

並列化コンパイラ評価用マルチプロセッサシミュレータ

5K-7

安里 彰*

河場 基行**

小沢 年弘*

* APC研究体 (富士通)

** (株)富士通研究所

1 はじめに

アドバンスド並列化コンパイラ研究体(以下, APC)富士通研究室では, 手続き間データ依存解析技術に基づく粗/中粒度並列化, 分散キャッシュ向け自動データ分散技術, 粗粒度タスクの投機的実行技術などをコアに並列化コンパイラの研究を進めている.

開発したコンパイラの評価は, コンパイルコードを実機で走行させて実測することが基本になるが, 我々はそれと並行してソフトウェアシミュレータによる評価も必要と考えている. なぜなら, ハードウェア PA データ等ではわからないシステムの内部挙動の観測によるボトルネック解析や新たなアイデアの実験を通じて, コンパイラによる並列化を促進するには, ソフトウェアシミュレーションが必須だからである.

我々はこの目的のため, 富士通研究所で開発したマルチプロセッサシミュレータ SEEDS[1] を拡張して並列化コンパイラの評価を行うことにした. 本論文では, まず SEEDS を簡単に紹介し, 続いて我々の研究でどのように SEEDS を活用していくか述べる.

2 SEEDS シミュレータ

2.1 シミュレータの概要

SEEDS は図 1 に示す構成のシミュレータである. 基本的には SPARC V9 命令セットに対応した SMP システムの実行型シミュレーション (EDS 方式) を行うが, モード切り換えによって, トレース型シミュレーション (TDS 方式) も行うことができる. また, プロセッサ内部のマイクロアーキテクチャ, キャッシュ構成, バスアーキテクチャ等のシステム属性は実行時パラメータとして設定可能である.

2.2 EDS 方式と TDS 方式

EDS 方式では SEEDS 内部メモリに展開したアプリケーション実行オブジェクトから順次命令をフェッチし, ISA レベルのエミュレーションを行う. 更に, システムシミュレータの内部に有する複数の CPU モデルとシステムモデルにおいてクロックレベルの詳細なタイミングシミュレーションを行う.

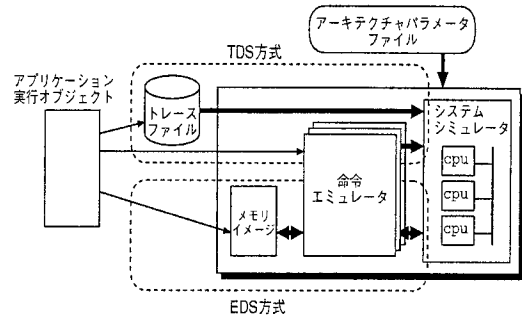


図 1: SEEDS の構成

EDS 方式は精度の高いシミュレーションが可能な反面, 動作速度が遅い欠点がある.

一方, TDS 方式は何らかのトレーサであらかじめ採取した¹実行トレースを入力とし, 実行トレースに格納された順に各命令の実行タイミングを計算していくことでシミュレーションを行う. 命令エミュレーションを行わないので高速である.

また, SEEDS の TDS シミュレーションでは更なる高速化を目指して, 単体 CPU モデルによる疑似並列化の機構を取り入れている. これは, fork/join 型の単純な並列プログラムを対象に, 図 2 のように, あらかじめトレース内に挿入してあるスレッドの fork と join を意味する特殊な命令を SEEDS の一つの CPU モデルが認識し, 他 CPU の分もまとめてタイミング計算を行うものである.

我々が研究対象としているプログラムは比較的単純な構成の科学技術計算プログラムなので, 上記疑似並列化を採り入れた TDS シミュレーションで誤差の小さい評価が行える場合が多いと考えている.

3 並列化コンパイラ研究のための SEEDS の機能強化

我々は SEEDS シミュレータを並列化コンパイラ評価用ツールとして利用するにあたり, 必要に応じて機能強化を行っている. 本章ではそれらの変更点について説明する.

A multiprocessor system simulator for evaluation of parallelizing compiler

Akira ASATO,¹ Motoyuki KAWABA,² Toshihiro OZAWA¹

1: Advanced Parallelizing Compiler Project (Fujitsu)

2: Fujitsu Laboratories LTD.

¹SEEDS 自身を用いて EDS 方式で採取することもできる

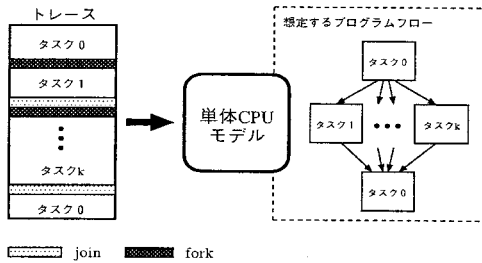


図 2: 単体 CPU による疑似並列処理

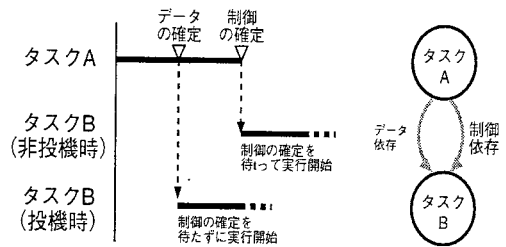


図 4: 投機的なタスク実行

3.1 OpenMP プログラムの並列化サポート

我々は OpenMP[2] 化された Fortran プログラムを主な研究対象としている。従って、上記の疑似並列化を実現するにあたり、OpenMP 仕様における *parallel sections* ディレクティブに対応させて fork/join 指示命令を実行トレースに挿入する仕組みが必要になる。SEEDS 利用環境にはこのための専用ライブラリが用意されており、我々は Fortran ソース内の *parallel sections* ディレクティブを自動的に専用ライブラリコールに置き換えるプリプロセッサによって、本機能の実現を図ることにした。

3.2 分散キャッシュの有効性評価

我々の研究テーマの一つに、中粒度並列実行の高度化がある。ここでは、手続き間依存関係解析によるプログラム並列性抽出と、データローカリティを考慮した分散キャッシュの有効利用が鍵になる。このうち SEEDS では特に後者の評価をサポートする機能を追加することにした。

分散キャッシュが有効に働くためには、データの false sharing によるキャッシュミスを極力減らすことが重要になる。この効果を厳密に評価するためには、同一キャッシュライン上の異なるデータを別々の CPU が write アクセスした事象を捉える必要がある。このため、SEEDS のキャッシュシミュレータ部分を拡張し、図 3 に示すように各ライン上のワード単位にアクセス履歴を管理して false sharing の発生状況を把握できるようにした。

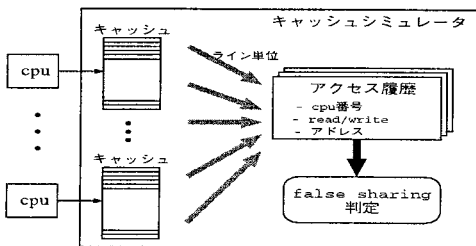


図 3: false sharing の認識

3.3 投機的タスク実行

我々の別の研究テーマに投機的なタスク実行がある。これは主として非ループ部の高速実行を狙ったもので、あるタスクの実行に必要なデータが揃った時点で、制御の確定を待たずに並列実行してしまうというのが基本的な考え方である。(図 4)

投機実行による全体的な性能向上の度合いは実機でも測定可能であるが、それだけでは詳細な分析は困難である。例えば投機の成否を議論するには、単に予測の成功率を調べるだけでは不十分で、投機タスクと依存関係を持つ後続タスクの実行タイミングに及ぼした影響まで含めたトータルな評価が必要になる。そこで、SEEDS を機能強化して、投機タスクおよびそれに依存するすべてのタスクについて、実行可能になる時刻や実行開始時刻をモニタできるようにした。

4 まとめと今後の課題

APC 富士通研究室で利用している SMP シミュレータ SEEDS および、その機能拡張について述べた。我々は SEEDS の提供する二種類のシミュレーション方式 (EDS, TDS) のうち後者を利用して研究を進めようとしている。これは、両者の間に存在するシミュレーション精度と速度のトレードオフに関し、我々が対象とする科学技術計算プログラムでは精度的な問題は小さく、速度を重視する方がよいとの判断に基づくものである。しかし、これはあくまでも仮説であり、早い時点で確認しなければならない。

また、SEEDS に加えた拡張は現状ではまだ不十分であり、研究が進むに従って、よりきめ細かな対応が要求されると思われる。将来的に必要な機能を早期に見極め、研究本体と連携した評価環境をタイムリーに構築していくことが重要と考えている。

参考文献

- [1] 大河原他, “計算機システムにおける分岐予測外れバスの影響”, SWoPP2001 予稿集.
- [2] OpenMP Architecture Review Board, “OpenMP Fortran Application Program Interface Version 2.0”, November 2000.