

依存関係を考慮したミラーサーバの高速同期方式

樋口 雅文 島 健一 高橋 克巳 三浦 信幸 横路 誠司

NTT ソフトウェア研究所

要旨

多くのコンテンツを持つ大規模サーバにとって、マスターサーバ上の変更をミラーサーバに即時に反映させるのは困難である。我々はマスターサーバ側での変更を検知し、それをミラーサーバ側に通知することによって、双方の負荷を小さく押さえたまま、変更されたファイルを即座にミラーサーバに転送する方式を考案し、実装した。さらにコンテンツを構成する各ファイルの依存関係を考慮し、転送のタイミングや順序を制御することによって、サーバ運用中にファイルの変更を行ってもコンテンツの整合性になるべく影響を与えないようにした。

High speed synchronization method of the contents on master server and mirror servers based on the dependency of the contents

Masafumi Higuchi Ken'ichi Shima Katsumi Takahashi Nobuyuki Miura Seiji Yokoji

NTT Software Laboratories

abstract

It is very difficult to update files on mirror servers soon after the files on master server have been changed. We developed an method for high speed synchronization of the contents on both an master server and mirror servers by detecting the changes on master server and notifying the changes of the mirror servers. Furthermore, we try to keep consistent the contents by controlling the timing and sequence of mirroring of the files, considering the dependency of the files.

1 はじめに

近年 Internet のユーザはかなりの勢いで増加し、WWW サーバをはじめとするネットワークサービスへのアクセスもうなぎ登りに増加している。その中で、著名なサーバへのアクセスはすでに一台のサーバマシンでは処理するのが難しいレベルに達している。そこで多くのアクセスがある著名なサーバでは、ミラーサーバを設けて負荷分散を行っている。ミラーサーバをネットワーク上の適切な位置に配置することで、サーバそのものの負荷分散だけではなく、ネットワークの負荷分散を図ることができる。

しかし、サーバのコンテンツは日々更新されていくものであるから、サーバが複数になると更新の手順が複雑になる。定期的にミラーを行うプログラムを走らせて、マスターサーバからミラーサーバへ更新されたデータのコピーを行うという方法が一般的であるが、この方法ではマスターサーバ上でなされた変更が、ミラーサーバ上に反映されるまでに時間がかかる場合がある。

そこで我々は、マスターサーバ上加えた変更が即座にミラーサーバ上にも反映されるよう、マスターサーバから更新通知を送るシステムを開発した。これによって、ミラーサーバを含む全サーバの内容を高速に同期させることができるだけでなく、同期に伴う負荷を減らすことも期待できる。本稿ではそのアプローチと実装の概要について述べる。

2 アプローチ

2.1 従来のミラー

従来のミラーリングに多く用いられているのは `mirror[1]` というプログラムである。このプログラムは起動されると以下のような動作を行い、指定されたファイルをミラーする。

1. マスターサーバに `ftp` で接続する
2. 必要なディレクトリ以下のファイルのリストを再帰的に取得する
3. 取得したリストから各ファイルのタイムスタンプとサイズを抽出する
4. 抽出したタイムスタンプやサイズと、自分が持っているファイルのタイムスタンプやサイズを比較し、マスターサーバ上でのファイル更新を検出する
5. 更新されたファイルをマスターサーバから取得する

このような動作を定期的に行い、マスターサーバとミラーサーバの内容を同一に保つようにする。しかし一般にはマスターサーバ上のファイルの更新周期と、ミラーが行われる周期とは一致しないため、マスターサーバ上では更新、追加されたファイルがミラーサーバ上では更新されない、存在しないという期間が必ず存在する。このような事態をなるべく防ぐためにはミラーの周期を短くし、頻繁にサーバの内容の同期をとることが必要になる。

しかし広域ネットワーク越しにミラーを行うのは、ネットワークとサーバの双方に負荷をかけることになるため、あまり頻繁に行うのは好ましいことではない。

マスターサーバとミラーサーバの内容を出来るだけ短時間に同期させようとした場合、まず問題となるのは2のファイルのリストを取得する部分である。更新されたファイルの数はマスターサーバに存在する全ファイルに比べるとごく少数のはずである。そのファイルを検出するために、全ファイルのリストを取得するのは無駄が大きい。さらに、ファイルのリストを取得するというのはサーバのファイルシステムにも大きな負荷をかけるため、マスターサーバのパフォーマンスを下げることもつながらる。ファイルシステムの負荷の問題は、大きなファイルシステムを持つ大規模なサーバほど顕著になる。

したがって、全ファイルシステムのリストを取得する以外の方法で、更新されたファイルを検出することができれば、効率的に更新されたファイルのみを転送することが可能になる。

さらに更新の検出をリアルタイムに行うことができれば、それをトリガにしてミラーサーバへのファイル転送もリアルタイムに行うことができ、マスターサーバとミラーサーバの内容が異なっている時間を短くすることも可能になる。

2.2 変更通知によるミラー

上に述べたような従来の `mirror` の欠点を補うため、我々は以下のような方式を考案した。

1. マスターサーバ上でのファイル更新を監視する
2. ファイル更新を検知すると、ミラーサーバに対して更新を通知する
3. 通知を受け取ったミラーサーバは、該当するファイルをマスターサーバから取得する

従来のミラーが主に「ミラーサーバがマスターサーバ上のファイルの更新を調べに行く」という形態だったのに対して、「マスターサーバがファイルの更新をミラーサーバに通知する」という形態をとったのが特徴である。

このような方式によってミラーを管理すると、以下のような利点がある。

- ファイルの同期遅れの低減

従来のミラーは定期的に起動されるという運用形態が普通であるため、最大でミラーの起動間隔と同じだけの時間、マスターサーバの内容とミラーサーバの内容とが不整合である可能性がある。マスターサーバ側でファイルが更新されるとすぐにミラー側に通知が行くため、双方のサーバの内容が不整合である時間が短くなる。

- CPU コストの低減

ファイルの更新を検査するために全てのファイルのタイムスタンプを取得する必要がない。したがって、ファイルシステムへの負荷は非常に少なくなることが期待できる。この負荷の軽減は大規模なサイトほど顕著である。また、ファイルのタイムスタンプを取得するのに要する時間やそれに要する CPU コストを節約できる。CPU の負荷やファイルシステムの負荷が軽くなるため、ファイル更新時のサーバのパフォーマンス低下を防ぐことができる。それによって、頻繁にファイル更新を行ってもサービスのパフォーマンスに影響が出にくくなる。

- 回線帯域幅の節約

従来のミラーでは、ファイルのタイムスタンプを比較してファイルの更新を検出するという方法を取っていたため、起動するたびに全ファイルの詳細なリストを転送し、それを比較するという動作が不可欠であった。本方式ではファイルのリストを必要としないため、それを転送するに要するネットワーク資源を節約できる。

上記のようなメリットがある反面、マスターサーバ上でのファイル更新を検出するため、ファイル更新の方法に何らかの制限を設ける必要がある。今回我々が実装したシステムでは `ftpd` を改造し、`ftp` の `PUT` を監視することでファイル更新を検知しているため、全てのファイル更新は `ftp` 経由で行う必要がある。

2.3 コンテンツ依存関係

ファイルの中には深く関連づけられ、相互に依存関係を持っているものがあり、一方が更新された場合それに伴って他方も必ず更新されるようなファイルの組があることが予想出来る。そのようなファイルの組のうち、一つ以上のファイルが更新されたらそれ以外のファイルもまとめて転送する。ファイルの組はあらかじめ指定しておく。

また `WWW` のリンク元とリンク先のように、一方が更新されただけでは内容に不整合を起こすような場合も考えられる。例えば、リンク先のファイルがまだ存在しないにも関わらず、そのファイルを参照するリンクがリンク元ファイルに書き込まれている場合などがそれに相当する。そのような状況に対処するため、リンク元ファイルが更新されただけではミラーへの転送を開始せず、必要な全てのファイルがそろった段階で全ファイルをまとめて転送する。

3 システムの概要

3.1 システムの機能

システムの動作を簡単に述べる。下記の番号は図 1 中の番号と対応している。

1. マスタサーバへの `ftp` アクセスを監視

`ftpd` は、自分自身へのアクセスを監視している。

2. `PUT` を検出

`ftp` の `PUT` アクセスがあるとそのファイルが更新されたものと見なして `notifier` を起動する。

3. ミラーへ通知

`notifier` はファイルの更新通知を全ミラーサーバに送る。どのミラーサーバに通知を送るかはあらかじめ設定しておく。

4. 通知を受け取ったミラーサーバがマスターサーバからファイルを取得

`notify receiver` は更新通知を受け取り、`retriever` を起動して該当するファイルをマスターサーバから取得する。

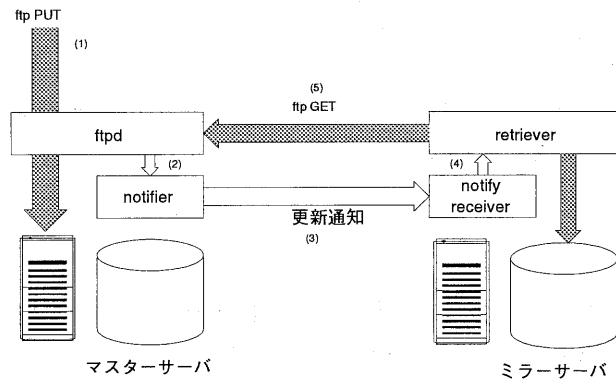


図 1: システム構成図

3.2 システム構成

本システムではマスターサーバ上でのファイル更新を `ftp` を用いて行う。 `ftpd` で `PUT` を監視し、ファイル更新を検出する。 `PUT` されたファイルは更新されたものと見なすことができるので、全ファイルのリストを取得することなく自動的に更新されたファイルを検出することができる。

- マスターサーバ側

- `ftpd`

- `wu-ftpd[2]` を改造し、 `PUT` を検出して `notifier` のサブルーチンを呼び出すようになっている。

- `notifier`

- `ftpd` がファイルの更新を検出すると、あらかじめ設定されたミラーサーバにファイル更新の通知を送る。

- ミラーサーバ側

- `notify receiver`

- `notifier` からの通知を受け取り、ファイル名を指定して `retriever` を起動する。

- `file retriever`

- マスターサーバに `ftp` でアクセスし、指定されたファイルを取得しローカルファイルシステムに置く。

3.3 通知メッセージ

ファイルの更新を知らせる通知メッセージは 1 ファイルにつき 1 メッセージで、各メッセージは以下の情報からなる。

- 変更されたファイルのタイムスタンプ

- マスターサーバの FQDN (Fully Qualified Domain Name)
- ファイルのパス名
- ファイルのサイズ

このメッセージで得られたファイルのパス名、タイムスタンプ、サイズから各ミラーサーバはそのファイルをミラーすべきか否かを判断する。

4 まとめ

我々はミラーサーバへのコンテンツの複製を、時間遅れを少なくしつつ効率的に行うため、マスターサーバ側から更新通知を行うシステムを考案し、実装した。ミラーサーバではその更新通知を受け取り、マスターサーバから更新されたファイルを取得する。本システムでは、マスターサーバからの更新通知で転送すべきファイルを識別するため、従来の `mirror` が行っていたような全ファイルのタイムスタンプを調べるといった、負荷の大きい作業は不要になる。

現在、更新通知を元にミラーを行う部分は実装済みであり、コンテンツ依存関係を考慮する部分を実装中である。

5 今後の研究予定

現在の実装では、更新通知は `unicast` を用いてミラーサーバに伝えられる。ミラーサーバが複数になった場合、同じ更新通知を複数回行うことは非効率的であるため、今後は `multicast` を利用してミラーサーバの数に関わりなく更新通知を行うことができるように改良する予定である。

また、現在は更新通知が何らかの理由でミラーサーバに届かなかった場合のことを考慮していない。そこで、送達確認及び再送の機構を設けて、全ての更新通知がミラーサーバに確実に届くようにする必要がある。

参考文献

- [1] `ftp://src.doc.ic.ac.uk/computing/archiving/mirror/mirror-2.8.tar.gz`
- [2] `ftp://wuarchive.wustl.edu/packages/wuarchive-ftpd/wu-ftpd-2.4.tar.gz`