

複数のミラーサーバを用いた Web アクセスの最適化に向けて

02T209 岩佐年倫（最所研究室）

本研究では、複数のミラーサーバに同時アクセスする際のサーバへの振り分け最適化およびサーバ高負荷時のアクセス制限を行うために必要なパラメータを得るための基礎的な実験を行った。さらに、その結果に基づいてサーバへの振り分けの最適化について検討した。

1. はじめに

現在、インターネットの広い普及に伴い、多くの Web サイトが生まれ、情報の早期伝達が可能となり、いつでも・どこでも・だれとでも情報をやり取りできる便利なユビキタス社会となっている。しかし、その一方でインターネット人口増加によるネットワーク負荷の増大やメンテナンスに伴うサービスの低下が問題となっている。このため、大手企業の人気サーバや検索エンジンでは、同一のサービスを行うミラーサーバを複数設置することで負荷分散を行っている。

このミラーサーバを効率的に利用するためのプロキシプログラムの研究を我々の研究室では行っており、既に同時アクセスが有効であることが分かっている。そこで本研究では、最適な割り付けの際の判断基準について、基礎的な実験を行い検討する。

2. ミラーサーバへの同時アクセス

Web ブラウザから HTTP リクエストを受け取り、それを Web サーバへ送る中継の役割を果たすのがプロキシである。同一サーバに対する複数のリクエストをプロキシでミラーサーバ群へ同時に振り分けて送ることにより、並列アクセスが可能となり、全体のアクセス時間を短縮できる [1]。リクエストをどのように振り分けるか、負荷軽減をどのようにするのかについての議論が必要である。

本研究では図 1 に示すように、リクエストのサイズによってサーバを選択する方法を検討する。あらかじめリソースサイズを取得しておき、サイズの大きいものは早いサーバへ振り分け、リソースが大容量の場合は、分割して取得する。リソースサイズを取得する方法には、GET メソッドと HEAD メソッドの 2 つの方法があり、状況に応じて使い分ける必要があり、それを検討するための実験を行う。

また、本論文では HTTP/1.1 [2] で拡張された機能である Range ヘッダフィールド、持続的接続、HTTP パイプライン機能を用いて最適化を図る。
○ Range・・・GET に付加することで、取得範囲を指定することができる。

- 持続的接続・・・TCP コネクションをクローズしないで、次のリクエストを処理できる。
- HTTP パイプライン・・・複数のリクエストを応答を待つことなく連続してサーバへ送る。

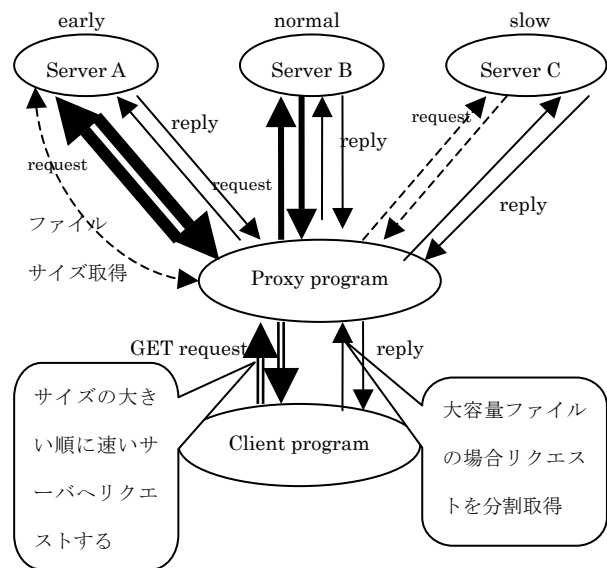


図 1 リソースサイズ別の振り分け接続

3. リソース情報の取得に関する実験

本実験では、GET と HEAD を用いてネットワーク距離の異なるサーバに対する応答時間差を調査する。対象ホストは、同一 LAN 内(早い)、同一キャンパス内(少し早い)、外部(遅い)である。クライアントとして、OS : Windows XP、速度:100Mbps、CPU : Intel Celeron(R)2.66GHz、メモリ:512MByte の PC を用いた。測定結果を図 2, 3, 4 に示す。

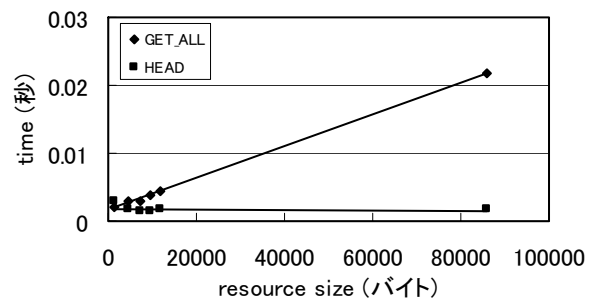


図2 同一 LAN 内

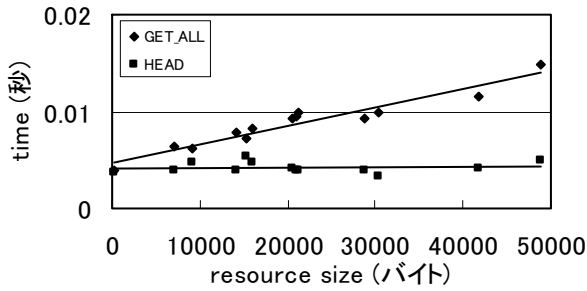


図3 同一キャンパス内

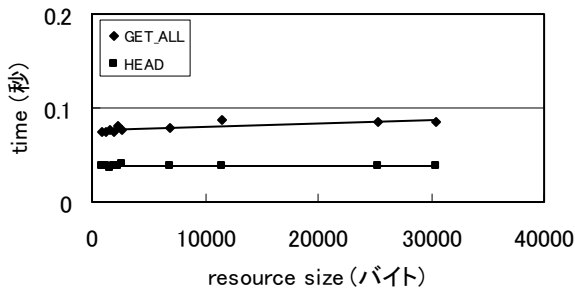


図4 外部

3つのグラフから TCP のオーバーヘッドが大きいことが見て取れる。パイプライン処理により TCP オーバーヘッドの影響を小さくできれば、HEAD を用いてサイズを取得し最適な割り当てを行うことも考えられる。

4. Web アクセス最適化の検討

実験結果を踏まえ、下記に示す方法を検討する。

◆ リソースのサイズで判断(図5)

Web ページを表示するときに、複数のリクエストのヘッダを HEAD でまとめて取得して、処理する方法。図5のように、リソースサイズがある大きさ(D)より小さい場合まとめて処理し、大きい場合分割取得する。

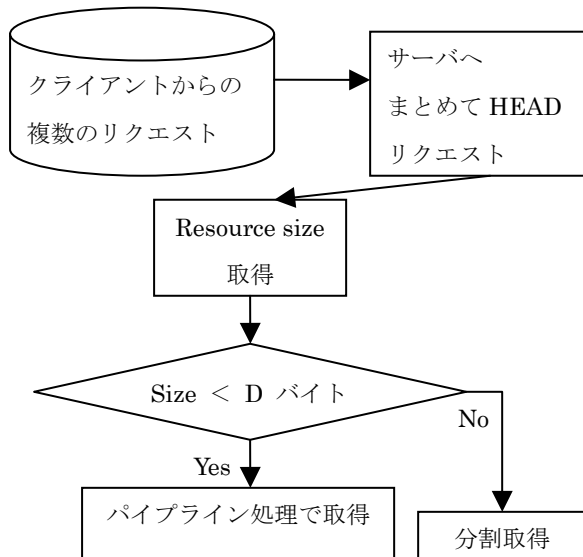


図5 リソースサイズで判断

◆ サーバの性能で判断

リクエスト量をサーバ性能に応じて割り付ける。たとえば、以下の手順で割り付ける。

- ① 複数の情報を HEAD でまとめて取得
- ② リクエストをサイズの昇順にする
- ③ サーバへ割り付ける
 - ・ 早いサーバ ← D バイト
 - ・ 少し早いサーバ ← 3/4 D バイト
 - ・ 遅いサーバ ← 1/4 D バイト
- ④ サイズが D バイト以上なら分割し③の処理
- ⑤ クライアントへ応答

◆ 拡張子で判断

Web ページに存在するリソースタイプには、主にテキスト・画像・動画ファイルなどがある。これらのメディアタイプに応じて、サイズを予測してリクエストをサーバへ振り分けることで効率的にリソースを取得する。

◆ 分割取得

サーバ間での取得時間の差が小さい場合 D バイトを取得する時間と D/2 バイトを取得する時間の差が HEAD でサイズを取得する時間より長ければ分割取得により全体の取得時間を短縮できる。このため各サーバにおけるリソースの大きさと取得時間の情報を保持する必要がある。リソースが D バイト以上なら分割取得する。

◆ 応答が遅いサーバ

ネットワーク状況は常に変動しており、突然サーバが極端に遅くなったり停止したりすることがある。これに対処するために以下のようなアクセス制限を行うことが考えられる。

- ・ いったん止める→再開の条件が必要
- ・ リクエスト数を半分に減らす
- ・ アクセス先の自動切り替え

5. まとめ

ネットワーク距離が異なるサーバに対するアクセス実験を行った。この結果、GET + Range が有効であることが分かった。しかし、大容量のリソースに対しては HEAD でサイズを取得し分割することが有効であるので、ファイルの大きさを推測する方法が必要である。たとえば、拡張子を用いることなどが考えられる。

今後の課題として、さらに大きなリソースや様々なサーバに対する実験、実際にプロキシを実装しての評価、HTTP/1.1 のその他の機能に対応する機構の検討などがある。

参考文献

- [1] 藤澤弘、“複数のミラーサーバを用いた高信頼高効率 Web アクセス機構の実装及び評価”、香川大学工学部、卒業論文、2002。
- [2] RFC2616、<http://www.faqs.org/rfcs/rfc2616.html>