

シリアル転送 DRAM インタフェースを持つプロセッサの キャッシュ構成に関する研究

高性能コンピューティング学講座 本多・近藤研究室

1253001 池田賢祐

主任指導教員：近藤正章

1 はじめに

近年 CPU と主記憶装置に用いられる DRAM のアクセス速度の性能差は拡大している。プロセッサ-DRAM 間のデータ転送速度を向上させるために、米 micron 社の発表した Hybrid Memory Cube (HMC) アーキテクチャ[1] にシリアル転送方式が実装されている。

シリアル転送方式を用いることにより、データ転送のクロック周波数を上げることができるため、1 リンクあたりのデータ転送レートを向上させることができる。また、シリアル転送リンクを複数本用いることで、通常の平行転送方式に比べ、合計のバンド幅も向上させることができる。シリアル転送方式により、1) 非常に多数のプロセッサ-DRAM 間チャネルを持つことができる (1 チャネルあたりのバンド幅は従来より小さくなる可能性がある)、2) シリアル/平行変換が必要になるためデータ転送にかかるレイテンシが増加する可能性がある、点で従来のメモリインタフェースとは特性が異なると予想される。そのため、関連する機構のアーキテクチャ構成について研究すべきところも多い。

本研究では、HMC を用いる将来のシステムに関し、プロセッサ内でデータを一時的に保持する記憶領域であるキャッシュメモリに着目し、上記のデータ転送に関する特性の違いを踏まえて、高性能を達成できるキャッシュ構成方式について検討する。

2 研究背景

ここでは、将来的にプロセッサ-DRAM 間のデータ転送として有望視されているシリアル転送を用いた DRAM 実装技術の一つである米 micron 社の HMC を紹介する。また従来の DRAM インタフェースにおけるキャッシュメモリ構成の関連研究について述べる。

2.1 シリアル転送技術を用いる

Hybrid Memory Cube アーキテクチャ

従来の DRAM システムにおいては平行転送が実装されていた。平行転送では、データ伝送線を何本も使用することになるため、データ到着のずれ(スキュー)や隣り合う伝送線の磁気干渉(クロストーク)といった問題

が生じる。そのため平行転送のままではデータ転送を高速化させることが難しい。

そこで近年では micron 社の Hybrid Memory Cube (HMC) のようにシリアル転送を用いたプロセッサ-DRAM 間データ転送技術が注目されている。図 1 は HMC の模式図である。HMC は DRAM の層が 3 次元的に 4~8 枚積み重なっており、各層は貫通電極によって接続されている。プロセッサと各 HMC はシリアル転送インタフェースによって接続されている。

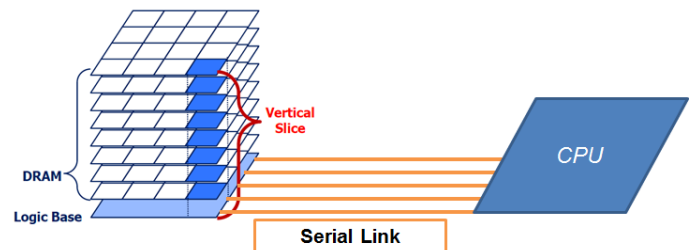


図 1: Hybrid Memory Cube (HMC) とプロセッサの接続図

HMC と従来の DRAM 実装技術の性能諸言を表 1 に示す。表 1 は、1 メモリモジュール単位の各種性能データであるが、メモリバンド幅に着目すると、DDR4 の 21.34 GB/s に対し、HMC は 128.00 GB/s もの性能を達成可能と言われている。

2.2 キャッシュメモリ構成に関する従来研究

特にシリアル転送時でのメモリ構成に限定しなれば、キャッシュと DRAM の構成に関する研究は数多く存在する。Biswas ら [2] はマルチコア環境において、あるプロセスにおいて利用されたデータを別のプロセスで利用するた

表 1: DDR と HMC の性能比較表

Technology	VDD	IDD	BW GB/s	Power (W)	mW/GB/s	pj/bit	real pj/bit
SDRAM PC133 1GB Module	3.3	1.50	1.06	4.96	4664.97	583.12	762
DDR-333 1GB Module	2.5	2.19	2.66	5.48	2057.06	257.13	245
DDRII-667 2GB Module	1.8	2.88	5.34	5.18	971.51	121.44	139
DDR3-1333 2GB Module	1.5	3.68	10.66	5.52	517.63	64.70	52
DDR4-2667 4GB Module	1.2	5.50	21.34	6.60	309.34	38.67	39
HMC, 4 DRAM w/ Logic	1.2	9.23	128.00	11.08	86.53	10.82	13.7

[出典 : J. T. Pawlowski (Micron), "Hybrid Memory Cube"
HOT CHIPS 23]

めの Mergeable Cache Architecture を考案し、DRAM へのアクセスの削減に成功した。また、Yoon ら [3] は実行しているアプリケーションの空間的局所性に応じて、キャッシュラインブロックサイズを変更できるようにし、かつ DRAM に仮想的に設けられたサブランクシステムを用いてメモリシステムのデータアクセス粒度を変更可能な Adaptive Granularity Memory Systems を考案した。これらのようにキャッシュの性能向上に関する研究は多いが、シリアル転送インタフェースを考慮した手法はまだ無い。

3 キャッシュメモリ提案手法の考案に向けて

パラレルインタフェースと異なりシリアルインタフェースでは、多数のメモリチャンネルを持つが 1 リンクあたりのデータ転送のレイテンシは増加する。そのため、ラストレベルキャッシュの最適な構成が従来のシステムと異なる可能性がある。例えば、シリアル転送ではキャッシュラインブロックサイズが大きいほどレイテンシが増加するため、従来よりもキャッシュのラインサイズを小さくする必要があるかもしれない。また、プリフェッチのアルゴリズムを再考する必要があると考えられる。本研究では特に、キャッシュライン転送の粒度に着目して研究を行う。現在検討中のキャッシュシステムは実行中のアプリケーションに応じて各ライン転送に用いるチャンネル数の割り当てを変えることや、最適なデータブロックサイズを検討するというものである。

4 多数のメモリチャンネルを持つシステムの予備実験

メモリチャンネル数と複数コアでの IPC を調べて、キャッシュ構成手法の考案に活用するために、CPU コア数を 2、メモリチャンネル数を 2, 4, 8 と変更させてシミュレーションを行った。使用しているシミュレータは gem5 シミュレータでサイクル精度のシミュレータである。図 2 がシミュレーション結果である。横軸はベンチマーク名で SPEC2006 から複数選択した。縦軸は各コアでベンチマークを実行した際の IPC の合計値である。

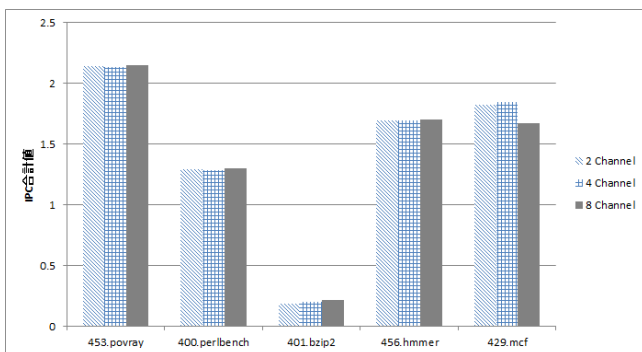


図 2: 2 コアで 2, 4, 8 チャンネルと実行した結果

シミュレータではチャンネル数の増加と共に IPC の増加がみられる。理論的にはチャンネル数の増加に従って IPC の合計値は増加するはずなのだが、429.mcf において 8 チャンネルの場合に IPC の低下が発生している。現在原因を調査中である。

5 今後の予定

1. メモリチャンネル数と CPU コア数を変更させてキャッシュメモリにどのような傾向が現れるか調査する。
2. シリアル転送時におけるキャッシュメモリ構成手法の考案を行う。
3. 提案手法をシミュレータ上で評価する..

コンピュータシステムではメモリチャンネル数を増加させるとメモリバンド幅の向上につながり、コア数を増加させるとメモリアccessの競合が増える。そこで 1. に関しては各コアでベンチマークを実行した際の IPC や L2 キャッシュミスといった数値について解析することで、提案手法考案に役立てる。

2. に関しては、キャッシュメモリ構成に関する論文を読み、CPU, DRAM, キャッシュに関する知識を深めていくことで、提案手法を考案する予定である。

3. に関しては、提案手法の考案ののちに、従来のシステムとの評価をどのように行うかが重要となる。そのために評価に最適な指標とベンチマークの選択が必要である。また、メモリアccessの頻度の高いベンチマーク実行の際に、メモリチャンネル数を増加させると IPC が低下する原因を追求する必要がある。

参考文献

- [1] “HotChips23 HMC - HC23.18.320-HybridCube-Pawlowski-Micron.pdf” <http://www.hotchips.org/wp-content/uploads/shc_archives/hc23HC23.18.3-memory-FPGAHC23.18.320-HybridCube-Pawlowski-Micron.pdf>
- [2] S. Biswas, *et al.*: Multi-Execution: Multicore Caching for Data-Similar Executions *Proc. 36th Annual International Symposium on Computer Architecture (ISCA'09)*, pp. 164-173, 2009.
- [3] D. H. Yoon, *et al.*: Adaptive Granularity Memory Systems: A Tradeoff between Storage Efficiency and Throughput. *Proc. 38th Annual International Symposium on Computer Architecture (ISCA'11)*, pp. 295-306, 2011.