

# 8 コアプロセッサのメモリアクセス性能

## Memory Access Performance of 8 Cores Processors

河辺 峻

KAWABE, Shun

### 要旨

プロセッサは現在マルチコア化による高速化が進んでいる。並列処理が可能なプログラムやスループットを主とする多重プログラムにとっては高速化が期待できる。しかしキャッシュ構成は複雑化しており、それぞれのコアが所有するキャッシュは、L1, L2, L3 に階層化されている。最新の Intel プロセッサチップでは、L1/L2 キャッシュはコア間で共有せず、L3 キャッシュはコア間で共有している。また複数のプロセッサチップを搭載するサーバでは、L3 キャッシュ間で情報の交信を行っている。このような構成において、キャッシュのコヒーレンシ（一貫性）を保ちながらメモリアクセス性能の低下がどのくらいになるかについて評価プログラムを作成して8 コアプロセッサ性能の特性を評価した。その結果、キャッシュのコヒーレンシ（一貫性）回路が他のコアで動作している場合、メモリからのアクセスが 1.2～2.0 倍程度遅くなることが分かった。

### 1. はじめに

現在、プロセッサのシングルコアの周波数が消費電力や発熱問題により頭打ちになり、マルチコアが普及しつつある。そしてこれからはさらにプロセッサのコア数が増える方向へと進んでいくと思われる。本研究では8 コアプロセッサを用いてキャッシュの整合性を取る論理回路が他のコアで動作している場合、メモリからのアクセスがどれくらい性能低下するかを Linux の C 言語を用いて分析プログラムを作成して評価する。

### 2. 8 コアプロセッサの性能評価

#### 2. 1 評価に用いるプロセッサ

図 1 に今回の評価で用いたプロセッサの構成図を示す。

Intel E5620 (Westmere-EP) プロセッサは図 1 に示すように、1 つのプロセッサに 4 つのコアがあり各コアごとに 32KB のデータおよび命令キャッシュと 256KB の L2 キャッシュを持ち、各コアが共有する 12MB の L3 キャッシュを持っている。周波数は 2.40GHz で TPD は 80W である。このプロセッサチップがボード上に 2 つ搭載されており、L3 コントローラでプロセッサ間の情報の交信を行っている。また、それぞれのコアが HT (Hyper Threading) 機能を持って

いる。このためプログラムからは論理的には 16 のプロセッサがあるように見える。

今回の評価では affinity 機能を用いて使用するコアを固定し、HT (Hyper Threading) 機能は使用しないようにした。また OS は Fedora14 (64b) を使用した。

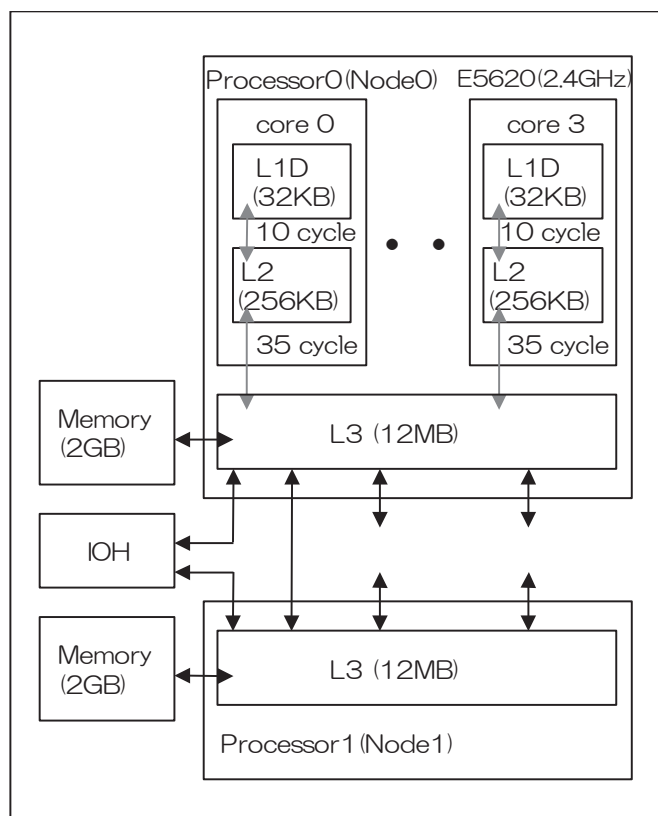


図 1 評価に用いたプロセッサ構成図

## 2. 2 8 コアの性能評価手法と結果

マルチコアでは他のコアがキャッシュコヒーレンシのチェックを行う動作をする場合でも、メモリアクセスが遅くなる場合があり、測定プログラムを作成して定量的に評価する。

### ■測定プログラムと測定結果

メモリ上の連続したエリア (64Bx8=512B) をとり、各 64B の先頭 1B をコア対応に 1B 毎に更新 (0xFF との排他的論理和) を行う。

core 0	core 1	core 2	core 3	core 4	core 5	core 6	core 7
	gbuf[256]	gbuf[128]	gbuf[384]	gbuf[64]	gbuf[320]	gbuf[192]	gbuf[0]

それぞれのコアが所有するキャッシュは、ブロックあるいはラインと呼ばれる 64B 単位にメモリからデータを持ってくる。各コアでは異なる 64B のデータを保持するため、キャッシュのコヒーレンシに関してはチェックする論理回路のみが動作する。

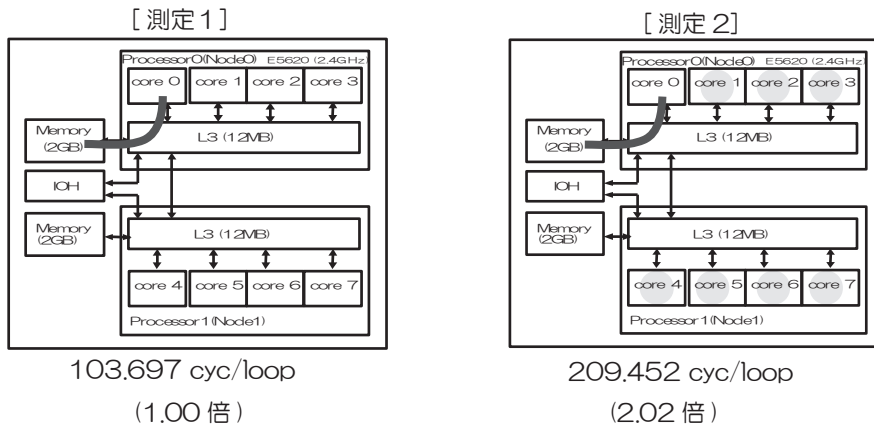
測定 1 : core 0 をメモリ 32MB の領域から 1loop あたり 64B の中の 8B を読み出し、loop ごとにメモリからライン転送が発生するプログラムのみを動作させる。キャッシュのコヒーレンシはなし。

測定 2 : core 1 ～ 7 を 1B 毎に更新するプログラムを動作させる。  
core 0 は測定 1 のプログラムを動作。

測定 3 : core 4 ～ 7 を 1B 毎に更新するプログラムを動作させる。  
core 0 は測定 1 のプログラムを動作。

測定 4 : core 6 ～ 7 を 1B 毎に更新するプログラムを動作させる。  
core 0 は測定 1 のプログラムを動作。

測定 5 : core 7 を 1B 毎に更新するプログラムを動作させる。  
core 0 は測定 1 のプログラムを動作。



● 1B 毎の更新中 core

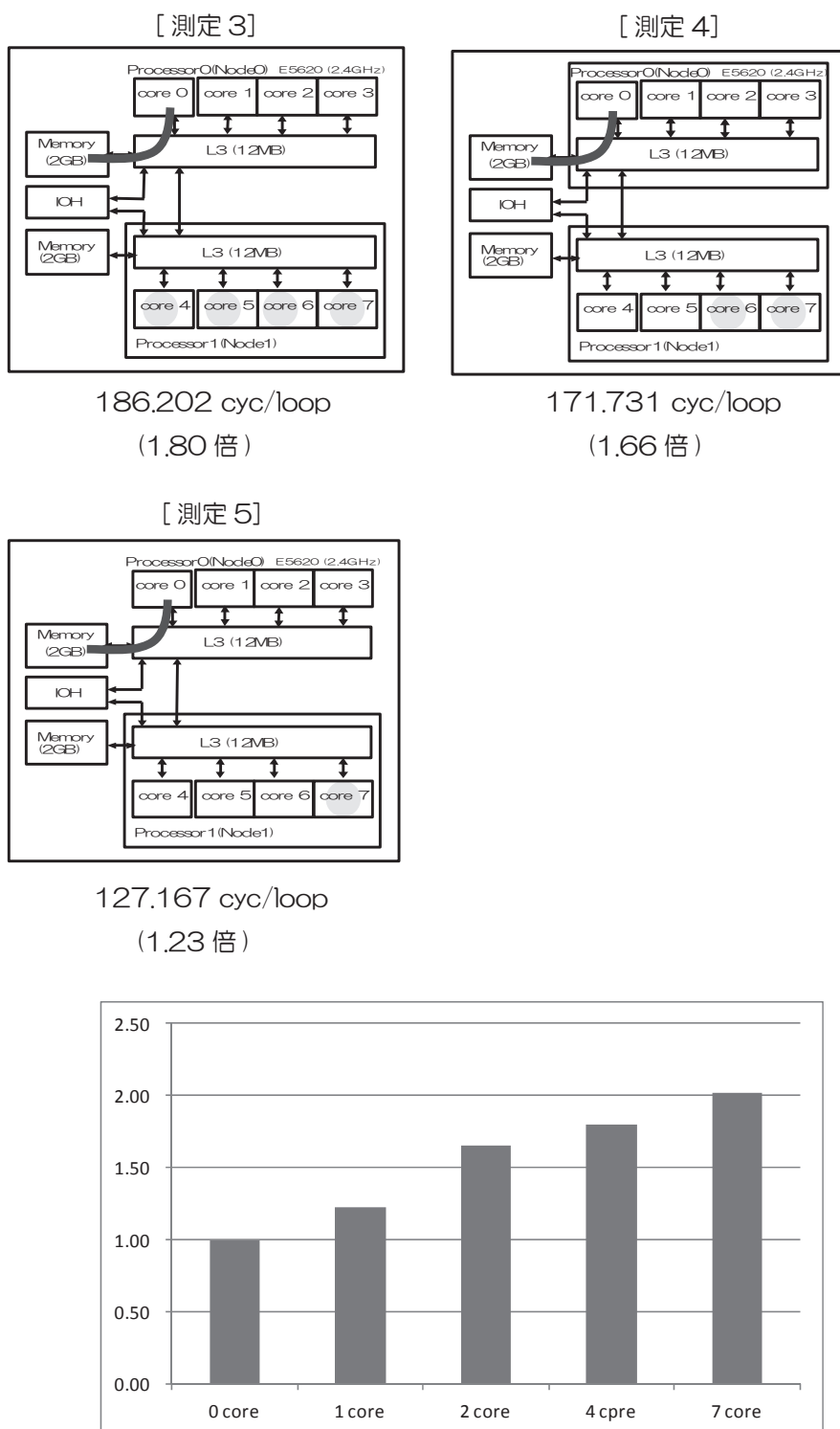


図 2 キャッシュコヒーレンシの論理回路の動作による性能低下比率

### ■ 考察

- ・測定 1 はキャッシュコヒーレンシの論理回路は動作しない。loop ごとにメモリからライン転送が発生するので loop 性能はメモリから L3 への転送時間が大部分と考えられる。
- ・測定 2 は core 1 ～ 7 を 1B 毎に更新するプログラムを動作させる。キャッシュコヒーレンシをチェックする論理回路は最大限動作する。測定 1 に対して loop 性能は 2.02 倍悪くなる。
- ・測定 3 は core 4 ～ 7 を 1B 毎に更新するプログラムを動作させる。キャッシュコヒーレンシをチェックする論理回路は Processor1 側で動作するが、Processor0 側にも影響を与えている。測定 1 に対して loop 性能は 1.80 倍悪くなる。
- ・測定 4 は core 6 ～ 7 を 1B 毎に更新するプログラムを動作させる。同様に測定 1 に対して loop 性能は 1.66 倍悪くなる。
- ・測定 5 は core 7 のみを 1B 毎に更新するプログラムを動作させる。他の core とはキャッシュコヒーレンシのチェックの論理回路が動作しているため測定 1 に対して loop 性能は 1.23 倍悪くなる。
- ・この結果から、他の core でキャッシュコヒーレンシをチェックする論理回路のみが動作しているプログラムが動作している場合（測定 2 ～ 5）でも、メモリアクセスが 1.2 ～ 2.0 倍程度遅くなる。

他のコアがキャッシュコヒーレンシのチェックを行う動作をする場合、性能が低下する理由は L3 にある Address Directory の取り合いによるものと思われる。従来のメインフレームなどでは他プロセッサからのキャッシュコヒーレンシのチェックは、Address Directory のコピーを持っていてそこで処理していたため、メモリからのアクセスの影響はあまり受けなかったが、Intel chip は Address Directory のコピーは持っていないようである。したがってマルチコアで L3 ミスが頻発すると性能に与える影響はより大きくなると考える。

### 3. 結論

マルチコアプロセッサのキャッシュ構造においては、キャッシュの一貫性を保つ処理の為、性能が大幅に低下する場合がある。この問題は既に文献[1] [2] [3] [4]で指摘してきた。これに対してプロセッサが進化するにつれて、このキャッシュのコヒーレンシ（一貫性）を保つためのオーバーヘッドは改善されているようにも見える。しかしながら今回の報告のように他の core でキャッシュコヒーレンシをチェックする論理回路が動作しているプログラムが動作している場合、メモリからのアクセスが 1.2 ～ 2.0 倍程度遅くなる。これはキャッシュコヒーレンシをチェックする論理回路が L3 の Address Directory を使用しているためと考えられる。Address Directory のコピーを持つことによりメモリからのアクセスの低下を改善できるが、現状ではそういう対応にはなっていない。

このためマルチスレッドを用いたアプリケーションプログラムを作成する場合、このような性能低下があることを十分注意してプログラミングを行わなければならない。

#### 4. 今後の課題

今後のプロセッサの進化は、1つのプロセッサに搭載するコアを増やしていく方向で進化していく可能性が高い。コアがさらに増えた場合、キャッシュの一貫性を保つ処理による性能低下はさらに大きくなるのではないかとと思われる。今後も引き続きプロセッサ特性を評価していきたいと思う。

#### 参考文献

- [1] 橋本壮広, 河辺峻: Intel 系デュアルコアプロセッサの性能解析手法, 明星大学情報学部紀要, 15, pp. 71-92 (2007)
- [2] 河辺峻, 佐藤祐治: マルチコアプロセッサの性能評価分析, 明星大学情報学部紀要, 17・18, pp. 41-46 (2010)
- [3] 河辺峻, 森下真次: 4 コアプロセッサの性能評価分析, 明星大学情報学部紀要, 19, pp. 15-20 (2011)
- [4] 河辺峻: 8 コアプロセッサの性能評価分析, 明星大学情報学部紀要, 20, pp. 15-20 (2012)