B-019

NUMA 構成の計算機におけるメモリ操作速度に着目した評価
Evaluation of Memory Access Performance in NUMA Architecture

島谷 隼生 † 山内 利宏 †† 谷口 秀夫 †† 佐藤 将也 †††
Toshiki Shimatani Toshihiro Yamauchi Hideo Taniguchi Masaya Sato

1. はじめに

クラウドコンピューティングの利用拡大に伴い，マルチコアの PU（Processing Unit）を複数搭載した計算機システムの利用が増加している．このような計算機システムは，NUMA（Non-Uniform Memory Access）構成であることが多い．NUMA 構成の計算機システムでは，コアのメモリ操作速度がコアとメモリの接続関係に依存しており，一定でない．そこで，本稿では，このコアのメモリ操作速度の違いに着目し，PU を複数搭載した NUMA 構成の計算機における性能分析を述べる．

2. 評価

2.1 観点

PU を複数搭載した計算機において，プログラムの存在するメモリ域を固定し，プログラムを実行するコアを変える．これにより，コアとメモリが近い場合と遠い場合のプログラムの処理時間を比較し，コアのメモリ操作速度の違いがプログラムの処理時間に与える影響を明らかにする．

NUMA 環境上では，コアのメモリ操作速度は，以下の 2 つの組み合わせによって変化する．

(1) AP（Application Program）と OS のプログラムが，どのメモリ域に存在するか
(2) AP と OS のプログラムが，どのコアで実行されるか

上記の要因の組み合わせにより，評価の種別を XY とする．X は AP と OS のプログラムが存在するメモリ域とし，Y は AP と OS のプログラムを実行するコアを有するプロセッサの位置（コア位置）とする．

例として，PU（1PU は 8 コアを搭載）を 4 基搭載した計算機における種別 12（X=1，Y=2）の様子を図 1 に示す．MM（Memory Module）は，末尾の数字が同じ PU に接続されたメモリである．種別名の前の数字が 1 であるため，AP と OS のプログラムは MM1 に該当するメモリ域に存在する．また，後の数字が 2 であるため，AP と OS のプログラムを PU2 のコアが実行する．

種別 1Z（Z = {0，1，2}），2Z，3Z は，いずれも 0Z のいずれかの評価種別と同じ関係になるため，種別 00，01，02，および 03 について，実測し評価する．なお，種別 0Z の測定では，MM0 のメモリ領域しか使わないように OS を制限する．

2.2 評価プログラム

3 つの評価プログラムを以下に説明する．

（A）メモリ操作処理 [1]：文献 [1] で使用したものと同じプログラムである．このプログラムでは，mmap() システムコールを用いて AP 領域に 32 MB(PU のラストレベルキャッシュよりも大きいサイズ) のバッファを確保し，AP が 4 B データをキャッシュライン (64 B) 間隔でバッファの終端まで書き込む．

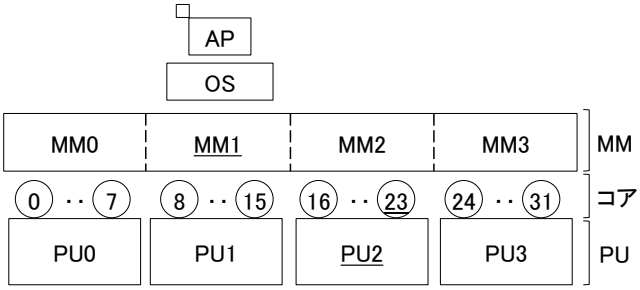

図 1 評価種別：種別 12（X=1，Y=2）

（B）hackbench：プロセスのスケジューリング性能を測定するベンチマークプログラムである．このプログラムでは，指定数のプロセスグループ内で生成されるプロセスの組 (送信プロセスと受信プロセス) 間でパイプ，またはソケットを介し，データ授受を行う．デフォルトの設定では，グループ数が 10，グループ内で生成されるプロセスの組数が 20，プロセス間で授受するデータサイズが 100 B，データ授受の繰り返し回数が 100 回である．今回の評価では，デフォルトの設定を利用する．

（C）memslap：インメモリキーバリュー DB である memcached 用のベンチマークプログラムである．このプログラムでは，memcached サーバに対して，ソケットを介して GET リクエストと PUT リクエストを繰り返す．デフォルトの設定では，GET リクエストと PUT リクエストの比が 9:1，キーサイズが 64 B，データベース内部に格納するデータサイズが 1024 B である．今回の評価では，デフォルトの設定を使用する．また，memcaced サーバと memslap クライアントはローカルループバック接続で通信し，同一の PU で実行する．

2.3 結果と考察

評価には，2 つの計算機を用いた．1 つは，Intel Xeon E5-2630 v3 (8 コア，2.4 GHz，20 MB L3 キャッシュ) を 2 基と 32 GB（DDR4-1866 4 GB x 8）のメモリ搭載した計算機 (以降，2PU) である．もう 1 つは，Intel Xeon Gold 6234 (8 コア，3.3 GHz，24.75 MB L3 キャッシュ) を 4 基と 384 GB（DDR4-2933 16 GB x 24）のメモリ搭載した計算機 (以降，4PU) である．また，OS は FreeBSD 11.3-RELEASE を使用した．

メモリ操作速度の違いによる処理時間差について，2PU の結果を図 2 に示し，4PU の結果を図 3 に示す．図 2 と図 3 より，以下のことが分かる．

† 岡山大学大学院自然科学研究科, Okayama University
†† 岡山大学学術研究院自然科学学域, Okayama University
††† 岡山県立大学情報工学部, Okayama Prefectural University

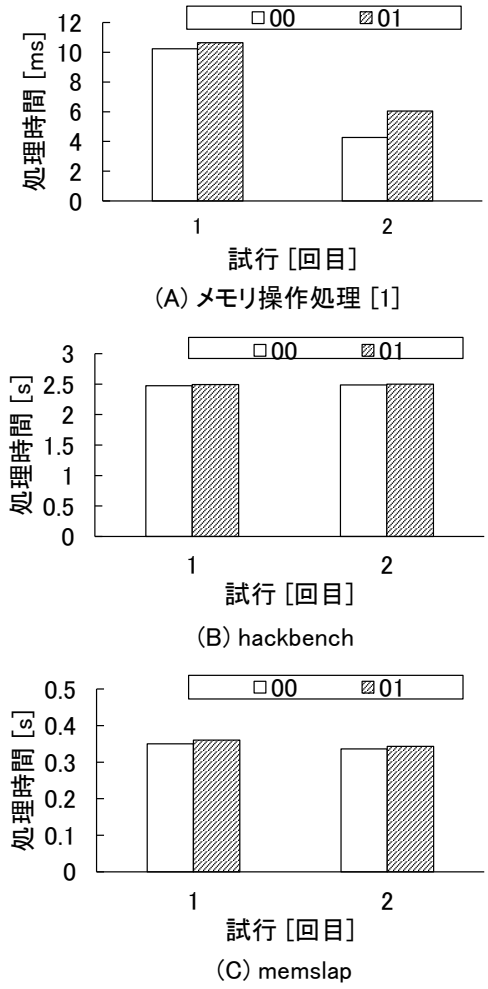

図 2　メモリ操作速度の違いによる処理時間差 (2PU)

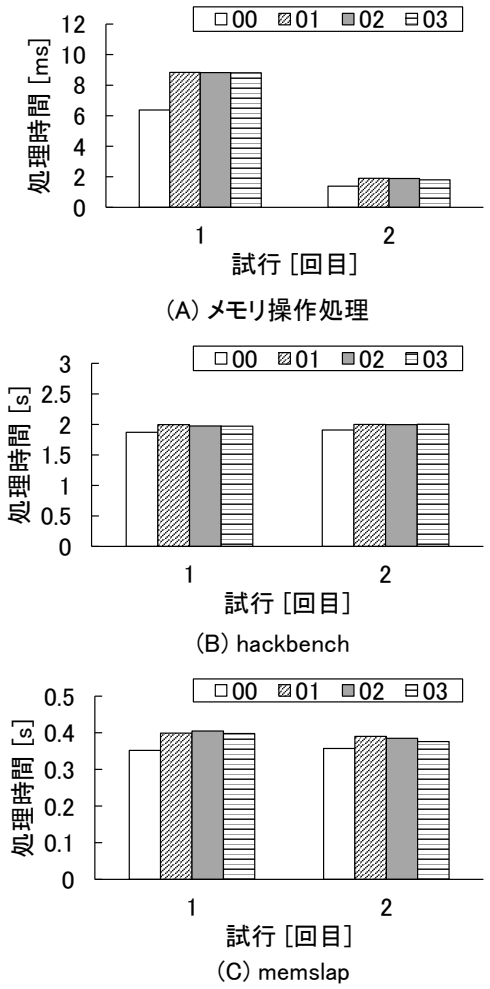

図 3　メモリ操作速度の違いによる処理時間差 (4PU)

（1）図 3 の (A) より，メモリ操作処理プログラムにおいて，1 回目の処理時間が 2 回目以降よりも長い．これは，プログラム内で測定を繰り返しており，データ書き込み時に，1 回目はページ例外が発生するものの，2 回目はページ例外が発生しないためである．また，これは文献 [1] の 2PU における結果と一致する．

（2）図 2，図 3 の (B) と (C) より，PU 数に関わらず，hackbench と memslap において，1 回目の処理時間が 2 回目と同様である．これは，メモリ操作処理プログラムと異なり，1 回の測定ごとにプログラムが終了しており，データ書き込み時に，1 回目と 2 回目でページ例外が同様に発生するためである．

（3）図 2 と図 3 より，PU 数に関わらず，コアとメモリが遠い場合（種別 01，02，および 03）の処理時間が近い場合（種別 00）より長い．例えば，図 3 の 4PU の場合，種別 00 と種別 01 の処理時間を比べると，メモリ操作処理プログラムで約 40 %（2.2 ms），hackbench で約 7 %（0.1 s），memslap で約 13 %（0.05 s）長くなる．これは，コアによるメモリ操作がすべて他 PU に接続されたメモリ（リモートメモリ）への操作となるためである．

（4）図 2 と図 3 より，評価プログラムに関わらず，4PU における種別 00 と他の種別の 1 回目の処理時間差は，2PU よりも大きい．例えば，hackbench の 1 回目において，2PU は約 1%（0.02 s）で 4PU は約 5%（0.12 s）である．これは，4PU ではリモートメモリへのメモリ操作速度の均一性を保つコストが 2PU よりも大きくなるためだと考えられる．つまり，PU 数が増加するほど，リモートメモリへのメモリ操作速度が遅くなると推測できる．

3. おわりに

PU を複数搭載した計算機において，コアのメモリ操作速度の違いがプログラムの処理時間に与える影響を評価した結果を述べた．評価では，メモリ操作処理 [1]，hackbench，および memslap の 3 つの評価プログラムを用いて，コアとメモリが近い場合と遠い場合の処理時間を比較した．

評価結果より，コアのメモリ操作速度の違いにより最大で 40 %の処理時間が長くなることを示した．また，PU 数が多い場合にコアとメモリが遠い場合のメモリ操作速度が遅くなることがわかった．

謝辞　本研究の一部は，JSPS KAKENHI 21K11830 による．

参考文献

[1] 島谷 隼生，佐藤 将也，谷口 秀夫：クラウドコンピューティングを支える仮想計算機技術の性能分析，情報処理学会研究報告，vol.2020-DPS-185, No.8, pp.1–6，2020.