メニーコアプロセッサにおける効率的なキャッシュシステム

入谷 優 †1　三好 健文 †2,†3　吉瀬 謙二 †2
東京工業大学 工学部情報工学科 †
東京工業大学 大学院情報理工学研究科 †2　独立行政法人科学技術振興機構 †3

1 はじめに

近年，1 チップ上に複数コアを搭載するマルチコアプロセッサが主流であり，今後コアの個数を更に増加させたメニーコアプロセッサの時代が到来すると考えられる．しかしながら，多くのコアが頻繁にメインメモリを参照する場合においては，そのアクセスレイテンシによってコア数に比例した性能を出す事が難しくなる．そのため，キャッシュの効率的な利用によりメインメモリへの参照を減らす事が，メニーコア上でプログラムを高速化するために重要である．

本稿では，キャッシュシステムをメニーコアアーキテクチャ上に実装し，キャッシュ容量の割り当て方によってメニーコア上のプログラムの実行速度がどのように変化するかについて評価を行う．

2 M-Core アーキテクチャ

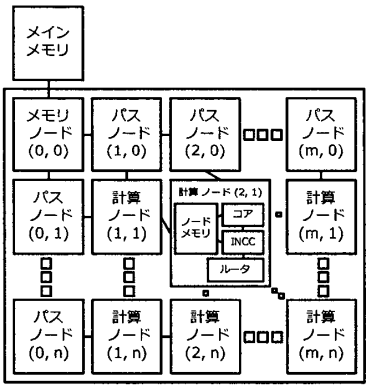

図 1: M-Core アーキテクチャの構成

本稿で対象とする M-Core アーキテクチャ[1] の構成を図 1 に示す．M-Core アーキテクチャは，多数のノードを 2 次元メッシュネットワークで接続した構成である．以下，(i, j) は X 座標が i，Y 座標が j である事を示す．

ノードには，計算ノード，メモリノード，パスノードの 3 種類のノードが存在する．計算ノードには，プログラムが実行される計算コア，ノード毎に用意されたメモリ領域であるノードメモリ，メモリとパケットを制御するコントローラである INCC，パケットを制御して他のノードと通信を行うためのルータが搭載されている．(0, 0) に位置するメモリノードには INCC とメインメモリが搭載されており，メインメモリへのアクセス要求はメモリノードによって処理される．(0, 0) を除く (0, *) 及び (*, 0) には，ルータのみが搭載されているパスノードが存在している．ルータは XY 次元順ルーティングによってルーティングを行う．

メインメモリへの読み書きはそれぞれ DMA_GET, DMA_PUT と呼ばれる DMA 要求をメモリノードへ送信する事によって行われる．本稿では，メインメモリへアクセスするプログラムのデータアクセス単位は固定長とし，計算コア間でのデータ共有は無いものとする．

3 キャッシュノード

各ノードにはノードメモリが搭載されているが，メインメモリに対するキャッシュは搭載されていない．そのため，メインメモリに対して頻繁にアクセスを行うプログラムでは，メインメモリのアクセスレイテンシのためにプログラムは充分な速度を得る事が出来ない．

そこで，メモリアクセスを効率化しプログラムの実行速度を向上させる事を目的に，キャッシュノードを導入する．キャッシュノードはキャッシュとして動作するノードであり，パスノードに代わって (0, *) に位置する．

3.1 キャッシュノードの構成

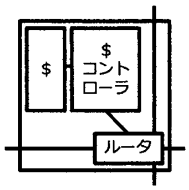

図 2: キャッシュノードの構成

キャッシュノードの構成を図 2 に示す．キャッシュノードにはルータ，キャッシュコントローラ，及びキャッシュ領域が搭載されている．キャッシュ領域は，4-way セットアソシアティブのキャッシュである．

3.2 キャッシュノードの動作

ここでは，DMA 要求が計算ノードから送られてきた時のキャッシュノードの動作について述べる．

3.2.1 DMA_PUT 要求

キャッシュノードにおけるキャッシュには，ライトスルー方式を採用している．キャッシュノードは，メインメモリに対する PUT 要求及び PUT するデータを計算ノードから受け取り，書き込み先アドレスが自身のキャッシュに存在すれば受け取ったデータをキャッシュ領域に書き込む．キャッシュに存在するしないにかかわらず，キャッシュノードは受け取った PUT 要求とデータをメインメモリに向けて再送する．

An Efficient Cache System for Many-Core Processors
Masaru IRITANI[†1], Takefumi MIYOSHI[†2,†3], Kenji KISE[†2]
†1 Depart. of Computer Science, Tokyo Institute of Technology
†2 Graduate School of Information Science and Engineering, Tokyo Institute of Technology
†3 Japan Science and Technology Agency

3.2.2 DMA_GET 要求

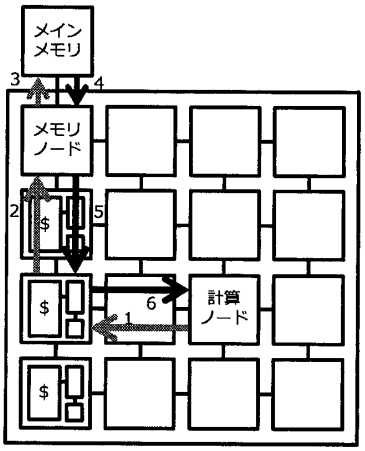

図 3: GET 要求時のキャッシュノードの動作

メインメモリに対する GET 要求が来た場合のキャッシュノードの動作を図 3 に示す．キャッシュノードは計算ノードメインメモリに対する GET 要求を受け取り (1)，メインメモリに代わって処理する．キャッシュミスが発生した場合には，キャッシュノードはメモリノードに対して GET 要求を発行する (2)．メモリノードは要求を受け取ってデータをメインメモリから読み出し (3, 4)，キャッシュノードへ送信する (5)．キャッシュノードはメモリノードからデータを受け取り，自身のキャッシュ領域からメインメモリから送られてきたデータをキャッシュ領域に格納してから，要求元の計算ノードに対して要求されたデータを送る．

メインメモリに対する DMA 要求は必ずキャッシュノードのルータを経由する．そのため，メインメモリにアクセスするプログラムはキャッシュノードの存在を意識する必要は無く，メインメモリに対して DMA 要求を送信する事で，透過的にキャッシュが動作する．

4 キャッシュシステムの評価

M-Core アーキテクチャのシミュレータである SimMc[1] において，キャッシュノードを利用した複数のキャッシュシステムを実装する．その上で，プログラムのデータセットサイズに偏りが有る場合と無い場合において，キャッシュ容量の割り当て方によってプログラムの実行性能にどのような影響が有るかについて評価する．

なお，メインメモリのアクセスレイテンシは 40 サイクル，キャッシュのアクセスレイテンシは 3 サイクルとする．キャッシュノードのキャッシュとは別に，各ノードは直前に GET した 16B のデータを保持しており，必要なデータがその中にある場合，メインメモリへ GET 要求を送信しない．

4.1 評価するキャッシュシステム

次の 3 つのキャッシュシステム上にて評価を行う．

LEFT (0, 1) から (0, 4) に 8KB のキャッシュノードを配置
INC1 (0, 1) から (0, 4) に 4KB，4KB，8KB，16KB のキャッシュノードを配置
INC2 (0, 1) から (0, 4) に 16B，16B，16B，32KB のキャッシュノードを配置

ここで，16B のキャッシュノードでは必ずキャッシュミスが発生する事に注意されたい．

4.2 ベンチマーク

評価には，クイックソート（qsort），FFT（fft），行列積（mm），LU 分解（lu）の 4 種類のベンチマークを用いる．それぞれのベンチマークについて，データセットの大きさに偏りが有るもの（DIFF）と無いもの（SAME）の 2 種類を用意し，計 8 つのベンチマークにおいて実行サイクル数を測定する．データセットサイズは，SAME では全ての計算コアで 16KB，DIFF では計算コア (4, 4) で 32KB，他の計算コアで 16KB である．

4.3 評価結果

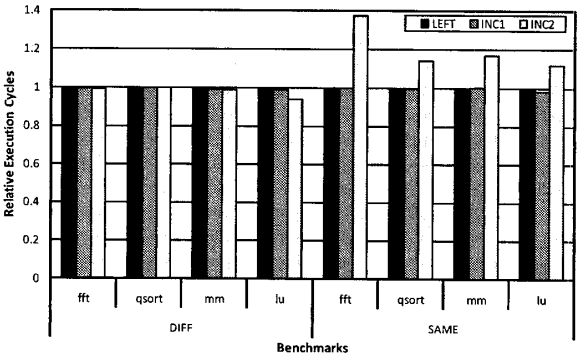

図 4: ベンチマークの相対実行サイクル数

評価結果を図 4 に示す．値は，各ベンチマークにおいて LEFT での実行サイクル数を 1 とした時の相対実行サイクル数である．データセットの偏りが有る場合には，lu で LEFT に比べて INC1 で 2%，INC2 で 6% 程度の性能向上が見られたほか，他のベンチマークにおいても僅かな性能向上が見られる．

一方で，データセットサイズが各計算ノードで均一な場合には，LEFT に比べ INC1 では僅かな性能低下が見られ，INC2 では平均して 20% の性能低下が見られる．

この結果から，データセットが非均一な場合には非均一なキャッシュ，データセットが均一な場合には均一なキャッシュを利用する事によって，性能向上が見込める場合が有る事が示される．

5 まとめ

メニーコアアーキテクチャ上にキャッシュノードと呼ばれるノードを導入する事でキャッシュシステムを実装し，キャッシュ容量の割り当て方によるプログラムの実行時間への影響について評価した．データセットの大きさに偏りが有る場合には，キャッシュ容量をその偏りに応じて適切に割り当てる事で，プログラムの実行速度を向上できる事を示した．

参考文献

[1] Koh UEHARA, et al. A Study of an Infrastructure for Research and Development of Many-Core Processors Workshop on Ultra Performance and Dependable Acceleration Systems held in conjunction with PDCAT'09