1D－3

暗号化ファイル共有システムの概要

高橋 俊成、新保 淳、室田 真男
(株)東芝 研究開発センター

1. はじめに

今日、電子媒体を利用したコミュニケーションが盛んであるが、その多くは電子メールや電子ニュースといった古い世代のツールが活躍している。一般にグループウェアと呼ばれる計算機サポートのツールまたはその環境は、今日の計算機ハードウェアの進歩やマルチメディア、ネットワークの発展を十分に活かすものを目指していると考えて良いが、見栄え良く多機能化しただけの「良くできたツール」は、UNIX等の透過な環境に慣れたユーザにとっては見かけ倒しのアプリに終る場合も多い。

我々は、共同作業の効率化を凝ったアプリケーションに求めるのではなく、汎用性の高い計算機システムとしての基盤作りによって本質的に解決することを目指し、分散共有データをファイル・システムのレベルで扱う一手法を提案する。

本手法によれば、主としてテキスト・ベースの共有ファイルの編集を、非同期に行うことができるため、ネットワークから切断される移動端末や、品質の保証されない無線ネットワーク等においても、同時編集が可能である。アクセス権の管理が厳密であり、履歴管理も容易である。さらには、ファイル・データを暗号化管理した場合においても、これら特徴(非同期編集、アクセス管理、履歴管理)を損なわない工夫がなされている。

2. 概要

共有ファイルの更新を、各利用者の変更部分のマージにより行うシステムのアイデアが知られている[1)]。我々はこのアイデアが非同期編集に適するものであることに着眼し、これを実現するとともに、さらにネットワーク上での共同編集に利用することを意識し、認証、秘匿の可能な形式に改良し、システム管理者からも保護できるものとした。認証および暗号化の詳細については別途発表される[4)]。

現在ネットワーク共同編集ファイルのアプリケーションとして実験的に開発しており、将来は次世代ファイル・システムとして実用化する予定である。

A design of privacy enhanced file sharing system
TAKAHASHI Toshinari, SHIMBO Atsushi, MUROTA Masao
Communication and Information Systems Research Labs. TOSHIBA Corp.

3. ファイルの共有および更新

サーバは共有データに対し、更新時の短時間なロックを除きユーザ間の排他制御は行わない。編集しようとするクライアントは任意の時刻にファイルを読み出し、その内容と時刻の記録を編集終了まで保存する。編集はサーバとは無関係にクライアントの持つツール(例えば従来からあるemacs)を用いて行う。編集が終了したら編集前と編集後の差分(diff)[2)]を文字単位で求め、取り出した時刻と共にサーバに送る。サーバはクライアントから送られた「挿入」または「削除」のデータを履歴の残る形式で共有ファイルに反映させる(マージ処理)。その際、挿入/削除 位置は、クライアントが最初に取り出した時刻情報を基にその時点における位置に変換する。

例えば図1の例では、client-Aが時刻t1に取り出したファイルを編集し、先頭からaの位置に長さbのデータを挿入し、cの位置から長さdのデータを削除する。これらの変更は時刻t1のファイルとのdiffにより検出される。そして時刻t3に変更をサーバに反映させる。各ブロックには生成時刻および消去時刻のtime stampが記されている。同様にclient-Bが時刻t2に取り出したファイルのeの位置に長さfのデータを挿入し、時刻t4に変更をサーバに反映させる。その際、位置eは取り出した時刻t2における位置を指すように、各ブロックを数えるか否かを各ブロックのtime stampを元に判断する。

この方法によれば、複数の人(プロセス)が同時にファイルの変更を行っても、各々の挿入/削除 のオペレーション内容は全て反映されるので、非同期に共同編集ができ、ネットワークの切断等による予期しないロックやデータの破壊は起こらない。

マージ処理において、複数人による同一部分への複数の挿入は、順番を保証しない連続した挿入とする。また、複数人による同一部分の削除は、先行した1回の削除として扱う。

この方法は、各変更における差分データを維持する弱い一貫性を保証するものである。同時編集により意味的な破綻が起こる場合もある[1)]が、関知しない。それによって、切断可能な環境における非同期共有編集を可能にすることが本研究の主旨である。

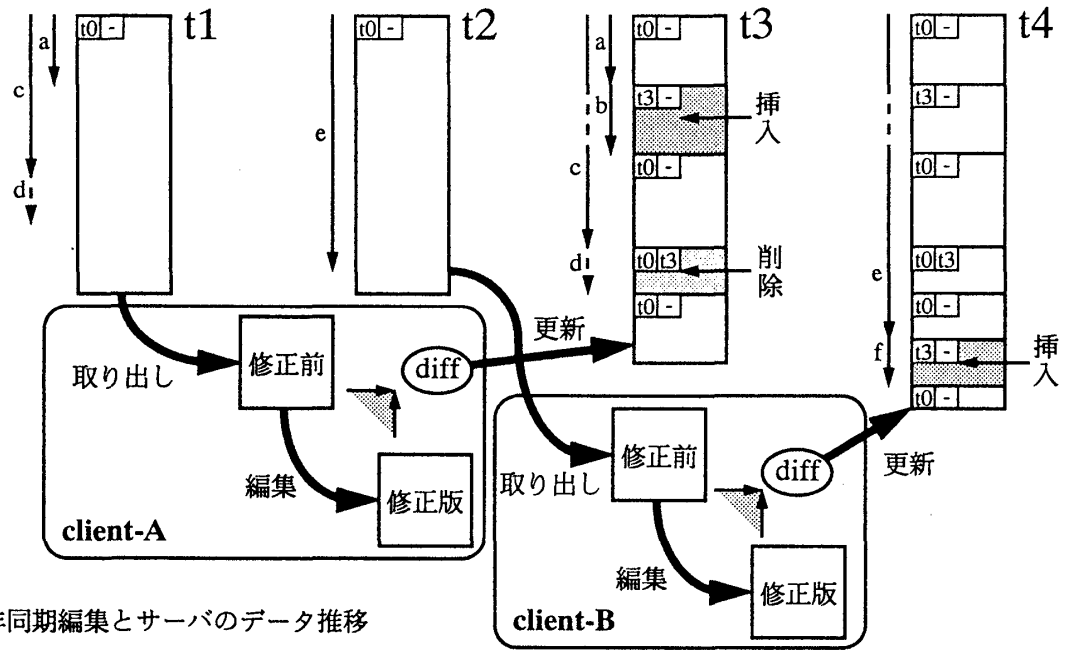

図1. 非同期編集とサーバのデータ推移

以上説明したマージ処理は、全てのクライアントによる編集開始から終了までの変更履歴をサーバが保持しているので可能となる。非同期編集の目的に限れば編集開始以前の変更履歴は破棄することも可能であるが、我々はこれをも保存することを原則とした。これにより、任意の時刻におけるファイル内容は各ブロックのtime stampを使って容易に知ることができるので、履歴管理機能も含んだ共有ファイルとして使うことができる。また、クライアントの編集開始時刻をサーバが保持する必要がないので、ステート・レスなサーバが提供できる。

4. nbbs (Network Bulletin board System)

本アイデアによるnbbsの開発を行っている。これはパソコン通信等に見られるBBSとは全く異なり、複数のメンバが共有ファイルを既存のエディタ(emacs,vi等)で編集し、情報交換やデータ・ベースとして用いるものである。認証、秘匿、非同期編集などのネットワーク特有の要求に最適なツールという意味でNetworkという語を付けた。共有エディタに改造するアプローチ[3]とも異なる。

ファイル名(nbbsのグループ名)を指定してnbbsを起動すると、Server/Client通信によりネットワーク上の任意のファイルを共有ファイルとして用い、各ファイルに登録されたユーザのみがアクセスを許可される。編集を終了すると変更部分がサーバに伝わり共有ファイルのデータが更新される。

このように、nbbs は今日の IP-reachability を有効に活用するとともに、ファイルという概念で統一されたUNIX等の扱い易さを生かすものである。

現在詳細仕様が固まり、αバージョンが完成しようとしている。

5. CFS (Collaborative File System)

本方式により協調作業向きのファイル・システムCFSを作ることが予定されている。例えば

vi u-tokyo.ac.jp:/usr/local/lib/meibo/suuri88.meibo

といった指定によって、従来のエディタを全く改造せずに共有ファイルを編集することである。

実装はNFSの上位互換化またはライブラリの入れ換えによる方法を検討中である。

6. おわりに

本発表のコンセプトにおいては、ある種の制約の下にデータ共有におけるアクセス管理、認証、秘匿、同時(非同期)編集を全て解決する。近年のネットワーク・コミュニケーションの広まりと無線通信の進歩に殊に馴染むものであると期待されている。

参考文献

[1] 高橋 俊成:平面型エディタによる立体志向データの編集, 情報処理学会 第32回プログラミング・シンポジウム, pp. 63-68 (1991).

[2] W. Miller and E.W.Myers: A File Comparison Program, Software Practice and Experience vol.15, No.11, pp.1025-1040 (1985).

[3] Issues in the Design of a Toolkit for Supporting Multiple Group Editors: Proceedings of the Spring 1993 USENIX Conference, Computing Systems, Vol. 6, No. 2, pp. 135-166 (1993).

[4] 新保 淳 他: 暗号化ファイル共有システムのセキュリティ機構, 情報処理学会第49回全国大会, 1D-2, (1994).