「光バスクラスタ計算機Euphoria」における同期機構の実現

2B－5

下山朋彦　濱口一正　福井俊之　柴山茂樹
キヤノン(株) 情報メディア研究所

1　はじめに

従来のワークステーション規模のノードを、光波長多重回線を用いて相互に接続し、ノード間でのメモリの共有を可能とした「光バスクラスタ計算機」のプロトタイプ“Euphoria”の開発・試作を行った[1][2]。本報告ではEuphoriaの同期機構について述べる。

2　Euphoriaのメモリシステム

Euphoriaは外部ノードのメモリを直接参照できるハードウェアによる分散共有メモリを搭載しており、光バス(光波長多重回線)を通じてシステム全体がアドレス空間を共有するNUMA型のメモリアーキテクチャを採用している。

メモリアクセスコスト低減のためキャッシュシステムを持ち、特に参照コストが高い外部ノードメモリのアクセス時間低減を図っている。複数のプロセッサ(ノード)がキャッシングを行うためキャッシュの一貫性保持機構が必要となるが、これはメインプロセッサであるPowerPC601のスヌープキャッシュ機構と、外付けのディレクトリ方式のキャッシュ一貫性保持機構を組み合わせることで対処している。

3　ノード内での同期

各ノード内でのプロセッサ間同期は、PowerPCのlwarx(load word and reserve indexed), stwcx.(store word conditional indexed)命令により行う。これらは特殊なread, write命令であり、次の動作を行う。

- lwarx命令発行：read と共にPowerPC内部の予約フラグをセットする。
- バススヌープ：lwarx したアドレスに対するwrite(キャッシュの無効化を含む)を検知した場合は予約フラグをリセットする。
- stwcx. 命令発行：予約フラグがセットされている場合はwriteを発行しstwcx. 成功フラグをセットする。リセットされている場合はwriteを発行せずstwcx. 成功フラグをリセットする。

例えばlwarx, stwcx. 命令によりTest&Setの同期操作を行う場合は、次のようにプログラムを書く。

```
LOOP:   lwarx (lock_var), r0
        stwcx. r1, (lock_var)
        if (stwcx fail) then LOOP
```

lwarxでreadした後、stwcx. でwriteする。readとwriteの間に他のプロセッサによりwriteが行われた場合には(stwcx. に失敗した場合には)lwarxからやり直す。つまりreadしたアドレスの内容が変更されていないことを確認してからwriteを行うことにより同期動作を保証する。

4　ノード間の同期

Euphoriaではノード間の同期も、分散共有メモリ上でのlwarx, stwcx. 命令を用いる。これは次のような理由からである。

- ノード内とノード間の同期を同じ方法で行いたい
- バスをロックする同期命令では光バス全体にロックがかかるため非効率である

だがlwarx, stwcx. 命令による同期はバストランザクションがスヌープできることを前提としており、バスを直接スヌープできないノード間での同期にはそのための機構の付加が必要となる。

Euphoriaではプロセッサにスヌープさせることが必要なバストランザクション(writeのバストランザクション)を、必要なノードにマルチキャストする機構を設けることによりlwarx, stwcx. 命令による同期を実現する。

各ノードのメモリはキャッシュブロック毎にnビットのタグ(n:ノード数)を持つ。このタグは各々のキャッシュブロックに対するwriteをどのノードにマルチキャストするかを記録する。writeのマルチキャストは、次の手順により行う。

Synchronization Mechanism of an Optical Bus Cluster Computer “Euphoria”
T. Shimoyama, K. Hamaguchi, T. Fukui and S. Shibayama
Media Technology Laboratory, Canon Inc.

(1) lwarx を行ったノードの記録：内部バスを監視し自ノードのメモリに対する lwarx 命令による read を検出する。検出後 read を実行したアドレス (キャッシュブロック) のタグに、lwarx 命令を実行したノードを記録する。

(2) write の検出：内部バスを監視し自ノードのメモリに対する write を検出する。検出後 write されたアドレスのタグを参照し、記録されている (lwarx 命令を行った) ノードにその write をマルチキャストする要求をネットワーク中央の光バスアービタに出す。

(3) write のマルチキャスト：要求を受けた光バスアービタは必要なノードに write をマルチキャストする。他のノードのプロセッサはそれをスヌープし、内部の予約フラグがセットされている場合はフラグをリセットする。

複数ノードで同時に同期命令が実行された場合、(3) で光バスアービタにより直列化されるので矛盾は生じない。

キャッシュ ON 時には、ディレクトリ方式のキャッシュの一貫性保持機構によりこれらの動作を実現する。このとき両者は次の対応を持つ。

	同期機構	キャッシュ
記録を行う場所	メモリのタグ	ディレクトリ
記録が必要な access	lwarx による read	全ての read
伝達が必要な access	全ての write	全ての write

lwarx 命令で read したアドレスは、そのノードにキャッシングされたものとして一貫性保持機構によりディレクトリに登録される ((1) に対応)。write が行われると、一貫性保持機構はディレクトリに登録されたノードに対して write をマルチキャストするように光バスアービタに要求を出す ((2) に対応)。光バスアービタは指示されたノードにマルチキャストし、そのキャッシュブロックの無効化を図り、同時にプロセッサ内部の予約フラグをリセットする ((3) に対応)。この様子を図1に示す。

キャッシュ OFF 時には、lwarx 命令と write についてのみ一貫性保持機構を動作させることで write のマルチキャストを行う。一貫性保持機構は lwarx 命令が実行された時にそのノードをディレクトリに記録する (キャッシュ ON 時とは異なり、lwarx 命令による read についてのみ記録する)。write が行われた時にディレクトリに記録されたノードにその write をマルチキャストする。

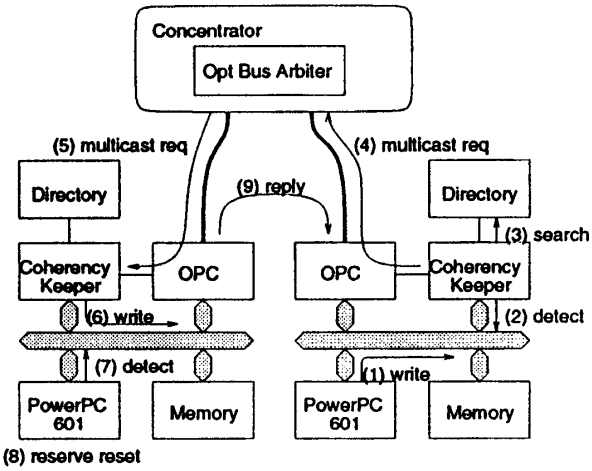

図 1: write のマルチキャスト

5 異なる CPU での同期

Euphoria は PowerPC をメイン CPU とし、MC68030 を I/O プロセッサとしたヘテロなマルチプロセッサシステムである。

PowerPC と MC68030 との間でも同期が必要となるが、MC68030 命令セット中に lwarx,stwcx. 命令はない。そこで I/O プロセッサ側に、lwarx,stwcx. 命令をエミュレートする回路を備えることでこれに対応した。

6 今後の課題

現在 Euphoria はノードが単体動作している段階である。メイン CPU と I/O プロセッサとの同期は I/O プロセッサの同期エミュレータを使用して行っており、これまでのところ良好な結果を得ている。今後、ノード間を接続する光バスを実装しノード間での同期の検証や、アプリケーションの同期パターンの調査を行う予定である。

参考文献

[1] 福井他 “光バスクラスタ計算機 Euphoria の開発 (1) 概要” 情報処理学会第 49 回全国大会 (1994).

[2] 下山他 “光バスクラスタ計算機 Euphoria の開発 (2) メモリアクセス機構” 情報処理学会第 49 回全国大会 (1994).