クラスタ型並列計算機のための並列化コンパイラ

4X－10

児島 彰　　土江 竜雄　　弘中 哲夫　　藤野 清次
広島市立大学 情報科学部 情報工学科

1　はじめに

並列計算機上で並列処理を行うには処理を適当な大きさの粒度に並列化し、それぞれの場面で適当な同期処理を行う必要がある。いろいろな大きさの粒度を含む処理を行うには、クラスタ型並列計算機が適している。本稿では、現在、構築中のクラスタ型並列計算機について述べ、この計算機ための並列化コンパイラについて述べる。この並列計算機では、ハードウェアによるバリア同期とクラスタ構造によるメモリの階層性を有する。同期コードの挿入方法、クラスタ内／外のメモリ割り付け、疎結合のクラスタ間の通信の方法について述べる。

2　クラスタ型並列計算機

今回ターゲットとしている開発中の並列計算機は、図１のように全体としてはクラスタ構造を取り、クラスタ内は4～16CPU の要素プロセッサと共有メモリを持つ。それぞれの要素プロセッサは共有メモリに対するキャッシュを持つ。細かい粒度の並列処理は自動並列化コンパイラでクラスタ内に割り付ける。また、クラスタ間は高速シリアルリンクで結合し、クラスタ間のデータの受渡しはパケット単位で通信プロセッサを介して行う。クラスタ単位へは比較的大きな粒度の並列化を割り付けることにした。

要素プロセッサは、今回新らたに設計した。命令フォーマットを図２に示す。一般的な RISC の命令アーキテクチャに、同期制御命令やキャッシュ制御命令を付加している。特徴的なのはレジスタ間演算命令など出現頻度の高い命令 (Type0) の一部分に、これらの命令がコプロセッサ制御命令として埋め込まれていて、通常の命令の実行と並行して実行可能な点である。これ

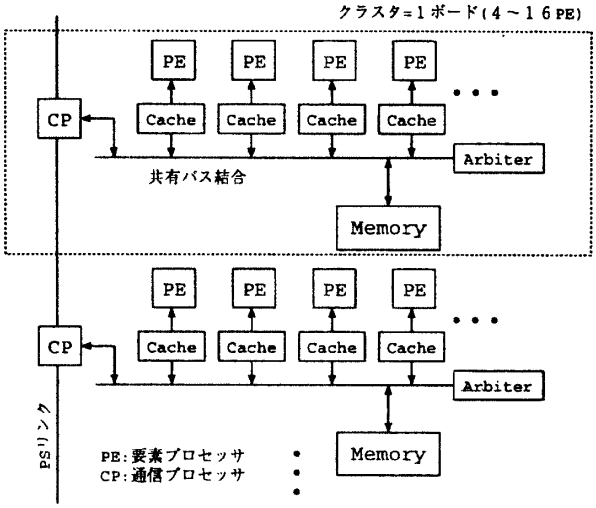

図 1: 全体図

により細粒度の並列化のための同期制御、キャッシュ制御を実現している。

3　バリア同期

クラスタ内には、細かい粒度の並列処理を割り付けるので、極めて密な同期を行う必要がある。共有メモリ上でのプログラムによる同期処理では、同期操作自体の処理時間が大きくなり、細粒度の並列処理では問題を生じる。Doacross などの処理でも、極めて密な同期を行う必要があるが [1]、最近の高速なプロセッサでは相対的に同期操作自体のコストが大きくなり、全体の速度に大きく影響を与える [5]。そこで、ハードウェアによるバリア同期を用意することにした。これまでにバリア同期には様々な手法が提案されているが [2][3]、松本の提案する Elastic Barrier[4] を基本に、一部改変したバリア同期機構を採用した。今回採用したのバリア同期機構を図３に示す。

4　並列化コンパイラ

開発中の並列計算機のため並列化コンパイラについて述べる。並列化コンパイラは以下のように構築する。

Parallelizing Compiler for Cluster Parallel Computer TK98
KOJIMA Akira, Tsuchie Tatsuo, HIRONAKA Tetsuo, FUJINO Seiji,
Department of Computer Engineering,
Faculty of Information Sciences, Hiroshima City University,
3-4-1, Ozuka-Higashi, Asaminami-ku,
Hiroshima City,731-31,Japan

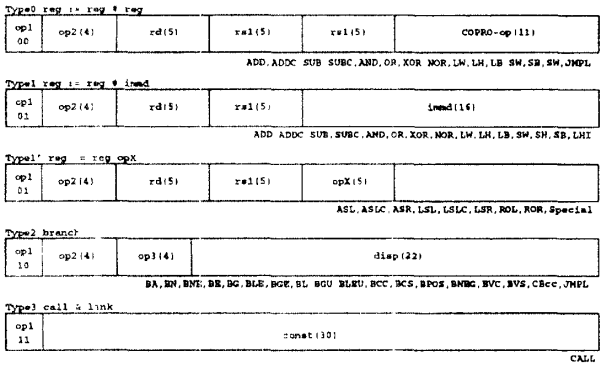

図 2: 命令フォーマット

- 当初はC言語をターゲットとするが、パーザ部分の置き換えで他の言語への対応も可能とする。
- 通常の逐次コンパイラとライブラリ関数による並列プログラミング環境を用意し、並列化パーツによる出力コードをこれに合わせることで並列化コンパイラとする。
- クラスタ間に割り付ける粒度の大きいタスクの分割は、関数単位でコンパイラ指示行により行う。
- クラスタ内の並列化は関数内の各部に対して基本的に自動で行う。
- クラスタ間に割り付けるデータの分割は、コンパイラ指示行により行い、クラスタ内でのデータ割り付けは、コンパイラが自動的に行う。
- コンパイラ指示行で指定されたクラスタ間のデータへの操作は、コンパイラが自動的にクラスタ間通信を挿入する。
- 最終段のピープホールオプティマイザに、キャッシュ制御、同期制御の命令をコンパクションする機能を付加する。

粒度の小さいクラスタ内の処理やデータの分割はコンパイラが自動で行うが、粒度の大きいクラスタ間への処理やデータの分割はユーザが行い、そのためのデータ転送処理などの細かい処理はコンパイラが行う。クラスタ間のデータ転送速度は、比較的大きいものになると高速であるが、細かいデータ転送は、オーバヘッドが大きいので、この点に考慮したプログラミングが必要である。

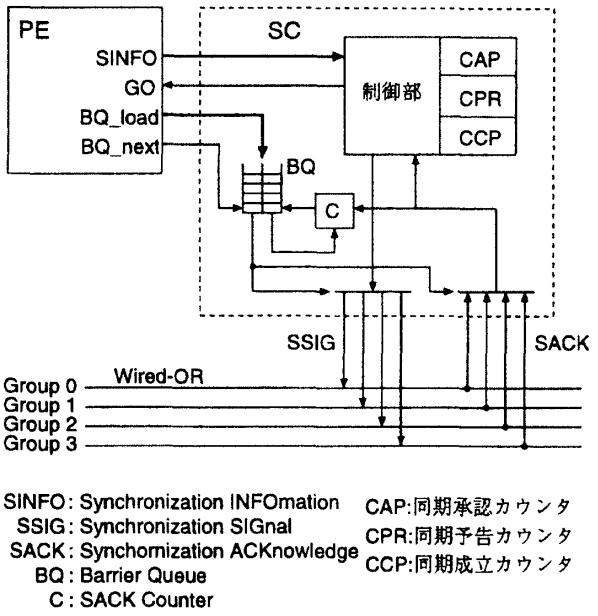

図 3: 同期ブロック

5 おわりに

開発中のクラスタ型並列計算機とこの計算機のための並列化コンパイラについて述べた。この並列計算機では、バリア同期や同期制御、キャッシュ制御の専用命令を有する。現在、要素プロセッサと同期制御部、キャッシュ制御部を一つにまとめたLSIを設計中で、Verilogによる記述、シミュレーションを行っている。

参考文献

[1] Ron Cytron: "Doacross: Beyond Vectorization for Multiprocessors (Extended Abstract)", Proc. of Int. Conf. on Parallel Processing pp.836-844 (Aug. 1986)

[2] Arenstorf,N.S and Jordan,H.F: "Comparing Barrier Algolisms", Parallel Computing, Vol.12 No.2 pp.157-170 North-Holland (Nov. 1989)

[3] Gupita.R: "The Fuzzy Barrier:A Mechanism for High Speed Synchronization of Processors", Proc. Third Int. Conf. on ASPLOS pp.54-63 (Apr. 1989)

[4] 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)

[5] 児島 彰, 藤野 清次, 弘中 哲夫, 高山 毅: "Doacross ループの実行時間と最適プロセッサ数", 情報処理学会 第53回全国大会 講演論文集 (1) pp.321-322 (Sep. 1996)