

UPCHMS の特徴解析

牧 晋広 岡本 秀輔 曾和 将容

電気通信大学大学院情報システム学研究科

概要

我々は、高速で大容量なメモリシステムを目的とした、階層化されたメモリ間のデータ転送を、プログラムにより1ワード単位で行う階層メモリシステムについて提案している。

本方式はプログラムにより階層メモリの内容を1ワード単位で操作するため、階層メモリの中身が完全に把握可能で、必要なデータだけを上位メモリに置くことができる。これらの転送処理は一般のプログラムと並列に処理されるため、ソフトウェア転送による実行時間の増加は隠蔽される。

本稿では、このメモリシステムで不明であったシステムの特徴について述べ、それらのこの効果を評価する。評価結果では、キャッシュを持つ計算機システム(CMS)との比較で、CMS以上に上位メモリ上のデータを有効に利用する、ソフトウェアデータ転送処理を並列に行うことによりそのオーバーヘッドを短縮する、ことによりCMSの最大1.42倍高速に実行することを示している。

The feature analyzing for UPCHMS

Nobuhiro Maki Syusuke Okamoto Masahiro Sowa

Graduate School of Information Systems, University of Electro-Communications

abstract

We've proposed a new hierarchical memory system, which consists of three level memories. This system has two kinds of programs. One is target program which executes arithmetic and logical part. Another is *data transfer programs* which control data mapping and transfer between different levels of hierarchical memories by one word. Furthermore, data transfers are done in parallel with the target program. Therefore, it is possible for this system to prepare necessary data into upper level memory before the target program refers to them.

This paper describes mainly the analyzing for this system's features.

1 はじめに

プロセッサの演算処理速度とメモリのアクセス速度には大きな隔たりがある [2]。現在ではこの隔たりを補うために、メインメモリ程大容量ではないが高速なメモリであるキャッシュメモリをプロセッサとメインメモリの間に配置している。しかし、キャッシュメモリには、キャッシュミスが発生するため、キャッシュメモリを持つ計算機システムではその処理能力が大幅に低下する [2, 4]。この原因の1つは、キャッシュメモリと主メモリの内容の入れ換えをLRU法などの固定的なアルゴリズムで行っていることにある。そこで、このキャッシュミスを回避するため、著者らはキャッシュメモリに相当するメモリを主メモリ同様に線形的なアドレスを与え、そのメモリにデータをプログラムにより1ワード単位で転送する階層メモリシステム『ユーザープログラム制御階層メモリシステム：UPCHMS』を提案している [3]。この方式では、1ワード単位でデータを転送するため無駄なデータ転送をしない、プログラムでキャッシュレベルメモリの内容を管理するため必要なデータだけをそのメモリに配置することができる、データ転送をそれぞれ専用のユニットが並列に行うためデータ転送をプログラムにより処理することによる処理時間の増加を減らすことができるという特徴を持つ。

簡単な構成における本システムの評価では、その有効性が示されている [3]。しかし、UPCHMSの特徴であるプログラムによる階層メモリの制御の効果、並列データ転送の効果の定量的な評価は示されていない。そこで本稿ではこれらに対処するため、UPCHMSの特徴を述べた後、それらの特徴を個別に定量的な評価を行う。

2 UPCHMS

2.1 ハードウェア構成とプログラムの基本動作

図1は、UPCHMSのブロック図である。PUは、一般の演算処理を行うプロセッサユニット、

IMはプログラムメモリ、DMはデータメモリである。DMは、超高速メモリVHM(レジスタに相当)、高速メモリHM(キャッシュメモリに相当)、主メモリMMの3階層により構成される。HM、MMメモリはそれぞれ独立な線形アドレスを持つ。

HU、MUは階層メモリ間のデータ転送処理を専門に行うプロセッサユニットで、HUはVHMとHM間のデータ転送、MUはHMとMM間のデータ転送を行う。HU、MUは、それぞれのIMに格納された専用のメモリ操作プログラムを実行する。tcは、トークンカウンタとよばれるカウンタで、ユニット間でトークンを送受することにより命令実行の同期を取るためのものである [5]。トークンとは1ビットの信号である。

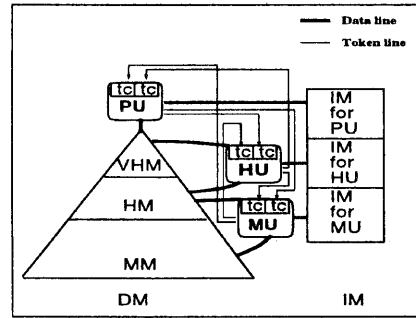


図1: UPCHMSのブロック図

図2はMMのm0番地からm5番地に、それぞれデータa1, a2, a3, a4, a5, a6が格納されている時、 $(a1+a2+a3)(a2-a1+a4+a5)-a3+a6$ の計算を行うプログラムである。ここでIPU, IHU, IMUはそれぞれPU, HU, MU用の命令流であり、IPUは主に計算処理、IHUはVHM-HM間のデータ転送処理、IMUはHM-MM間のデータ転送処理を行う。各命令間に付けられたアークは実行の先行関係を表し、例えば、命令M3とH3の関係では、M3命令終了後にH3命令が実行可能となることを意味している。ここで命令M3のldh m2,h2は、MMのm2番地のデータをHMのh2番地に転送するロード命令、命令H3のldv h2,v3はHMのh2番地のデータ

を VHM の v3 番地に転送する命令、命令 P2 の add v2,v3,v2 は VHM の v2 番地と v3 番地の内容を加算してその結果を v2 番地に格納する命令である。

図 2 で、M2, H2, P1 までの命令の実行が完了しているとしてプログラムの動作を説明する。この状態では、MM の m2 番地のデータ a3 は M3 命令によって HM の h2 番地に転送され、転送されたデータは、H3 命令によって VHM の v3 番地に転送され P2 命令により処理される。P2 により演算処理されたデータは、H4 命令により VHM から HM に書き戻される。P3 命令は P2 命令終了後すぐ実行可能であるので、H4 と P3 命令は並列に実行される。

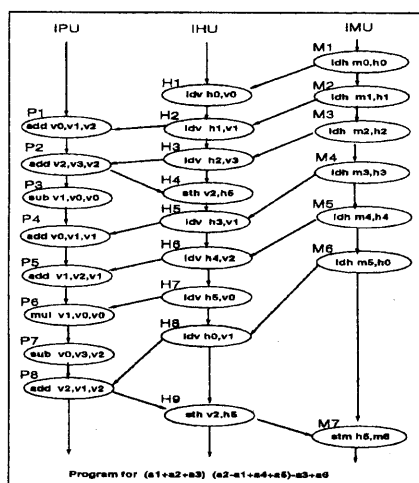


図 2: UPCHMS のプログラム

3 UPCHMS の特徴

UPCHMS の特徴には、プログラムによるデータの 1 ワード転送、並列データ転送の 2 つがある。次にこれらの特徴について述べる。

3.1 プログラムによるデータの 1 ワード転送

ユーザプログラムによる 1 ワード単位のデータ転送は、以下を可能にする。

- 1) 必要なデータだけを HM に転送できる
- 2) HM 上に転送された将来利用されるデータを MM に戻さない

これらの特徴が HM のデータの利用率を向上させることができる。ここで HM のデータの利用率を向上させるとは、HM 上のデータをできる限り多くプロセッサが使用することである。

1) は、1 ワードデータ転送が可能にする。ブロックによるデータ転送では、転送されるデータでプロセッサが利用しないものが同時に転送される可能性がある。一方、1 ワードデータ転送では将来使用されるデータだけを転送できるため、HM の内容を将来使用するデータだけにすることが可能である。これは、無駄なデータが HM がない分、ブロック転送する時より相対的に HM を大きく使うことができ、そこに更に多くの将来利用されるデータを HM に配置可能になり、結果としては、HM 上のデータの利用率が向上する。

2) は、プログラムによるデータ転送が可能にする。

UPCHMS のデータ転送はプログラム制御であるため、HM 上のデータを任意の配置にスケジューリングできる。そのため、HM が飽和してデータの入れ換えをする必要が生じる場合でも、HM 上のデータで将来利用されないもしくは利用されるまでに十分な時間のあるものから優先的に MM に書き戻すことができる。

3.2 データの並列転送

UPCHMS は、HU, MU のプログラムによりデータの転送を行う。HU および MU のメモリ操作処理を PU が行うと PU の処理量が増加するために、結果としては実行時間が既存の CMS 以上に増加する可能性がある。

そこでUPCHMSでは、HU、MUを独立したユニットとして設けそれぞれ並列に動作させる。これによりデータ転送処理からPUを開放させることができ、データ転送をプログラムにより行う処理時間の増加を回避することができる。

4 UPCHMSの特徴解析

上記で示したようにUPCHMSには2つの特徴がある。ここではUPCHMSの特徴による効果を個別に評価していく。評価は、UPCHMSと既存のキャッシュメモリを持つ基礎的な計算機システムCMSと比較することで行う。UPCHMSは開発段階で、基礎的な特性もわかっていない。そこで基礎的な特性をみるため、UPCHMSの各ユニットにはパイプライン処理などの付加的な高速化機構をいれない。この設定に合わせCMSでも命令の処理にパイプライン処理等をいれない。

4.1 評価設定

シミュレーションにおいて前提としたUPCHMS、CMSの各階層のメモリの容量およびアクセス転送時間を表1に示す。メモリはVHM、HM、MMとも1ワード32bitとする。メモリサイズが商用のそれに比べ小さいのは、UPCHMS用のソフトウェア開発環境（コンパイラなど）が未開拓の現状では実用規模のプログラムの開発が難しく、開発可能なプログラムでUPCHMSの基礎特性を得るための処置である。ブロック転送時のMMアクセス時間は、ブロック転送の高速性を考慮して1ワードあたり平均50nsとする。命令の実行時間を表2に示す。基本的にUPCHMSとCMSの設定値を同じ条件のもとで決めている。HU、MUに命令実行時間が二つあるのは、命令が間接アドレス指定かどうかによるものである。

CMSは、HMと同容量のキャッシュメモリを持ち、LRU置換、write through、フルアソシエイティブキャッシュ、1ブロック4ワードとしている[1]。

命令メモリは理想的なものとし、HMのアクセス時間で1命令をフェッチできるものとする。

前提としたプログラムは、実行開始前にすべてのものがメモリに格納されているとし、HMもそれらに静的に割り当てられるとした。

評価プログラムには、さまざまなプログラムに対する性能を見るため、異なるメモリ参照動作を示すものを選んだ。プログラミングに際しては、UPCHMS、CMSとも特別非効率な動きをすることは避けるが、特別な最適化を施さないとする。表3にその名前、内容、および特徴を示す。これらは手動で記述される。

表1: UPCHMS, CMSのメモリの容量および各階層のアクセス時間

		VHM,reg	HM,cache	MM
UPCHMS	memory size (word)	16	128	65536
CMS	access time (ns)	5	20	100

表2: UPCHMS, CMSの各種命令の実行時間 (ns)

UPCHMS	PU	HU		MU	
		間接あり	間接なし	間接あり	間接なし
CMS	60	load&store			
		hit		miss	
	60	65	80	175	155
		80		260	

4.2 評価プログラム別特性評価

図3は、UPCHMSによる実行時間の短縮を示している。図ではCMSによる実行時間を1としている。UPCHMSの実行時間がCMSのそれよりもすべての評価プログラムで小さくなっており、UPCHMSは最高でCMSの70.1パーセント（mulスピードアップで1.42倍）、最低でも76.1パーセント（srch 1.31倍）CMSに比べ高速であることがわかる。

- プログラムによる1ワードデータ転送

表 3: 評価プログラムの名前, 内容, 特徴

Program	Description	Characteristics
add	sum up data stream	データの再利用なし, 逐次にデータを参照, プログラムの解析可能
srch	binary search	データの再利用なし, 離散的にデータを参照, プログラムの解析不可
mul	multiple matrixes(AxB)	データの再利用あり, 再利用されるデータの時間的間隔が異なる プログラムの解析可能

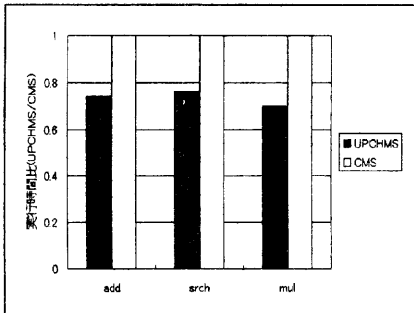


図 3: UPCHMS と CMS の実行時間比

上節で, プログラムによる 1 ワード単位のデータ転送は, HM のデータの利用率が向上するを述べた. ここでは, その HM の利用率を評価する.

HM のデータの利用率とは, プロセッサが必要とするデータ数に対する HM 上のデータの比で表される. 表 4 は, この利用率を示している. Arc は UPCHMS(UPC) および CMS を, Ref. は評価プログラムのデータ参照回数を, Trans. は HM もしくはキャッシュメモリに転送するデータ数を, Avail. は HM のデータの利用率で, Ref./Trans. を示している.

結果をみると Avail. で, UPCHMS の方が全ての評価プログラムで CMS 以上に高い数値を示していることがわかる. 最大で srch の 4 倍, 最低で add の 1 倍である. これは, UPCHMS が HM に転送されたデータを CMS 以上に無駄なく利用することを意味している. このことは, mul などのようにデータの参照に再利用性があり CMS においても Avail. の値が他の結果より良い (1

以上) ようなプログラムでも, UPCHMS の方がそれ以上に Avail. が高いことからわかる.

表 4: UPCHMS と CMS のデータの利用率

Prog	Arc	Ref.	Trans.	Avail.
add	UPC	1001	1001	1.00
add	CMS	1001	1004	1.00
srch	UPC	15	15	1.00
srch	CMS	15	60	0.25
mul	UPC	129600	70320	1.84
mul	CMS	129600	123664	1.05

● データの並列転送

UPCHMS の並列転送の効果を見る. HU, MU 各ユニットの効果を分離してみるため, UPCHMS のデータ転送処理を部分的に並列に行わない 2 つの UPCHMS を定義し, 以下に示す.

- U1: 全てのデータ転送を逐次的に行う UPCHMS
- U2: VHM, HM 間のデータ転送を逐次的に, HM, MM 間のデータ転送を並列に行う UPCHMS

U1 と U2 の違いは, MM と HM 間のデータ転送を並列に行うかどうかであり, U2 と UPCHMS の違いは, VHM へのデータ転送を並列に行うかどうかである. U1 と U2 を比較することで, HM へのデータ転送の並列効果がわかり,

U2 と UPCHMS を比較することで、VHM へのデータ転送の並列効果がわかる。

図4にUPCHMS, U1, U2の比較結果を示す。縦軸は、UPCHMSの実行時間を1とした時の実行時間を、横軸は、評価プログラムを示している。

これを見ると、各階層への並列データ転送の効果は、ほぼ同じ程度現れていることがわかる。具体的には、U1とU2の比較では、23から45パーセント程度、U2とUPCHMSの比較では、20から38パーセント程度実行時間が増加している。また、UPCHMSとU1との比較は、データ転送を逐次的に行う場合を意味するが、この場合は、60から80パーセント程度実行時間が増加することがわかる。すなわちUPCHMSでは逐次的に処理をすると実行期間が60パーセント以上増加するデータ転送を並列に行うことで、その処理時間を短縮していることがわかる。

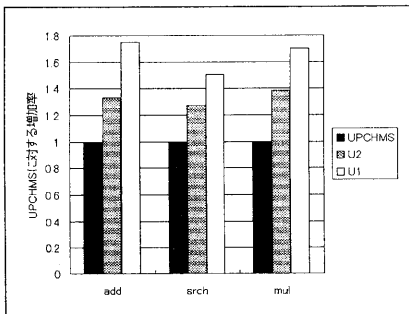


図4: UPCHMSの並列転送の効果

5 おわりに

以上本論文は、キャッシュメモリに相当するメモリにプログラムで1ワード単位でデータを転送するUPCHMSについて述べ、特徴を挙げ、特徴別に評価を行った。

この方式では、プログラマが階層メモリの内容をすべて見ることができると、不必要なデータをHMに転送しない、処理プログラムにあった最適なタイミングでデータを転送できる、不

必要なデータを早期に追い出すなど通じて、容量に制限のある上位階層のメモリを有効に利用することが可能となる。

本論文では、UPCHMSの特徴である2つの特徴、プログラムによるデータの1ワード転送、データの並列転送について評価した。

UPCHMSとCMSとの評価プログラムによる比較では、最大1.42倍最低でも1.31倍高速に処理した。プログラムによるデータの1ワード転送では、キャッシュレベルのメモリに転送されたデータの利用頻度の調査を行い、どの評価プログラムに対してもCMS以上に有効にデータを利用することを示した。データの並列転送では、逐次に行うと60パーセント以上処理時間が増加するデータ転送を並列に行うことで隠蔽ができることを示した。

以上の結果より階層メモリ間のデータ転送を全てソフトウェアで、1ワード単位で行うUPCHMSの特徴は有効に働いていると考えられる。

参考文献

- [1] Alvin, R. and David, A.: Cache Profiling and the SPEC Benchmarks: A CASE Study., *COMPUTER*, Vol. 27, No. 10, pp. 15-26 (1994).
- [2] Hennesy, J. L. and Patterson, D. A.: *Computer Architecture: A Quantitative Approach.*, Morgan Kaufmann (1990).
- [3] 牧晋広, 岡本秀輔, 曾和将容: ユーザプログラム制御階層メモリシステム, 情報処理学会論文誌, Vol. 37, No. 10, pp. 1512-1526 (1996).
- [4] Simmons, M. and Wasserman, H.: Performance Evaluation of an Optimized Scalar with Two Vector Processors., *Proc. of Supercomputing*, pp. 132-141 (1990).
- [5] 高木浩光, 河村忠明, 有田隆也, 曾和将容: 問題の持つ先行関係だけを保証する高速な静的実行順序制御機構, 並列情報処理シンポジウム JSP, Vol. 1, pp. 57-64 (1990).