

動いているお茶の水1号

6K-6

— 大域同期機構の評価 —

中里 学 松本 尚 平木 敬

東京大学理学部情報科学科

1 はじめに

汎用細粒度並列計算機お茶の水1号 [4] は市販の RISC プロセッサの他に、外部回路として FPGA(Field Programmable Gate Array) を用いて、細粒度並列処理を効率的にサポートするための通信・同期機構を搭載している。大域同期機構(Global Synchronizer)[2] は、システムバスに負担をかけずに、通信・同期を効率的に支援する役割を持つ。大域同期の一つとしてバリア型同期がある。バリア型同期は、同期に参加する全てのプロセッサが、同時に待たされるもので、比較的簡単に構成できる。お茶の水1号上に一般化されたバリア型同期機構である Elastic Barrier[3] を実装して、プログラムを実行させた。本稿では、性能測定の結果について述べ、その有効性を示す。

2 大域同期機構の構成

共有バス結合のマルチプロセッサシステムでは、スヌープキャッシュやメモリ自体を高機能化するなどの試みがされてきたが、いずれも共有バスの飽和が性能低下の原因となっていた。共有バス共有メモリ型マルチプロセッサであるお茶の水1号でも、メモリベースのデータ駆動同期機構や大域構造体先行フェッチ機構により、プロセッサ間の効率的な通信・同期のための手段を提供しているが、共有バスアクセスの増大が性能向上のためのボトルネックになってしまう。このため、お茶の水1号では大域同期機構のための FPGA(Xilinx 社 XC4010:10,000 ゲート相当) を1つ(以下GSと呼ぶ)用意して、共有バスを使わない通信・同期機構を持たせている。通信・同期のためにバスアクセスが頻繁に起こるアプリケーションにおいても、プロセッサが効率良く動作するためには、より効率的な通信・同期機構の存在

が不可欠である。

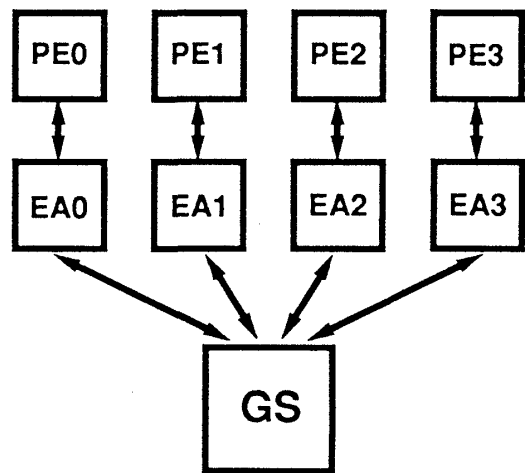


図1: 大域同期機構(GS)の構成図

現在のところ実装されているのは、バリア型同期機構である。バリア型同期機構は、きわめて軽い同期機構であり、ハードウェアコストが少なく済むという特長をもつ。Elastic Barrier は、細粒度タスクの静的スケジューリングと組み合わせて用いられ、バリア型同期機構の中で最も効率が良いものである。図1に示すように、プロセッサとは直接信号のやりとりをできない構成になっているため、GS は各プロセッサの横に接続している外部回路である External Agent(EA) と通信する。Elastic Barrier は3種類の同期情報を使用し、同期の多重化を許すので、それらをカウンタで管理する方式である。具体的には、3つの情報に対し、プロセッサは以下の動作をとる。

- APRV(同期の承認) 特定のアドレスへのストア。プロセッサはブロックされない。
- PREQ(同期の予告) 特定のアドレスへのストア。プロセッサはブロックされない。
- RREQ(同期の要求) 特定のアドレスへのロード。同期が成立してないと成立するまでブロックされる。

OCHANOMIZ-1: Performance Evaluation of Global Synchronization Mechanism
 Gaku NAKAZATO 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: {duke, tm, hiraki}@is.s.u-tokyo.ac.jp

EA と GS ではプロセッサの出力する命令に対し、同期成立の検出とカウンタによるタイミング制御を協調して行う。より速く同期成立を検出し、より早くプロセッサを動かして、オーバーヘッドを削減するために、EA 内部にカウンタを設けてタイミング制御を行っている。GS の内部では同期条件の成立判定を行う。実際に、プロセッサがブロックされるのは、同期成立を要求する RREQ の時だけであるが、この場合も、すでに GS が同期条件成立を検出していれば、プロセッサは通常の LOAD 命令と同じ遅延のみで実行を続けられる。

3 Elastic Barrier の性能評価

Elastic Barrier をアプリケーションプログラムに対して使用し、実行時間を測定した結果が表1である。アプリケーションを gcc コンパイラを用いてコンパイルし、細粒度並列計算機最適化コンパイラ: OP.1[1] の出力コードを付加して、実行コードを生成した。OP.1 は、Elastic Barrier 用の同期コード (RREQ, PREQ, APRV) を自動的に生成する最適化コンパイラである。Elastic Barrier の有効性を比較するために、共有変数を使用するソフトウェアでバリアを行った場合の実行時間を測定した。ソフトウェアによるバリアは、共有フラグと各プロセッサが持つフラグとで同期を成立させる方法である。プロセッサは共有フラグを取ってきて、自分のフラグと比較して1増やし(または減らし)、同期に参加するプロセッサ数と一致するまでスピンして待つものである。プログラムには、Elastic Barrier 用コードの中の APRV と RREQ を上記のフラグによる同期条件の成立コードに置き換えたものを使用した。Σプログラム(表1中: Σ) は、10,000 までの和を求めるプログラムである。

	Elastic Barrier	ソフトウェアバリア
Σ	0.024	0.083

表 1: バリアの性能比較 (単位:sec)

この例から、ハードウェアを用いた Elastic Barrier はソフトウェアによるバリア同期に対して数倍の性能向上が可能であることが分かった。ソフトウェアによるメモリバリアでは、同期地点に到着する度に共有フラグの値を取りに行き、値を変えた後もスピンして待つために、バスアクセスが増大し、Elastic Barrier に比べてかなり性能が悪くなっている。Elastic Barrier の方は、同期を真に要求した時 (RREQ) に、同期が成立していないと初めてプロセッサはブロックされるため、プロセッサが同期待ちをする時間がきわめて軽減される。

4 おわりに

お茶の水1号上に Elastic Barrier を実装し、性能を測定した。単純なバリア同期方式との比較により、オーバーヘッドのきわめて少ない効率的な同期機構であることを示した。今後は、アプリケーションの種類を増やし、他のバリア型同期機構との比較を行っていく予定である。さらに、もう一つの通信・同期機構である共有レジスタも実現する予定である。

謝辞

お茶の水1号の作成に当たり、チップを供給して頂いた日本電気株式会社に感謝致します。

参考文献

- [1] 稲垣 達氏, 松本 尚, 平木 敬, “システムの階層的並列性を統一的に扱う最適化コンパイラ,” 信学技報 CPSY-94-40, pp. 105-112, July 1994.
- [2] 中里 学, 松本 尚, 平木 敬, “細粒度並列計算機お茶の水1号 - 大域同期機構 -,” 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. 61-62, Oct. 1993.
- [3] 松本 尚, “Elastic Barrier: 一般化されたバリア型同期機構,” 情報処理学会論文誌, vol. 32, no. 7, pp. 886-896, July 1991.
- [4] 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.