B_038

遅延書き込みを用いた暗号化ファイルシステムの高速化手法の提案

Proposal of Fast Cryptographic File System by Using Delayed Write

松田 直人† 田端 利宏†
Naoto Matsuda Toshihiro Tabata

1．まえがき

現在，計算機は小型化や低価格化により，いたるところに普及している．また，ネットワークの普及により，多くの計算機がネットワークに接続し，情報のやり取りを行っている．さらに，多くの情報が電子化され，計算機に保存されている状況にある．しかし，システムの脆弱性の悪用や，コンピュータウィルスの感染，計算機の盗難などによって，電子化された情報の漏洩が問題になっている．この情報漏洩の対策として，情報を暗号化する方法が有効である．そこで，暗号化ファイルシステムが開発されているが，その処理速度は，通常のファイルシステムと比べて，低速であることが指摘されている[1].

本稿では，独自のキャッシュ機構を用いた遅延書き込み機能を使用することによって，暗号化ファイルシステムを高速化する手法を提案する．

2．CFS

CFS（Cryptographic File System）[2]は，CFS デーモンと呼ばれる NFS サーバとして実装されている．利用者は暗号化を行うディレクトリを CFS 経由でマウントすることにより，透過的にデータの暗号化と復号を行うことができる．CFS のデータの流れを図 1 に示す．暗号化を行う場合は，データは，応用プログラム（以降，AP と略す）からカーネル，カーネルから CFS に渡され，そこで暗号化される．その後，再びカーネルを通してファイルシステムに渡され，ディスクへ保存される．復号を行う場合も同様に，カーネルを通して CFS に渡されて復号される．そして，再びカーネルを通して AP へデータが渡される．

暗号化ファイルシステムは，暗号化と復号により大きく性能が低下する．その他に CFS 固有の問題点として，ネットワークによるオーバーヘッドや，カーネル空間とユーザ空間でのデータの授受やコンテキストスイッチが頻繁に発生することによって，性能が低下してしまうことが挙げられる．

3．遅延書き込み機能

3.1 基本機構

メモリへのアクセス速度に比べて，ディスクへのアクセス速度は，非常に低速である．このため，書き込み要求をそのつど処理していては，全体の性能が低下してしまう．そこで，効率のよいデータ I/O 方法として，遅延書き込みを CFS に実装する．遅延書き込みとは，データの書き込み要求をすぐに処理するのではなく，一時的にメモリ上に溜めておいて，後でまとめて書き込む方法である．この方法を用いることで，ディスクへのアクセス回数の削減やコンテキストスイッチ回数の削減，システムコール発行回数の削減が可能であるため，処理効率が上がり，全体のパフォーマンスを向上させることが可能である．また，書き込み要求を出した AP に対して，すぐに処理を返すことが可能である利点や，キャッシュ内のデータを読み込み時に使用することにより，読み込みを高速に行うことが可能な利点もある．

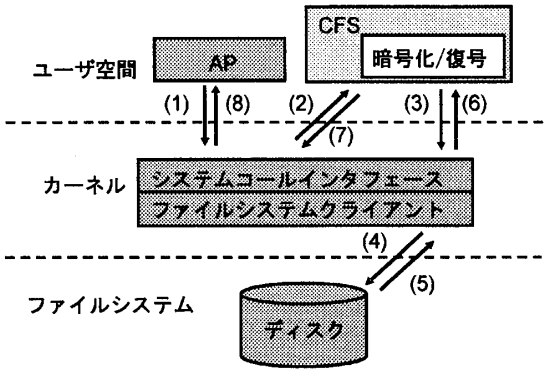

図 1 CFS の概要

3.2 課題と対処

遅延書き込みを実現するための検討課題として，データの保存形式，キャッシュを行ったデータをディスクへ書く契機と量，暗号化の契機，及びキャッシュを行ったデータに対して読み込み要求が来たときの対処がある．以降で，各課題について述べる．

（1）データの保存形式

遅延書き込みを行うためには，データをメモリ上に一時的に保存しておく必要がある．このため，データの保存形式は重要な課題である．ファイルへの書き込みを行う場合，ファイル名を基にファイルをオープンする．次に，オフセット値を基にして，書き込む位置を決める．最後に，書き込むデータと，そのバイト数を基にして，ファイルへの書き込みを行う．よって，最低限保存しなければならない情報は，ファイル名，オフセット値，データ長，実際に書き込むデータの 4 種類であるため，これらの情報を構造体にて保存する．以降，この構造体をノードと表記する．

また，以下に示す 2 種類のキューによってノードを管理する手法をとる．

（a）clean キュー

未使用のノード，もしくは，すでにデータをディスクへ書き込んだノードを管理するキューである．

（b）dirty キュー

データに変更があったが，ディスクへ書き込まれていないノードを管理するキューである．

（2）キャッシュを行ったデータをディスクへ書く契機と量

†岡山大学 大学院自然科学研究科，Graduate School of Natural Science and Technology，Okayama University

キャッシュを行ったデータは，できるだけ早くディスクへ書き込む必要がある．しかし，ディスクへの書き込みによって利用者のジョブの妨げになってはならない．そこで，CPU と ディスクの空き時間を利用して書き込むようにする．具体的には，データを書き込むプログラムを用意し，通常よりも低い優先度で走行させ，ある一定のタイミングで書き込むようにする．詳細なタイミングと量は，実装時にパラメータを調整し，適切な値を見つける必要があるため，今後の課題とする．

（3）暗号化の契機

暗号化の契機として以下の2種類の方式が考えられる．

（a） メモリ上に保存するときに暗号化を行う

（b） ディスクへ書き込むときに暗号化を行う

方式（b）は，ノードのメモリ上のデータについては，暗号化を行わないため，方式（a）よりも高速に動作すると思われる．しかし，メモリ上のデータは暗号化されていないため，メモリがダンプされると，暗号化を行う情報が読めてしまう問題がある．そこで方式（a）を用いる．

（4）キャッシュを行ったデータに対して読み込み要求が来たときの対処

AP がキャッシュを行ったデータに対し，読み込み要求を出した場合，キャッシュしてあるデータを返すことができれば，ディスクへアクセスする必要がなくなるため高速な処理が可能である．そこで，ディスクへの書き込みを行っていないデータに対して読み込み要求が来たときには，キャッシュしているデータを返すようにする．

4．実装と評価

4.1 実装

遅延書き込みの効果を検証するために，基本的な評価を行った．このために，遅延書き込み機構を CFS 1.4.1 上に実装した．CFS は，NFS サーバとして動作するため，ファイル属性の取得や，ファイルへの書き込みなどの一連のファイル操作ごとにそれぞれ手続きが用意されている．そこで，手続きの一部を変更することで，遅延機構を実現した．具体的には，書き込み用手続きを，キャッシュへ保存するように変更した．また，読み込み用手続きは，キャッシュを探索し，要求するデータが見つかればそのデータを返し，見つからなければディスクへ読みにいくように変更した．また，キャッシュ内のデータを定期的にディスクへ書き込む別 AP の作成も行った．

4.2 評価

評価は，CPU: Pentium III 866MHz，メモリ: 384MB，OS: FreeBSD 4.3-RELEASE の計算機上で，ベンチマークソフト PostMark を実行させた．PostMark は，メールサーバのような多数のサイズの小さいファイルを扱うシステムを想定している．PostMark を初期の設定である 500 個のファイルを生成し，500 回のトランザクションを行う設定で実行した．また，実装したキャッシュは 8KB の領域を 1000 個確保している．実験結果を図2，図3に示し，説明する．

図2から，遅延書き込みを行う場合では，遅延書き込みを行わない場合に比べて，合計時間は約 15%，トランザクション時間は約 20%処理時間を短縮できることがわかる．これは，ディスクへのアクセスを行う必要がないため，応答時間 を短縮できるためである．

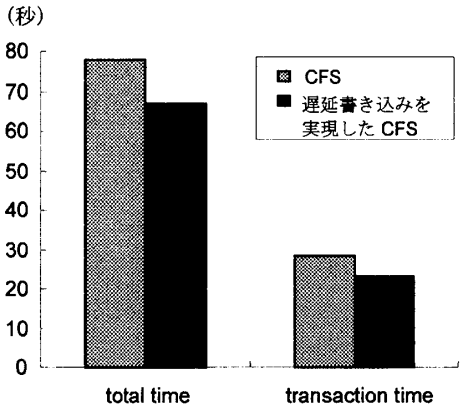

図2 PostMark による評価

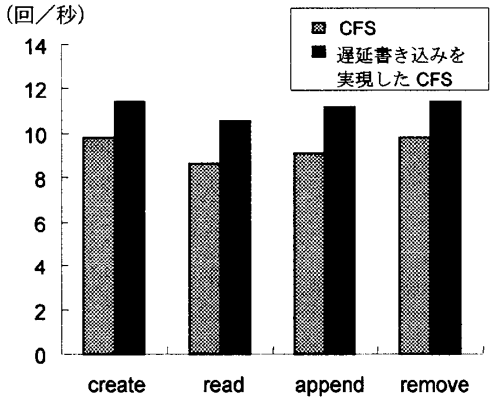

図3 PostMark による詳細な評価

図3から，書き込みだけでなく，ファイルの生成（create），読み込み（read），及び削除（remove）操作も高速化されていることがわかる．これは，キャッシュの効果により，多くのファイル操作の応答時間を短縮できるためである．

5．おわりに

本稿では，遅延書き込みを用いることにより，暗号化ファイルシステムを高速化する手法を提案した．提案手法の有効性を確認するため，CFS に遅延書き込み機能を実装し，PostMark を用いて評価を行った．評価の結果，ディスクアクセス回数を削減することにより，処理時間を約 15%短縮できることを示した．

参考文献

[1] Charles P. Wright, Jay Dave, Erez Zadok, “Cryptographic File Systems Performance: What You Don't Know Can Hurt You”, IEEE Security in Storage Workshop, pp. 47-61, 2003.

[2] Matt Blaze, “A Cryptographic File System for Unix”, Proceedings of the first ACM Conference on Computer and Communications Security, pp. 158-165, 1993.