

柔軟なグループ管理が可能な ファイル同期システムの開発

06T242 高石諒 (最所研究室)

同期するファイルをグループにまとめ、複数のコンピュータ間で同期するかどうかをグループ単位で設定できる、柔軟なファイル同期システムの設計と評価について述べる。

1 システム概要

現在、個人が自分用のコンピュータを様々な場所で複数台所有することが珍しくない。この際に、別々のコンピュータで、あるファイルを一貫性を持たせて閲覧・編集したいという欲求がある。このような欲求を解決するために、ネットワークを使ってファイルを同期する Dropbox[1] のようなファイル同期システムが使われる。本研究では、ファイルの作成や削除、更新をバックグラウンドで検知してネットワーク経由で同期し、さらに、同期するファイルをグループ単位で管理することでコンピュータ毎に同期ファイルを選択できるシステムを提案する。本研究で提案するファイル同期システムを持つ主な機能は以下の通りである。

同期ファイルのグループ管理機能 同期するファイルをグループにまとめ、グループ単位で同期

ファイルシステムイベント監視機能 バックグラウンドでファイルの作成や削除、更新等の検知と同期

2 システム設計

本システムは、図1に示すように、サーバ・クライアント型のシステムである。サーバ・クライアント型にした理由は、サーバを経由して同期することで、サーバが起動していれば同期を行うクライアントが全て起動していなくても同期できるからである。サーバ・クライアント型にした理由は、ある場所のコンピュータで同期を行う際、サーバさえ起動していれば同期を行う事ができるからである。各クライアントは、サーバとファイルの同期を行う。つまり、クライアント同士のファイル同期は、サーバを介して行われる。

同期ファイルのグループ管理機能とは、複数のコンピュータ間で同期を行いたいファイルやディレクトリを既存システムよりも柔軟に管理するための機能である。一つ以上のファイルやディレクトリを、グループと呼ぶ集団に所属させ、グループ単位で同期を行う。グループは複数個持つことができるので、コンピュータがあるグループを同期するかしないか選択することができる。また、グループにはファイル一個から所属

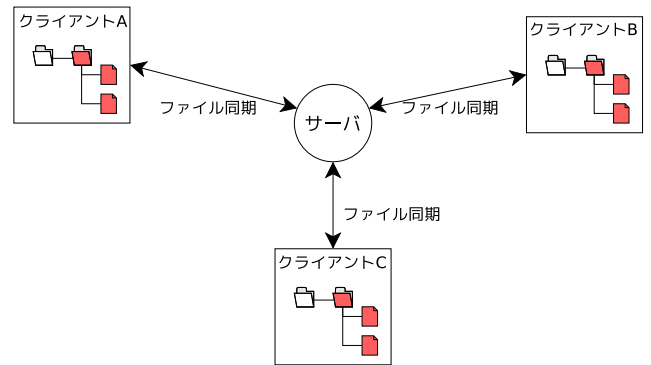


図1: システム概要図

させることができるのでファイル単位の同期が可能である。この機能により、既存のファイル同期システムではできない、必要なファイルのみの同期や、ファイルを移動することなく同期が可能になる。例えば、図2に示すように、コンピュータAとコンピュータBはグループAとグループBを、コンピュータAとコンピュータCはグループCを同期している。

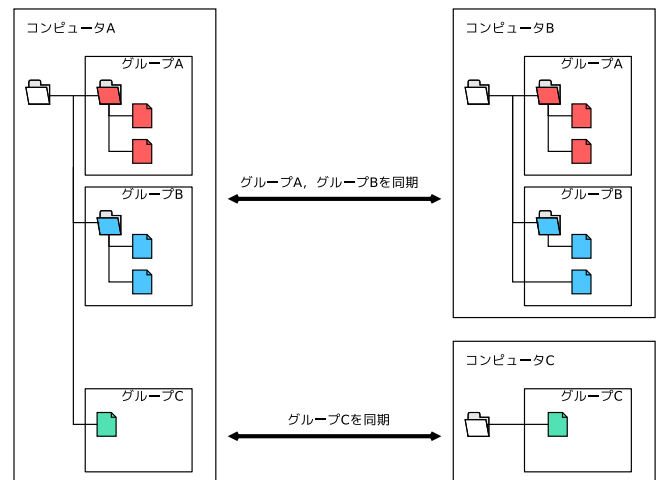


図2: 同期ファイルのグループ管理概要

ファイルシステムイベント監視機能とは、ファイルのオープン・クローズ、ファイルやディレクトリの作成・削除、ファイルの更新などのファイルシステムイ

イベントを監視する機能である。同期ファイルを監視し、それらに対するファイルシステムイベントの発生を検知し、イベントに応じてサーバと同期を行うかどうかを決定する。

3 評価

設計に基づいて本システムを実装し、同期中にサーバプログラム・クライアントプログラムの CPU 使用率がどのように変化するか実験を行った。この実験の目的は、同期するファイル数が増えた場合、全てのファイルの同期を行う際にどの程度 CPU を消費するのか調べ、実用に耐えうるかどうかを確認することである。実験は以下の 2 パターンを行った。

パターン A 性能の高いコンピュータをクライアントに、性能の低いコンピュータをサーバにして計測

パターン B パターン A の逆で計測

パターン A の結果を図 3 に、パターン B を図 4 に示す。

図 3 に示されるように、クライアントプログラムの CPU 使用率は、性能の低い CPU を用いた場合でも、最大で 30% 程度であり、この程度ならば許容範囲であるといえる。サーバプログラムの CPU 使用率がファイル数の増加に伴い増加した理由は、今回の実験がクライアントからサーバにファイルを送信するものであるため、サーバプログラムのファイルや DB への書き込みが膨大な量になるからであると考えられるが、詳細な調査が必要である。図 3 に示されるように、サーバプログラムの CPU 使用率は、性能の低い CPU を用いた場合、同期ファイル数が 10000 個になると 80% 以上 CPU を使用している。この時、サーバの他のプロセスに影響を与える可能性が考えられるため、プログラムを改善して CPU 使用率を低く抑えるようにするか、CPU に高性能な CPU を使用する等の対策が必要である。また、パターン B のサーバの CPU 使用率が、

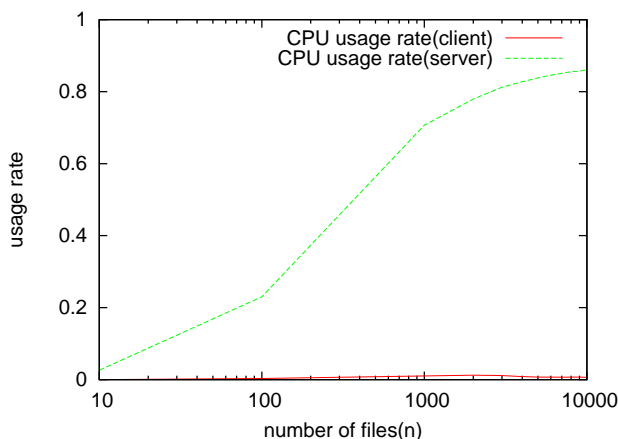


図 3: 同期中の CPU 使用率 (パターン A)

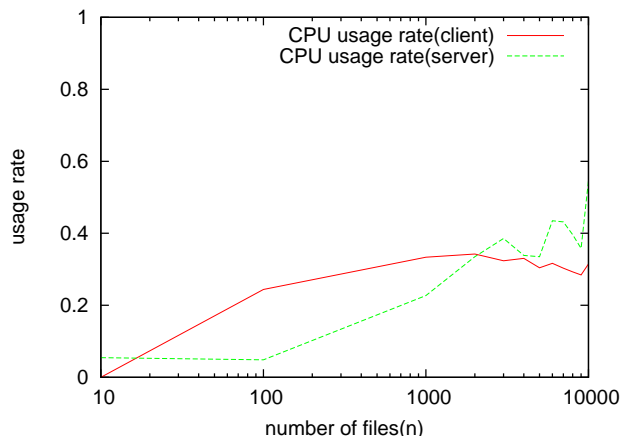


図 4: 同期中の CPU 使用率 (パターン B)

ファイル数 2000 個から 10000 個にかけて、上下する結果となったが、原因を特定することはできなかった。また、同期にかかる時間や同期中のメモリ使用量についても評価を行ったが、実用上問題はなかった。

4 まとめ

ファイルをグループで管理するファイル同期システムを提案した。このシステムは、同期ファイルのグループ管理機能とファイルシステムイベント監視機能で構成される。本システムは、サーバとクライアントが双方向同期を行う事で、間接的にクライアント同士のファイル同期を実現している。また、評価実験を行い、実用性について確認することができた。本システムの今後の主な課題を以下に述べる。

エラー処理・セキュリティ機能 今回の実装ではエラー対策は行っていない。また、セキュリティ機能も実装してない、しかし、実用するためにはこれらの実装が不可欠となる。

グループ操作やファイル追加時の負荷の軽減 今回の実装では、グループを新規に作成したり、グループにファイルを追加すると、クライアントは再起動する仕組みのため、これらの操作を行うと、毎回サーバと全体同期を行う事になる。この時、処理が膨大になり、システムへ負荷がかかるため、改良する必要がある。

他の OS への移植 今回の実装では、Linux 上のみでしか動作しない。しかし、多くの既存のファイル同期ソフトウェアは、マルチプラットフォームであり、本システムも、実用することを考えると Windows や UNIX への移植を行う必要がある。

参考文献

- [1] Dropbox. <http://www.dropbox.com/>.