクへの IO がボトルネックとなり、クライアントのうちのいくつかは目的のデータを取得できないことがあった。そこで、ポート番号を変えて複数のプロキシサーバを立ち上げ、各 EL92 に対してそれぞれ処理を分担させる方法を考えた。もちろん、キャッシュデータやログを蓄えるディスクはそれぞれ SCSI ユニットを変え、ディスクへの負荷が分散するようにした。また複数のサーバプログラムが OS の仕組みとして別々の CPU に割り振られる。しかし、複数のサーバがそれぞれ同じデータを取得しに行ってしまうは無駄である。できればキャッシュデータを共有しあった方がよい。そこで、プロキシサーバ間でキャッシュデータを共有し合う機能を有する squid を採用したのである。

今回はクライアントの数、一斉にアクセスする URL の数、プロキシサーバの数をそれぞれ変えて実験を行なった。ただし、実時間を測定しているので、同じパラメータの実験を複数回 (4 回) 行ない、その平均値を取ることにした。

Table 1: 実験結果

- ・ クライアント数 104 (= 13 クライアント × 8)
- ・ 転送データ 76.7Kbyte

		URL 数				
		1	4	8	13	26
サーバ の数	1	7.759	8.033	8.168	8.198	8.238
	4	8.552	8.643	8.411	8.701	8.939
	8	8.383	8.569	8.786	8.578	8.727
	16	8.692	8.716	9.149	8.784	9.078

3. 結果

実験結果は Table 1 である。しかしこの実験結果は著者の予想を裏切るものであった。著者の予想は次のようである。URL の数が一つの場合は、プロキシサーバの数が多いとサーバ間の通信がネックとなるので、プロキシサーバも一つの方がよいであろう。一方、URL が複数になれば、それに伴ってプロキシサーバも複数であれば良いだろう。しかし、もちろんプロキシサーバの数が多すぎれば URL やクライアントの数に関わらず、サーバ間の通信がネックとなって良いパフォーマンスを得ることはできないだろう。つまり処理の負荷分散とサーバ間の通信負荷とのバランスで最適な値が分かるのではないかと期待していたのである。ところが実際には Table 1 に見られるように値にばらつきがあり、規則性がほとんど見られない。

なぜこのような結果になったのか理由として考えられる点は以下である。

- 実験した URL 数やサーバ数の項目数が少なかったこと
- DNS への問い合わせ(子プロセス dnsserver)にかかる時間にばらつきがあったこと
- 複数の URL が実は同じ WWW サーバ上の別データという特殊なものであったこと

このような結果をふまえ、発表ではその後の実験結果を発表する予定である。

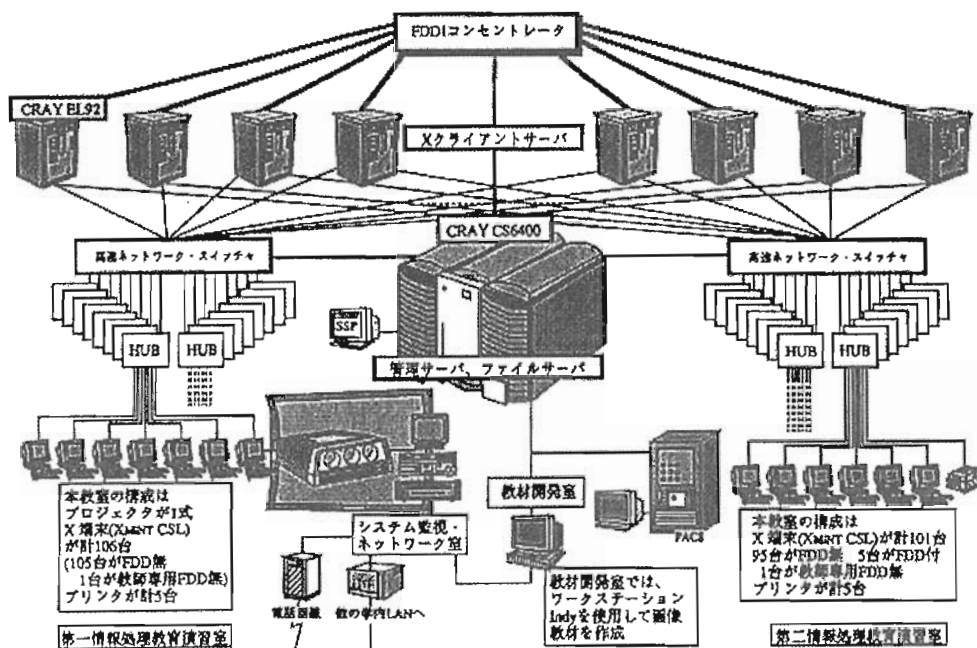


図1: システム構成図