オペレーティング・システム　47－5
（１９９０．６．８）

大容量高速記憶装置（ＬＨＳ）の基本方式

西　直樹、大野直哉、妹尾義樹、中崎良成、藤原良二[1]、実宝　昭[2]、増本健三[2]

日本電気㈱Ｃ＆Ｃシステム研究所、[1]日本電気コンピュータシステム㈱、
[2]日本電気㈱コンピュータ技術本部

本稿では、大容量高速記憶装置ＬＨＳ（Large-capacity High-speed Storage）の記憶管理方式を報告する。ＬＨＳは階層記憶システムであり、半導体記憶装置とその他の比較的低速な記憶装置（磁気ディスク、光ディスク、磁気テープ等）から構成されている。ＬＨＳは一種の仮想記憶システムとなっており、この基本方式を「ワンレベル・ファイル・ストア」と称している。ＬＨＳ実験システムをスーパーコンピュータの拡張記憶として構築し、大規模ＬＵ分解（３万２千元）プログラムを用いて評価した。全経過時間（１０時間４０分）に占めるＬＨＳとのデータ転送時間比は２１．７％であり、ＬＨＳが大規模数値計算環境で良好な性能を発揮することが確認されている。

The Architecture of LHS
(Large-capacity High-speed Storage)

Naoki NISHI, Naoya OHNO, Yoshiki SEO, Ryousei NAKAZAKI,
Ryouji FUJIWARA[1], Akira JIPPOU[2], Kenzou MASUMOTO[2]

C&C Systems Research Laboratories, NEC Corporation
[1]NEC Computer Systems Corporation
[2]Computer Engineering Division, NEC Corporation

4-1-1, Miyazaki, Miyamae-ku, Kawasaki, Kanagawa 213 JAPAN

This paper presents memory management techniques developed for the LHS(Large-capacity High-speed Stogage). The LHS is a hierarchical file system with two kinds of file devices; a high-speed semiconductor storage and other relatively slower file devices(such as magnetic disks, optical disks and magnetic tapes). The LHS architecuture is called "ONE LEVEL FILE STORE", which is a kind of virtual memory system. An experimental system was developed as an extended storage for supercomputers. An LU decomposition program was also designed for the LHS evaluation. This program solved a huge dence matrix of 32K²(8GB) with 10 hours and 40 minitues. The LHS data transfer time is 21.7% of all consuming time, which means the LHS is an effective file system for supercomputing environments.

1．はじめに

本報告は、通産省スーパーコンプロジェクトの一貫として研究開発された、大容量高速記憶装置ＬＨＳ（Large-capacity High-speed Storage）に関するものである*。ＬＨＳ開発の目的は、「スーパーコンピュータに見合った大容量かつ高速な記憶装置」を研究することにあり、本目的の達成のため、ＬＳＩテクノロジや実装技術を含めたハードウェア装置開発、新しい入出力アーキテクチャを実現するための制御ソフトウェア開発、新装置／新方式の有効性実証のためのアプリケーション開発を行った。本稿では、このＬＨＳでＯＳ機能として重要な役割を果たしている、ワンレベル・ファイル・ストア方式に基づくファイル・システム統合アーキテクチャを中心に報告する。

2．ワンレベル・ファイル・ストアの狙い

計算機の記憶階層分類において最も一般的なのは、レジスタ、キャッシュ、メモリ、ディスク（Ｉ／Ｏ）といったハードウェアの物理的構成から見た記憶階層分類である。しかしながら、論理的な記憶階層分類は、ソフト／システム設計者の観点により大きく異なる。例えば、コンパイラ設計者はレジスタとメモリが主要な着目点であり、レジスタの最適利用のために骨身を削ることになる。リンカ設計者はコンパイラとＯＳの間で最適かつ拡張性のある空間分割／配置に励む。

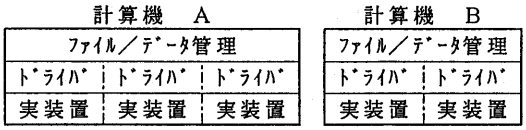
同様な意味で、ＯＳ設計者から見た記憶階層は、メモリ系とファイル系が大きな分かれ目であろう。ほとんどの計算機ＯＳはメモリとファイルを異なるものとして利用者プログラムに提供している。また、ＯＳ内部においてもメモリ系とファイル系の独立性は高く、各々の目的／実現技術は異なったものとなる。他方、ＯＳにおいてメモリとファイルを統合した環境として multics に端を発するワンレベル・ストアがあり、ＩＢＭ社のいくつかのシステムで実際に実現されている［１，２］。ＬＨＳのワンレベル・ファイル・ストアは、メモリとファイルは別物として捉えるが、ファイル系に関してより統合された環境をサポートすることを狙ったものである。ＬＨＳの基本構想を定めるにあたって認識した問題点／目標を以下に示す。

①入出力速度／容量の向上

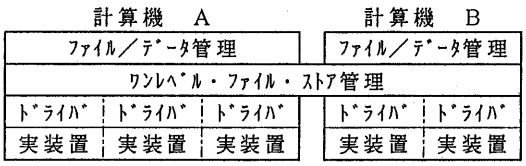
Ｉ／Ｏ性能は、入出力装置自身の特性に依存するところが大きい。半導体ディスク、磁気ディスク、光ディスク、磁気テープ等、転送速度と容量、記憶特性の異なる様々な記憶装置が存在する。この特性の差異を出来るだけ吸収し、より高速、かつ大容量のファイル・システムを実現することがＬＨＳの目的である。

②ファイル・システム設計の階層化

一般的にＯＳが異なると、ファイルシステムも異なる。新しいＯＳには新しいファイルシステムを構築する必要があるが、ファイルシステムの下位層のプログラム機能は、異種ＯＳであってもほぼ同等機能である場合が多い。



（a）従来のファイル管理構築法



（b）ワンレベル・ファイル・ストア環境

図１　ファイルシステムの構築

図１にワンレベル・ファイル・ストアを用いたファイル管理法を示す。従来、ファイルシステムは実入出力装置に対するドライバの上位に構築され、基本的には２階層の構造となっている。ワンレベル・ファイル・ストアでは、これを３階層に広げ、ワンレベル・ファイル・ストア管理を中間層として設ける。ワンレベル・ファイル・ストアが実現する機能は、データ管理中の物理バッファＩ／Ｏ機能と、ファイル管理における名前と実体との対応を保持する機能である。これにより、異なるファイルシステムや装置ドライバの構築を標準化し、かつ実現を容易にすることが可能となる。大型計算機等では、ファイル／データ管理層として、通常のファイル、仮想記憶のＢＫＳＴ（BacKing STore: 仮想記憶のページアウト用二次記憶装置）、データベース管理等、目的に応じた複数のファイル管理システムが同時に動作しており、これらの構築も容易になる。また、最上位のファイル／データ管理をネットワーク上で分散させることも可能である。実際、今回試作したＬＨＳでは、ファイル管理／データ管理層が動作する計算機とワンレベル・ファイル・ストア以下のファイル記憶が

---

*本研究は通商産業省工業技術院大型プロジェクトの一貫として新エネルギー・産業技術開発機構（ＮＥＤＯ）から委託を受けて、実施したものである。

	ディスクキャッシュ	半導体ディスク	拡張記憶
記憶装置構成法		ＤＲＡＭインタリーブ	
ハード的なアクセス方法	Ｉ／Ｏ命令による	Ｉ／Ｏ命令による	ＣＰＵ命令による
ＯＳ内での捉え方 (ｱﾌﾟﾘｹｰｼｮﾝへの見せ方)	基本的にはソフトからは意識する必要なし	ディスクと互換性をもたせるために、ドライバで、シリンダやトラックをシミュレート	半導体ディスクと同じ見せ方に加え、ディスクをシミュレートしないファイル（従来ファイルと非互換）も実現

表１　半導体入出力装置の分類

動作する計算機を分離しており、両者の間は専用ハードウェア・インタフェースで結合している。

③多種／多様な半導体記憶装置の一元化

現在、汎用大型計算機やスーパーコンピュータには３種類の半導体記憶装置が存在する（ディスク・キャッシュ、半導体ディスク、拡張記憶）。これらの装置本体は、比較的安価な DRAM をインタリーブして使用して構成しているものの、ハードウェアの細部や、ソフト上での取扱いが異なり、ユーザからみた場合には別物になっている（表１参照）。ＬＨＳではこれらを一元化することを図る。

３．ワンレベル・ファイル・ストア

前記の目的を考慮し、ワンレベル・ファイル・ストアは、高速な半導体記憶装置とその他の入出力装置を階層接続した構成を基本としている。また、論理的には上位ファイル管理／データ管理とのインタフェースを以下のように定めている。

①アドレス空間

図２にワンレベル・ファイル・ストアのアドレス空間を示す。ワンレベル・ファイル・ストアは2^{68}Ｂの論理記憶空間を有し、ＭＢ単位でアドレス可能である。アドレス空間（48bit）は、３２ビットの空間ＩＤ（SPID: SPace IDentifier）と１６ビットの空間内ブロックアドレス（BID: Block IDentifier）からなる。ファイル管理層からの空間取得要求に際してアドレス空間を確保し、確保した３２ビットのアドレス空間番号（ＳＰＩＤ）を引き渡す。

ファイル／データ管理層は、空間へのアクセスに先立ち、空間の使用をワンレベル・ファイル・ストアに通知する。OPEN 要求された空間に対して１６ビットの一時的な空間番号（CSPID: Current SPace IDentifier）を割り当てる。ブロックの転送、ステージング等以降のアクセスは、このＣＳＰＩＤとブロック内アドレスＢＩＤの３２ビットのアドレスにより行われる。即ちファイル／データ管理層は、同時に最高６４Ｋ個迄の空間に対してアクセスすることが可能である。

このアドレッシング方式は、基本的にセグメンテーションを用いたものであり、セグメントの最大サイズは６４GBに限定される。６４GBを越える大空間取得要求に対しては、ワンレベル・ファイル・ストア側が連続した SPID を割り当てることで、大空間の使用を可能にしている。

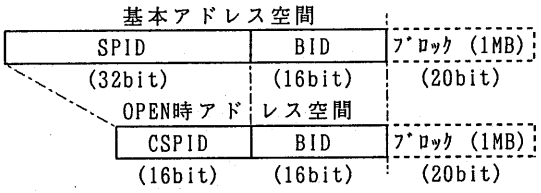


図２　アドレス空間

②記憶管理用コマンド

上記のアドレス空間に対して行うことが可能な操作には以下のものがある。

・空間管理コマンド
　－空間の取得／解放
　－空間の OPEN/CLOSE
　－ステージイン／アウト
　　（＋その他制御用コマンド）
・空間アクセス・コマンド
　－空間ページへの READ/WRITE

ここで、ワンレベル・ファイル・ストアにおける空間の考え方について説明する。ワンレベル・ファイル・ストアにおいて、空間は属性を備える。属性には以下のようなものがある。

・性能属性
　－低速、中速、高速
・記憶属性
　－パーマネント／テンポラリ
・アクセス属性
　－シーケンシャル、ランダム
・信頼性属性
　－ジャーナル、二重化等

属性の指定は空間生成時に行う。この属性の指定／変更等は、空間の OPEN 時にも行えることが望ましいが、現段階では OPEN 時の指定は許していない。これは、空間情報を保持するための制御テーブル構造が空間属性によって異なるためであるが、将来的には OPEN 時の指定／変更を可能にしたいと考えている。

４．ＨＰＰ－ＬＨＳ実験システム

図３にＨＰＰ－ＬＨＳシステムの構成を示す。ＨＰＰ（High-speed Parallel Processors）は富士通㈱殿が制作された並列スーパーコンであり、４台のローカルメモリを有するベクトルプロセッサ（２GFLOPS×３台、４GFLOPS×１台）と、これらを接続する共有記憶装置から構成されている。この共有記憶装置にＬＨＳも接続される。ＬＨＳの記憶装置は半導体記憶部（４GB）とディスク装置群から構成されている。ＨＰＰ－ＬＨＳを結合しているデータ転送パスは１．５GB/secの転送能力を有している。また、ＬＨＳのような仮想化ファイルシステムでは、ある論理アドレスに対応するデータが高速メモリ部に存在するのか、ディスク装置部に存在するのかを極めて高速に判定する必要がある。この目的のため、ハードウェア・アドレス変換ユニットを備えて、高速メモリ部に HIT するデータに対しては、制御プロセッサを介入させることなく、直接ハードウェアでデータ転送を行うことを可能としている。

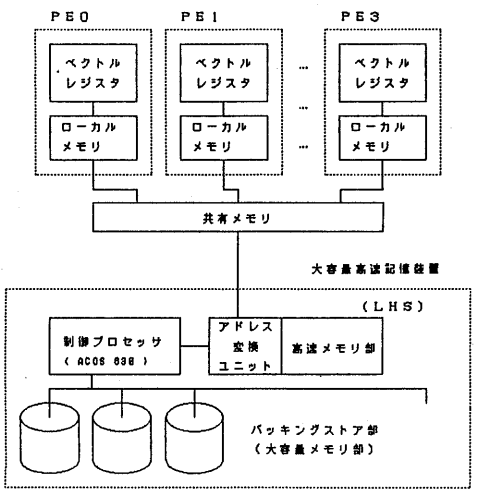


図３　ＨＰＰ－ＬＨＳシステムの構成

アドレス変換ユニットの論理構成を図４に示し、変換動作を説明する。ＨＰＰから転送要求とともに与えられた OPEN 中の空間アドレス（32bit）は、以下の手順で高速メモリ部に対象データが存在するかどうかがチェックされる。

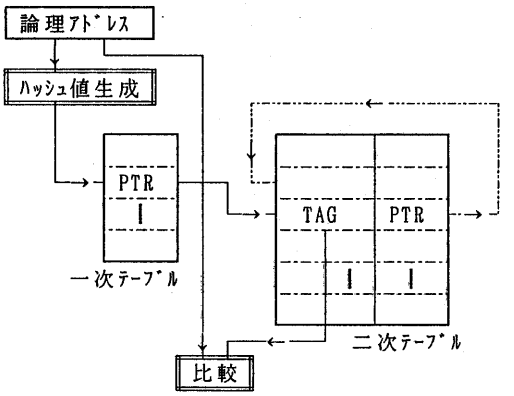


図４　アドレス変換ユニットの構成

①：論理アドレスをハッシングし、一次テーブルへのポインタを求める。
②：①で得られたポインタを用いて一次テーブルを参照し、二次テーブルへのポインタを求める。
③：②で得られたポインタを用いて二次テーブルを参照する。二次テーブル内のエントリには、
　TAG フィールド（論理アドレス）、
　PTR フィールド（二次テーブルへのポインタ）
　FLG フィールド（該当エントリの TAG/PTR の有効／無効を示す）
　の３つのフィールドが有り、これらを用いて次の動作を行う。

・TAG/PTR が無効であれば MISS HIT と判定
・TAG がチェック対象の論理アドレスと一致すれば、HIT と判定。
・TAG がチェック対象の論理アドレスと一致しない場合、PTR を再び二次テーブルへのポインタとして③の頭に戻る。

ここに示した変換方式は、いわゆる「逆引きテーブル」であり、二次テーブルのエントリ番号が、高速メモリ部の物理アドレスに対応する。即ち、二次テーブルのエントリ数は、ＬＨＳの高速メモ

リ部（４GB）の全ページ数である４Ｋエントリに等しい。また、本方式において、一次テーブルは高速メモリ部の特定ページに障害が発生した場合にページ単位での切り離しを可能とするために設けられている。このアドレス変換ユニットは、超高速のアドレス変換を可能とするためにＳｉＬＳＩに加えて３種のＧａＡｓ素子を使用し、これらを液浸冷却する方式が採用されている。

次に、ＨＰＰ－ＬＨＳシステムの構成をファイルシステムの面から説明する（図５）。ＨＰＰ－ＬＨＳ間は、物理的に２種類のパスにより接続されている。第一のパスは空間管理系コマンドパスで、9,600 bps 無手順回線を用いている。通常、空間の作成／削除等のコマンドは、アプリケーション中の出現頻度が少なく、実験システムでは簡単な接続法を採用している。第二のパスは、空間アクセス系コマンド（READ/WRITE）の要求パスであり、転送要求は頻繁に発生、かつ処理遅延は性能低下を招くため、専用ハードウェアインタフェースを用いている。この第二のパスは１ＭＢブロックのデータ転送パスと共用しており、６４Ｂのデータバス巾を備えている。

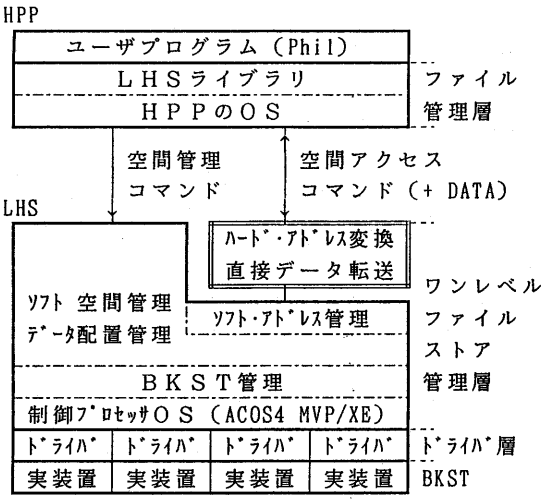


図５　ＬＨＳファイルシステム構造

ファイルシステムとして、最上位のファイル／データ管理層は、ＨＰＰ上で動作する。アプリケーション中文字列で表現されるファイル名等をワンレベル・ファイルの空間番号に対応させることは、ＨＰＰ側で行われる。また、本来この層で、ファイルの論理レコードと物理レコードの対応をとる必要があるが、今回の実験システムでは暫定的に論理レコードと物理レコードは等しいものとし、ユーザプログラムのデータ領域との直接データ転送を行っている。基本的に今回のインプリメンテーションでは、データ管理にあたる機能は、ＨＰＰ上の主力言語である Phil に付属するＩ／Ｏ用ランタイム・サポート・ルーチンの形で提供されている。ファイルシステムの内、ワンレベル・ファイルを実現する層以下は、ＬＨＳの制御プロセッサ内で実現されている。

ここで、ＨＰＰ上のアプリケーション・プログラムの動作が、システム全体としてどのように動作するかを説明しよう。

・空間の確保

ファイル名と記憶容量をパラメタとして空間を作成する。実験システムにおいては、空間属性はパーマネントなシーケンシャル空間として固定されている。ＬＨＳ側では新規作成した空間番号を登録するとともに、この空間を管理するために必要となる最小限のページテーブル・エントリを作成する。

・空間の OPEN

ファイル名を用いて空間を OPEN する。ＨＰＰ上でファイル名は空間番号（SPID）に置き換えられる。ＬＨＳは OPEN された SPID に対して CSPID を与える。また、OPEN された空間に対応するページテーブルの一部を活性化し以降のアクセスに備える。

・空間へのアクセス

ＨＰＰから発せられたアクセス要求は、ＬＨＳのハードウェア・アドレス変換ユニットでチェックされ、HIT しているならばハードウェアで直接データ転送を行う。MISS HIT した場合には、制御プロセッサによりアドレス変換テーブルの再設定や、必要であればディスクとのデータ転送が行われる。新規アドレスに対するアクセスが発生した場合、この時点でページ・テーブル・エントリが割り当てられる。また、新規にデータがディスクに追い出される場合、ＢＫＳＴの割り当てもこの時点で行われる。

・空間の CLOSE

ＨＰＰから CLOSE 要求が発せられると、ＬＨＳは処理要求受付後、直ちにＨＰＰに完了報告を通知する。CLOSE 処理に伴って必要となる高速メモリ部からディスクへのデータ追い出しは遅延をともなって処理される。ディスクへの追い出しが完了しない間に対象空間が再度 OPEN された場合、高速メモリ部に残存していたデータはそのまま再活用されることになる。

・空間の削除

ＨＰＰ側ファイル管理ではファイル名が抹消される。ＬＨＳ側においても、削除された空間に対して与えていた実資源（高速メモリ、ＢＫＳＴ）を回収し、ページテーブル等の空間管理のために保持していた各種テーブルも解放する。

５．ＬＨＳ制御ソフトウェアの構造

制御プログラムの構造を図５に示す。制御プログラムは大きく４つのＩ／Ｏ管理モジュールと、５つの内部処理モジュールから構成されている。

ＴＩＯ：オペレータ端末Ｉ／Ｏ
ＦＩＯ：空間管理コマンド用Ｉ／Ｏ
ＸＩＯ：新装置管理
ＤＩＯ：並列ディスクＩ／Ｏ
ＣＩＭ：制御ソフト全体管理
ＣＩＰ：コマンド・インタプリタ
ＡＴＭ：アドレス変換管理
ＳＰＭ：空間管理
ＴＩＭ：タイマ

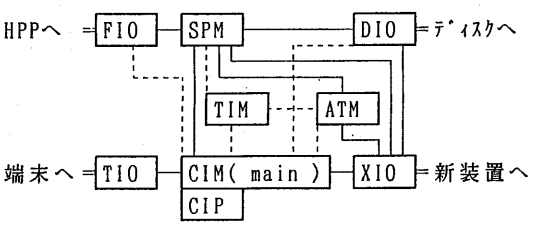


図５　ＬＨＳ制御ソフトの構造

すべてのモジュールは並列動作可能であることを前提としており、合計１８個のタスクで実現している。タスク間通信はメッセージ交換を基本としているが、ユーザ・データ・ブロック（１ＭＢ）等の大量データに関しては、バッファアドレスをメッセージとし、特に必要がない限りデータ自身のコピーは行わない。今回このような構造を採用したのは、ＬＨＳ制御ソフトを分散環境でも効率的に動作可能としておくためである。

次に、ワンレベル・ファイル・ストアの実現を図っている空間管理部について説明する。空間管理における特徴として、次の４点があげられる。

・論理ＢＫＳＴの提供
・空間管理テーブルの仮想化
・空間属性に応じた空間管理テーブル構造の採用
・動的データ配置制御

以下、この４点について説明する。

①論理ＢＫＳＴの提供

ＬＨＳに使用される物理的なＢＫＳＴ装置は、制御ソフト内で、論理的に１ＭＢページからなる論理ＢＫＳＴ装置として仮想化されて取り扱われている。本構成により記憶管理の上位層は、ＢＫＳＴが並列ディスク化されていることや、二重化されていること等を意識しないですむ構造になっている。論理ＢＫＳＴと物理ＢＫＳＴとのマッピングはシステム構成定義によりユーザが規定することが可能である。

②空間管理テーブルの仮想化

ワンレベル・ファイル・ストアにおけるユーザアドレス空間の最大量は2^{68}Ｂである。仮に、この超巨大な記憶空間（2^{48}ページ）を保持するために必要となる管理用テーブルを１ページ当たり４Ｂとするならば、管理オーバヘッドは3.8^{-6}（$2^{50}/2^{68}$）となり、極めて効率的な記憶管理が実現可能である。しかし、ここで問題となるのは記憶管理テーブルの絶対量自身である。相対的には優秀な記憶効率が得られるとしても、記憶管理テーブルに2^{50}Ｂが必要であり、記憶管理テーブル自身の記憶管理が最大の問題点となる。

この問題に関し、ＬＨＳでは記憶管理テーブルを仮想化することで対応している。図６に示すように、制御プロセッサメモリ上には、高速メモリ部とＢＫＳＴ間のデータ転送用バッファ（１ＭＢ×４枚）に加えて、記憶管理テーブル用のキャッシュ領域（４ＫＢページサイズ）を用意している。このキャッシュを使用するのは、OPEN 中の空間のページ・テーブル、及び、BKST 管理用テーブルである。これらのテーブルは、４ＫＢ単位に制御プロセッサ・メモリ上に展開され、必要に応じてディスク上のテーブルと交換される。ページ枠は、リファレンス・カウンタとページ・ロック機構を併用し、ＬＲＵアルゴリズムで管理されている。

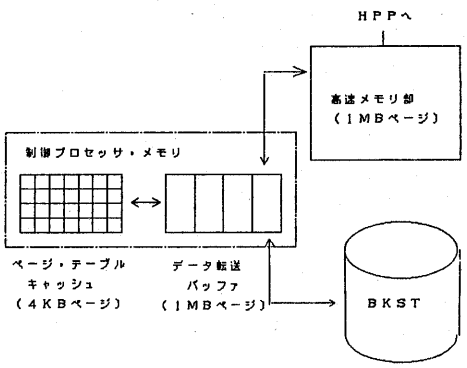


図６　ページテーブルの仮想化

③空間属性に応じた空間管理
テーブル構造の採用

前項で示したように、ＬＨＳにおいてはユーザデータ自身が仮想記憶化されるとともに、空間管理用のテーブルも仮想化されている。従って、空間管理テーブル自身のミスヒットを少なくすることも重要となる。そこで、ＬＨＳでは空間属性に応じて空間管理テーブルを異なる形式で表現している。

図７はシーケンシャル空間に対する空間管理テーブルを示している。シーケンシャル空間に対するページテーブルは BID をオフセットとして用いる標準的なテーブル構成である。ページテーブルは４ＫＢ単位にディスクと交換されるが、シーケンシャル・アクセスされる限りページテーブル・ミスは多発しない。

他方、ランダムアクセス空間に対しては、図８に示すテーブル形式を採用している。この形式は基本的にハードウェア・アドレス変換テーブルと同じ形式であり、ハッシングとポインタを用いて検索を行う。このテーブル形式はランダム空間中のすべてのアドレスが実際に使用された場合には効率が悪いが、スパース空間として使用された場合に効果を発揮する。即ち、実際に書き込みが発生している空間アドレスに対してのみ、ページテーブル・エントリが割り当てられる。

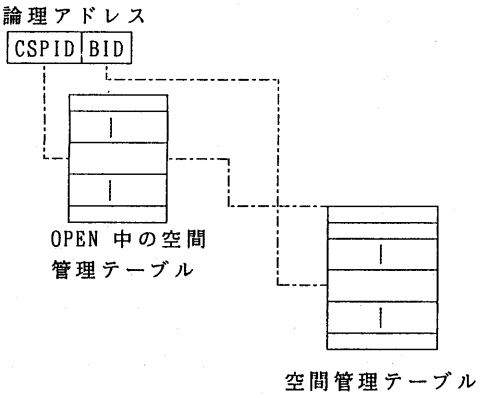


図７　シーケンシャル・アクセス・ファイルの管理

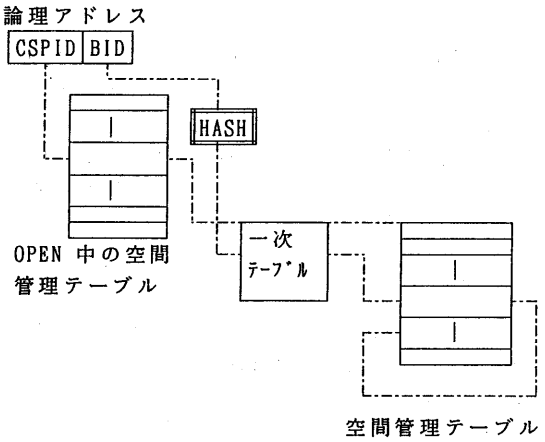


図８　ランダム・アクセス・ファイルの管理

④動的データ配置制御

高速メモリ部とＢＫＳＴ部の最適な記憶配置を目的とし、動的データ配置制御を行うために２種類の機構の取り込みを図っている。

・アプリケーションからのステージイン／アウト要求に基づいた記憶配置制御

ＬＨＳに対する記憶管理用コマンドの一種として、ステージイン／アウト・コマンドが用意されている。このコマンドは、アプリケーションプログラムから記憶配置制御を行う場合に利用する。また、ＯＳの内部からこの機能を使うと有効な場合もある。例えばＪＯＢ管理において、ＪＯＢ実行開始に先だって、ユーザデータセットをプリロードして、処理高速化を図ることが可能となる。

・ＬＨＳに対する記憶参照履歴を用いた、自動データ配置制御

ＬＨＳは、ＨＰＰから参照されたアドレスの履歴をバッファ中に保持している。このバッファは、ＦＩＦＯとして管理されており、バッファ更新は、高速メモリ部へのミスヒット発生時、あるいは高速メモリ部へのヒットが１Ｋ回程度発生した時点を契機として行われる。また、このバッファの各エントリは、OPEN 中の空間ごとにリンクされている。基本的にはこの参照履歴情報を用いて自動データ配置制御を行うことが可能である（現状では未完の機能）。一般にシーケンシャル空間に対するプリフェッチ等が考えられるが、より多くの参照パターンに対して自動配置アルゴリズムを検討して順次機能拡張をしていく必要がある。この機構の性質上、空間単位での最適化と全体としての最適化のバランスの取り方にも工夫する必要がある。

６．性能評価

ＨＰＰとＬＨＳを結合したシステムとして評価した結果を示す。評価システムは平成元年１１月に富士通㈱沼津工場において実際に結合し、システム検査を実施後、各種評価を実施した。評価は、基本性能特性評価とアプリケーション時性能評価の２段階にわけて行っている。表２に基本性能を

示す。この結果は、ＨＰＰ上の性能測定プログラムを用いて経過時間ベースで測定したものである。従って、この結果は、ＨＰＰ上のＩ／Ｏサポートルーチン、ＬＨＳ制御ソフト、ＨＰＰとＬＨＳ双方のハードウェア等のすべてのオーバヘッドが含まれた実効性能である。

性能測定ｹｰｽ	処理時間 (msec)	処理速度 (MB/sec)
高速ﾒﾓﾘ登録時	268	3.73
ﾘｰﾄﾞ・ﾋｯﾄ時	0.964	1,037
ﾗｲﾄ・ﾋｯﾄ時	0.973	1,028
ﾃﾞｨｽｸ書き込み時	898	1.13
ﾃﾞｨｽｸ読み込み時	856	1.17
ﾃﾞｨｽｸ・ｽﾜｯﾌﾟ時	1,358(2MB)	1.47

表２　基本性能特性

結果を解析する。まず、ヒット時性能であるが、ＨＰＰ－ＬＨＳ間のデータ転送パス性能１．５GB/secに対して、実効性能は１．０３GB/secとなっている。実時間にして３００μsec程度の損失が発生している。これは、アドレス変換ユニット等の各種付加ハードウェア・オーバヘッドと、ＨＰＰ側の性能測定プログラム（Ｉ／Ｏサポート・ルーチンを含む）のオーバヘッドの影響と考えられる。次に、ＬＨＳ側ディスクにデータ・アクセスが発生した場合であるが、この場合の実効性能は１ＭＢ強にまで低下しており、ヒット時性能と比較すると１０００倍近い性能差を生じている。ディスク自身は３MB/secチャネル４本を用い、４スピンドルに対し、１６ＫＢ単位の並列アクセスを行っている。アドレス変換テーブルの再設定と、制御ソフトのオーバヘッドで４００～５００msecをロスしていると考えられる。今後の改良の第一課題は、ミスヒット時性能向上を図ることである。

次に、アプリケーション実行時性能として、世界最大規模の密行列に対するＬＵ分解を実行させた場合の性能を表３に示す。このＬＵ分解プログラムは、ＨＰＰの４台の並列プロセッサと４階層のメモリ（ローカル、共有、ＬＨＳ高速部、ＬＨＳディスク）を最適に利用するために新たに開発したアルゴリズムを採用している。３万２千元の行列サイズの場合、ＨＰＰ－ＬＨＳシステムでの全経過時間（１０時間４０分）に対し、ＬＨＳとのデータ転送所用時間は２１．７％である。残りはＨＰＰのＣＰＵ経過時間とＨＰＰ内のローカル－共有メモリ間データ転送によって消費されている。ＬＨＳとの転送時間の大半は、高速メモリ部とディスク間の転送であり、ここでもディスク性能の向上が重用な課題であることが示されている。

行列ｻｲｽﾞ		$16,348^2$	$32,768^2$
経過時間(sec)		7,586	38,387
HPPのﾛｰｶﾙ	転送量(GB)	624	2,496
ﾒﾓﾘと共有	時間(sec)	3,761	15,044
ﾒﾓﾘ間転送	速度(MB/sec)	166	166
ＬＨＳ	転送量(GB)	69	274
HIT 時	時間(sec)	66	263
の転送	速度(MB/sec)	1,040	1,040
ＬＨＳ	転送量(GB)	1.9	7.8
MISS HIT	時間(sec)	1,741	6,965
時の転送	速度(MB/sec)	1.1	1.1
LHS高速ﾒﾓﾘ部登録(sec)		274	1,097

表３　アプリケーション（ＬＵ分解）性能

７．おわりに

本稿では、大容量高速記憶装置ＬＨＳの記憶管理方式とその基本構想であるワンレベル・ファイル・ストアに関して報告した。ワンレベル・ファイル・ストアは、ファイル・システム構築の統合を狙った方式であり、半導体記憶とその他の低速記憶装置を階層接続することにより、大容量かつ実効的に高速な記憶装置の実現を目指している。ＨＰＰとＬＨＳから構成される実験システムにおいて評価を実施した。ＬＨＳの大容量性を生かし、世界最大規模のＬＵ分解の実行を行った。全経過時間に占めるＬＨＳとのデータ転送時間比は２１．７％であり、良好な性能が得られている。

今後の課題として、ＬＨＳの二次記憶装置のデータ転送速度を向上させることがあげられる。また、ワンレベル・ファイル・ストア方式として考案したすべての機構が既に実現出来たわけではない。属性空間とその特性を生かした効率的な記憶管理法の検討を進めていくことが重要である。

謝辞

本研究の実施にあたりご指導頂いた、C&Cｼｽﾃﾑ研究所 石黒所長、ｺﾝﾋﾟｭｰﾀ・ｼｽﾃﾑ研究部 小池部長に感謝します。ＬＵ分解ﾌﾟﾛｸﾞﾗﾑの開発を行って頂いた日本電気技術情報ｼｽﾃﾑ開発㈱の白戸氏に感謝します。また、本プロジェクト遂行にあたり様々なご協力を頂いた関係諸機関各位に深く御礼申し上げます。

参考文献

[1] E. I. Organick, "The Multics System: An Examination of Its Structure," Cambridge, the MIT Press, 1972.

[2] G. G. Henry et al., "IBM System 38 Technical Developments," General System Division, IBM Corp., 1987.