

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成24年 6月 4日現在

機関番号：13302

研究種目：若手研究（B）

研究期間：2009～2011

課題番号：21700056

研究課題名（和文） リアルタイム依存解析による実用的な並列命令処理方式の構築

研究課題名（英文） Designing a Practical Processing Model of Parallel Instructions Using Real-time Dependence Analysis

研究代表者

佐藤 幸紀（SATO YUKINORI）

北陸先端科学技術大学院大学・情報社会基盤研究センター・助教

研究者番号：30452113

研究成果の概要（和文）：

本研究では、アプリケーションプログラムに対して生産的に高度な並列化を行うことを支援するためにリアルタイムデータ依存解析システムを研究開発し、プログラムに内在する並列性をユーザーに提示する仕組みを構築することを目指した。動的バイナリ変換システムを用いて本システムを構築し、ベンチマークプログラムにより実用性や精度に関する評価を行った。それらの結果、検出したデータ依存関係から生産的に並列な領域を抽出できることを解明した。

研究成果の概要（英文）：

In this research, we developed a real-time data dependence analysis system that can help advanced parallel computing and parallelization of application programs by providing inherent parallelisms in a program for users. We implemented the analysis system on a dynamic binary translator, and evaluated the productivity and the accuracy using benchmark programs. From these results, we found that we can productively extract parallel regions from the obtained data dependencies and these regions can be utilized for parallelization in an effective manner.

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2009年度	1,200,000	360,000	1,560,000
2010年度	1,000,000	300,000	1,300,000
2011年度	1,000,000	300,000	1,300,000
年度			
年度			
総計	3,200,000	960,000	4,160,000

研究分野： 計算機アーキテクチャ

科研費の分科・細目： 情報学・計算機システム・ネットワーク

キーワード： データ依存解析、リアルタイム依存解析、ループ並列性、並列命令処理方式、ランタイム並列化、実行時プロファイリング

1. 研究開始当初の背景

歴史的にCPUの性能向上はCPU動作周波数の向上に大きく依存してきた。しかしながら、消費電力や信頼性の観点から動作周波数の向上が困難となった為、将来的にもCPU性能

向上を可能とする並列処理技術とそれを支えるアーキテクチャ技術に高い関心が集まっている。

近年、新しい並列処理アーキテクチャ技術としてメニーコアプロセッサやSIMD、FPGA

といったアクセラレータを利用した高度な並列処理が注目されている。このような高度な並列処理により性能向上を目指す場合、ハードウェア上に高度で強力な演算能力を備える並列処理エンジンを実現する技術よりも、その理論性能を引き出すための技術が重要になると予測されている。高度な並列処理はアプリケーションの実装毎に変化する並列化するべき対象を的確に見つけ出すこと、そして、その部分を並列実行する適切な手段を見出すことが必要な解空間の広い発見的で困難な作業である。さらに、年々進行するアプリケーションプログラムの大規模・複雑化やプログラミング生産性低下という問題に加えて高度な並列化を考える必要があり、それらを容易化する技術への期待は極めて高い。

このような技術動向の状況において計算機アーキテクチャ研究に求められるものは、高い並列効率と生産性を両立する並列命令処理方式を確立していくことである。そこで、将来的に自動で命令レベルよりも粗粒度な並列部分を抽出しハードウェアに展開する機構に発展させることを視野に入れつつ、現行のスーパースカラが着目する局所的なデータ依存解析の枠を超える大域的なメモリ依存解析を核とするデータ依存解析システムの開発が望まれている。

2. 研究の目的

本研究の目的は実行時プロファイリング技術をリアルタイム依存解析技術へと発展させ、検出した依存関係から高い並列効率と生産性を両立する並列処理方式の形成を支援するツールとして提供することである。

この目的を実現する為に既存の実行時プロファイリングシステムを発展させて命令レベルの依存情報をリアルタイムに検出する手法、依存解析結果に基づき並列性をリアルタイムに抽出する手法を研究開発する。

リアルタイム依存解析は時間的にも空間的にも大きく変化するプログラムの挙動やポインタによる間接メモリアクセスを解析する非常に強力かつ実用的な手法である。本研究では、命令レベルよりも大きな粒度でありコード実行のホットスポットを形成するループに着目し、ループ階層間の依存解析をリアルタイムに行うことを試みる。リアルタイム依存解析は命令レベル並列性を超える自動並列化の為には不可欠な技術であり、ムーアの法則に従い増加し続ける計算ノード単体性能の効率的利用の為に最も重要な要素技術である。加えて、実際のアプリケーションプログラムから実用的な並列命令処理方式を構築していくことはムーアの法則と同じスピードでCPU処理性能を向上していくために必要不可欠となる技術である。

本研究においては、本手法をハードウェア実装も視野に入れた実行時プロファイリング機構として設計するとともに、実環境上に評価システムを構築し、実践的に得られる並列性を評価することにより、その並列処理効率と実用性を明らかにすることを目指す。

3. 研究の方法

本研究では、実行時プロファイリングを利用してプログラムからリアルタイムに依存関係を検出し並列部分を抽出する機構を設計する。実行時プロファイリングにはPinというツールを用いる。Pinは仮想マシン技術とJIT(Just In Time)コンパイラ技術により任意の実行バイナリコードを動的にコード変換するDBT(Dynamic Binary Translation)のためのツールの一種である。研究代表者佐藤はこれまでにPinを利用してループ構造検出機構を構築してきたため、本研究では開発済みのループ構造検出機構を拡張し、高い並列効率と生産性とリアルタイム性を実現する並列命令処理機構に発展させる。

本研究を効率的に遂行する為に、実際の計算機システム上で起こっている現象を忠実に再現することを第一に行う。現在の並列処理のプロセスはプログラミングからハードウェア制御に至るまで多階層のレイアをまたぐ複雑なものである。この複雑な並列処理プロセスにおいて実際に起こっている現象を忠実に再現し、現象の本質を把握することに努める。その本質的な理解こそが最適化や予測に基づく投機的処理への応用に役立つといえる。そこで、実システム上で起こっている現象ということで、現在のアプリケーションを反映したベンチマークプログラムの利用や最新技術を取り込んだコンパイラにより生成したバイナリコードを入力としたリアルタイム依存解析と並列性抽出を行う。このような実環境における現象を命令レベルで忠実に再現することこそが実用性の高い命令処理方式実現の第一歩であるという方針の下、リアルタイム依存解析と並列性抽出機構を実装する。

4. 研究成果

本研究では、3年間の期間でリアルタイム依存解析と並列性抽出機構の研究開発に取り組み、検出した依存関係から高い並列効率と生産性を両立する並列処理方式の形成を支援するツールとして提供することを目指した。以下、年度毎の研究成果を述べる

平成21年度はリアルタイム依存解析部分の基本設計とその実行時プロファイリングによる実装を行い、実用的な並列命令処理方式の構築のための基礎データの収集に取り組んだ。リアルタイム依存解析部分の設計に

においては、ループ階層構造に着目して効率的に依存関係を把握するために、ループ階層構造を実行時に検出する手法の詳細を検討した。具体的には、ループ階層構造に関連する研究資料を該当分野の国際会議等から広く収集し、現状の技術水準の把握とリアルタイム依存解析への応用を前提とした実行時ループ階層構造検出機構の基本要件を確認した。これらの結果より、ループ階層構造を正確に理解するためには自然ループ(natural loop)を検出することが不可欠であり、ランタイムにどのように自然ループを検出するかという問題に取り組む必要があるという知見が得られた。また、自然ループのランタイム検出の1つの実装として、実際に実行時プロファイリングを用いて、後方分岐命令とその飛び先のターゲットとなる命令の区間をその区間の先頭命令がその区間の全ての命令を支配しているループ区間と仮定することにより自然ループを検出しループ階層構造を実行時に抽出可能であることを確認した。加えて、リアルタイム依存解析部のデータ依存関係の抽出については命令レベルにて実際にメモリアクセス情報を抽出することを行った。メモリアクセス情報を命令レベルで解析するためにはメモリアクセスに関する膨大なデータをハンドリングする必要があるという課題に対して、ループ階層構造に着目してループ区間を単位にデータ依存関係の把握を試みた。評価実験の結果、ループを単位とする並列部分の推定の有効性と実現可能性を確認した。

平成 22 年度は正確なループ構造をコード実行時に抽出可能な手法の設計および実装による評価に取り組んだ。ループ構造抽出部の設計の過程でループネスト構造を正確に抽出するためには reducible loop に分類される自然ループ(natural loop)だけを想定するのでは不十分という知見が得られたので、irreducible loop の検出も可能なループ抽出法に基づき実行バイナリコードを実行する際にループの動的な挙動をモニタリング可能な機構を設計した。また、ランタイム自動最適化技術に関連する研究資料を該当分野の国際会議等から広く収集し、現状の技術水準の把握と動的バイナリ変換(Dynamic Binary Translation)システムへの応用を前提とした実行時ループ階層構造検出機構の基本要件を確認し、システム上に実装した。実装したシステムの評価の結果、ソースコード上のループ構造と矛盾しない正確な構造が抽出されていること、バイナリコード実行中随時ループネスト構造が追跡されること、関数呼び出しをまたぎ動的に実行されるループネスト構造を L-CCT (Loop-Call Context Tree)として効率的に表現可能であること、SPEC CPU ベンチマークにおいて実行時ループ

検出なしのネイティブのコード実行と比べて平均4倍程度のオーバーヘッドにて実行時ループの検出が可能であることが示された。加えて、各ループのイテレーションの数(ループとリップカウント)を実行時に検出する手法を設計し、SPMD (Single Program Multiple Data) の並列モデルで記述される MPI プログラムにおいて並列化されているループ部分をコード実行時に検出することを試みた。評価実験の結果、ループにおいて並列化されている MPI プログラムから並列部分をコンパイル時のオプション指定やソースコードの参照をすることなく並列化されているループ部分を検出できることを確認した。

平成23年度は、リアルタイムのメモリ依存検出手法の詳細設計と検出されるメモリ依存情報に基づくリアルタイムの並列性抽出手法への適応を視野に入れた評価を行った。アプリケーションのバイナリコードを実行するのに合わせてリアルタイムにメモリ依存を検出するために最後にメモリライトアクセスを行ったメモリアドレスを効率的に記録するデータ構造を開発し、メモリ依存を検出するためのオーバーヘッドを時間とメモリの面から評価した。SPEC CPU2006ベンチマークのRefデータセットを利用して評価を行った結果、現実的なメモリオーバーヘッドで実時間のうちにメモリ依存を検出できることが分かった。さらに、リアルタイムなメモリ依存情報に基づく結果から実用的な命令処理方式のモデルとしてどのようなものが考えられるかという考察を行った。この過程で、メモリ依存関係を効率的に表現するために、関数呼び出しとループネスト構造を表現するL-CCT形式をメモリ依存を含んだLCCT+M (Loop-Call Context Tree with Memory)形式に拡張し、関数やループノード間のメモリを介したデータ依存をソースコードを読むことなしで理解できるようにした。また、関数やループノード間の並列性に加えて、ループ反復間に存在するループ並列性を解析できるようにループ反復をモニタする機能を追加することを行った。これらの結果、アプリケーションバイナリコードを実行する際にリアルタイムに依存を検出し、メモリ依存のない並列性を持つと考えられる部分を抽出できることが分かった。

以上のように、本研究の目的であったリアルタイム依存解析と実用的な並列命令処理の解析方式を開発するという事は達成できた。平成23年度は本研究課題の最終年度であったため、これまでに開発した要素技術や得られた知見をまとめ、ACM主催の国際会議CF' 11や国内の研究会において成果の報告を

行い、これらの成果を発表した。

5. 主な発表論文等

[雑誌論文] (計 2 件)

1. Yukinori Sato, Yasushi Inoguchi and Tadao Nakamura. On-the-fly Detection of Precise Loop Nests across Procedures on a Dynamic Binary Translation System. Proceedings of 2011 ACM International Conference on Computing Frontiers, 2011, pp. 25:1–25:10, 電子ジャーナル (DOI=10.1145/2016604.2016634), 査読有

2. Yukinori Sato. HPC systems at JAIST and development of dynamic loop monitoring tools toward runtime parallelization. High Performance Computing on Vector Systems 2011, 2012, pp. 65–78, 査読無

[学会発表] (計 7 件)

1. 佐藤幸紀, 井口寧, 中村維男. バイナリトランスレーションによるループ反復間のデータ依存解析. 第 133 回ハイパフォーマンスコンピューティング研究発表会, 2012. 3. 26, 神戸市

2. 佐藤幸紀, 井口寧, 中村維男. Loop-call context tree を用いたランタイムデータフロー解析. 2011 年並列/分散/協調処理に関する『鹿児島』サマー・ワークショップ, 2011. 7. 27, 鹿児島市

3. Yukinori Sato, Yasushi Inoguchi and Tadao Nakamura. On-the-fly Detection of Precise Loop Nests across Procedures on a Dynamic Binary Translation System. 2011 ACM International Conference on Computing Frontiers, 2011. 5. 4, イタリア イスキア

4. 佐藤幸紀, 井口寧, 中村維男. 動的バイナリトランスレーションによるループネスト検出とプログラムチューニング支援への応用. 第 18 回ハイパフォーマンスコンピューティングとアーキテクチャの評価に関する北海道ワークショップ (HOKKE-18), 2010. 12. 16, 北海道大学 (北海道)

5. Yukinori Sato. HPC systems at JAIST and development of dynamic loop monitoring tools toward runtime parallelization. 13th Teraflop Workshop (招待講演), 2010. 10. 21, 東北大学 (宮城県)

6. Yukinori Sato, Tadao Nakamura. Profiling the dynamic behavior of nested loops using the loop-call context tree. WISH: 2nd Workshop on Infrastructures for Software/ Hardware co-design, held in conjunction with 2010 International Symposium on Code Generation and Optimization (CGO), 2010. 4. 25, カナダ トロント

7. 佐藤幸紀, 中村維男. 実行時データ依存解析によるループ階層構造に着目した並列性抽出. 2009 年並列/分散/協調処理に関する『仙台』サマー・ワークショップ SWoPP2009, 2009. 8. 4, 仙台市

[図書] (計 0 件)

[産業財産権]

○出願状況 (計 0 件)

○取得状況 (計 0 件)

[その他]

ホームページ等

www.jaist.ac.jp/~yukinori

6. 研究組織

(1) 研究代表者

佐藤 幸紀 (SATO YUKINORI)

北陸先端科学技術大学院大学・情報社会基盤研究センター・助教

研究者番号: 30452113