

# ファイル常駐キャッシング機能の提案と評価

3F-4

中野裕彦、道明誠一、新井利明、吉澤康文  
 (株)日立製作所 システム開発研究所

## 1. はじめに

多くの分散ファイルシステムは、ファイルサーバが保管するファイルをクライアントにキャッシングすることでファイルアクセスを高速化している。ここでの課題は、キャッシングするファイルを適切に選択することである。通常は、キャッシュヒット率の向上を目的として、LRU方式を採用している。

しかし、分散ファイルシステムでは、キャッシュヒット率だけではなく、キャッシュミス時のアクセス時間も考慮する必要がある。分散環境では、ファイルの存在場所によって、遠隔アクセスに要する時間が大きく異なる。例えば、高速なLANで接続しているファイルサーバと比べて、低速なWANで接続しているファイルサーバへのファイルアクセスは、数十～数百倍時間がかかる。このような状況では、WANで接続したファイルサーバが保管するファイルを優先的にキャッシングすることで、平均アクセス時間を短縮できる。

そこで、本報告では、ユーザが指定したファイルをクライアントのキャッシュに常駐する機能を提案する。ユーザは、アクセスする可能性があるファイルのうち、アクセスに時間がかかると影響の大きいファイルを常駐させることで、ファイルの平均アクセス時間およびプログラムの実行時間を短縮することができる。

分散環境では、ネットワーク障害やサーバのダウンなどが原因で、ファイルをキャッシングできない場合がある。また、キャッシングできたとしても、他のクライアントがファイルを更新した場合、キャ

ッシュが無効化されてしまう。これらの状況においてもファイルの常駐を保証する機能として自動再キャッシング機能を提案する。そして、この機能をOSF<sup>1</sup>/DCE (Distributed Computing Environment) の分散ファイルシステムであるDFS (Distributed File System) [1] 上に実現した。

以下、常駐キャッシング機能とその効果を述べる。

## 2. 常駐キャッシング機能

図1に、常駐キャッシング機能の構成を示す。キャッシュマネージャは、ユーザが常駐指定したファイルをキャッシングし、ファイル状態記録テーブルに登録する。通常、キャッシュマネージャは、LRU方式に従ってキャッシュを置き換えるが、常駐指定されたファイルに対しては、置き換えの対象としない。

ネットワーク障害やサーバのダウンなどが原因で、常駐指定されたファイルをキャッシングできない場合がある。また、キャッシングできたとしても、他のクライアントがファイルを更新した場合、キャッシュは無効化されてしまう。キャッシュマネージャは、ファイルの状態を表1に示す6つに分類し、ファイル状態記録テーブルに登録する。ユーザは、ファイル状態記録テーブルを参照することで、常駐指定したファイルの状態を調べることができる。

not-accessibleとrevoked状態のファイルに対しては、再度、キャッシングを試みることで、キャッシングに成功する可能性がある。そこで、これらの状態のファイルを自動的に再キャッシングする自動再キャッシング機能を導入した。

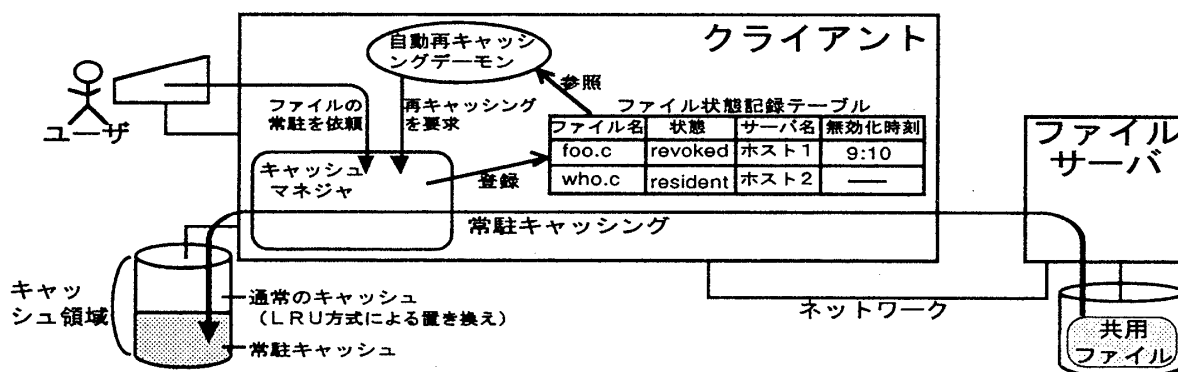


図1 常駐キャッシング機能の構成

<sup>1</sup>OSFは、Open Software Foundationの商標です。

表1 ファイルの状態一覧

項番	状態	説明
1	resident	常駐キャッシング中
2	revoked	キャッシュが無効化された
3	not-DFS	DFSのファイルでないため、キャッシング不能
4	not-accessible	サーバにアクセスできないため、キャッシング不能
5	not-exist	ファイルが存在しないため、キャッシング不能
6	access-denied	Readの権限がないため、キャッシング不能

### 2. 1 自動再キャッシング機能

自動再キャッシングデーモンは、ファイル状態記録テーブルを参照して、not-accessibleおよびrevoked状態のファイルを再キャッシングする。

not-accessible状態のファイルに対しては、そのファイルを保管しているファイルサーバとの接続を定期的に調査する。接続が確認でき次第、ファイルの再キャッシングを行なう。

revoked状態のファイルに対しては、無効化されたから時間Sだけ経過してから再キャッシングを行なう。ユーザは、時間Sの値を使用環境に応じて変更することができる。時間Sを短くすると、キャッシュの存在確率が向上し、ファイルアクセスを高速化できる。しかし、他のクライアントが一定期間連続してファイルを更新する状況では、再キャッシングしても、すぐに無効化されてしまう。この結果、再キャッシングとキャッシュの無効化を繰り返し、パフォーマンスが著しく劣化する。このような状況では、時間Sを長めに設定し、しばらくの間、再キャッシングを見合わせる必要がある。

### 3. 常駐キャッシングの評価

常駐キャッシングの目的は、遠隔アクセスに時間がかかるファイルをキャッシュに常駐することで、ファイルアクセスを高速化することである。しかし、あまり多数のファイルを常駐すると、LRU方式で置き換えが行なわれる通常のキャッシュの容量が減少する。その結果、キャッシュミスが増大し、かえってアクセス性能が劣化する恐れがある。

本章では、常駐キャッシングの効果をシミュレーションにより評価する。更に、常駐するファイルがキャッシュ全体に占める割合の限界値を求める。

#### 3. 1 評価方法

同じサイズの1000個のファイルが存在し、950個のファイルは、遠隔アクセスに100msかかり、残りの50個のファイル(以下、ファイルSと呼ぶ)は、遠隔アクセスに200msかかる場合

を想定する。また、キャッシュアクセスに要する時間は、50msとする。クライアントは、以下の分布に従って、これらのファイルにアクセスする：

全ファイルを最近アクセスした順に整理し、

- i) 先頭からn番目 ( $1 \leq n < 100$ ) のファイルにアクセスする確率を、 $7.0/n\%$
- ii) 先頭から100番目以降のファイルにアクセスする確率を、一律0.070%とする。

上記条件の元で、次の2つのキャッシュ置き換え方式を採用した場合のキャッシュヒット率と平均アクセス時間を算出する。

- (1) LRU方式。
- (2) ファイルSをキャッシュに常駐し、残りのファイルは、LRU方式で置き換える。

#### 3. 2 評価結果

シミュレーションの結果を図2に示す。クライアントが150個以上のファイルをキャッシングできる場合、50個のファイルSをキャッシュに常駐しても、キャッシュミスがほとんど増加しない。この結果、他のファイルに比べて遠隔アクセスに2倍時間がかかるファイルSをキャッシュに常駐することで、平均アクセス時間を約3ms短縮できる。

しかし、100個以下しかキャッシングできない状況で、50個のファイルSを常駐すると、キャッシュミスが急増し、LRU方式に比べて平均アクセス時間が長くなる。従って、常駐するファイルをキャッシュ全体の50%以下におさえる必要がある。

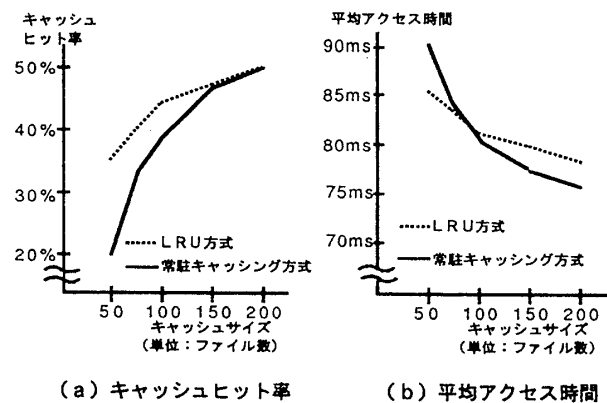


図2 シミュレーション結果

#### 4. おわりに

アクセスに時間がかかるファイルをキャッシュに常駐する機能を提案し、シミュレーションにより平均アクセス時間の向上を確認した。

#### 参考文献

- [1] M.L.Kazar et al. : DEcorum File System Architectural Overview, USENIX Conference Proceedings, pp.151-164 (1990 Summer)