

ユーザプログラム制御階層メモリシステム

牧 晋 広[†] 岡 本 秀 輔[†] 曾 和 将 容[†]

キャッシュメモリに相当するメモリへのデータ転送を、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を提案する。本方式は、アプリケーションプログラムで必要とされるデータだけを、キャッシュメモリ相当のメモリに転送するため、従来のキャッシュメモリシステム以上に効率的なデータ転送が行える。簡単な性能評価では、従来のキャッシュメモリシステムの1.76倍高速になった。

User Program Controlled Hierarchical Memory System

NOBUHIRO MAKI,[†] SHUSUKE OKAMOTO[†] and MASAHIRO SOWA[†]

We propose a new hierarchical memory system. In this memory system, two extra programs transfer the data among any hierarchical memories by one word. This system makes cache like memory utilization better than conventional cache memory systems, because the data which application programs just need are transferred to upper level memory. Simple benchmarks show that this system can execute the application programs 1.76 times faster than conventional cache memory systems.

1. はじめに

キャッシュメモリシステム（以後CMSとよぶ）ではキャッシュミスが発生し、計算機システムの処理能力を下げる¹⁾。我々はこの問題を解決するために、キャッシュメモリの内容をプログラムで操作する方式を提案してきた²⁾。この方式では、データの振舞いを知るプログラマがキャッシュメモリを操作するため、キャッシュミスが発生する前に、アプリケーションプログラムが必要とするデータを、前もってキャッシュメモリに転送することが可能となり、キャッシュミスを減少させることができた。しかしこれまでの方式では、ブロック単位によるキャッシュメモリへのデータ転送を行っていたために、不必要なデータをも転送し、キャッシュメモリを無駄に使用していた。

本稿では、上記の無駄をなくすために、転送を1ワード単位で行う階層メモリシステムについて検討した。このシステムを以後ユーザプログラム制御階層メモリシステム（User Program Controlled Hierarchical Memory System: UPCHMS）とよぶ。

2. 基本構造

図1は、UPCHMSのブロック図である。PUはプロセッサユニット、IMはプログラムメモリ、DMはデータメモリを示している。DMは、レジスタに相当する超高速メモリ（VHM）、キャッシュメモリに相当する高速メモリ（HM）、主メモリ（MM）の3階層よりなる。

PUはVHMのみを対象として一般の演算を行う処理ユニットであり、HUはVHMとHM間のデータ転送、MUはHMとMM間のデータ転送をそれぞれ行う転送ユニットである。転送ユニットは、それぞれIMに格納されたプログラムによってデータ転送を行う。ユニット内のtcは、トークンカウンタとよばれるユニット間の同期機構である³⁾。

3. UPCHMSの動作

図2はMMのData番地から順番に格納されているデータa1, a2, a3, a4を用いて、 $(a1+a2) \times (a3-a4)$ を計算するプログラムである。ここでIPU, IHU, IMUはそれぞれPU, HU, MU用の命令流であり、各命令間に付けられたアークは、実行の先行関係を表す。M2とH2のアーク上にある点は、トークンとよばれるもので、このトークンの送受により命令間で同期を

[†] 電気通信大学大学院情報システム学研究科
Graduate School of Information Systems, University of
Electro-Communications

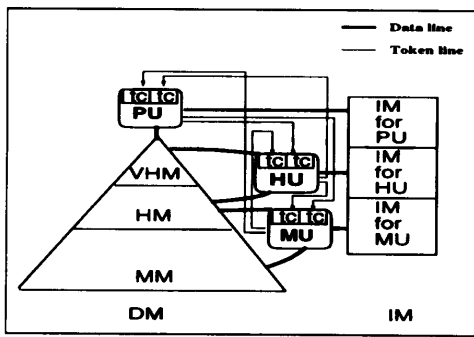


図1 UPCHMSのブロック図
Fig. 1 Diagram of UPCHMS.

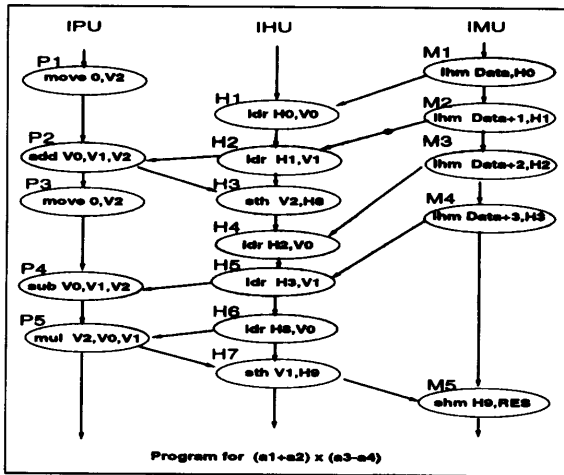


図2 UPCHMSのプログラム
Fig. 2 Program for UPCHMS.

とる。

今、図2で、P1, H1, M1の命令の処理が終了しているものとする。このとき、IPUのP2命令で必要とされるデータa2は、MMのData+1番地からM2命令によってHMに転送される。転送されたデータは、H2命令によってVHMのV1番地に転送されP2命令により使われる。

IPUの命令が必要とするデータが、使用される直前に決定される場合、MMからVHMへのデータの転送が間に合わずにIPU命令がデータを待つ状態になることがある。これはデータ待ち(Data Waiting: DW)とよばれており、CMSのキャッシュミスに相当する。これは、階層メモリスシステムが持つ性質であり、回避することは不可能である。この場合、UPCHMSでは必要なデータを決定できた時点から転送を開始することができるので、このDWを最小にすることができる。これに対し、CMSではキャッシュメモリにデータがないことを検出してはじめてデータ転送を開始するので、待ち時間が相対的に長くなる。

表1 シミュレーションで用いたUPCHMS, BUPCHMS, CMSの命令実行時間 (ns)

Table 1 Execution time of instructions for UPCHMS, BUPCHMS, and CMS.

| | PU | HU | MU | Others |
|---------|-----|------------|-------------|-------------------------------------|
| UPCHMS | 60 | 60 | 160 | |
| BUPCHMS | 60 | 60 | 260 | |
| CMS | CPU | load&store | | LRU write-through 1 block=4 word |
| | 60 | hit 80 | miss 220 | |

4. 簡単な性能解析

UPCHMSの性能評価をシミュレーションにより行った。シミュレーションでは、UPCHMSとCMSおよびブロック転送を行うことだけがUPCHMSと異なるUPCHMS(以下BUPCHMSとよぶ)との比較を行った。表1は、シミュレーションで用いたUPCHMSとCMSおよびBUPCHMSの1命令の実行時間(ns)を示している。UPCHMS, CMSおよびBUPCHMSはすべて、1ワードを32bitとし、CMSおよびBUPCHMSでは、1ブロック4ワード単位で転送を行うこととした。ブロック転送と1ワード転送との違いを考慮して、CMSおよびBUPCHMSの1ワードあたりの転送速度をUPCHMSの2倍とした。CMSは、ワークステーション等で一般的に用いられているキャッシュメモリスシステムで、フルアソシエイティブキャッシュ、write through方式、LRU置換とした。

アプリケーションプログラムは、10万項の数列の単純演算add、データX,Yそれぞれの個数が1万の分散の演算V(XY)、10万個のデータの2分探索searchである。UPCHMS, BUPCHMSおよびCMSのアプリケーションプログラムは、基本的に同じ処理を行うが、対象としているアーキテクチャが違うためその分だけ形式などが異なっている。UPCHMSとBUPCHMSのプログラムは、現状ではUPCHMSコンパイラが未開発なので直接アセンブリ言語で記述される。

表2はシミュレーション結果である(表中の時間の単位は、 μs)。表中のExe-timeは実行時間を示す。DWは、UPCHMSおよびBUPCHMSではDW時間の総和であり、CMSではキャッシュペナルティ時間の総和である。Spd-upは、スピードアップであり、CMSの実行時間に対するスピードアップ率を示す。Miss ratioは、CMSのキャッシュミス率である。

表2のシミュレーション結果によれば、UPCHMSのスピードアップはCMSの1.28から1.76であり、データ待ち時間はCMSの3.52から168に対して、

表2 シミュレーションの結果
Table 2 Results of simulations.

| | program | Exe-time (μ s) | DW (μ s) | Spd-up | Miss ratio (%) |
|---------|---------|------------------------|------------------|--------|-------------------|
| UPCHMS | add | 300 | 0 | 1.76 | |
| | V(XY) | 6000 | 0 | 1.49 | |
| | search | 11.4 | 0.495 | 1.28 | |
| BUPCHMS | add | 300 | 0.075 | 1.76 | |
| | V(XY) | 6000 | 0.030 | 1.49 | |
| | search | 13.1 | 1.365 | 1.14 | |
| CMS | add | 528 | 168 | 1 | 25 |
| | V(XY) | 8940 | 37 | 1 | 25 |
| | search | 14.6 | 3.52 | 1 | 100 |

UPCHMS では 0 から 0.495 と減少している。add, V(XY) は、逐次的に MM からデータを参照するため、データの転送動作が単純である。その結果 HU, MU の処理が PU の演算処理よりも十分に少なくなり、HU, MU のデータ転送処理が PU の演算処理に対して先行される。このことが PU のデータ待ちを 0 にしていると考えられる。一方 CMS では、次に必要なデータを先行して用意することは不可能であるため、キャッシュミスが 25% 発生している。search は、プログラム実行中に必要なデータが決定されるため、MM からのデータ転送の開始が遅れ、データ待ちを 0 にできなかったと考えられる。CMS での search ではキャッシュミス率が 100 パーセントであり、このため転送時間が増加し、UPCHMS に比べデータ待ちが約 7 倍長いという結果となった。

表 2 で UPCHMS と BUPCHMS について考える。どちらも CMS より実行時間およびデータ待ち時間は短い。UPCHMS と BUPCHMS の実行時間は、add, V(XY) で同じ、search では UPCHMS の方が 1.15 倍高速である。また、データ待ち時間は、add, V(XY) で UPCHMS が 0 であるのに対し、BUPCHMS では少しあり、search では UPCHMS が BUPCHMS の約 3 分の 1 に縮小している。add, V(XY) は BUPCHMS に有利なプログラムで、これらは逐次的にデータを参照する。そのため BUPCHMS ではブロック転送されたデータを無駄なく使用する。その結果 UPCHMS と BUPCHMS の実行時間に差が生じなかった。これに対し、search では離散的にデータが参照されるため、BUPCHMS でブロック転送されたデータのうち使用されるデータは少なく、この非効率的な転送が実行時間の差となって現れたと考えられる。

5. おわりに

1 命令により 1 ワードを基本として転送するユーザプログラム制御階層メモリシステムについて述べた。

1 ワード転送はブロック転送に比べて不必要なデータを転送しない一方、相対的にワードあたりの転送速度が遅いという欠点がある。しかし、シミュレーション結果によると、実行速度は最大 CMS の 1.76 倍に、BUPCHMS の 1.15 倍になることが分かり、また、キャッシュミス時間は 0 になることが分かった。これらの結果から UPCHMS は十分な性能が期待できることが明らかになった。

UPCHMS では、MM から HM への転送を MU のプログラムが行うため UPCHMS のプログラム容量は CMS のそれよりも 50 パーセント程度増加した。また、データの転送処理をプログラマが記述するので、その分プログラマに負担がかかるが、このことは UPCHMS のコンパイラを開発することで解決されると考えている。現在、この UPCHMS コンパイラについても研究中である。

参考文献

- 1) Hennesy, J.L. and Patterson, D.A.: *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann (1990).
- 2) 佐藤正樹, 有田隆也, 曾和将容: 並列処理によるキャッシュ操作の明示化, 並列シンポジウム JSPP, Vol.1, pp.1-7 (1990).
- 3) 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 JSPP, Vol.1, pp.57-64 (1990).

(平成 8 年 3 月 4 日受付)

(平成 8 年 7 月 4 日採録)

牧 晋広 (学生会員)



昭和 44 年生。平成 5 年名古屋工業大学工学部電気情報工学科卒業、平成 7 年電気通信大学大学院情報システム学研究科修了、同大学院博士後期課程在学中。計算機システムの記憶装置の高速化に関する研究に従事。情報処理学会会員。

**岡本 秀輔 (正会員)**

平成元年成蹊大学工学部経営工学科卒業。平成3年同大学院博士前期課程修了。平成6年同大学院博士後期課程修了。工学博士。平成6年電気通信大学大学院情報システム学研究科助手。並列処理言語およびその処理系、命令レベル並列プロセッサの研究に従事。IEEE, 情報処理学会, 各会員。

**曾和 将容 (正会員)**

昭和49年名古屋大学大学院博士課程(電気電子専攻)修了。同年群馬大学工学部情報工学科助手。昭和51年助教授。昭和62年名古屋工業大学教授。平成5年電気通信大学大学院情報システム学研究科教授。この間、並列処理、計算機アーキテクチャ、特にデータフロー計算機、コントロールフロー計算機など次世代コンピュータの研究に従事。工学博士。IEEE, ACM, 情報処理学会各会員。