

1 H-5 密結合並列計算機における『過去共有データ空間』とそのアドレッシングモードによる実現

星野浩志 北野博 鈴木敏浩 富澤眞樹 五十嵐智 阿刀田央一 斎藤延男
東京農工大学 工学部 電子情報工学科

1. はじめに

共有アドレス空間を持つ並列計算機ではデータ領域を、共有データと私的データに分けて考えられてきた。しかし共有と私的の間にある第3の種類(過去共有データ)があることには、あまり注意が払われていない。

たとえば盤を使うゲームの次の手を、子プロセスを使って並列に探索する場合を考える。最初の盤の状態は、すべての手で共有である。しかし次の手が盤に反映されたとき、それぞれの手で盤の状態は私的になってほしい(図1)。このようなデータを実現するには、結局データを各子プロセスの私的領域にコピーする方法をとらなければならない。並列計算機ではこのコピーはバスにかなりの負担をかける。

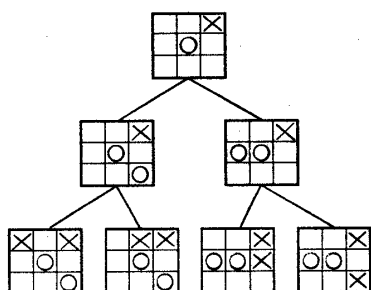


図1 ゲーム盤と探索木

このようなデータは、バックトラックを含む深さ優先探索を、幅優先探索になおして並列計算機で実行しようとするときに生じるものである。有名な例は、論理型言語の処理をOR並列化するとき多重重束縛される変数で、Prolog専用機での解決案が示されている[1]。

本論文では、このような種類のデータおよびその領域を、『過去共有データ』『過去共有領域』等と呼び、共有アドレス型並列計算機一般の問題として考える。そして、プロセッサエレメントに特定のアドレッシングモードを持たせることによって、過去共有データを効率よく扱う方法を提案する。また、過去共有データとキャッシングとの関係についても論じる。

2. 過去共有データの定義

『過去共有』は、呼出し型の密結合並列計算機を土台にして考えている。われわれはすでに、呼出しにより並列プロセスを動的に生成する機構を命令で用意した密結合並列計算機を提案し、実験、検証した[2]。

プロセスの呼出し関係を『親→子』と考え、それぞれの環境を『過去』『現在』と呼ぶことにする。『過去共有デ

ータ』は、一つの『過去』を共有している複数の『現在』である。直感的には一つの『過去』から複数の『現在』が派生して、独立に進行していくと考えるとわかりやすい。『過去』を基準にすると、『未来』が多重に発散していくと考えることもできる(図2)。

過去共有データの外見上のふるまいは、次のように定義することができる。

- (1) 現在データの初期値は、過去データの値である。
- (2) 現在データを変更しても、過去データに影響しない。
- (3) 複数の現在データは互いに独立である。

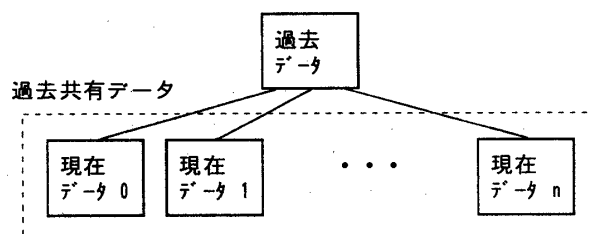


図2 過去データと現在データ

これは値渡し(引数)の引き数とまったく同じ性質である。たしかに過去共有データを、値渡しのコピーによって実現することは可能である。しかし大量データの値渡しは、並列計算機にとって致命的な負荷になることがある。これは逐次計算でも同じで、大きな配列等の値渡しはあまり使われない。

値渡しは親では引き数の多重化をせず、なおかつ子ではいつの間にか多重化されているという便利なものである。大量データでこれを効率よく実現できれば、負荷がおもくて採用されなかったアルゴリズムが使われる可能性もある。

3. 過去共有の実現

3.1 方針

データ全部をコピーするのが大変なので、まえもって現在データの領域だけ確保しておき、参照要求時に必要な部分をコピーする方式を考える。この方式では、参照されない部分のコピーをしなくてよい分負荷が軽くなる。大量の値渡しデータの一部分だけが参照され、しかもどれが参照されるか不確定な場合に有効である。

現在データでまだ参照されていない部分の値は不定であるが、参照と同時に過去データの値のコピーで確定されるので、論理的には確定している。過去共有データを多段に用いる場合が生じるが、値が確定である過去データまで逆戻りしてコピーする(図3)。現在データに対する書込は、現在データだけで閉じて確定する。

以上のようにして(1)~(3)の性質を満たすデータを実現できる。この技法を『データ継承』と呼ぶことにする。データ継承は、過去共有を実現するための一技法である。

ただし、この方式には制限がある。子プロセスを起動した後、親プロセスが自分のデータを変更すると、(1)の条

Past Sharing Data on Tightly Coupled Multiprocessor and Its Implementation with Addressing Mode
Hiroshi HOSHINO, Hiroshi KITANO, Toshihiro SUZUKI
Masaki TOMISAWA, Satoshi IGARASHI, Oichi ATODA
Nobuo SAITO
Tokyo University of Agriculture and Technology

件を満たさない場合が生じる。これは子プロセスが過去データを参照するために起きる問題である。また継承してきたデータを、過去のどの深さまで確定するか等、選択の余地がある。

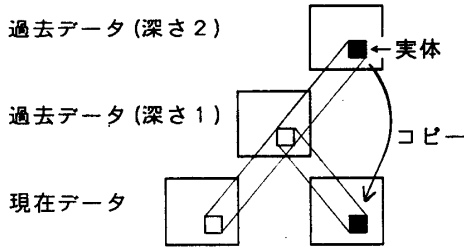


図3 データ継承による参照

3.2 アドレッシングモードによる実現

データ継承の処理はソフトウェアで行うことができ、このためのライブラリを提供することも可能である。しかし過去共有データの参照は、速いのが望ましい。また本来これはデータ参照の方式の一種であるので、アドレッシングモードとしてハードウェアが用意すべきものであろう。

アドレッシングモードでデータ継承の動作を実現した場合、ソフトウェアで単に過去共有領域を参照するだけで、ハードウェアによって値の確定などの処理がなされる。このような処理するためのハードウェアを『データ継承ユニット (DIU)』と呼ぶことにする。

3.3 過去共有領域の実現例

著者らの研究室では現在、過去共有の管理構造とデータ継承ユニットの試験機を作成中である。

[1] 管理構造

ある領域を過去共有領域として使うためには、そのための管理構造をメモリ中に用意する必要がある。一つの過去共有領域を管理するために、図4に示す三つが必要である。

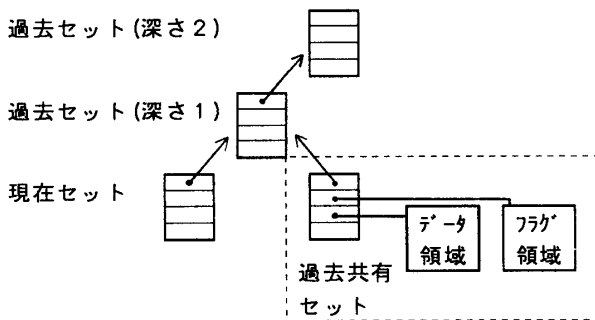


図4 過去共有領域の管理構造

データ領域は、過去共有データの領域である。フラグ領域は、このデータ1バイト毎に用意したフラグの領域である。フラグは valid, dirty の2ビットから成る。過去共有セットは、これら二つの領域のエントリである (表1)。

表1 過去共有セットの構成

要素	意味
Ancestor	過去データのセットへのポインタ
Fptr	フラグ領域へのポインタ
Dptr	データ領域へのポインタ
Dsize	データ領域のサイズ

[2] データ継承ユニット

データ継承ユニットは、各 PE 内のプロセッサのバスに直接つながっている。バスに過去共有領域の参照・書き込みが現れたときに、そのバスサイクル内でデータ継承処理をしてしまう。本来プロセッサの一部として、取り込むべきものである。

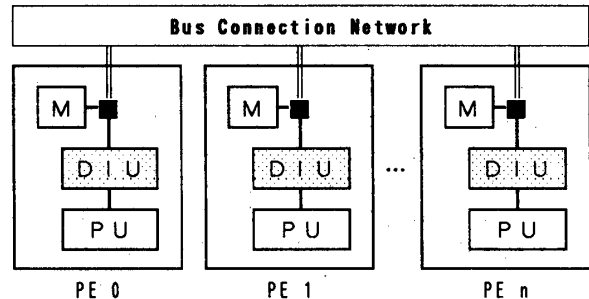


図5 データ継承ユニットの位置

データ継承ユニットは、過去共有セットと同じ4本構成のレジスタセットを持つ。このレジスタに過去共有セットが登録されていれば、その過去共有領域はアクティブになり、アドレスバスでバウンダリチェックをする。チェックにかかった参照は、そのバスサイクル内でデータ継承操作をして、データバスに確定した値を返す。

レジスタの操作のために、過去共有セットをレジスタセットに登録する命令を用意する。またこのレジスタは一般のレジスタと同様に、プロセススイッチの管理対象になる。

4. キャッシングとの関係

データ参照時に実体をコピーしてもってくるという方法は、キャッシングの技法と同じである。すなわちデータ継承は、過去データをプロセッサに近いところの現在領域にキャッシングする技法に他ならない。書き込みに対するローカル性を保っているため、一貫性管理の必要もなく、キャッシングに適したデータである。

過去共有データでは、現在を過去に書き戻すことまでは定義していないが、必要ならばキャッシュのコピーバック技法を使ってこれを実現することも考えられる。

5. おわりに

過去共有データとそのアドレッシングモードによる実現について述べた。今後の課題は、実行評価と過去共有データに適したアルゴリズムの調査、および著者らの研究室で作成した並列記述言語C// [3]でのサポートである。

参考文献

[1] A. Ciepielewski, S. Haridi : Formal Method for OR-Parallel Execution of Logic Programs, IFIPS, 1983.
 [2] 富澤ほか: 手続きフロー型並列計算機における分散型組込み制御機構, 信学論 (D), J71-D, 10, pp.1921-1930 (1988).
 [3] 富澤ほか: 手続きフロー型並列計算機の制御機構と並列記述言語, 情報処理学会計算機アーキテクチャ研究会資料, 79-15 (1989).