

ミラーサーバへの自動転送機能を有する 分散 WWW サーバシステム

1 F-4

杉山正一

高井昌彰 山本強

北海道大学大学院工学研究科

北海道大学大型計算機センター

1 はじめに

インターネットの急速な発展に伴い、WWWサーバアクセスの高速化が重要な課題となっている[1]。その解決方法の1つとして、ミラーサーバを提供する方法がある。しかし、ミラーサーバを用いる方法では、どのミラーサーバを利用するかをクライアント側で選択する必要がある。また、どのミラーサーバが利用されるかはクライアント任せなので、特定のミラーサーバにアクセスが集中してしまい、サーバ側が想定していた負荷分散効果が得られない可能性がある。

本研究では、メインサーバと複数のミラーサーバからなる分散WWWサーバ群を用いる際に、クライアントを自動的に負荷の軽いWWWサーバに転送することで、負荷分散を実現するシステムを提案する。これによりサーバ側の能動的な負荷分散が実現でき、クライアント側は意識せずにミラーサーバを利用できるというメリットが期待できる。

2 システムの構成

分散WWWサーバシステム内の各サーバは、スケジューラ、ファイル管理デーモン、WWWサーバデーモン(以下httpd)から構成される。図1に2つのサーバからなるシステムの構成を示す。

スケジューラはクライアントからの接続を受け取り、各サーバのCPU使用率やTCPセッション数などを考慮して、どのサーバにクライアントを転送するかを決定する。スケジューラの動作については3章で詳しく述べる。

ファイル管理デーモンは、メインサーバからミラーサーバへファイルコピーを行い、必要に

じてリンクの書き換えを行う。ファイル管理デーモンの動作については4章で詳しく述べる。

httpdはスケジューラから転送されてきたクライアントの要求に対してHTTPに基づいた処理を行う。httpdは一般に普及している物をそのまま使用することができる。通常httpdはポート番号80を開きクライアントの接続を待つが、本システムでは、このポートをスケジューラが使用するので、httpdは他のポートを使用する。

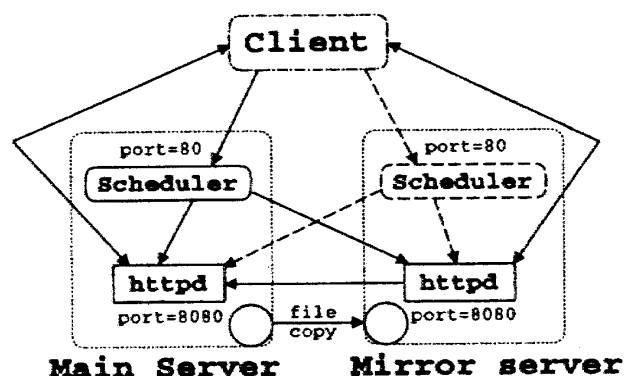


Fig. 1 システムの構成

3 スケジューラの動作

スケジューラはHTTPのウェルノウンポートであるポート番号80を開き、クライアントからの接続を待つ。クライアントからのHTTPリクエストを受け取ると、スケジューラは、どのファイルが要求されているかを解析する。メインサーバにしか存在しないファイルが要求された場合は、メインサーバ上のhttpdにクライアントの要求を転送する。ミラーサーバにも存在するファイルが要求された場合は、スケジューリングポリシーに従ってクライアントを転送するサーバを決定する。

スケジューリングポリシーには、CPU使用率やTCPセッション数などの負荷が一番軽いサーバに転送する方法と、ラウンドロビンで転送する方法があり、サーバ管理側で選択できる。

どちらの場合でも、クライアントが最初にサーバ群に接続してきたときにのみスケジューラが接続を受け取るが、その後、クライアントは転送先サーバの httpd と直接 HTTP のやり取りを行い、スケジューラはクライアントと httpd とのやり取りには一切関与しない。このような方法を取る理由の1つは、すべての接続にスケジューラが関与すると、スケジューラを通して処理を行うことによるオーバーヘッドが大きくなるためである。もう1つの理由は、同じサーバに接続している間の負荷は、それほど変化しないことが推測されるからである。

クライアントの要求をスケジューラによって決定されたサーバに転送する方法として、HTTP の Location ヘッダを用いる。Location ヘッダは、"Location: URL" という形式を取る。クライアントの要求を受け取ったスケジューラは、Location ヘッダの URL 部に転送先サーバ上のファイルを指定し、クライアントに対して HTTP レスポンスを返す。この HTTP レスポンスを受け取ったクライアントプログラムが、Location ヘッダで指定されている URL に対して、スケジューラに送った要求と同じ物を自動的に送ることを利用して、要求の転送を行なう。この過程は、クライアントにはほとんど意識されずに行われる。図2にスケジューラが返す HTTP レスポンスの例を示す。

```
HTTP/1.0 302 Moved Temporarily
Date: Tue, 17 Mar 1998 14:00:00 GMT
Server: MasakazuServer/0.4698
Location: http://www.foo.ac.jp:8080/times.html
Content-Type: text/html
```

Fig. 2 HTTP レスポンスの例

4 ファイル一貫性の保持

単純にすべてのファイルをメインサーバからミラーサーバに転送することはできない。なぜなら、CGIなどによって同じ名前のファイルに2ヶ所以上で同時に書き込みが起こる可能性があるからである。このような場合、メインサーバからファイルがコピーされると、ミラーサーバのファイルに対する変更は失われてしまう。解決法として、すべてのサーバで同期を取りながらファイルの変更を監視する方法が考えられるが、サーバ間で頻繁に通信が起こるため、効率的でない。

本システムでは以下のようなクラス分けを行うことで、ファイルの一貫性を保持する。

- コピー可能クラス
- コピー不可能クラス
- カウンタクラス

コピー可能クラスに属するファイルは、メインサーバでファイルに変更が加えられたときに、ファイルがミラーサーバへとコピーされる。ミラーサーバ上にあるファイルは上書きされてしまうので、同時に複数のサーバでファイルの変更が起こりうるファイルは、このクラスには含めない。

コピー不可能クラスに属するファイルは、ミラーサーバにはコピーされないため、複数のサーバでファイルの変更が起こる可能性はない。しかし、ミラーサーバでこのクラスのファイルが要求されたときに、ミラーサーバ上には要求されたファイルが存在しないという問題が起こり得る。解決法として、このクラスに属するファイルへのリンクを含むファイルをミラーサーバに置く際に、リンク先のアドレスをメインサーバ上のファイルを指す絶対アドレスに書き換えることで、メインサーバ上のファイルにアクセスさせる方法を取る。このクラスに属するファイルの例としては、チャットや掲示板のログファイルが挙げられる。

カウンタクラスに属するファイルは、一定時間ごとにメインサーバに集められ、カウンタの集計を行って、再び各サーバに戻される。

ファイルのデフォルトのクラスはコピー可能クラスとする。他のクラスにする場合は、ファイル作成者にファイル名を宣言してもらい、ファイル管理デーモンがクラスに応じた処理を行う。

5 まとめ

本研究では、メインサーバと複数のミラーサーバからなる WWW サーバ群に対して、自動的に負荷分散を実現するシステムを提案した。現在、負荷分散の効果を検証する実験システムの実装を行っている。

参考文献

- [1] Azer Bestavros, "WWW Traffic Reduction and Load Balancing through Server-Based Caching", *IEEE Concurrency*, Vol.5, No.1, pp.56-67, 1997