# RAIDシステム内蔵型NAS(5) –キャッシュメモリ制御-
# (Improvement of Cache Memory Performance for Embedded NAS)

須藤 敦之　坂口 明彦　山崎 康雄‡
(株)日立製作所 中央研究所§

## 1. はじめに

情報システムの利用拡大により、システムが保持するデータ容量は膨大なものとなっている。その膨大なデータを管理するコストを削減するため、ストレージ統合が求められている。これまで、SAN(Storage Area Network)接続による管理の統合が進められてきたが、近年、LAN(Local Area Network)上に配置することが可能な NAS(Network Attached Storage)の利用も増大しつつある。より大規模なストレージ統合実現のため、NAS のデータアクセス性能に対する要求はますます高まっている。

そこで、RAID システムに NAS ブレードを内蔵する RAID システム内蔵型 NAS において、NAS サーバ OS と RAID コントローラとが協調して動作するデータアクセス性能向上方法を検討した。とくに NAS サーバ OS のファイルシステムの情報を利用した RAID システムのキャッシュメモリ制御方法についての実装および評価を行った。

## 2. システム構成

RAID システム内蔵型 NAS は、RAID システム内に NAS ブレードと呼ぶ専用ボードを実装する。これにより、1 台の RAID システムで SAN/NAS の 2 つの機能を提供する。システム構成を図 1 に示す。

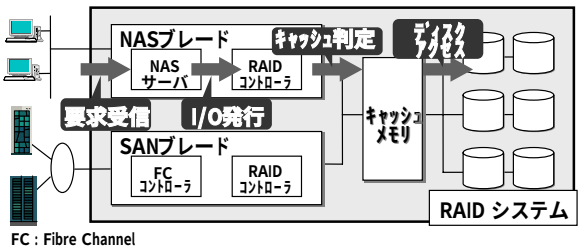


**図 1 RAID システム内蔵型 NAS システム構成**

RAID システムは、大容量のディスクドライブ、ディスクドライブ上のデータを一時的に配置するキャッシュメモリ、およびこれらの制御を行う RAID コントローラを装備している。

また、NAS ブレードには、ネットワークと接続し I/O 要求を処理する NAS サーバと RAID コントローラを搭載する。NAS サーバは Linux の改良版によって実装しており、NFS はカーネルスレッドにより実現した。

ファイルアクセス要求処理は以下のようになる。

1) NAS サーバがネットワークからの NFS 要求を受信
2) ファイルシステムへの要求として I/O を発行
3) RAID コントローラが要求された I/O のデータがキャッシュメモリ上にあるかを判定
   3-1) あればそのデータを使用
   3-2) なければ、ディスクドライブから一旦キャッシュメモリに転送
4) NAS サーバに要求されたデータを送る
5) NAS サーバから要求元への応答データを送信

## 3. 従来のキャッシュメモリ制御方式と課題

キャッシュメモリ制御は、Read 要求の処理と Write 要求の処理との 2 つに分けられる。いずれの要求の場合であっても、まず要求されたデータがキャッシュメモリ上に存在するかを判定し、もし存在すれば、そのデータを使用して Read/Write 要求を処理する。存在しない場合は、要求されたデータを置くための領域を確保してから、要求の処理を行う。

Read 要求の場合、データが存在しなければディスクドライブからデータを転送する必要があるため、キャッシュメモリに存在するかしないかで、要求に対する応答時間は大きく変化する。

一方、Write 要求の場合、キャッシュメモリ上に存在する場合としない場合とはそれほど変わらない。なぜなら、キャッシュメモリからディスクドライブへのデータ書き込みはバックグラウンドで実行されるため、キャッシュメモリにデータが転送された時点で要求が完了するためである。

このように、Read 要求がキャッシュメモリヒットするか、しないかは大きくファイルアクセス性能に影響を与える。これまで RAID システムでキャッシュメモリヒット率を上げる方法として、ディスクの一部の領域をキャッシュメモリ上に常駐する機能が提供されてきた。しかし、NAS ではそのような機能を使用することはできない。なぜなら、キャッシュメモリ上へ常駐する領域は、管理者があらかじめ指定する必要があるのに対して、ファイルシステム

‡ Atsushi Sutoh, Akihiko Sakaguchi, Yasuo Yamasaki
§ Hitachi, Ltd., Central Research Laboratory

上のデータの配置はファイルの生成/書き込み時に動的に変更されるためである。

## 4. データ種別によるキャッシュメモリ制御方式

NAS ではファイルシステムを使用してデータにアクセスする。そこで、ファイルシステムの情報を利用したキャッシュメモリのヒット率向上方法について検討をおこなった。

ファイルシステムは、ファイルやディレクトリを管理するためのメタデータと、ファイルの内容そのものであるユーザデータとの 2 種類のデータを用いる。そこで、これらメタデータおよびユーザデータのデータ種別に着目し、標準的なファイルアクセスベンチマーク実行時のディスクアクセスパターンを解析した。採取した情報は、データ種別、LBA(Logical Block Address)、アクセス回数である。結果を図 2 に示す。

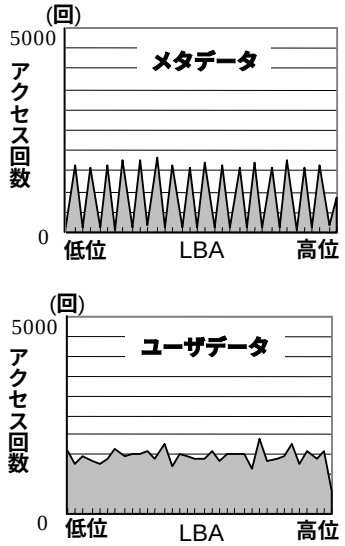


図 2　データ種別毎のディスクアクセス回数

ユーザデータはディスク上の領域に一様にアクセスが発生しているが、メタデータは一部の LBA に集中してアクセスしていることがわかる。このため、メタデータをキャッシュメモリ上に長時間保持できれば、キャッシュメモリヒット率を向上させることができると考えられる。しかし、現状のキャッシュメモリ制御では、メタ/ユーザデータを区別することなくキャッシュしているため、多量のアクセスが発生するユーザデータによって、メタデータがキャッシュメモリ上から追い出されてしまう。

そこで、メタデータとユーザデータとでキャッシュメモリ上での管理を区別する方式を設計した。方式の概要を図 3 に示す。本方式では、NAS サーバ上のファイルシステムが利用するデータ種別を RAID コントローラまで渡し、そのデータに基づいてキャッシュメモリ上のデータの配置を制御する。より具体的には、ファイルシステムがメタ/ユーザデータの区別をデバイスドライバに渡し、デバイスドライバは RAID コントローラ用の I/O コマンドフォーマットにそのデータ種別を入れて I/O 要求を発行する。そして I/O 要求を受け取った RAID コントローラは、データ種別に応じてメタデータ用キャッシュメモリおよびユーザデータ用キャッシュメモリを使い分ける。このように RAID システムのキャッシュメモリを管理することで、メタデータがより長時間キャッシュメモリ上に存在し、キャッシュメモリヒット率の向上が期待できる。

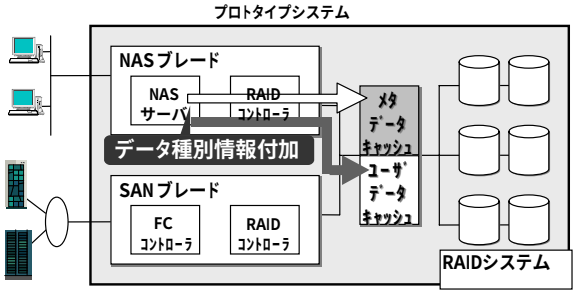


図 3　データ種別によるキャッシュメモリ制御

## 5. 評価

データ種別を利用したキャッシュメモリ制御方式を実装したプロトタイプを用いて、本方式の評価を行った。評価には、一般的なファイルアクセスを模擬したベンチマークを用い、以下のような設定で行った。

A)　データ種別を考慮しない
B)　メタ/ユーザのデータ種別をキャッシュメモリ管理に利用

設定 A、B に同様の負荷を与えた測定の結果を図 4 に示す。データ種別を用いる方式により、キャッシュメモリヒット率はおよそ 1.3 倍に上昇した。

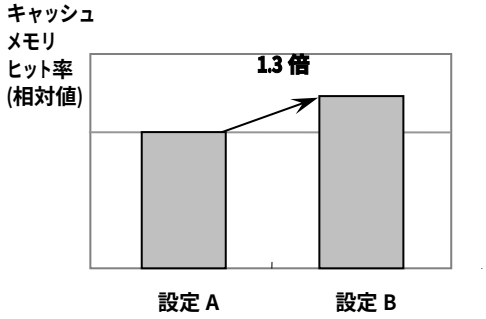


図 4　キャッシュメモリヒット率の比較

## 6. おわりに

ファイルシステムのデータ種別情報を用いたキャッシュメモリ制御方式により、RAID システムのキャッシュメモリヒット率が向上することを実証した。