RC-005

DMA で主記憶をアクセスする CPU における不連続アクセスの連続化

田邊 昇* 太田 淳† 金 美善† 中條 拓伯†

1 はじめに

いくつかの重要アプリケーションでは，メモリアクセスの空間的局所性が乏しく，不連続アクセスが性能ネックである．例えば NAS CG ベンチマークはリストアクセスが性能ネックである．Wisconsin ベンチマークは主記憶上にデータベースが配置された場合，等間隔アクセスが性能ネックである。これらは，キャッシュベースの CPU では例えば 128 バイトのキャッシュラインの中に有効なデータが 8 または 4 バイトしかない非効率的なアクセスが発生するため著しい性能低下があった．

上記の問題の解決のため，これまで筆者らはキャッシュベースの COTS の CPU やマザーボードをそのまま用いることが可能でメモリスロットに装着可能なベクトル型のプリフェッチ機能を有するメモリモジュールである DIMMnet-2[1][2][3] および DIMMnet-3[4] の研究開発を行ってきた．

一方，Cell Broadband Engine(Cell/B.E.)[5][6] や SpursEngine のようにキャッシュを持たない単純な CPU コアを多数内蔵するマルチコア CPU が注目されている．これらは小容量のローカルメモリを持ち，主記憶とローカルメモリの間を DMA 転送によりデータ転送することでキャッシュの代用をさせる．これらの CPU ではキャッシュラインと同等の内部バス転送単位とアプリケーションの不整合の問題だけでなく、DMA 起動やバス調停のためのオーバーヘッドが存在し、キャッシュベースの CPU 以上に不連続アクセスの問題が深刻である．

DMA で主記憶をアクセスする CPU における不連続アクセスの高速化に関する従来研究は数少ないが，Cell/B.E. における DMA リスト [7] はその一つである．しかし，特にバースト長が小さい不連続アクセスが支配的なアプリケーションにおいては効果が限定的である．よって，さらなる高効率を実現できるアーキテクチャの開発が望まれる．

本論文では DMA で主記憶をアクセスする CPU における不連続アクセスに伴う上記の課題の解決方法を提案し，その評価を東芝 Cell リファレンスセット (以下 CRS) 上で行なった．以下，第 2 章で DMA で主記憶をアクセスする CPU とその一例である Cell/B.E. およびその開発環境である CRS の概要について紹介する．第 3 章で上記アーキテクチャをとる CPU の不連続アクセスにおける課題を述べる．第 4 章で上記の課題の解決法を提案する．第 5 章では提案方式の性能評価について述べ，第 6 章で関連研究について述べ，第 7 章でまとめる．

2 DMA で主記憶をアクセスする CPU

DMA で主記憶をアクセスする CPU としては IBM，ソニー，ソニー・コンピュータエンタテインメント，東芝が共同で開発した Cell Broadband Engine(Cell/B.E.) や，東芝の SpursEngine などがある．これらは，CPU コアを単純化して多数チップ内に内蔵することにより，チップ内の演算性能を向上させるとともに，データ転送をキャッシュと比較してプログラマから制御しやすいものとすることで，実行性能チューニングの可能性を高めている．図 1 に，Cell/B.E. の構成を示す．

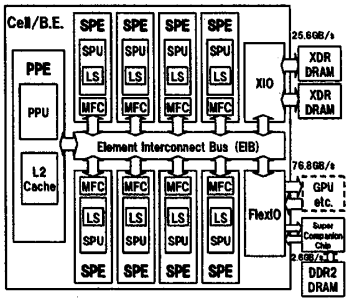

図 1 Cell Broadband Engine (Cell/B.E.) の構成

1. マルチコア・アーキテクチャ・デザインを採用
2. 8 個の演算に特化したコア SPE と，1 個の汎用コア PPE を搭載
3. 各 SPE は SIMD 型演算処理ユニット，128 個の 128 ビットレジスタファイル，および 256KB の Local Strage(LS) を有す
4. 外付けの XDR DRAM ベースの主記憶は XIO を介して接続しており，他の外部チップは I/O Interface(FlexIO) を介して接続
5. PPE，8 個の SPE，主記憶，および他の外部チップの相互間データ転送には，超高速データ転送バス Element Interconnect Bus(EIB) が用いられる

Cell/B.E. が搭載する 8 個の SPE は，それぞれ LS を持ち，実行するコードやデータをすべて LS に格納する．しかし，SPE は直接主記憶にアクセスできないため，必要に応じて，演算に必要なデータなどを主記憶から LS へ DMA(Direct Memory Access) 転送しなければならない．

3 解決すべき課題

本章では DMA で主記憶をアクセスする CPU における不連続アクセスに関連する課題について Cell/B.E. を例に述べる．

3.1 DMA コマンドオーバーヘッド

DMA コマンドを発行するには少なからずソフトウェアオーバーヘッドが存在するので，細粒度の DMA 転送が頻繁に発生するアプリケーションの性能は制約される．この問題は Cell/B.E. にも実装されている DMA リストを用いることによりある程度軽減することが可能である．

* (株) 東芝，研究開発センター
† 東京農工大学

3.2 内部バスの調停オーバーヘッド

Cell/B.E. のように調停回路から内部バスのアクセス権利を取ってから DMA 転送を行なう種類の CPU では，少なからず調停オーバーヘッドが存在するので，細粒度の DMA 転送が頻繁に発生するアプリケーションの性能は制約される．この問題は Cell/B.E. にも実装されている DMA リストを用いても軽減することができない．

3.3 内部バスの転送単位との兼ね合い

Cell/B.E. では前述の調停オーバーヘッドとの兼ね合いからも長めのバースト転送における内部バスの転送効率を向上させるために，内部バスの最低転送単位を 128 バイトに設定されている．ところが，NAS CG ベンチマークや Wisconsin ベンチマークに代表されるいくつかの重要アプリケーションではアプリケーション上での転送単位は 8 バイトまたは 4 バイトの不連続アクセスとなる．このため，DMA リストを用いて DMA コマンドオーバーヘッドを軽減したとしても，このようなアプリケーションにおけるバスの実効バンド幅は 8/128 または 4/128 に低下してしまう．

4 提案方式

本章では上記の問題を解決するための解決策として，DMA で主記憶をアクセスする CPU への外付けハードウェア追加，そのコンパニオンチップへの改良および同 CPU への改良について提案する．

4.1 提案方式の基本コンセプト

DMA で主記憶をアクセスする CPU における前章での課題の解決策として，DIMMnet-2 と同様の連続化ハードウェア（分散/収集機構）を外部メモリに近い場所に追加することを提案する．提案方式の基本コンセプトを図 2 に示す．

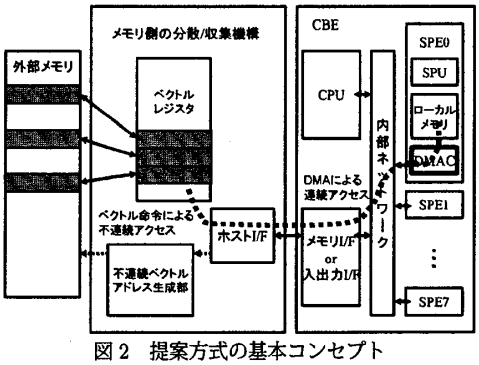

図 2 提案方式の基本コンセプト

表 1 に DIMMnet-2 の主なベクトル型アクセスコマンドを示す．このうち，等間隔ロード/ストア，リストロード/ストアのコマンドが不連続アクセスの連続化を実行するものである．ロード系が外部メモリから一種のベクトルレジスタである Prefetch Window への収集 (Gather) 処理を行い，ストア系が一種のベクトルレジスタである Write Window からの外部メモリへの分散 (Scatter) 処理を行なう．

本提案は，このような分散/収集処理が可能な追加ハードウェアを DMA で主記憶をアクセスする CPU のメモリサイドに設けることにより，CPU 内部での細切れな DMA 転送コマンド発行を抑制し，それに伴う内部転送資源の浪費や効率低下に伴う性能低下を抑制するものである．

表 1 DIMMnet-2 の主なベクトルコマンド

ロード	連続	VL
	等間隔	VLS
	リスト	VLI
ストア	連続	VS
	等間隔	VSS
	リスト	VSI

4.2 ハード的な実装方式

4.2.1 DIMMnet 装着による方式

DIMMnet-2 や DIMMnet-3 は COTS の CPU やチップセット (コンパニオンチップ)，マザーボードに改造をすることなく，メモリスロットに後付けで装着することで不連続アクセスの連続化機能を追加することができる．

現状の DIMMnet-3 は CRS の SO-DIMM スロットに装着可能な子基板を有しており，前述の基本コンセプトを CRS に実現可能である．ただし，CRS の SO-DIMM スロットはピークバンド幅が 2.56GB/s に留まっており，その十倍のバンド幅である XDR DRAM による主記憶に比べてバンド幅が低いので，その効果は限定的であるものと考えられる．

一方，CRS 上では XDR DRAM がメインボード上に直接実装されているが，XDR DRAM 自体は技術的にはメモリモジュールの形態での実装が可能である．よって XDR DRAM のメモリスロットを装備した Cell/B.E. 関連機器においては，XDR DRAM インタフェースを有する DIMMnet 子基板を開発することで，高い主記憶バンド幅を背景にした基本コンセプトを実現できる可能性がある．

4.2.2 コンパニオンチップ改良による方式

Cell/B.E. 自体には FlexIO という上記の SO-DIMM スロットよりも高いバンド幅を有する入出力ポートが存在する．ノースブリッジに相当するコンパニオンチップやマザーボードの新規開発が必要になるが，FlexIO インタフェースで動作する連続化ハードウェアと拡張メモリを実装することで，Cell/B.E. 自体には改造を加えることなく，前述の基本コンセプトを実現可能である．

4.2.3 CPU チップ改良による方式

東芝による SpursEngine や IBM による RoadRunner 向け CPU など，Cell/B.E. の派生製品である改良型 CPU の開発事例がいくつかある．このようなケースでは CPU チップにマイナーな改造を加えることで，従来の Cell/B.E. に付加価値を加えることができる．

そのような派生 CPU の開発の際に，本提案のハードウェアを主記憶コントローラや，入出力コントローラの中に実装することで，本提案のコンセプトを高性能に実現することが可能であると考えられる．

4.3 ソフト面での改造方法

上記の提案方式におけるソフトウェアの改造においては以下のような方針で行なう．

1. 主記憶とローカルメモリの間で細かいデータサイズで行なわれる多数回の DMA コマンドの繰り返しを，主記憶との間で Prefetch Window に収集/Write Window から分散する少数回のベクトルロードコマンドと，Window とローカ

ルメモリの間で基本的には Window サイズで行なわれる少数回の DMA コマンドに変更する.

2. DMA で主記憶をアクセスする CPU にはキャッシュがないため，Pentium4 等のキャッシュベースの CPU 向けの改造の際に必要になるキャッシュライフフラッシュ命令の挿入は不要である.

5 性能評価

本章では，Cell/B.E. の主記憶側 (主記憶コントローラ内または XDR DRAM の場所) に DIMMnet-2 同様の Gather 回路がある状態を仮定して，等間隔アクセスを主体とする処理として Wisconsin ベンチマークを用い，CRS 上で性能を評価した．表 2 に実機評価の評価環境を，表 3 にコンパイル環境を示す.

表 2 評価環境 (CRS 実機)

モデル	Cell リファレンスセット
CPU	Cell B.E. / 3.2GHz
チップセット	TOSHIBA Super Companion Chip
主記憶	XDR 512MB ECC 対応

表 3 コンパイル環境

GCC バージョン	ppu-gcc / spu-gcc 3.4.1
コンパイルオプション	PPE: -O3 -m32 SPE: -O3

5.1 評価に用いたベンチマーク

本研究では，Wisconsin ベンチマークを用いて等間隔アクセスによる主記憶データベースの高速化の評価を行った.

Wisconsin ベンチマークでは 15 個の属性からなるタプルが 10K 個，または 1K 個から構成されるデータベースが検索対象になる．1 個のタプルは 15 個の属性が 4 バイトのデータまたはポインタからなる (合計 60 バイト)．本評価ではタプル数を 1K 個にして，データが全部 Cell 上の 1 個の SPE のローカルメモリ 256KB のローカルメモリに入る状態で評価を行った．本評価では参考文献 [2] において他の複数のクエリーと傾向が同じということで遅延変動の評価に用いられていた以下の最小値を検索するクエリーを用いた.

(Q7) select MIN(unique2) from tenk1

評価に際しては上記クエリーを Cell/B.E. 上の PPU と 1 個の SPU で動作する C 言語で記述し、これをオリジナルプログラムとした．これを DIMMnet のベクトルコマンドを用いるように改造して，比較評価を行った．ハードウェアで実行する部分を記述したエミュレーション版を作成し，その実行結果を確認し，改造の妥当性をチェックした.

5.2 ゼロ遅延モデルでの検索性能

まずは，ハードウェアによる外部メモリアクセスが理想的で，ベクトル等間隔ロード関数コール直後 1 回目のコマンド完了フラグチェックまでに Prefetch Window へのロードが終わってしまうほど十分にハードウェアが低遅延な場合 (これをゼロ遅延モデルと呼ぶことにする) に相当する加速率を測定する．本測定においてはハードウェア部の記述をしている関数の中身をコメントアウトすることで，ハードウェア部の性能不足に起因する遅延がゼロになった状態の実行時間を再現して，ベクトルコマンド起動に関するオーバーヘッドを含んだ性能を測定した.

上記クエリー Q7 に関して，ゼロ遅延モデルの評価を行った結果を図 3 で示す．測定結果は，クエリー Q7 の性能を，オリジナルのベンチマークの実行時間と比較したときの相対値である加速率で示している．ここで，Prefetch Window のサイズは 512B，つまり 4 バイトのデータを 128 個分に固定している．PW1 は Prefetch Window を単純に 1 枚用いた場合の加速率で，PW2 は Prefetch Window を 2 枚用いた場合で，original list は Cell/B.E. 自身のデータギャザー機構を使った場合の加速率である.

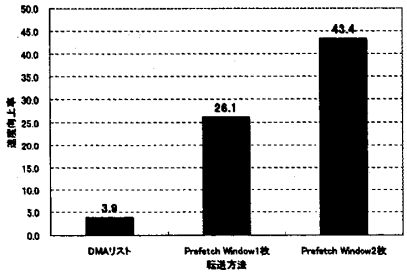

図 3 ゼロ遅延モデルでの Wisconsin Q7 検索性能

その結果，オリジナルに比べ，Prefetch Window を 1 枚だけ用いた場合は 26.1 倍，Prefetch Window を 2 枚用いた場合は 43.4 倍の加速率が得られた．そして，現状の Cell/B.E. のデータギャザー機構である DMAlist 機構を使った場合に比べても，Prefetch Window を 1 枚だけ用いた場合は 6.7 倍，Prefetch Window を 2 枚だけ用いた場合は 11.1 倍の加速率が得られた.

5.3 プリフェッチにかかる遅延時間の影響

DIMMnet 上の外部メモリから Prefetch Window までの等間隔ベクトルロード実行にかかる時間を変化させたときの性能の変化を測定した．仮想的に変動させる遅延は，ベクトルコマンドが実行する処理を記述した関数をコメントアウトし，代わりにベクトルコマンドの実行が消費する時間に対応する回数の空きループの反復回数によって変化させた遅延を挿入することで実現する．ここで，空きループと遅延時間の関係はあらかじめパフォーマンスカウンタによる予備実験で確認しておく．本測定における対象もクエリー Q7 である．その結果を図 4 で示す.

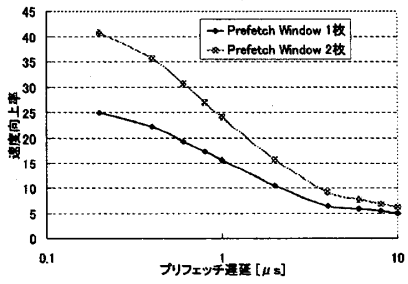

図 4 プリフェッチの遅延を変動させた時の Q7 処理性能

その結果，512 バイトのデータのプリフェッチにかかる時間が 1 μ秒以下ならば十分に高い加速率が得られることがわかった．外部メモリの種類の違い (DDR，DDR2，DDR3，XDR など) やそのバンク数によって不連続データのプリフェッチにかかる時間が変動するが，100MHz の DDR ベースの 2 バンクのメモリである DIMMnet-2 上では 1 μ秒強であり，Cell/B.E. に

採用されているXDR DRAMなどのより高速なメモリでは，それより大幅に少ない遅延を想定できるため，本方式は有望であると言える．

また，キャッシュベースのCPUにおける評価結果[2]と同様に，Prefetch Windowが1枚に比べ2枚の場合，等間隔ロードコマンド実行にかかる時間に対する耐性が強かった．タプル数1Kという小規模なデータベースでの実験では，Prefetch Windowのサイズや枚数を増やす効果は少ないと思われるが，よりタプル数が多いデータベースを検索する場合には，これらの測定パラメータの変更により効率向上が期待できると考えられる．

6 関連研究

メモリコントローラを改善することによる不連続アクセスの高速化に関する従来研究にはImpulse[8]，SDT[9]がある．しかし，これらはCell/B.E.のようなDMAで主記憶をアクセスするCPUへの適用を提案するものでもない上，その種のCPUと組み合わせた場合における効果を評価したものでもない．

キャッシュベースのCPUにおける不連続アクセスの高速化に関する従来研究には筆者等が行なったDIMMnet-2を用いた研究がある．NAS CGによるリストアクセスの高速化[1]や，Wisconsinベンチマークによる等間隔アクセスの高速化[2]が評価されている．しかし，これらはDMAで主記憶をアクセスするCPUにおける評価ではない．また，キャッシュベースのCPUに適用した場合は，キャッシュラインの無効化が必要であり，性能向上は無効化処理によって効果が半減してしまう．

DMAで主記憶をアクセスするCPUにおける不連続アクセスの高速化に関する従来研究は数少ないが，Cell/B.E.におけるDMAリスト[7]はその一つである．しかし，DMAリストでは内部バス調停オーバーヘッドが回避できないことや，内部バスの最小転送単位が128バイトであるためキャッシュベースの転送における転送効率の悪化と同様に有効なデータの割合が低い状況に陥るので，特にバースト長が小さい不連続アクセスが支配的なアプリケーションにおいては効果が限定的である．

7 おわりに

本論文では，メモリ側に配置されたGatherハードウェアによる，DMAで主記憶をアクセスするCPU上での不連続アクセスの連続化を提案し，その効果を東芝Cellリファレンスセット(CRS)上で測定した．

Cell/B.E.の主記憶側(主記憶コントローラ内またはXDR DRAMの場所)にDIMMnet-2同様のGather回路がある状態を仮定して，等間隔アクセスを主体とする処理としてWisconsinベンチマークを用い，CRSの実機上でのソフトウェアエミュレーションと人工的遅延挿入によって性能を評価した．Wisconsinベンチマークでは，データベースのある属性に対する検索処理を行う際には等間隔アクセスとなる．本研究では，DIMMnet-2の等間隔アクセス命令VLSを検索処理に適用した．

その結果，プリフェッチがプリフェッチ完了フラグ確認より前に終わる実装を仮定した場合，最小値を検索する問合せ処理が単純なDMAを繰り返す場合に比べ，Prefetch Windowを1枚だけ用いた場合は26.1倍，Prefetch Windowを2枚用いた場合は43.4倍の加速率が得られた．DMAリストを用いるプログラムに対しても6.7倍高速化され，Prefetch Windowを2枚用いると11.1倍高速化されることがわかった．

また，プリフェッチにかかる遅延を変動させた場合は，512バイトのデータのプリフェッチにかかる時間が1 μ秒以下ならば十分に高い加速率が得られることがわかった．100MHzのDDRベースの2バンクのメモリであるDIMMnet-2上では1 μ秒強であり，Cell/B.E.に採用されているXDR DRAMなどのより高速なメモリでは，それより大幅に少ない遅延を想定できるため，本方式は有望であると言える．

ローカルメモリには納まりきらない主記憶上の大きなデータベースに対する性能評価は今後の課題である．また，Prefetch Windowのサイズや枚数を増やし，DMA転送効率の向上や，プリフェッチにかかる時間をさらに隠蔽することも課題として挙げられる．また，本研究で明らかになった有効性を背景に，大規模データ可視化装置への本手法の応用が予定されている．その詳細設計と試作・評価は今後の課題である．

参考文献

[1] 田邊, 安藤, 箱崎, 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によるPC上での間接参照の高速化”, 情報処理学会論文誌コンピューティングシステム,Vol. 46, No. SIG12 (ACS11), pp. 1-12(Aug. 2005).

[2] 田邊, 羅, 中條, 箱崎, 安藤, 土肥, 宮代, 北村, 天野: プリフェッチ機能を有するメモリモジュールによる等間隔アクセスの高速化，ハイパフォーマンスコンピューティングと計算科学シンポジウム(HPCS2006), p.55-62(Jan. 2006).

[3] 北村, 濱田, 宮部, 伊澤, 宮代, 田邊, 中條, 天野: DIMMnet-2ネットワークインターフェースコントローラの設計と実装，情報処理学会論文誌，Vol. 46, No. SIG12 (ACS11), pp.13-26 (Aug. 2005).

[4] 田邊, 北村, 宮部, 宮代, 天野, 羅, 中條: 主記憶以外に大容量メモリを有するメモリ/ネットワークアーキテクチャ，情報処理学会計算機アーキテクチャ研究会，2007-ARC-172-27, pp.157-162 (Mar. 2007).

[5] 東芝セミコンダクター社： “Cell Broadband Engine”, http://www.semicon.toshiba.co.jp/product/micro/cell/index.html

[6] Cell User's Group： “Cell 関連情報”, https://www.cellusersgroup.com/modules/product/

[7] M. Kistler, M. Perrone, and F. Petrini: "Cell Multiprocessor Communication Network: Built for Speed" IEEE Micro, vol. 26(3) pp. 10-23, May-June 2006.

[8] Carter, Hsieh, Stoller, Swanson, Zhang, Brunvand, Davis, Kuo, Kuramkote, Parker, Schaelicke and Tateyama： “Impulse : Building a Smarter Memory Controller”, International Symposium on High Performance Computer Architecture (HPCA-5), pp.70-79 (Jan. 1999)

[9] K.Tanaka, T.Fukawa： “Highly Functional Memory Architecture for Large-Scale Data Applications”, International Workshop on Innovative Architecture for Future Generation High-Performance Processors and Systems (IWIA2004), pp.109-118 (Jan. 2004)

[10] Jim Gray. The Benchmark Handbook. Morgan Kaufmann, 1993.