

動的最適化のためのCPUマイニングの試み

横田 隆史 須賀 弘一 大津 金光 馬場 敬信[†]

宇都宮大学工学部情報工学科

1 はじめに

プログラムの実行を高速化するためのアプローチとして、これまで2種類の方法が採られてきた。ひとつはハードウェア資源を投入して可能な限りの並列性を引き出す並列化のアプローチであり、もうひとつは、プログラムの実行上無駄な部分をそぎ落とす最適化のアプローチである。マイクロプロセッサは、これまで、動作周波数の向上と並列化の相乗効果により飛躍的な性能向上を果してきた。そして今後さらなるブレークスルーを達成するものとして最適化技術に期待が寄せられている^[1]。

ここで我々が注目しているのは、コンパイラによる静的な最適化技術ではなく、プログラムの実行挙動を把握した上で、その挙動に合わせて最適な方式を探る動的最適化の技術である。そこで我々は、S/W, H/W両面からの究極の動的最適化を目指し、メタレベル計算原理による動的最適化と柔構造計算機システムYAWARAを提案している^[2]。

動的最適化を行う際に鍵となるのは、プログラムの実行挙動の把握と、効果的な対処方法の決定を迅速かつ正確に行うことである。そのためにハードウェアによりプロファイル情報を取得する試みが活発に行われている^[3, 4]。また、^[5]で試みられているような比較的高コストの分岐予測も提案されている。

これまでのアプローチでは、上のように、狙った結果を効率的に得られるよう、的を絞った資源投入を行うことで成果を得ていた。これに対して我々は、YAWARAシステムで究極の動的最適化システムを目指すために、動的最適化のメカニズムをメタレベル計算原理として一本化して扱うアプローチをとる。従来目的を絞って導入されていたメカニズムを汎化し、それらを統合することでさらなる高次元の最適化を実現するためである。

具体的なアプローチのひとつとして、まず、従来の最適化手法・予測手法に因われず、プログラムを実行して行く中でCPU内で発生する様々なイベントの間の関連性、規則性を抽出できないかと考えた。その第一歩として、CPU内で発生するイベントの記録を時系列データとみなして、そこにデータマイニング技術を適用することを試みた。

本稿では、動的最適化に適用するためのCPUマイニングについて検討し、最初のステップとして現在得られている結果を紹介する。

2 時系列データのマイニング

データマイニングは膨大な量のデータベースを対象として、そこから一定の法則性・規則性を引き出すために考案された解析技法の総称であり、現在、様々な分野で顕著な成果が得られている^[6]。我々は、動的最適化のためのプログラム挙動の解析手法として、これまで顧みられることのなかった関連性・規則性を抽出できることを期待し、データマイニング技術の適用を試みた。

CPU内でプログラムが実行される際には、プログラムの進行に伴い、命令語のフェッチ、メモリへのデータのアクセス(ロード/ストア)、キャッシュミス、分岐、割り込みなど様々なイベントが発生する。これらのイベントは、もちろん独立に発生するものではなく、プログラムの記述や与えられるデータの性質に従って時系列に発生するものである。このため時系列データのマイニングに着目し、その基本的なアルゴリズムである *AprioriAll*^[7] をCPUイベント列に適用することにした。

時系列データのマイニングでは、たとえば、どの顧客がいつ何を購入したかの履歴を保存しているデータベースを想定する。顧客ごとに購入履歴をまとめ、購入品を時系列に表現するリストを作成する。ここで購入品目は数値化しておく。顧客数だけのリストができる。時系列データマイニングも目標は、顧客の時系列リストの中で頻出するサブパターン(アイテムセット *itemset* と呼ばれる)を抽出することにある。

ここでサブパターンの出現頻度が問われることになる。時系列データマイニングでは、出現頻度をサポートと呼ぶ。サブシーケンスを探索する過程で最低サポートに満たないものは外される。*AprioriAll*では、まず最低サポートを満足する1要素のアイテムセット (*Literalitemset*: *large itemset*) をリストアップする。そして1要素のアイテムセットを組み合わせることで、より多くの要素数を持つ *Literalitemset* を導く。このとき、生成した *Literalitemset* のサポートを求め、最低サポートを満足していなければ消去する。この繰り返しにより、最低サポートを満たす最大の *Literalitemset* を求める。最後に、この過程で得られた全ての *Literalitemset* に対し、包含関係にあるものを除外すれば、目的とする頻出サブシーケンスの集合が得られる。

3 CPUマイニング

3.1 時系列データマイニングの適用

CPUマイニングの概念図を図1に示す。時系列データマイニングを用いることにより、頻出するサブパターンを抽出することが期待できる。ただし、以下の点に留意してアレンジする必要がある。

顧客の概念が無い： 時系列データマイニングでは顧客ごとの時系列データをもとにしていた。

Preliminary Study on CPU Mining as a Novel Dynamic Optimization Technique

[†] Takashi Yokota, Koichi Suga, Kanemitsu Ootsu, and Takanobu Baba, Department of Information Science, Faculty of Engineering, Utsunomiya University

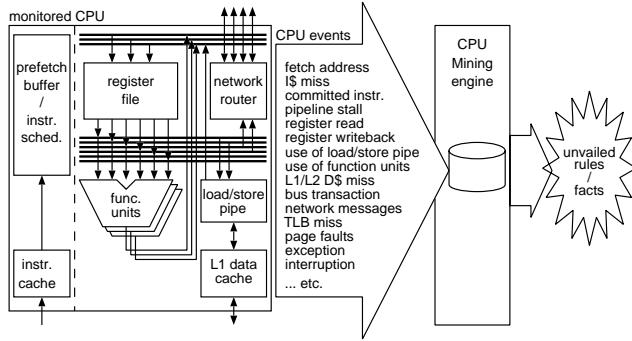


図 1: CPU マイニングの概念図

繰り返し出現を考える： 時系列データマイニングではアイテムの繰り返し出現は暗黙的に許容していなかった。イベントの発生順序が前後する： CPU マイニングでは、out-of-order 実行等により必ずしも順序が確定しない。時系列データが長い： CPU マイニングでは、対象イベントを選ばないと時系列データが長くなる。

3.2 CPU マイニングの試み

図 1 中に示されているように、様々なイベントを使用することが考えられるが、今回、以下の 3 通りで試行した。

- (a) 分岐イベント：分岐が行われたとき (命令フェッチアドレスが不連続になったとき) を検出し、その時のアドレスを 10 個まとめてひとつのリストとした。ホットパスの検出が期待できる。
- (b) 連続実行される命令のオペコード：上記の分岐イベントの間で実行される命令のオペコードをまとめた。処理の特徴が抽出できるものと期待される。
- (c) 命令のアドレス列：上記 (b) と同様。ただしオペコードではなく、フェッチアドレスとした。

前節での検討結果を受け、時系列データマイニングのアルゴリズムに CPU マイニング向けの改良を施した。CPU イベントの取得のため、SimpleScalar シミュレータにより実行ログを得て、それを CPU マイニングへの入力としている。

現在、逆行列および SPECint95 の各プログラムに対して CPU マイニングの適用を試みている。図 2 は、逆行列プログラムについて、上記の (a) 分岐イベントを使用した場合のマイニング結果である。図の左側は機械語のリストと、基本ブロックを示す。右側は検出されたアイテムセットとそのサポートを示している。たとえば (2 4) は、基本ブロック (2) (b) (4) の順で実行されるシーケンスを示しており、そのサポートは 4.8% であったことを示している。

図 2 からたとえば以下のことが導ける。基本ブロック (5) の実行頻度は非常に高い ((5) = 99.8%)。しかしながら (5) だけが長く連続して実行される頻度は多くない ((5 5 5 5 5 5 5 5 5) = 30.8%)。

4 おわりに

動的最適化のためのプログラムの挙動把握手法のひとつとして、CPU マイニングを提案した。CPU マイニン

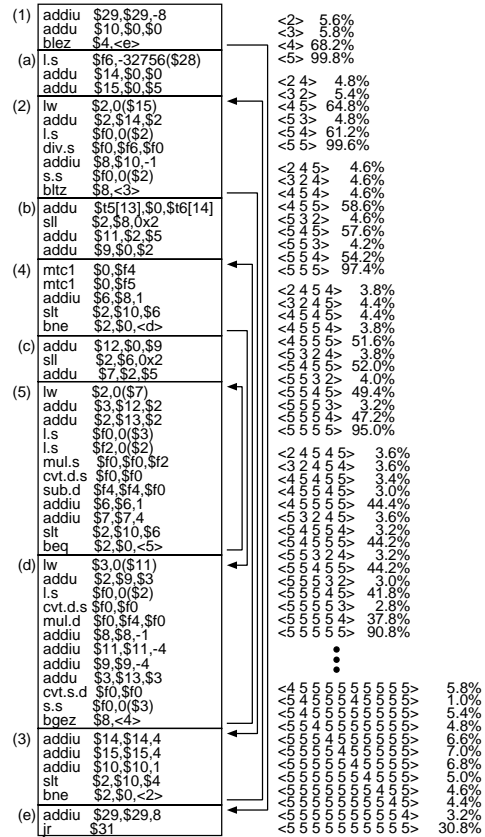


図 2: 逆行列の CPU マイニング結果

グは、プログラムの実行中に CPU 内で生じるイベントに対して時系列データマイニング手法を適用するものである。本稿では、その基本的な考え方を述べ、いくつかのプログラムに適用した結果を報告した。

今後は、YAWARA システム上での実現を目指し、CPU マイニング向けにさらにアルゴリズムを改善するとともに、適用対象の CPU イベントの種類を増し、適用可能性を探っていく。

謝辞 本研究は、一部日本学術振興会科学研究費補助金 (基盤研究 (B)14380135, 同 (C)14580362, 若手研究 14780186) の援助による。

参考文献

- [1] S. J. Patel, S. S. Lumetta, "rePlay: A Hardware Framework for Dynamic Optimization," IEEE Trans. on Comput., Vol.50, No.6, pp.590-608, June 2001.
- [2] 馬場, 横田, 大津, "YAWARA: メタレベル計算原理に基づく柔構造計算機システム," 情処全大 6G-1 ~ 3, 2003.
- [3] J. Dean, et al., "ProfileMe: Hardware Support for Instruction-Level Profiling on Out-of-Order Processors," Proc. the 30th Ann. Int'l Symp. on Microarchitecture (Micro'97), pp.292-302, Dec. 1997.
- [4] C. B. Ailles, G. S. Sohi, "A Programmable Co-processor for Profiling," Proc. HPCA-7, pp.241-252, Jan. 2001.
- [5] M. Kampe, et al., "The FAB Predictor: Using Fourier Analysis to Predict the Outcome of Conditional Branches," Proc. HPCA'02, pp.223-232, 2002.
- [6] M. J. A. Berry, G. S. Linoff, "Data Mining Techniques: for Marketing, Sales, and Customer Support," John Wiley & Sons, Inc., 1997.
- [7] R. Agrawal and R. Srikant, "Mining Sequential Patterns," IBM Research Report RJ 9910, Oct. 1994.