4P－6　並列推論エンジンPIE64の要素プロセッサUNIRED-IIの並列プログラムでの評価

渡辺 正泰　日高 康雄　小池汎平，田中英彦
東京大学工学部

1　はじめに

並列推論エンジン PIE64 の要素プロセッサである UNIRED-II は、Committed-Choice 型言語 Fleng を効率良く実行することを目指して設計されたプロセッサである。UNIRED-II は基本的には RISC 型パイプラインを持つ汎用プロセッサであるが、次の2つの特徴を持つ。

- パイプライン共有マルチコンテキストプロセッサである。
- Fleng のための複合命令を持つ。

本稿では PIE64 実機上で Fleng で記述された並列プログラムを用い、UNIRED-II の評価を上記2点について行なう。

2　並列推論エンジン PIE64

PIE64 は 64 台の推論ユニット (IU) を持つ並列計算機である。各 IU は低レイテンシの通信と自動負荷分散機構を特徴とするネットワークによって結ばれている。UNIRED-II は、IU の要素プロセッサとして用いられている。

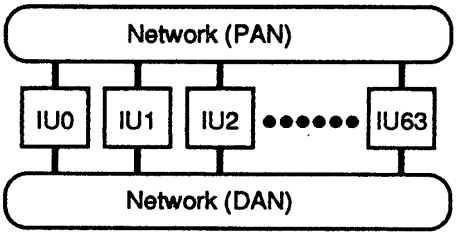

図 1: PIE64 の摸式図

Evaluation of UNIRED-II:A computaional processor of PIE64, using Parallel applications
Masahiro WATANABE, Yasuo HIDAKA,
Hanpei KOIKE and Hidehiko TANAKA
Faculty of Engineering,the University of Tokyo

3　UNIRED-II のマルチコンテキスト処理

並列計算機では通信のレイテンシ隠蔽が重要な問題となる。レイテンシ隠蔽の手法としてスレッドの切替を行なうマルチスレッド処理があり、高速なスレッドの切替を行なう機構としてマルチコンテキスト処理がある。

UNIRED-II のマルチコンテキスト処理では、各クロック毎に実行可能なコンテキストから命令単位にスケジューリングを行なってパイプラインに投入する。これにより高速なスレッド切替と、同一コンテキストの依存関係のある命令がパイプライン上で離されることによるパイプラインストールの低減が実現される。

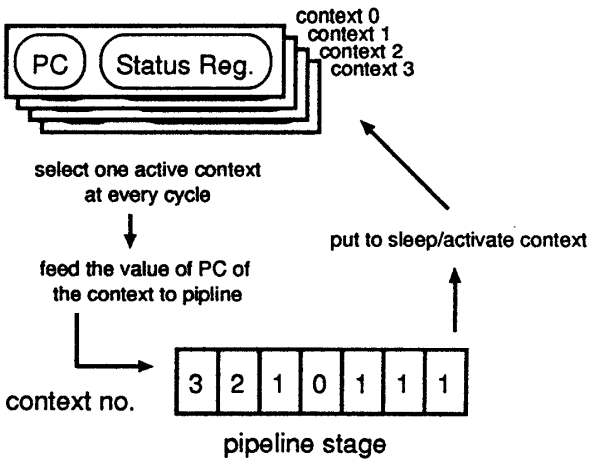

図 2: マルチコンテキスト処理の概要

このようなマルチコンテキスト処理を行なうプロセッサには HEP,MASA[1] などがあげられるが、UNIRED-II では同一コンテキストからの連続した命令の投入を可能にすることで、並列度が十分でないときにも極端な性能低下を招かないようになっている。

4　UNIRED-II の複合命令

PIE64 がターゲットとしている言語 Fleng では、変数のデレファレンスとタグチェックを効率良く行なうことが必要である。そのために UNIRED-II には次の複合命令が実装されている。

これらの複合命令は、複数回のメモリアクセスと自分自身への分岐を行なうが、UNIRED-II ではマルチコン

derf	変数のデレファレンス
dfcl	デレファレンスとリスト型であるかのチェック
dfcc	デレファレンスと定数型であるかのチェック
dcll	dfcl に加え、先頭ワードの読みだしを行なう

表 1: UNIRED-II の derf 系複合命令

テキスト処理を利用して、パイプラインストールを起こさないように実装されている。

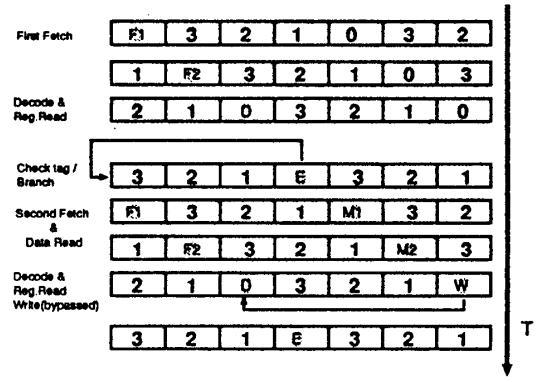

図 3: derf 系複合命令のパイプライン動作

5 PIE64 を用いた評価

これら UNIRED-II の 2 つの特徴の有効性を評価するために、PIE64 実機上で Fleng でかかれた並列プログラムを実行し、以下の計測を行なった。

1 コンテキスト数に対するパイプラインストール時間
2 複合命令を使用した場合と、使用しない場合の実行時間

計測には並列度の高いプログラム (queen12) と、それほど高くないプログラム (prime20000) を用いた。複合命令に関しては、これらのプログラムに含まれる derf,dcll を同等の処理を行なう複数の命令に置き換えた。

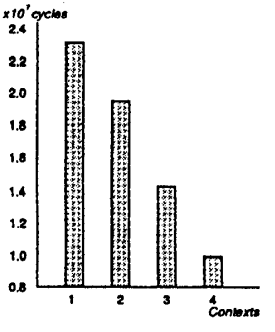

図 4: prime20000 の load ストール時間

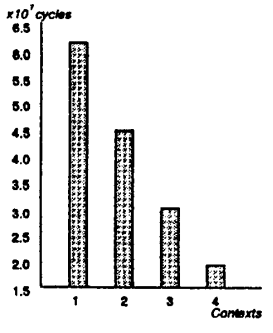

図 5: queen12 の load ストール時間

derf	dcll	time (sec)	speedup
○	○	2.174	1.0
×	○	2.071	1.05
×	×	2.014	1.08

表 2: prime20000 の実行時間

derf	dcll	time (sec)	speedup
○	○	3.241	1.0
×	○	2.994	1.08
×	×	2.791	1.16

表 3: queen12 の実行時間

6 考察

複合命令に関しては、マルチコンテキスト処理を利用してパイプラインを乱さないように実装することで、同等の処理を通常の命令で行なった場合より高い性能が得られることが確認された。ただその値は島田 [2] のシミュレーションによって予想された値より低かった。これは島田がネットワークを想定していなかったことによると考えられる。

パイプラインストールの低減については、コンテキスト数の増加につれ、依存関係のある命令が離される可能性が高くなるためにパイプラインストール時間が減少していることが確認できる。primes の方が減少の割合が小さいのは、primes は queen に比べ並列性が低く、queen がほぼ最大コンテキスト数を保つのに対し、primes は最大コンテキスト数を保てないために起こる。

UNIRED-II はキャッシュを持たないため今回はマルチコンテキスト処理がキャッシュに与える影響を評価していないが、今後はキャッシュの影響を考慮に入れた解析が必要になるだろう。

参考文献

[1] 島田 健太郎, 小池 汎平, 田中 英彦, "並列推論マシン用推論プロセッサの研究", 東京大学工学部 博士論文, 1993

[2] Halstead R. and Fujita T, "MASA: A Multithreaded Processor Architecure for Parallel Symbolic Computing", Proc. of the 15th Annual International Symposium on Computer Architecture, pp.443–451, 1988

[3] James Laudon, Anoop Guputa and Mark Horowitz, "Interleaving: A Multithreading Technique Targeting Multiprocessors and Workstations", Proc. of ASPLOS-VI, 1994