

動いているお茶の水1号

6K-4

— 大域構造体先行フェッチ機構の評価 —

大津 金光

松本 尚

平木 敬

東京大学理学部情報科学科

1 はじめに

従来のキャッシュメモリを用いたシステムでは参照の局所性を利用して効率的な処理を実現している。しかしながら、使用するデータセットのサイズがキャッシュのサイズに納まり切れない状況においては、キャッシュメモリのみでは性能を出すことは不可能である。このような場合には、使用するデータセットに内在する規則性を利用して必要とされるデータを予めプリフェッチしてくる方法が非常に有効である。お茶の水1号(OCHANOMIZ-1: Omnipotent Concurrency-Handling Architecture with Novel OptiMIZers-1)では巨大な構造体データへの順次アクセスという状況に問題を絞り、大域構造体先行フェッチ機構(GSPF: Global Structure Pre-Fetch)を実装した。本稿ではGSPF機構の実機上での性能評価を行なう。

2 構成

図1にお茶の水1号の全体の構成図を示す。お茶の水1号は高性能マイクロプロセッサ NEC VR4400を要素プロセッサとして使用し、低オーバーヘッドの同期機構や通信機構により効率の良い細粒度並列処理を目指した密結合並列計算機システムのテストベッドである[1]。主要なコンポーネントである外部エージェント、メモリコントローラ、ホストインタフェース、大域同期機構にはXilinx社FPGA(XC4010)を使用しており、柔軟な実装を行なうことが可能になっている。GSPF機構は外部エージェントである2個のFPGAを用いて制御される。図2は本機構の構成図である。GSPF機構の動作を簡単に説明すると次のようになる[2]。GSPF機構は先ずプロセッサから外部エージェントに対して先行フェッチを行なうべきデータセットのベースアドレス、転送ブロック数、ストライドを伝えることで起動される。外部エージェントは与えられた情報を基にプロセッサとは独立にメモリからデータのフェッチし、ローカルバッファ

に格納する作業を指定ブロックサイズ分行なう。プロセッサからデータの要求が来た際、その時点で既にデータがフェッチ済みであった場合は即座にプロセッサにデータを返し、まだデータのフェッチが終わっていない場合はプロセッサをデータの到着まで待たせておく。プロセッサが要求するデータがフェッチ済みであるかどうかの判定にはフェッチしたデータ数を保持しているカウンタを用いて行なう。そのためプロセッサはデータの要求を先頭から順に行なう必要がある。

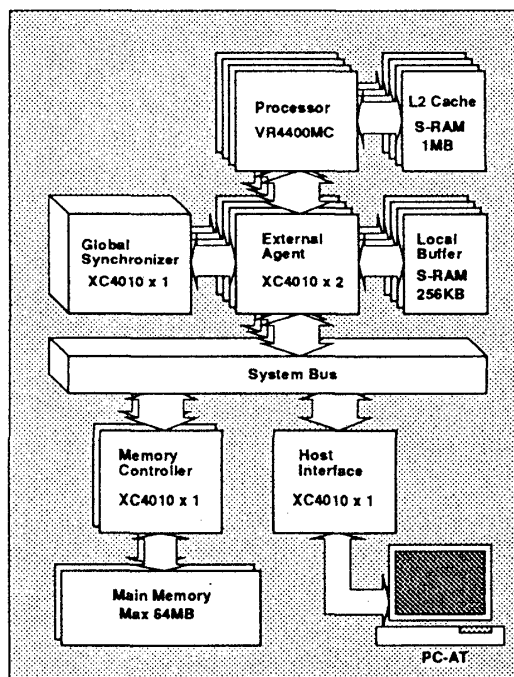


図1: お茶の水1号の構成図

3 性能評価

図3は内積計算を用いてキャッシュメモリのみを使用した場合とGSPFを使用した場合での性能比較を行なったものである。横軸にはプロセッサ台数、縦軸には1台でキャッシュメモリのみを使用した場合を1として正規化した実行時間をとった。各プロセッサ台数についてC,G,GBはユニットストライドで配列にアクセスした場合、4C,4G,4GBはストライド数4でアクセスした場合のそれぞれの性能を示す。Cはキャッシュメモリのみを使用して計算を行なった場合で、GはGSPF

Performance Evaluation of Global Structure Pre-Fetch Mechanism

Kanemitsu OOTSU Takashi MATSUMOTO Kei HIRAKI

Department of Information Science, Faculty of Science, the University of Tokyo

7-3-1 Hongo, Bunkyo-ku, Tokyo, 113, Japan

E-mail: {kim, tm, hiraki}@is.s.u-tokyo.ac.jp

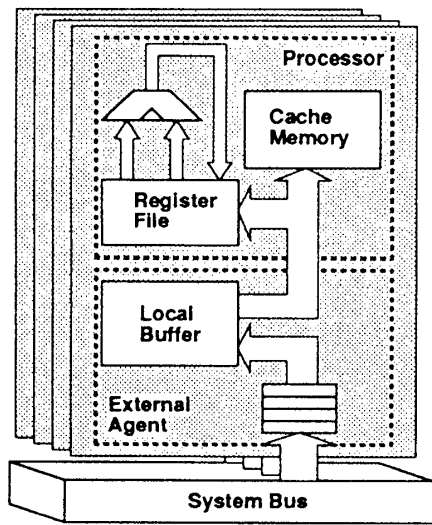


図 2: GSPF 機構構成図

機構を用いてフェッチしたデータをプロセッサに1データずつ受け渡す場合、GBはプロセッサに1ブロックずつ受け渡す場合を示している。

ユニットストライドの場合にはGBが最も性能が良く、キャッシュのみを用いた場合に比較して2割近い性能の向上が見られた。これはプリフェッチによる効果とプロセッサにデータをまとめて1ブロックずつ受け渡す際のペナルティをそのブロック中で実際に使用されるデータの量の多さで補うことで相殺される効果によるものである。ストライド4の場合には4Gが4GBを上回り最も性能が良い。これはプロセッサにデータをまとめて1ブロックずつ受け渡すペナルティをブロック中で使用するデータの量で補うことが出来なかったことによる結果である。この結果により、GSPF機構を使用する際には1ブロック中で使用するデータの量が多い場合はプロセッサへのデータの転送をブロック単位で行ない、逆に少ない場合は1語ずつで行なった方がよいことが分かる。

内積の計算では構造体は入力方向のみなので出力方向の構造体も含んだプログラムとしてSAXPY($a * X[i] + Y[i]$ を計算する)を用いて性能比較を行なった。図4がその結果のグラフである。図3と同様にCはキャッシュメモリのみを用いた場合、GはGSPFのみを用いた場合である。GBはGSPFで採ってきたデータをブロック単位でプロセッサに渡し、プロセッサからのデータもブロック単位で行なった場合であり、GB2はプロセッサからのデータのみ1データずつ受け渡す場合である。

この例でもGSPF機構を用いることで1割以上の性能の向上が見られた。GBとGB2の間に性能の差がほとんど見られないのは、GBの出力データを含むブロックを一度メモリからキャッシュに転送するペナルティとGB2の1データずつしか転送できないペナルティがほぼ同じになったためと推測できる。GBに関してはユニットストライドでアクセスを行なう場合は

メモリからキャッシュへのデータ転送は全くの無駄である。そのような場合はダミーのデータを返してやることで性能の向上が十分に予測できる。

Normalized Exec Time

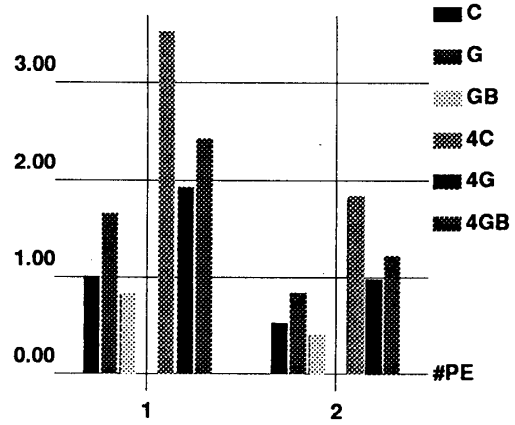


図 3: 内積 計算

Normalized Exec Time

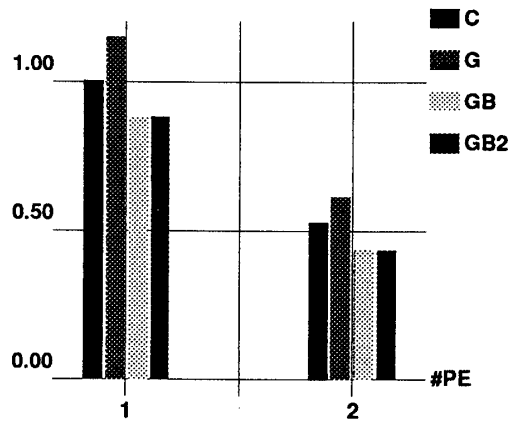


図 4: SAXPY

4 おわりに

細粒度並列計算機お茶の水1号実機上でGSPF機構の性能評価を簡単なアプリケーションを用いて行なった。今後はさらに複雑なアプリケーションを用いた性能評価を行なうと共に、機構の拡張を行なう予定である。

参考文献

- [1] 戸塚 米太郎, 大津 金光, 中里 学, 秋葉 智弘, 松本 尚, 平木 敬, “汎用細粒度並列計算機: お茶の水1号 — 構成と性能評価 —,” 並列処理シンポジウム JSPP'94, pp. 73-80, May 1994.
- [2] 大津 金光, 松本 尚, 平木 敬, “細粒度並列計算機お茶の水1号 — 大域構造体先行フェッチ機構 —,” 第47回情報処理学会全国大会講演論文集(6) ハードウェアシステム, pp. 57-58, Oct. 1993.