UltraSPARC 命令レベルシミュレータ UltraSim*

2F－2

河場 基行†　木村 康則

(株) 富士通研究所‡

e-mail: kawaba@flab.fujitsu.co.jp

1 はじめに

ソフトウェアのプログラムコードを最適化し速度向上を図るソフトウェアチューニングでは，性能ボトルネックになっているプログラム部分の特定とその要因を抽出する必要がある．特に，ハードウェアによって実行遅延を軽減できないプログラム部分（命令流が滞ってしまう部分）の特定は，ソフトウェアチューニングの鍵となる．ところが UltraSPARC に代表されるスーパスカラプロセッサでは，非順序命令実行完了機能や分岐予測機能などによって，キャッシュミスやパイプラインストールなどによる実行遅延を軽減してしまうためチューニングを施すべきプログラム部分の特定が以前にも増して困難になりつつある．

我々は UltraSPARC 上のソフトウェアのチューニング支援を目的とした命令レベルシミュレータ UltraSim を作成した．本シミュレータはソフトウェア開発の支援のみならず，コンパイラの最適化手法の発見にも有用である．ここでは，UltraSim が提供する情報及び UltraSim を用いたソフトウェアチューニング例を示す．

2 UltraSim

2.1 処理方式

UltraSim は命令レベルトレースドリブンシミュレータである．アプリケーションの実行履歴から UltraSPARC-I のパイプラインの挙動をシミュレーションする．(図 1 参照) 入力となる実行履歴は，SHADOW[1]によってオンメモリで採取されたものと，ファイル形式のものとをサポートしている．キャッシュ

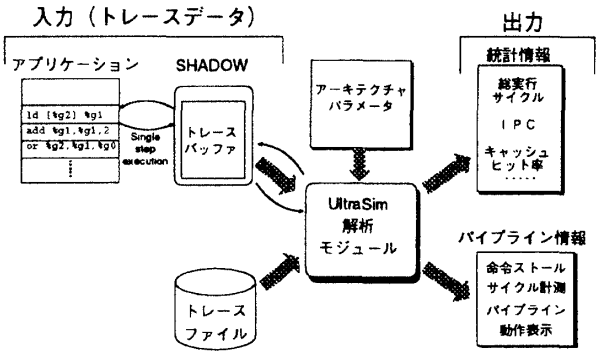

図 1: UltraSim の構成

サイズ，メモリアクセスレイテンシ等のアーキテクチャパラメータを変更することができる．これによってソフトウェアチューニングの効果の上限を見積もる事が可能となる．

2.2 提供する情報

UltraSim ではソフトウェアチューニングをサポートするために，実行サイクル数・キャッシュのヒット率・分岐予測正解率などの統計情報以外に，以下の２つの情報を提供する．

命令ストールサイクル計測　実行命令数が多くても，命令流に滞りがなければ（つまりストールしているサイクルが短ければ），チューニングの余地は少ない．全実行を通してストールしているサイクル数の多い命令が，パフォーマンスチューニングのターゲットになると考える．UltraSim では総ストールサイクル数の多い命令を出力機能を持っている．図 2 に出力例を示す．

パイプライン状態表示　パイプライン図を表示する機能を持っている．(図3) パイプライン状態はストールの原因も同時に表示される．パイプラインストールの様子をパイプライン図に組み込むことによって，ストールの直接的/間接的要因を直感的に把握しやす

---

* Trace-driven instruction level simulator for UltraSPARC: UltraSim

† Motoyuki KAWABA

‡ FUJITSU LABORATORIES LTD.

[1] Sun Microsystems, Inc によって提供されているトレーサ

```
Stall   Freq   PC        penalty Instruction
------  -----  --------  ------- -----------
100006  50000  0x266ac   2.0     subcc %i4, %i5, %g0
60000   20000  0x103e0   3.0     sra   %o0, 0, %o0
50040   50000  0x266c0   1.0     xor   %l3, %i4, %l3
50004   50000  0x266a8   1.0     ld    [%i1+%i0], %i4
```

Stall: 総ストールサイクル　Freq: 実行回数
penalty: 1実行当たりのストールサイクル

図 2: ストールサイクル表示

```
Pipeline          PC       LD/ST    DP   Instruction
--------------    --------  -------- ---- ------------------
FDGtEEENNW        00025248 000437ae 4637 ldsh  [%o5+29],%o5
FDGttttENNNNW     0002524c          4640 subcc %o5,0,%g0
 FDGtttEENNNW     00025250          4640 bg,a  0x2526c
```

Pipeline：パイプラインの状態を示す.
F,D,G,E,N,W はパイプラインステージ
t は真依存によるパイプラインストール
LD/ST：ロードストアによるアクセスアドレス
DP: 命令がディスパッチされた時刻

図 3: パイプライン状態表示

くなっている．先の命令ストールサイクル計測機能によってピックアップした箇所を，この機能によって検証することにより，容易にパフォーマンスボトルネック箇所及び要因検出を行うことができる．

2.3 シミュレーション精度/速度

SPEC92 ベンチマークをターゲットにしてシミュレーション精度を測定した．実機での実行時間と，シミュレーションによる実行時間見積もりとの差を比較した．シミュレーション誤差は SPECint92 で 3.67%，SPECfp92 で 0.84% となった．

シミュレーション結果をソフトウェアにフィードバックするためには，シミュレーション速度も重要となる．命令ストールサイクル計測機能を用いた場合，UltraSim は毎秒 20,000～30,000 命令をシミュレートする．(S-4/20(HyperSPARC 150MHz) で計測) これは実機の 1/3000～1/4000 の速度に相当する．

3 ソフトウェアチューニング例

UltraSim を用いたソフトウェアチューニング例を示す．図 4 は UltraSim によるパイプライン状態表示である．ここでは最後の命令 (add 命令) の実行ストールに注目する．add 命令は先行の ld 命令に対して真依存関係があることが分かる．また ld 命令は，先行ストア命令との間にメモリ干渉を引き起こしている．

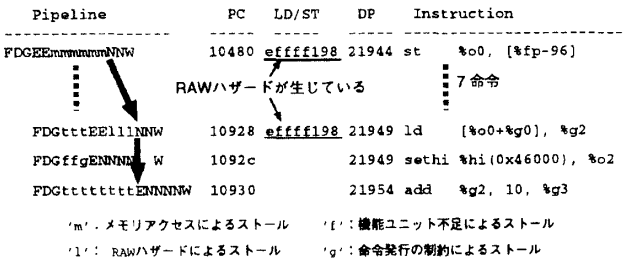

図 4: ADD 命令の解析

このため ld 命令は，先行のストア命令が完了しないと実行することができない．このことから add 命令のストールサイクルを削減するためには，① ロード命令と add 命令との間に挿入する，② ストア命令とロード命令の間に挿入する，という 2 つの手法があることが分かる．

ストア命令と add 命令との間には 9 命令もあり，静的なソースコード解析から後者のチューニング手法を発見することが難しい．UltraSim のようなソフトウェアの動的な振る舞いの解析ツールによる支援が必須である．

4 まとめ

UltraSPARC 上のソフトウェアチューニングを支援する命令レベルシミュレータ UltraSim を紹介し，シミュレータによるソフトウェアチューニングの例を示した．

今後ハードウェアが複雑になると，ソフトウェアの動的な振舞（特にメモリアクセスによる遅れや命令フェッチの遅れ）の予想/解析が困難になる．今回提案した UltraSim のように，ソフトウェアの微視的な振る舞いを直感的に分かりやすい形式で表示するソフトウェア解析ツールが，ソフトウェアチューニングにとって有効なツールとなると考えられる．

参考文献

[1] "UltraSPARC Programmer Reference Manual", Sun MicroSystems, 1995

[2] 志村 浩也 他.「スーパスカラプロセッサの性能評価 - Paratool - 」, 情報処理学会研究会報告 93-ARC-102-1, 1993