C-005

スレッドレベル並列投機実行のためのデータ値予測機構
A Value Prediction Mechanism for a Thread-Level Speculation on a Multiprocessor

小路 勇気† 布目 淳‡ 平田 博章‡ 柴山 潔‡
Yuki Shoji Atsushi Nunome Hiroaki Hirata Kiyoshi Shibayama

1. はじめに

マルチコアプロセッサが商用のマイクロプロセッサとして市販される一方で, プログラムからスレッドレベルの並列性を抽出するのは困難な現状にある. そこで, 我々は, プログラムから粒度の粗い並列性を抽出し, スレッドとして投機的に並列実行を行うマルチプロセッサシステムを開発中である. スレッドレベル並列性を抽出するためには, スレッド間の依存関係によるハザードを回避しなければならない. 本稿では, メモリ上のデータのアクセスに対して動的に依存関係の解析を行い, 特にフロー依存によるハザードを last value 方式[1]と stride value 方式[2]のデータ値予測によって回避するための機構を提案する.

2. データ値予測機構

2.1 システム概要

我々は, 投機実行中に更新した値を 1 次キャッシュに格納し, また, スレッド間の逆依存と出力依存のデータ依存関係に起因するハザードを除去する**メモリリネーミング機構**[3]を既に提案している. 本稿では, さらにフロー依存によるハザードを回避するためのデータ値予測機構を提案する. 本機構を用いるマルチプロセッサシステムの構成を図 1 に示す. 共有バスで接続した各プロセッサ上でスレッドを実行する. プログラム中の並列化対象のループに対して, その各イタレーションをスレッドとして実行する. $i-1$ 番目までのイタレーションの実行が完了しているものとすると, i 番目のイタレーションを実行するスレッドを**確定スレッド**と呼び, これと並列に, $i+1$ 番目以降のイタレーションを投機的に実行するスレッドを**投機スレッド**と呼ぶことにする. また, リング接続の制御線（依存解析用制御線）を用いることによって, 投機データ間のバージョンの前後関係を管理・制御する.

各 1 次キャッシュには, 値履歴テーブル**(VHT; Value History Table)**を併設する. VHT の各エントリには, データのアドレス（タグ）, 予測した値, 差分値のほかに, 予測の有効性や予測を行ったか否かを示す情報を格納する. 2.1.1 で述べる 1 次キャッシュの状態ビットによって, データ値予測を行うか否かを判断する.

なお, 本稿では, 簡単のため, 各スレッドでは 1 個のメモリロケーションに対して 2 回以上の書き込みは行わないものと仮定する.

2.1.1 キャッシュのアクセス状態ビット

メモリ上のデータアクセスに対するスレッド間の依存解析を行うために, 1 次キャッシュの各ラインごとに以下のアクセス状態ビットを設ける.

†京都工芸繊維大学大学院工芸科学研究科情報工学専攻
‡京都工芸繊維大学情報工学・人間科学系
Kyoto Institute of Technology

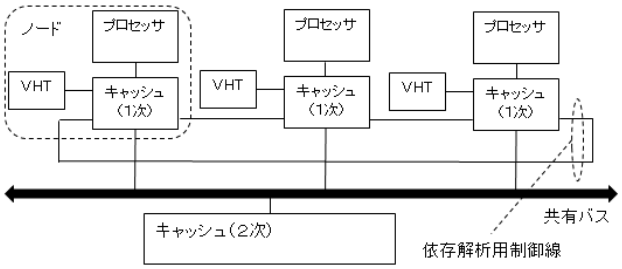

図 1 システム構成

- **W(Written)ビット**：ライン内の各バイト領域に対応付けて設け, そのバイト領域に書き込みを行ったことを示す.
- **FR(First Read)ビット**：ライン中の少なくとも 1 個のメモリロケーションに対して, そのスレッドにおける最初のアクセスが読み出しであったことを示す.
- **O(Obsolete)ビット**：ライン中のまだ書き込みを行っていないメモリ領域に対して, 先行スレッドが書き込みを行ったことを示す.

プロセッサが各自の 1 次キャッシュに書き込みを行うとき, 対応する W ビットをセットする. また, W ビットがセットされていない領域から読み出す場合, FR ビットをセットする.

一方, 先行スレッドを実行しているプロセッサが書き込みを行ったことをバススヌーピングによって検出すると, ハザードの検出処理を行う. FR ビットがセットされておらず, かつ, その書き込みアドレスに対応する W ビットがセットされていなければ, O ビットをセットする.

2.1.2 データ値予測方式

投機スレッドの読み出し要求に対して FR ビットをセットするとき, データ値予測を行う. メモリアドレスをもとに, ダイレクトマッピング方式によって VHT のエントリにアクセスする. 予測元のデータをどのキャッシュからロードしてくるかによって, データの予測値を求める方法が異なる. $n(n \geq 1)$ 個前の先行スレッドのキャッシュから元データを読み出す場合のその元データ値をA, VHT 内の差分をD, としたとき, 予測値Pを以下のように求める.

・W ビットがセットされているとき, $P = A + (n-1) \cdot D$
・W ビットがセットされていないとき, $P = A + n \cdot D$

2.1.3 値履歴テーブル（VHT）の管理

あるノードでの書き込みは, 共有バスを介して他のノードのキャッシュにブロードキャストする. 同じアドレスのデータを保持するそれらのキャッシュでは, VHT 内の情報を参照して, 既に予測を行っていた場合には予測値の検証を行う. 書き込まれる値が VHT 内に記録した予測値と異な

Copyright © 2015 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.

っていれば，スレッドをアボートする．同時に複数の投機スレッドがアボートするとき，その中で最も先行するスレッドのノードのキャッシュが，新たな差分値を計算する．書き込まれる値をM，VHT に登録していた古い差分値と予測値をそれぞれD_{old}，Pとすると，新たな差分値D_{new}は$D_{new} = M - (P - D_{old})$で求まる．この値で自ノードの VHT の差分値を更新するとともに，その情報を共有バスを介して他ノードの VHT にブロードキャストする．

データ値予測が何度も外れるメモリロケーションに対しては，予測を行わないよう指示する情報を VHT の状態に登録することで対応する．

3. 調査

3.1 調査環境

スレッド間の依存関係についてシミュレーションによる調査を行い，メモリデータに対するデータ値予測の有効性を見積もる．PowerPC[4]の命令セットを対象として，マシン命令レベルでプログラム実行のシミュレーションを行う．テストプログラムには SPEC2006 ベンチマーク[3]の中の 5 個のプログラム（入力データセットは test）を用いる．それぞれの調査対象プログラムの主要部を構成する最外ループを並列化の対象とし，今回はそのループについて，最初の 100 個のイタレーション（ループの繰り返し回数が 100 に満たない場合はすべてのイタレーション）を対象に調査する．

3.2 調査結果

各スレッドがアクセスするメモリロケーションを，(a) 逆依存または出力依存によるハザードが生じうるメモリロケーション，(b) フロー依存によるハザードが生じうるメモリロケーション，(c) 依存関係によるハザードが生じないメモリロケーション，に分類して，それぞれの割合を調べた結果を図 2 に示す．ただし，並列実行を開始した後にスタックフレーム領域に割り当てられるメモリロケーションに関しては，計測しないものとする．図 2 より，逆依存/出力依存によるハザードが多く生じることがわかり，メモリリネーミングによってそれらを取り除くことにより，並列性抽出の機会が劇的に増加するものと期待できる．

また，フロー依存によるハザードが生じうるメモリロケーションのうち，(1) last value 予測でハザードを回避できるものの割合，(2) stride value 予測でハザードを回避できるものの割合，(3) データ値予測では回避できないものの割合，を調べた結果を図 3 に示す．データ値予測によって，433.milc ではフロー依存によるハザードをすべて回避できることがわかる．453.povray では大幅に，また，450.soplex と 482.sphin3 では半数近くのハザードを回避できる．しかし，444.namd では，データ値予測を行ってもほとんど効果がない．以上より，すべてのアプリケーションプログラムに対してではないものの，データ値予測は非常に有効であると言える．ただし，データ値予測を行うにあたって多数のメモリロケーションの情報を保存する必要があり（482.sphinx3 で少なくとも 2329 個），本方式の実装において，VHT の容量などのハードウェアコストについて検討する必要がある．

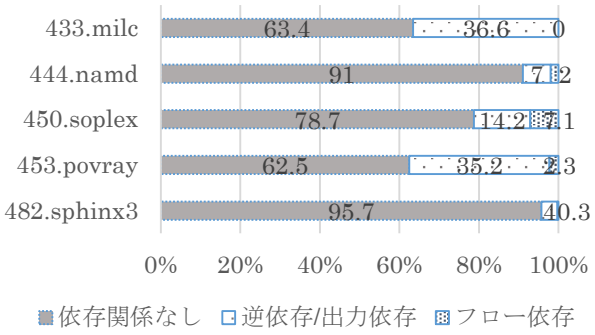

図 2　依存関係によるハザードの発生割合

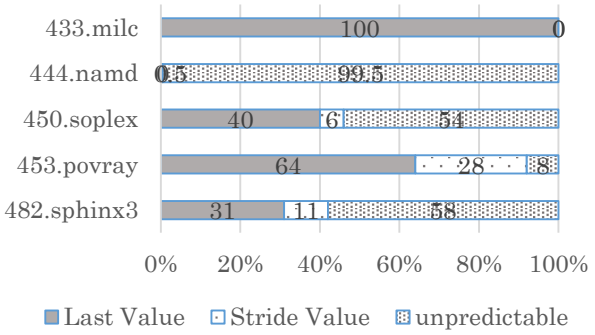

図 3　値予測によるハザードの回避割合

4. むすび

本稿では，並列投機実行スレッド実行のためのデータ値予測機構の概要について述べた．今後は，1 個のメモリロケーションに対してスレッド内で複数回の書き込みを行う場合や，データ値予測を行うべきか否かをメモリロケーションごとに判別する方式も含めて，データ値予測機構の詳細設計を行う予定である．

謝辞

本研究の一部は日本学術振興会科学研究費補助金（基盤研究（C）25330058, 15K00169）の補助による．また，本稿での調査では，IBM Power System S812L を使用した．

参考文献

[1] M. H. Lipasti, C. B. Wilkerson, and J. P. Shen, “Value Locality and Load Value Prediction, ” Proceeding of the 7th international conference on Architectural Support for Programming Language and Operating Systems, pp. 138-147(1996)

[2] K. Wang, and M. Franklin, “Highly Accurate Date Value Prediction Using Hybrid Predictors, ” Proceeding of the 30th international Symposium on Microarchitecture, pp. 281-290(1997)

[3] 平田 博章，藤井 崇弘，藤 皓平，森田 清隆，布目 淳，柴山 潔，“スレッドレベル並列投機実行のためのメモリリネーミング機構”，第 11 回情報科学技術フォーラム論文集，Vol. 1, pp. 31-34 (2012).

[4] Freescale Semiconductor, “Programming Environments Manual for 32-Bit Implementations of the PowerPC Architecture” (2001)

Copyright © 2015 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.