ディスクシステムにおけるキャッシュメモリ高可用化制御方式

3G－5

森下 昇，山本 彰
(株) 日立製作所システム開発研究所

1 はじめに

近年、無停止化要求の高まりの中で、ディスクシステムは、各構成要素、たとえば、プロセッサ、キャッシュメモリ [1]、ディスク [2] 等を冗長構成とし、可用性を確保している。各構成要素に障害が発生した場合、障害の発生した部位の閉塞処理と回復処理を実行する必要がある。ディスクシステムを無停止化するためには、この閉塞処理と回復処理を、ホストコンピュータからのリード / ライト要求と並行して実行できなければならない。

このような状況をふまえ、複数のプロセッサと複数の閉塞単位からなるキャッシュメモリを有するディスクシステムへの適用を目的とした、キャッシュメモリ高可用化制御方式を開発した。本方式では、キャッシュ部分障害時の障害部位の閉塞処理、および、プロセッサ障害時に発生するキャッシュメモリ管理情報の矛盾状態の回復処理を、ホストコンピュータからのリード / ライト要求を受け付けながら実行することを可能とする。

2 ディスクシステムの構成

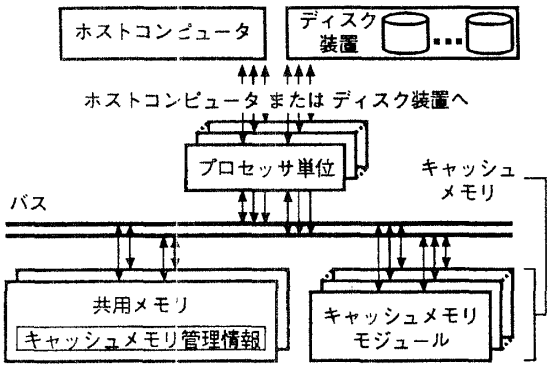

図 1: ディスクシステムの構成

本方式を実現したディスクシステムの構成を図 1に示す。複数のプロセッサ、閉塞単位である複数のモジュールからなるキャッシュメモリ、キャッシュメモリ管理情報を格納する共用メモリを共通のバスにより接続している。共用メモリは高信頼化のため 2 重化しており、キャッシュメモリ管理情報等プロセッサ間の制御情報を格納する。また、リード / ライト処理は、キャッシュメモリを介して実行する。

3 キャッシュメモリ管理構造

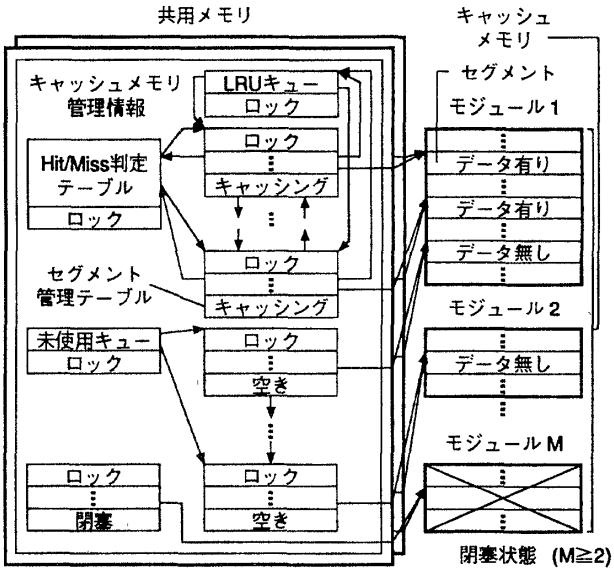

図 2: キャッシュメモリ管理構造

キャッシュメモリ管理構造を図 2に示す。キャッシュメモリ管理情報内にはキャッシュメモリの割り当て単位であるセグメント毎にセグメント管理テーブルを設けている。セグメント管理テーブルには、3 つの状態が存在する。

キャッシング状態 データを格納している状態。ディスク内の領域に対応した Hit/Miss 判定テーブルと、データの効率的な管理のために双方向ポインタの LRU キューに接続する。

空き状態 データを格納していない状態。未使用キューに接続する。

閉塞状態 モジュールの閉塞により使用禁止となっている状態。どこにも接続しない。

キャッシュメモリの割り当て / 解放処理時の性能を考慮し、未使用キューは単方向ポインタで構成している。また、プロセッサ間の排他処理のため、セグメント管理テーブル、Hit/Miss 判定テーブル、各キューにはロックを設けており、各情報の参照時または更新時に取得する。

4 キャッシュメモリ高可用化制御方式

ディスクシステムの無停止化のために、リード / ライト処理との同時実行性、高可用化のために異常終了時の再実行性の実現を前提条件とした。

4.1 キャッシュメモリモジュール閉塞処理方式

アクセスエラーが多発した場合、当該モジュールを閉塞する。閉塞処理の具体的な内容は、キャッシング

A High Available Control Method of a Cache Memory for Disk Systems
Noboru MORISHITA, Akira YAMAMOTO
Systems Development Laboratory, Hitachi, Ltd.

状態、および、空き状態の閉塞対象モジュールのセグメント管理テーブルを見い出し、閉塞状態とすることである。

しかし、未使用キューが単方向ポインタのため、空き状態のセグメント管理テーブルを、直接キューの途中から外すことはできない。このため、図3に示すように、まず、キャッシング状態のセグメント管理テーブルを閉塞状態として(ステップ1)、次に、未使用キューの先頭からサーチを行い、閉塞対象モジュールのセグメント管理テーブルを閉塞状態にする(ステップ2)方法をとった。

一方、閉塞処理を実行中には、リード/ライト処理に対し、閉塞対象モジュールの空き状態のセグメント管理テーブルの新規割り当て(キャッシング状態への遷移)を禁止した。

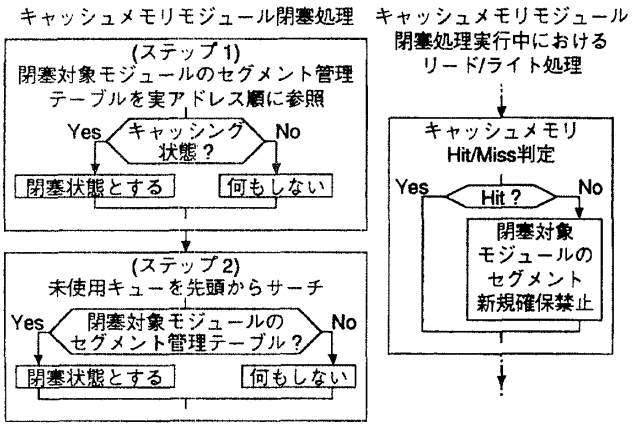

図 3: キャッシュメモリモジュール閉塞処理方式

リード/ライト処理との同時実行性は、キャッシュメモリ管理情報内の各情報に設けたロックにより実現した。また、再実行性を確保するため、異常終了した場合、再実行はキャッシュメモリモジュール閉塞処理の最初から行うこととした。

また、未使用キューのサーチ時のキューロック時間は、1回あたり1ms程度とし、リード/ライト処理への影響を抑えた。

4.2 キャッシュメモリ管理情報矛盾状態回復処理方式

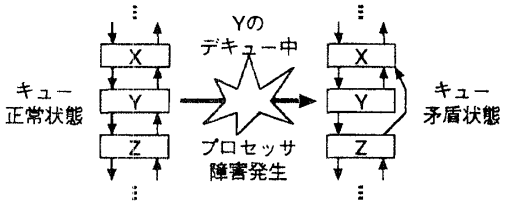

図 4: デキュー時のプロセッサ障害

図4に示すように、キャッシュメモリ管理情報のキューを操作中のプロセッサが障害を発生すると、キューが矛盾状態に陥る。この状態を図5に示す方式により、正常状態に回復する。

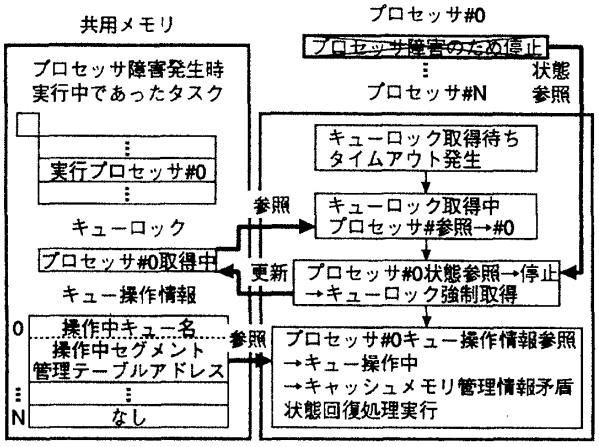

図 5: キャッシュメモリ管理情報矛盾状態回復処理方式

以下、具体的に説明する。プロセッサ#0が、キュー操作中に障害を起こしたものとする。なお、共用メモリに、キュー操作時の操作内容を設定するキュー操作情報を設ける。回復処理の実行契機は、プロセッサ#Nが、障害が発生したプロセッサ#0によって取得されているキューロックを取得しようとして、ロック取得待ちタイムアウトが発生した時である。このとき、取得しようとしたキューロックを取得中のプロセッサ#0の状態を参照し、プロセッサ#0が停止しているため、このキューロックを強制取得する。強制取得後、プロセッサ#0のキュー操作情報を参照する。キュー操作中であったならば、キューの矛盾状態発生箇所を特定し、状態に応じてキュー操作を進めるかあるいは戻すことで正常状態に回復する。以上により、キューの矛盾状態を回復することができる。

本回復処理は、リード/ライト処理の中のロックタイムアウト処理として実行され、かつ、回復時間そのものも短いため、性能的な影響は小さい。また、再実行性を確保するため、回復処理中のキュー操作時にもキュー操作情報を設定することとした。

5 まとめ

ディスクシステムの無停止化の一環として、複数のプロセッサと複数の閉塞単位からなるキャッシュメモリを有するディスクシステムへの適用を目的とした、キャッシュメモリ高可用化制御方式を提案した。

参考文献

[1] Smith,A.J.,"Disk Cache - Miss Ratio Analysis and Design Considerations",ACM Transactions on Computer Systems,Vol.3,No.3,pp.161-203,August 1985.

[2] David A.Patterson et.al, "A Case for Redundant Arrays of Inexpensive Disks(RAID)", ACM SIGMOD conference proceedings,Chicago,IL., pp.103-116,June 1-3 1988.