B-003

# NAS CG ベンチマークによるプリフェッチ機能付きメモリモジュールの性能評価

# Performance Evaluation of Memory Module with Prefetch Function by NAS CG Benchmark

箱崎 博孝† 安藤 宏† 田邊 昇‡ 土肥 康孝† 中條 拓伯§ 天野 英晴¶
Hirotaka Hakozaki Hiroshi Ando Noboru Tanabe Yasunori Dohi Hironori Nakajo Hideharu Amano

## 1. はじめに

昨今の半導体技術やアーキテクチャの進歩により，COTS（Comercial Off-The-Shelf）である Pentium4 などのマイクロプロセッサ（MPU）の演算処理能力は，そのコストパフォーマンスに加え，ここ数年で飛躍的に向上している．この進歩はこの先も当分，Moore の法則により維持されると言われている．

一方，PC における I/O バスやネットワークインターフェイス（NIC）に対する進歩は，一般ユーザーの要求が低かったため，PCI バス[1]が非常に長い間スタンダードであった．よって NIC の発展は，Moore の法則に従って発展してきたメモリや MPU に比べ遅れをとり，発展スケジュールもかけ離れたものになってしまった（図 1）．

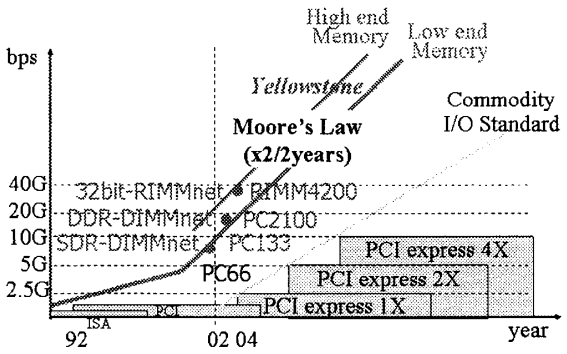


図 1: 標準 I/O と MEMOnet[2] の発展スケジュール

今日，ベクトル型スーパーコンピュータのユーザーは PC クラスタに移り，HPC（High Performance Computing）サーバにおいても，コストパフォーマンスの高い IA-64 や x86 サーバが急成長し，その市場は年々減少している．HPC の期待はコストパフォーマンスの高い PC ベースのクラスタコンピュータに集まってきている．

しかしながら，COTS 部品である Pentium4 などの MPU のほとんどは，キャッシュアーキテクチャに基づいている．キャッシュアーキテクチャは，主記憶の帯域幅の脆弱さを隠蔽するためのものであり，ベクトル型スーパーコンピュータと比べると，低コストの PC における主記憶は，例えば，リレーショナルデータベースや回路シミュレーションなどの不連続アクセスが必要なアプリケーションにおいては，キャッシュが効かなく，その演算能力が充分に発揮できない[3]．

近年のクラスタ技術に応用されている MPI や PVM など，小さなパケットを多数やり取りする場合，通信レイテンシが大きいとクラスタの本来持つ性能が得られない。クラスタコンピューティングには，Myrinet などの低レイテンシネットワークを用いたものや，Gigabit Ethernet など汎用高速ネットワークを用いたものがある。これらは何れも PCI バスに接続されるタイプのインターフェイスを用いるが，PCI バスでは光インタコネクションの持つ大きなバンド幅およびレイテンシともに力不足である。一部サーバなどで用いられている PCI-X[1] は多少高価であり，汎用 PC に搭載された例も少なくは無いが，PCI Express[1] の登場により，今後標準になる可能性も少ない．

このような背景の中，筆者らはメモリスロットに搭載されるネットワークインターフェイス[2][4] やプリフェッチ機能付きメモリモジュール[5][6] を提案してきた．

今年，次世代の I/O バス PCI Express が登場し今後の動向が注目されているが，当分は 1 レーン (x1) 構成を汎用の I/O スロット (現在の PCI スロットの置き換え) に，x16 をグラフィックス用 (現在の AGP スロットの置き換え) に，それぞれ使われる見込みだ．また，I/O バスや NIC の進歩のスケジュールはメモリや MPU に比べ遅れをとっており，それらのバンド幅が MPU の性能に追いつく見込みは少ない．本報告では新たな問題サイズの実行結果と，バス遅延を考慮した提案システムの有用性を述べる．

## 2. 提案方法

### 2.1 キャッシュアーキテクチャの弱点

キャッシュを用いた CPU を使って間接配列参照 (Gather 処理) を行うと以下の問題が発生する．

1. ポインタがメモリと CPU の間を往復することに伴うメモリバンド幅消費
2. 有効データが少ないために起こるメモリバンド幅の浪費
3. 有効データが少ないために起こるキャッシュエントリの浪費

†横浜国立大学
‡（株）東芝, 研究開発センター
§東京農工大学
¶慶應義塾大学

4. 有効データが少ないために起こる TLB エントリの浪費

キャッシュを用いた CPU を使って間接配列参照 (Gather 処理) を行った場合の動作と問題点を図 2 に示す.

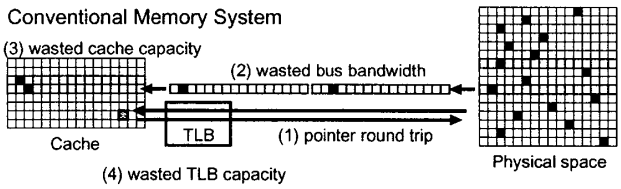


図 2: 間接配列参照における問題点

### 2.2 プリフェッチ機能付きメモリモジュール

これらの弱点を軽減するために，本研究ではプリフェッチ機能付きメモリモジュールを提案してきた.

メモリ空間にマップされたメモリモジュール側にあるバッファ(プリフェッチバッファ) へのプリフェッチコマンドを発行することによって，ホスト CPU から利用確率が高い状態に整えられたデータ群に対してブロックアクセスを行うことが出来るようになる.

その結果，キャッシュ・TLB・FSB・メモリバスの利用効率が向上する．メモリモジュールは着脱可能なので，CPU やチップセットを改造することなく，高性能かつ低価格な COTS を HPC 向けコンピュータとして有効に活用できる.

なお，プリフェッチコマンドには種々の実装法がありうるが，本研究ではベクトル転送命令について検討する．本報告で検討するのはベクトル間接ロード命令で，配列間接参照をベクトルレジスタに対して行う命令である.

図 3 は提案メモリモジュールにおける書き込み用と読み出し用のバッファとして考案されたプリフェッチ機能付き Window メモリであり，これらがベクトルレジスタとして用いられる．64bit データ毎にフラグがついている.

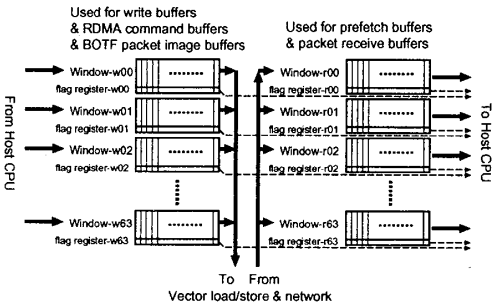


図 3: プリフェッチ機能付き Window メモリ

これらを用いて，間接配列参照を行った場合の動作と高速化の原理を図 4 に示す.

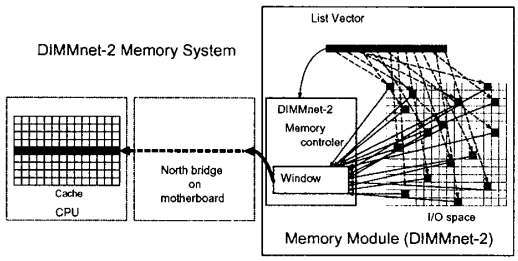


図 4: 提案システムにおける間接配列参照

## 3. NAS CG におけるソフト的対応

### 3.1 NAS CG ベンチマーク

NAS CG は NASA が並列コンピュータのために開発した NAS Parallel Benchmarks(NPB) の一つで，正値対称な大規模疎行列の最小固有値を求めるための共役勾配法を実行しているプログラムである.

各問題には問題サイズの異なる 5 つのクラス，S(ample)，W(orkstation)，A，B，C が定義されている.

NAS CG(シリアル版) のカーネル部分は以下の通りであり，殆どの処理時間がこの部分で消費されるため，この部分の高速化が重要である．その部分にはリストベクトル colidx[k] による間接参照がある.

```
for (j = 1; j <= lastrow-firstrow+1; j++) {
   sum = 0.0;
   for (k = rowstr[j]; k < rowstr[j+1]; k++) {
      sum = sum + a[k]*p[colidx[k]];
   }
   w[j] = sum;
}
```

ラインサイズ 128 バイトのキャッシュを内蔵する Pentium4 においては，クラス S や W ではキャッシュに大半の配列要素が載るのでメモリバンド幅の問題は生じないが，クラス B では二次キャッシュからもあふれ出て大半が主記憶へのアクセスとなる．非零要素が十分に疎らであるため，p[] の間接参照を行う際に 1 ラインの中の他の非零要素はほとんどの場合入っていないと考えられる．よって，p[] の間接参照はメモリバンド幅を消費してしまい，性能はメモリバンド幅がボトルネックとなる.

## 4. 性能評価

### 4.1 評価環境

性能評価は，RWCP による C 言語 NPB CG を提案メモリモジュールで使用できるように改造した [6] ものを使用する．改造に使用したパラメータ設定を表 1 に示す．また，表 2 に性能評価を行ったマシンの仕様を示す.

### 4.2 提案システムの性能

図 5 にハードを理想化した場合の提案メモリモジュールによる NAS CG の実行速度確認プログラムの実行結果を示す．キャッシュから溢れ出ると思われるクラス B，C に対して，

1. Original：RWCP 版 C 言語表記 NAS CG

2. Clflush：CLFLUSH 命令を用いた提案システム

3. NoClflush：CLFLUSH 命令を用いない提案システム

の 3 種類の測定をした.

表 1: パラメータ仕様

| | |
|---|---|
| Array copied to the module | p[], colidx[] |
| ♯ of used vector register for p | 1 |
| ♯ of used vector register for colidx | 1 |
| ♯ of used vector register | 1 |
| Words/vector register | 64 |
| Capacity/vector register | 512B |
| Used vector command | VLI (Vector indirect load) VL(Vector load) VS(Vector store) |
| Polling loss for VLI completion | 1word read/command |

表 2: 評価環境

| | |
|---|---|
| 機種名 | Dell Precision360 |
| CPU | Pentium4 |
| FSB 周波数 | 800MHz |
| コア周波数 | 2.4GHz |
| L1 キャッシュ容量 | 8KB |
| L2 キャッシュ容量 | 512KB |
| L1 キャッシュラインサイズ | 64B |
| L2 キャッシュラインサイズ | 128B |
| メモリ種類 | PC3200 (DDR SDRAM) |
| メモリバス本数 | 4 |
| メモリ容量 | 4GB |
| OS | Linux 2.4.20-8 |
| コンパイラ | gcc 3.2.2 |
| 最適化オプション | -O3 |

#### 4.2.1 クラス C における性能向上

提案システムを用いてクラス B で実装した結果，CLFLUSH を用いた場合において 74%の加速が得られた。更にクラス C においてはクラス B を上回る 159.8%の高速化が実現された．これはクラス B よりも問題サイズの大きいクラス C の方がキャッシュから溢れる割合がより多く，提案システムを十分に活かした更なる効果が得られたためと考えられる．

#### 4.2.2 CLFLUSH 影響

CLFLUSH 命令を全てコメントアウトした場合，クラス B，C それぞれにおいて 269.8%，479.9%の加速が得られた．よって CLFLUSH 命令の実行時間や，その命令

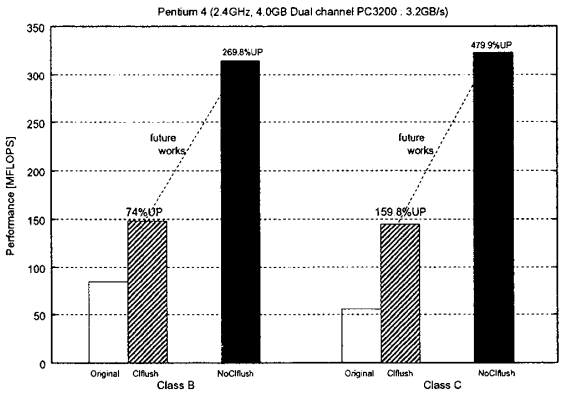


図 5: 提案システムの性能と CLFLUSH の影響

による影響が考えられ，それらを最大限排除できれば上記のような加速が得られると考えられる．

### 4.3 バス遅延を考慮した評価

#### 4.3.1 ポーリング時間

今回ポーリングによる遅延時間を空ループにより表現し，実行プログラムに加えることで，ポーリングによる遅延による影響下での提案システムの性能を測定した．

#### 4.3.2 バス遅延

メモリバスの遅延は，uncached 属性に設定されたメモリ領域へのリードを評価環境の下で実測した値を用いた．

一方，今回使用した PCI Express における遅延時間は，今後は Intel(R)925/915[7] チップセット搭載マザーボードが市場に出ていなく実測が不可能なため表 3 に表した行程でシリアル転送される事と，

表 3: PCI-Express 転送行程

| 送信側 | 受信側 |
|---|---|
| Header 生成 | PIPE インターフェイス |
| 送信バッファ制御 | デスクランブリング |
| FlowControl Credit 検査 | レーン to レーンデスキュー |
| VirtualChannel 調停 | 受信デフレーミング制御 |
| TLP フレーミング | シーケンスナンバー検査 |
| ECRC 生成 | LCRC 検査 |
| LCRC 生成 | パケット認識 |
| 送信パケット調停 | ECRC 検査 |
| 送信フレーミング制御 | 受信バッファ制御 |
| スクランブリング | アプリケーション IF |
| PIPE インターフェイス | |

これら各項目が 1〜2 クロックサイクルで動き，周波数 125MHz，更に PCI バスへのアクセス時に発生する

遅延時間からデータ転送部を差し引いた時間を約 200ns と見積もり，以上を踏まえ約 500ns～600ns と見積もった予測値を使用している．

#### 4.3.3 バス遅延を考慮した性能

バスによる遅延を加味した，クラス B，C の結果を図 6 に示す．

この結果から提案システムにおけるポーリング時間による影響は小さく，100ns と 600ns の値を比較すると，CLFLUSH 命令をコメントアウトしない場合において，クラス B，C でそれぞれ約 35%，60%程が PCI-Express よりも早く，CLFLUSH 命令をコメントアウトした場合においては，それぞれ 119%，128%提案システムのほうが速いことが解った．よって本提案システムを使用するメリットを実証できたと言えるが，やはり CLFLUSH 命令の影響は大きく，今後の課題となる．

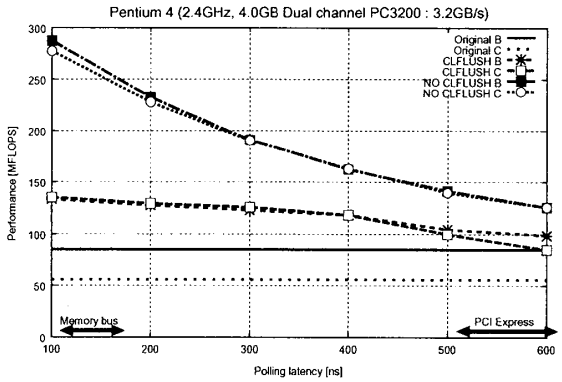


図 6: バス遅延を考慮した性能

## 5. まとめ

プリフェッチ機能付きメモリモジュール及びメモリスロット搭載型ネットワークインターフェイスにおける性能評価を NAS CG ベンチマークを用いて行い，それら提案システムの有用性を示してきた．

ソフトウェアオーバーヘッドを考慮した上で，理想的な提案システムをハード化した今回の実験方法において，クラス B においては 74%増の高速化を示したが，より問題サイズの大きいクラス C においては 160%増しの，より大きな高速化を達成出来ることが解った．これは問題サイズが大きいほど，よりキャッシュから溢れ，主記憶からのアクセスが頻繁に起こる為，提案システムの効果が出たものと考えられる．

更に，提案システムと PCI-Express をポーリングによる遅延時間を考慮して性能比較した結果，提案システムの方がクラス B において 35%，クラス C においては 60%程高速であり，比較的低遅延であるメモリバスを用いた方が性能は高い．

一方，CLFLUSH 命令による影響を改善することが出来たならば，提案システムにおいてクラス B で 313.93MFLOPS，クラス C においては 322.77MFLOPS に高速化が可能であるといえ，ポーリング遅延による影響下でも PCI-Express より 130%増の高速化が期待できる．よって CLFLUSH 命令による影響を最小化することが今後の課題である．



## 参考文献

[1] PCI-SIG, http://www.pcisig.com

[2] 田邊, 山本, 工藤：“メモリロットに搭載されるネットワークインターフェイス MEMnet”, 情報処理学会研究報告, 99-ARC-134（SWoPP'99）, pp.73-78 (Aug. 1998)

[3] 萩原, 梅澤：“パーソナルスーパーコンピュータ SX-6i”, 情報処理, Vol.44,No.3, pp.277-282 (Mar. 2003)

[4] 田邊, 濱田, 中條, 天野：“メモリロット装着型ネットワークインタフェース DIMMnet-2 の構想”, 情報処理学会計算機アーキテクチャ研究会, 2003-ARC-152, pp.61-66 (Mar. 2003)

[5] 田邊, 土肥, 中條, 天野：“プリフェッチ機能を有するメモリモジュール”, 情報処理学会計算機アーキテクチャ研究会, 2003-ARC-154, pp.139-144 (Aug. 2003)

[6] 田邊, 中武, 箱崎, 土肥, 中條, 天野：“プリフェッチ機能付きメモリモジュールによる不連続アクセスの連続化”, 情報処理学会計算機アーキテクチャ研究会, 2004-ARC-157, pp.139-144（Mar.2004）

[7] Intel, http://www.intel.co.jp/