# ＰＣサーバ向け「ハイブリットＲＡＩＤ」の開発（３）
# 2Ｆ－9　～可変長セグメントを用いたキャッシュ管理方式～

荒川 敬史，大枝 高，松並 直人，兼田 泰典，八木沢 育哉
（株）日立製作所　システム開発研究所

## 1　はじめに

PCサーバ向け「ハイブリッドRAID」は、RAID制御、キャッシュ管理等をソフトウェア（デバイスドライバ）によってPC本体のCPUを用いて行い、パリティ生成等をアクセラレータハードで行う。またデバイスドライバでPCの主メモリの一部を確保しキャッシュとして使用する。このキャッシュの管理方式において、メモリ領域の管理単位（セグメント）を可変長とした。可変長セグメント方式の採用により、キャッシュとしてのメモリ領域を100%有効に使用することができる（表１）。

ハイブリッドRAIDで採用しているディレイドパリティ生成方式はキャッシュにパリティを格納して、RAID5のライト性能を向上させる。このライト性能向上の効果は、キャッシュのヒット率が高いほど大きい。一般には高いキャッシュヒット率を得るためには、大きな容量のキャッシュが必要となる。しかしハイブリッドRAIDではPCの主メモリの一部をキャッシュとして割り当てて使用するため、大容量のキャッシュを持つことは期待できない。そのため高々数MBのメモリ領域でもキャッシュとして有効に使用できる方式が必要となった。

| | セグメント長 | メモリ領域使用効率 |
|---|---|---|
| 固定長セグメント | 2KB | 83.3% |
| | 4KB | 62.5% |
| | 8KB | 37.5% |
| 可変長セグメント | 可変 | 100.0% |

表１　メモリ領域使用効率（アクセス長4KB）

Development of "Hybrid RAID" for PC Servers(3)
Cache Management with Variable-length Segments
H.Arakawa, T.Oeda, N.Matsunami,
Y.Kaneda and I.Yagisawa
Systems Development Laboratory
Hitachi, Ltd.

## 2　固定長セグメント方式でのメモリ使用効率

ハイブリッドRAIDのキャッシュは、メモリ領域をセグメントに分割して管理することで実現する。セグメント長を固定長とすると、アクセスの範囲がセグメント境界に一致しない場合、セグメント内に不使用領域が発生する（図１）。この不使用領域の大きさはアクセス範囲とセグメントのずれに依存する。このずれが小さくなるようにセグメントを設定することが望ましいが、アクセス長は分布に偏りが見られるのに対し、アクセス範囲はアクセス対象ファイルの配置状態などに依存し、一定の傾向が見られるとは限らない。

OSが要求する標準的なアクセス長を4KBとすると、メモリ領域の使用効率に関して表１のような値が求められる。ただし固定長セグメント方式について、アクセス範囲とセグメントのずれには一定の偏りはなく、アクセスはシーケンシャルなものではないとした。セグメント長を小さくすれば不使用領域を小さくできるが、これは１アクセス当たりのセグメント数の増加を招く。１アクセス当たりに処理するセグメント数を抑える場合、固定長セグメント方式ではキャッシュとしてのメモリ領域の使用効率は低下する。

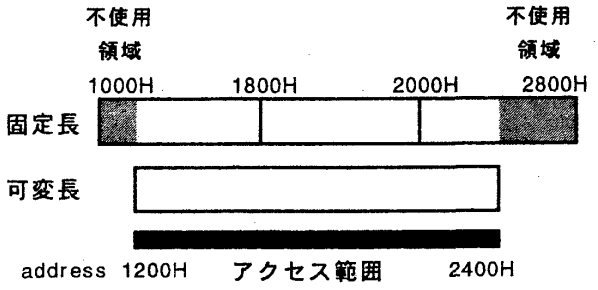


図１　固定長セグメント方式と
可変長セグメント方式

## 3　可変長セグメント方式の効果

上記の使用効率を上げるため、ハイブリッドRAIDでは、キャッシュのセグメント長をアクセスにあわせて決定する可変長セグメント方式を採用した。可変長セグメント方式では使用していない領域を管理下に置き、上位からのアクセスにあわせてそのつど割り当てる（図１）。すなわちセグメントは常に必要な長さだけ割り当てられる。セグメント内で不使用領域が発生することはない。これにより使用効率は100%となり、キャッシュとして使用するメモリ領域の大きさに対し、最大限のヒット率が得られる。

## 4　可変長セグメント方式の構成

可変長セグメント方式を実現するため設けた、可変長セグメント方式の構成上の特徴は、主に３つある。

１つは、ディスク上アドレスから任意のセグメントを検索するテーブルの管理方式である。可変長セグメント方式では、このテーブルをセグメント毎に管理することはできない。固定長セグメント方式と異なり、セグメントとディスク上アドレスとの対応が不規則となるからである。よって、このテーブルではセグメントを最小アクセス単位（512バイト）毎に管理し、検索できるようにした。

２つめは、使用していない領域（未使用の領域）の管理である。未使用領域を管理するためのテーブルを設け、未使用セグメントとして登録するようにした。キャッシュとして使用するメモリ領域は初期化の時点では一つの未使用セグメントとしてテーブルに登録し、アクセスに応じて複数のセグメントに分割して割り当て、使用する。使用するセグメントはこのテーブルから削除する。いまだ使用しない残りの領域は未使用セグメントとしてテーブル上に登録し続ける。格納している内容をディスク上に反映した（使用済みの）セグメントもこのテーブルに未使用セグメントとして登録し再利用を待つ。このテーブルは未使用セグメントの大きさで検索でき、セグメント割り当てに対し、最適な未使用セグメントを選択できる。

３つめは、メモリアドレス上で隣接するセグメントの管理である。隣接するセグメントを管理することで、メモリ領域の細分化を防ぐようにした。個々のセグメントに対し、隣接するセグメントを結ぶリスト構造を設けた。アクセスに応じて１個のセグメント（すなわち連続するメモリ領域）を分割する際には、このリスト構造を用いて分割後のセグメント同士を関連付ける。使用済みとなったセグメントを未使用セグメント管理テーブルに登録する際には、このリスト構造を用いて隣接するセグメントの使用状況を調べる。隣接するセグメントも未使用状態であれば、これらのセグメントを結合して一つのセグメントに戻す。

以上の構成により可変長セグメント方式のキャッシュを実現した。

## 5　まとめ

キャッシュのセグメント長をアクセスに応じて決定する可変長セグメント方式を採用することによって、キャッシュとして使用するメモリ領域の使用効率を100%とした。これによりPCの主メモリの一部を使用する比較的小容量のキャッシュにおいても最大限のヒット率を得られるようにした。

## 参考文献

[1]David A. Patterson, Garth Gibson, and Randy H. Karz. A Case for redundant arrays of inexpensive disks (RAID), Proc. of SIGMOD(Cicago, IL), June 1988.