

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

藤澤 弘 (最所研究室)

あらまし

ミラーリングが行われているサーバの負荷分散および停止時に対処するために、本研究においては、信頼性が高く、柔軟な接続を実現するプロキシ機構を提案し、その評価を行った。その結果、提案した機構の有効性を確認した。

1 はじめに

急激なインターネットの拡がりは多くの Web サイトを生み出し、企業や有名な検索サイトでは、サーバのミラーリングを行っている。また、DNS Balance[1]においては、サーバの性能、負荷状態、ネットワークの負荷などの情報を用いて負荷分散を行っているが、サーバの停止には対応できない。

このような問題に対し、本研究では、高い信頼性と効率的なデータ取得をミラーリングされたサーバ群に行うプロキシを用いた通信機構を提案し実装した。

評価の結果、提案したプロキシプログラムの有効性を確認できた。

2 提案プロキシ機構を用いたアクセス方法

アクセスするデータの性質、サーバやネットワークの状態に応じてミラーリングされたサーバの選択、アクセス方法の変更などを行い、ミラーリングされた Web サーバに対する要求を処理する。

2.1 高信頼アクセス機構

プロキシ機構はミラーリングされているサーバにアクセスするために、ミラーサーバ群のアドレスを保持している。接続中のミラーサーバが停止すると、他のミラーサーバに自動的に切り替える (図 1)。

2.2 柔軟なアクセス機構

プロキシ機構では、以下に示す 4 つのアクセス方法を提供し、効率性の向上、取得時間の短縮を行っている。

選択接続

ミラーサーバ群に対して、定期的に応答速度を調べておき、最も早いミラーサーバに対して要求を送る。

振り分け接続

連続した要求を、振り分けて同時に要求する。画像や音声などの複数のコンテンツからなる Web ページを効率的に開く場合に特に有効である (図 2)。

分割取得

1 つのファイルに対する要求を、複数のミラーサーバに異なる部分を分割して要求し、プロキシにより合成してクライアントに渡す。巨大なファイルを取得する時に有効である。

多重接続

同じ要求を複数のミラーサーバに送信し、応答のうち最も適したものを選択する。適したものとしては、最も早く到達したもの、最終更新日時が最新のものが挙げられる。

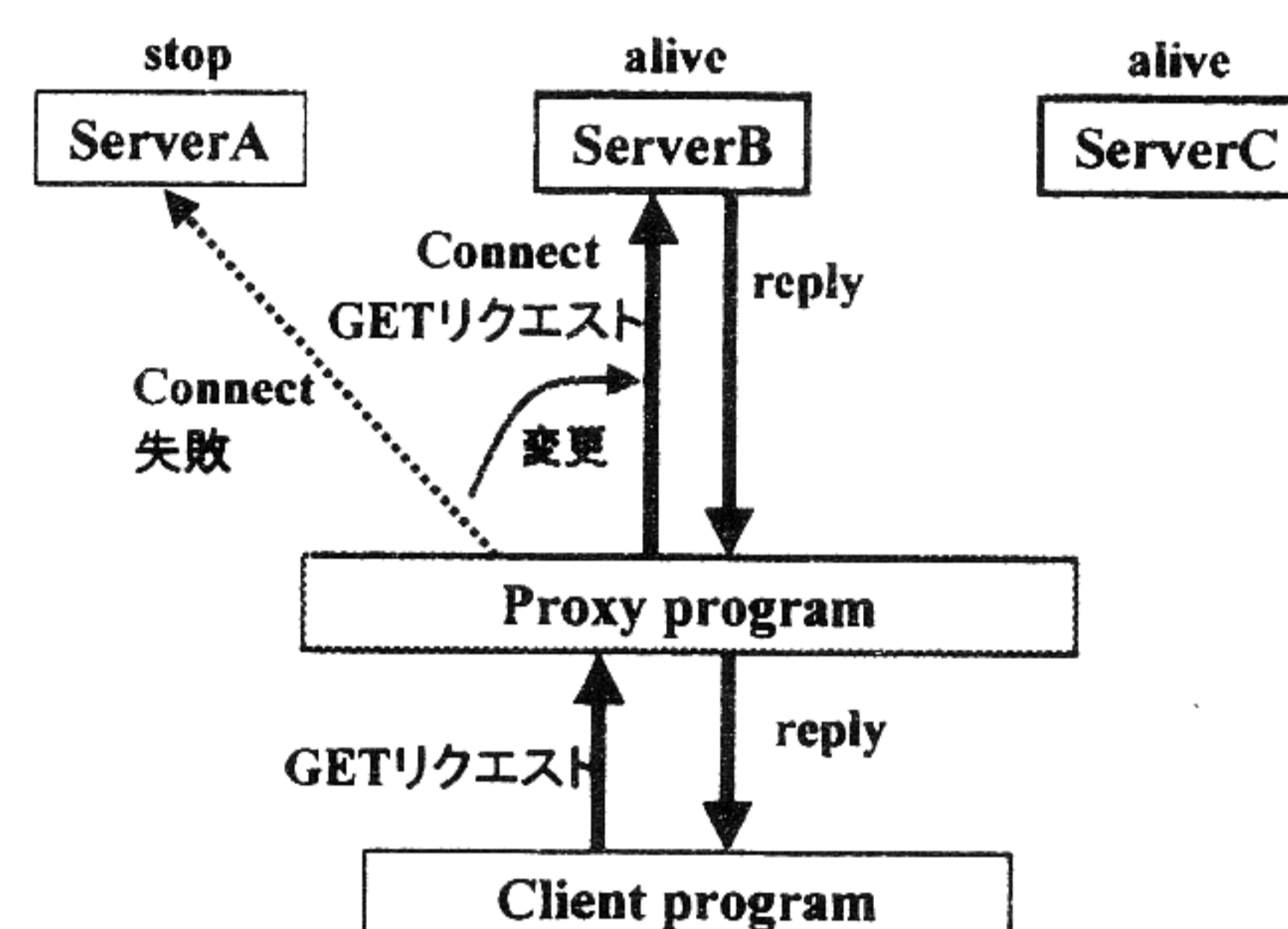


図 1: 高信頼アクセス機構によるサーバの自動切り替え

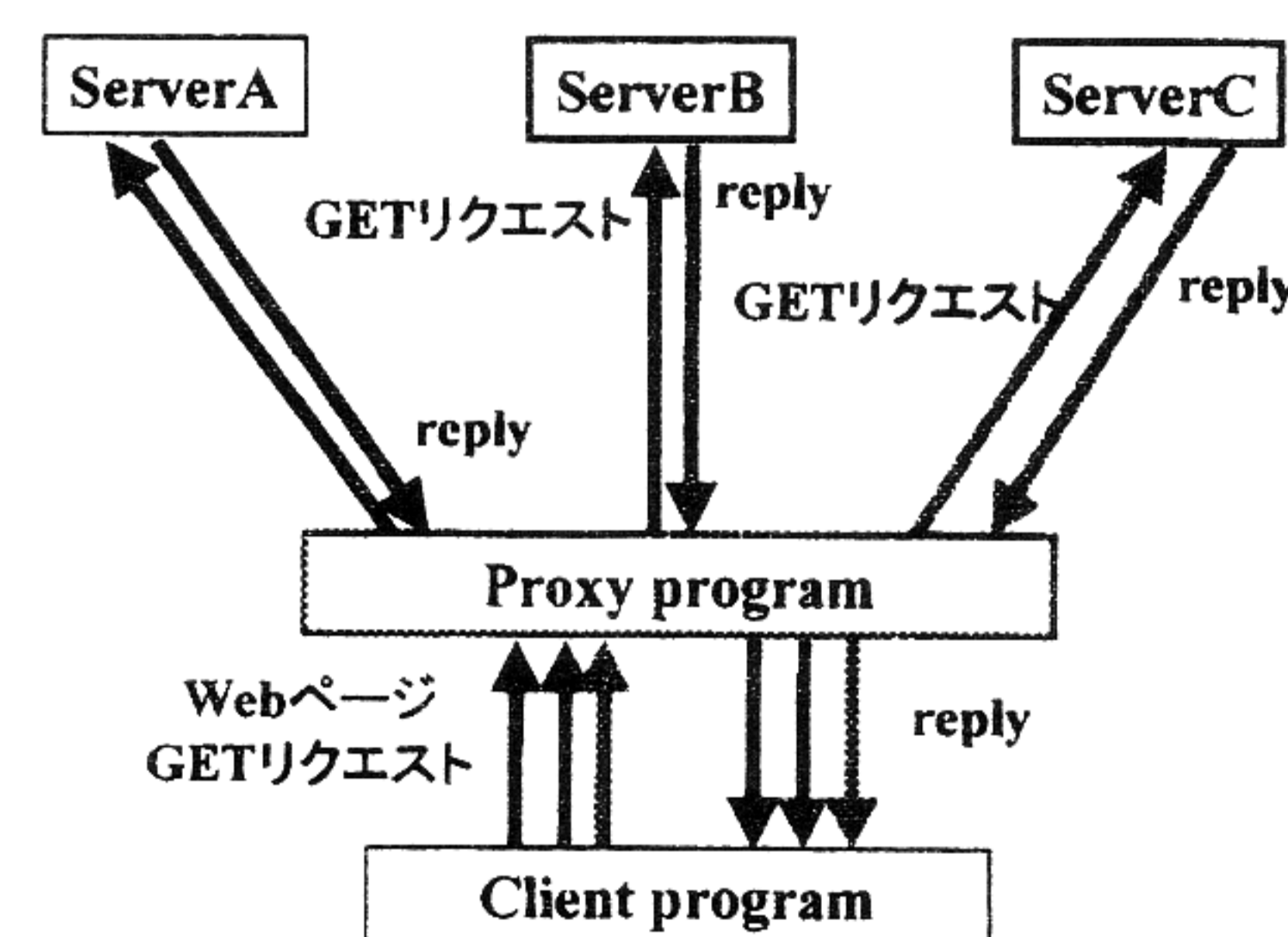


図 2: 振り分け接続によるサーバの自動切り替え

表 1: 転送時間の比較

	case1	case2	case3	case4	case 5
振り分け接続 (ミリ秒)	5,985	24,298	25,369	22,924	8,132
通常接続 (ミリ秒)	5,838	25,403	27,106	24,160	10,480
振り分け接続/通常接続 (%)	102.5	95.7	93.6	94.88	77.6

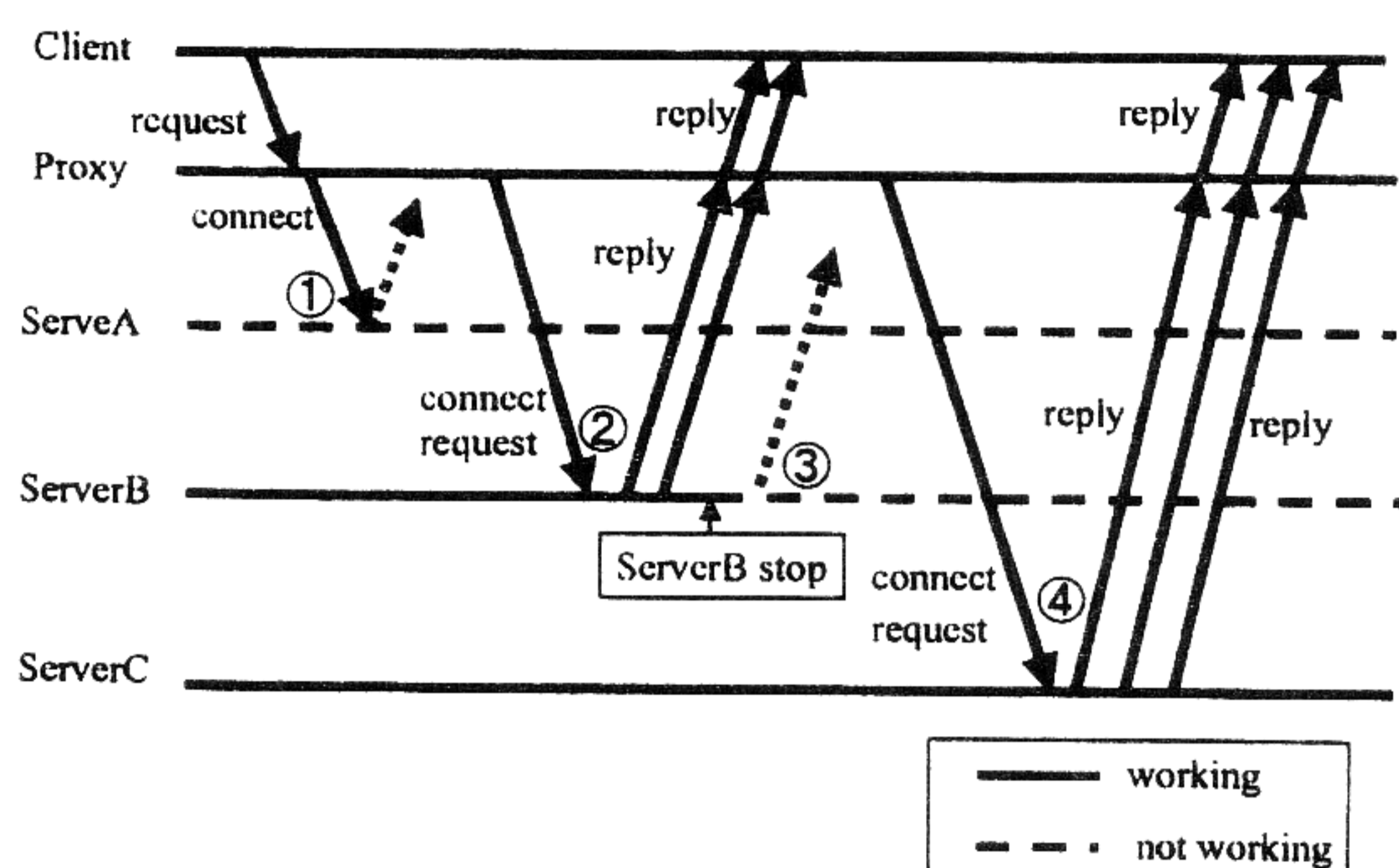


図 3: 高信頼アクセス機構の手順

3 プロトコル設計

紙面の制限により、高信頼アクセス機構の通信手順 (図 3) のみ説明する。

接続しようとしたサーバが停止中の場合

①のように、停止しているサーバ Server A に対しアクセスを行った場合は、接続に失敗した時点ですぐさま別のミラーサーバに接続し直す。

接続中のサーバが突然停止した場合

②のように、Server B にクライアントからのリクエストを送り、そのリプライを受け取っている途中で、③でサーバが停止してしまう場合も、同じようにミラーサーバに接続し直す。ただし、すでに Server B からリプライの一部を受け取っているため、④でミラーサーバ Server C には、Server B から受け取ったサイズ以降を要求する。

以上により、クライアントは、ミラーサーバ群の一部のサーバが停止しているのかにかかわらず、正常にファイル取得が行える。

4 評価

4.1 高信頼アクセス機構

実際に複数のローカルサーバを用いて実験した。接続中のサーバを停止させると、他のサーバに再接続し、ファイル取得の処理を継続していることを確認できた。

4.2 柔軟なアクセス機構

選択接続、振り分け接続を、検索サイトなどで実験を行い、取得時間が短縮していることを確認した。

検索サイト グーグル (www.google.com) の画像数の異なる複数のページを振り分け接続を用いて開き、その時間を測定した。その結果を表 1 に示す。多くの場合、振り分け接続が効率的に働いていることが分かった。

また、ローカルサーバ 2 台を使い、分割接続の実験を行った。ファイルサイズと取得時間の関係を図 4 に示す。図 4 より、分割接続を行うとどのようなサイズのファイルであっても取得時間を半分近くに短縮できることが分かった。

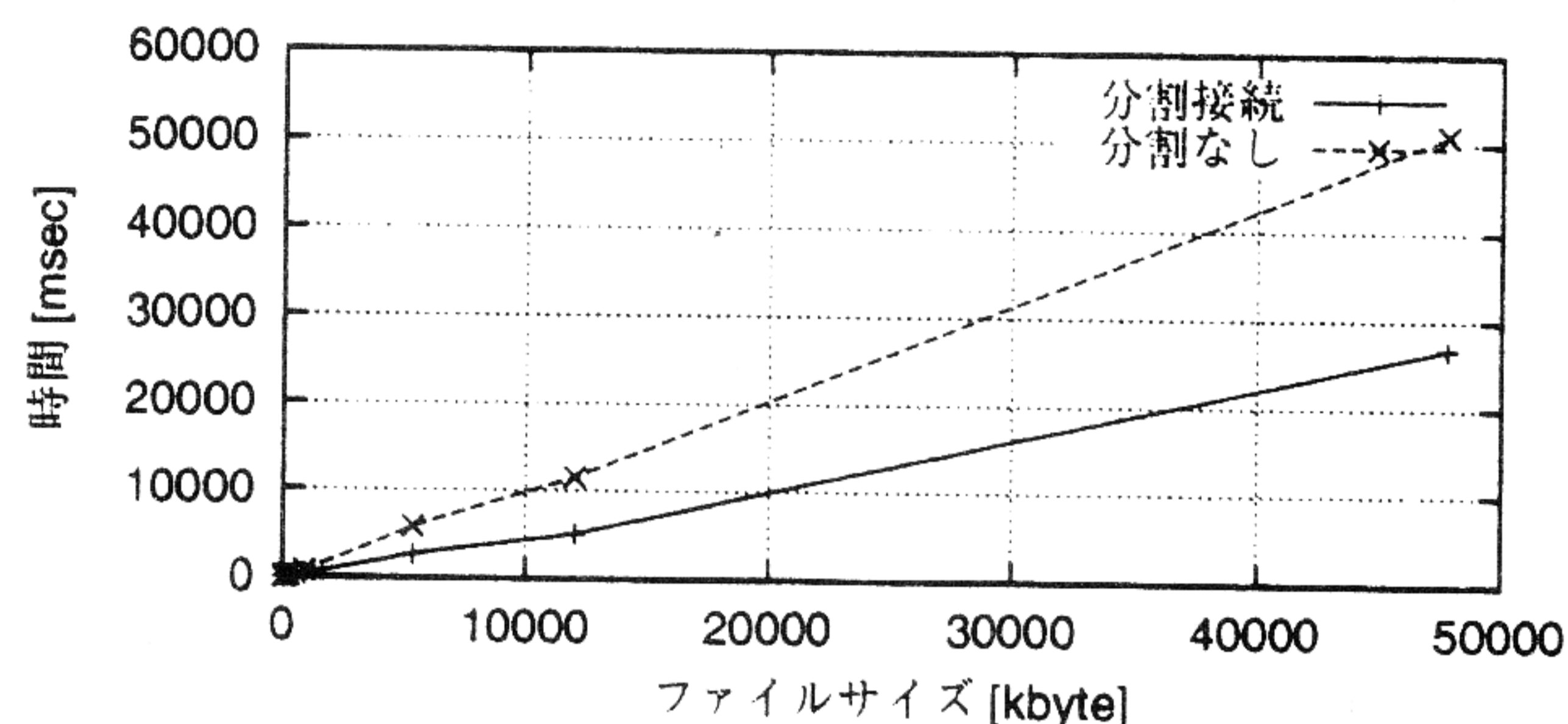


図 4: 分割接続の結果

5 まとめ

実験の結果、通信エラー時に自動的にサーバが切り替わることを確認した。また、複数の要求が同時に発生したときに複数のサーバに対してそれらの要求を振り分けることにより、多くの場合通信時間を短縮できることを確認した。

今後の課題として、動的データである cgi への対応、永続接続を用いた連続アクセスの効率向上、接続方式の自動選択等が挙げられる。将来的にはこの機能を取り込み、さらに高効率のアクセスを考えていきたい。

参考文献

- [1] DNS Balance オープンラボ,
<http://openlab.ring.gr.jp/dns.balance/>