

シングルチップマルチプロセッサにおける マルチグレイン並列処理

内田 貴之 木村 啓二
小高 剛 笠原 博徳

早稲田大学工学部電気電子情報工学科

〒 169-8555 東京都新宿区大久保 3-4-1 TEL:03-5286-3371

E-mail: {uchida,kimura,kodaka,kasahara}@oscar.elec.waseda.ac.jp

半導体集積度の向上と共にチップ上の資源の有効利用によるスケラブルな性能向上を目指す、次世代マイクロプロセッサアーキテクチャおよびそのソフトウェア技術(特にコンパイラ技術)の開発が求められている。このような次世代マイクロプロセッサアーキテクチャとしては、従来のスーパースカラ、VLIWのような命令レベル並列性のみでなく、異なる粒度の並列性を階層的に組合せプログラム全体より並列性を抽出できるマルチグレイン並列処理を実現するシングルチップマルチプロセッサ(SCM)が実効性能を高め集積度向上に対しスケラブルな性能向上が可能なアーキテクチャとして有望であると考えられる。本論文では、SPEC2000fp及びSPEC95fpより5本のプログラムに対しマルチグレイン並列処理を適用し、SCMの性能評価を行った。その結果、4プロセッサを搭載したSCMはマルチグレイン並列処理により1プロセッサの場合に対して1.4~3.8倍の速度向上が得られることが確かめられた。

Multigrain Parallel Processing on Single Chip Multiprocessor

TAKAYUKI UCHIDA, KEIJI KIMURA, TAKESHI KODAKA
and HIRONORI KASAHARA

Dept. of Electrical, Electronics and Computer Engineering, Waseda University

3-4-1 Ohkubo Shinjuku-ku, Tokyo 169-8555, Japan Tel: +81-3-5286-3371

E-mail: {uchida,kimura,kodaka,kasahara}@oscar.elec.waseda.ac.jp

With the advances in semiconductor integration technology, efficient use of transistors on a chip and scalable performance improvement have been demanded. To satisfy this demand, many researches on next generation microprocessor architectures and its software, especially compilers, have been performed. In these next generation microprocessor architectures, a single chip multiprocessor(SCM) using multigrain parallel processing, which hierarchically exploits different level of parallelism from the whole program, is one of the most promising architectures. This paper evaluates performance of the SCM architectures for multigrain parallel processing, using five application programs from SPEC2000fp and SPEC95fp. The evaluation shows that a four-processor cores SCM using multigrain parallel processing gives us 1.4 to 3.8 times larger speed up against a simple processor.

1 はじめに

半導体集積技術の進歩により、複数のプロセッサコアを1チップ上に集積したシングルチップマルチプロセッサ^{1)~6)}(SCM)が次世代のマイクロプロセッサとして注目を集めている。複数のプロセッサコアあるいはメモリを1チップ上に搭載することで、プロセッサコア間およびプロセッサコア-メモリ間の高スループット・低レイテンシデータ転送や、低マルチスレッディングオーバーヘッドが実現できる。現在、SCMアーキテクチャは活発に研究されており、MAJC¹⁾のようにマルチメディア向けに2基のVLIWコアを集積したもの、MP98²⁾のように1チップ上に4CPUを搭載しマルチスレッド処理に

よる高速化をしているもの、Power4⁷⁾のように高性能サーバ向けに2基のスーパースカラプロセッサコアがオンチップL2キャッシュを共有する構成のもの、またHydra, Multiscalar, Superthreadedのようなスーパースカラプロセッサコアを複数搭載し、スレッド投機実行をサポートするアーキテクチャで命令レベル並列性とスレッドレベル並列性を利用するもの^{8)~10)}などがある。

一方、筆者等は手続き型言語の並列処理を前提として、複数命令レベルでの並列処理である(近)細粒度並列処理¹¹⁾より大きな並列性を持つルーピタレーションレベルの中粒度並列処理¹²⁾及びサブルーチンあるいはループ、基本ブロック間の粗粒度

並列性^{13),14)}を階層的に組み合わせることで使用することにより高い実効性能を達成することができるマルチグレイン並列処理を提案している¹⁵⁾。このマルチグレイン並列処理をSCMに適用することにより、命令レベル並列性より飛躍的に大きな並列性を利用できより高い実効性能と今後の集積度向上に対してもスケラブルな性能向上が実現可能となる¹⁶⁾。

本論文では、実アプリケーションに対してマルチグレイン並列処理を適用し、OSCAR型SCMアーキテクチャの性能を評価したので、その結果について述べる。

以下、2節でマルチグレイン並列処理について、3節で本論文で評価するSCMについて説明し、4節でこれらのアーキテクチャにマルチグレイン並列処理を適用して評価した結果についてを述べる。

2 マルチグレイン並列処理

本節では、OSCAR型シングルチップマルチプロセッサで扱うマルチグレイン並列処理技術について述べる。

マルチグレイン並列処理¹⁷⁾とは、ループやサブルーチン等の粗粒度タスク間の並列処理を利用する粗粒度タスク並列処理(マクロデータフロー処理)、ループイタレーションレベルの並列処理である中粒度並列処理、基本ブロック内部のステートメントレベルの並列性を利用する近細粒度並列処理を階層的に組み合わせるプログラム全域にわたる並列処理を行なう手法である。

2.1 粗粒度タスク並列処理

(マクロデータフロー処理)

マクロデータフロー処理では、ソースとなるプログラムを疑似代入文ブロック(BPA)、繰り返しブロック(RB)、サブルーチンブロック(SB)の三種類の粗粒度タスク(マクロタスク(MT))に分割する。MT生成後、コンパイラはBPA, RB, SB, 等のMT間のコントロールフローとデータ依存を解析し、それらを表したマクロフローグラフ(MFG)¹⁸⁾を生成する。さらにMFGからMT間の並列性を最早実行可能条件解析¹⁸⁾により引きだし、その結果をマクロタスクグラフ(MTG)¹⁸⁾として表現する。その後、コンパイラはMTG上のMTをプロセッサ

あるいはプロセッサグループ(PG)に割り当てる。

2.2 中粒度並列処理(ループ並列処理)

PGに割り当てられたMTがDoall可能なRBである場合、このRBはPG内のプロセッサエレメント(PE)に対して、イタレーションレベルで割り当てられ並列実行される。

2.3 近細粒度並列処理

PGに割り当てられたMTが、BPAや中粒度並列処理を適用できずループボディ部に粗粒度並列性のないRBである場合、それらはステートメントレベルのタスクに分割され、PG内のPEにより並列処理される。

近細粒度並列処理においては、BPA内のステートメント、もしくは複数ステートメントから構成される疑似代入文を一つの近細粒度タスクとして定義する。近細粒度タスクは、PG内の各PEへスタティックスケジューリングされる。

OSCAR Fortran コンパイラ¹⁶⁾における近細粒度タスクスケジューリング手法として、データ転送オーバーヘッドを考慮し実行時間を最小化するヒューリスティックアルゴリズムであるCP/DT/MISF法、CP/ETF/MISF法、ETF/CP法、あるいはDT/CP法¹⁸⁾の4手法が実装されており、同一プログラムに対し4手法を適用し最良のスケジュールを選ぶ方式を取っている。

スケジューリング後、コンパイラはPEに割り当てられたタスクに対応する命令列を順番に並べ、データ転送命令や同期命令を必要な箇所に挿入することにより、各PEのマシンコードを生成する。近細粒度タスク間の同期にはバージョンナンバー法を用い、同期フラグの受信は受信側PEのビジーウェイトによって行なわれる。この時、OSCAR型マルチグレインSCMのように分散共有メモリ(DSM)を持つアーキテクチャでは、データ転送及び同期フラグのセットは受信側DSMへの直接ストアの形で行われ(3.1節)、受信側PEのビジーウェイトも自PE上のDSMへのビジーウェイトになるので、プロセッサ間相互接続網のバンド幅低下は起こらない。

3 シングルチップマルチプロセッサアーキテクチャ¹⁹⁾

本節では、今回評価を行ったシングルチップマルチプロセッサ (SCM) アーキテクチャおよびそのプロセッサコアについて述べる。

3.1 メモリアーキテクチャ

OSCAR 型マルチグレイン SCM のネットワークおよびメモリアーキテクチャは、図1のように CPU、データ転送ユニット (DTU)、ローカルプログラムメモリ (LPM)、ローカルデータメモリ (LDM)、および分散共有メモリ (DSM) を持つプロセッシングエレメント (PE) を相互接続網 (バス結合、クロスバ結合など) で接続し 1 チップ上に搭載したアーキテクチャである。今回の評価では、データ転送を CPU とオーバーラップして行える DTU は利用せず、また、PE 間相互結合網については今回は 3 本バスを利用した。

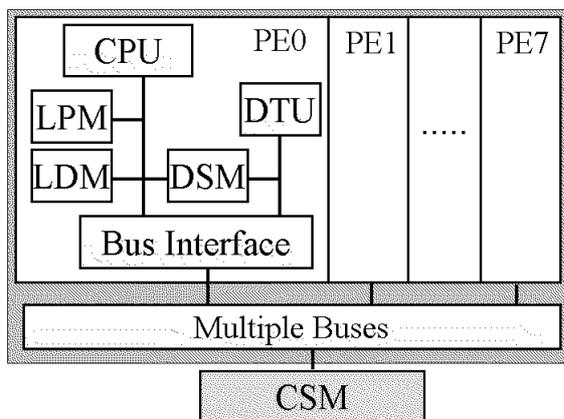


図 1: OSCAR 型 SCM アーキテクチャ

LPM は各々の CPU で実行するプログラムを格納し、今回の評価では 1 クロックでアクセスできるものとする。同様に、LDM は PE 固有のデータを保持するために使用し、その容量は 1PE あたり 1M バイト、LDM のアクセスレイテンシは 1 クロックとする。DSM は、自 PE と他 PE の双方から同時にアクセス可能なマルチポートメモリであり、異なる PE 上の近細粒度タスク間データ転送や同期、マクロデータフロー処理におけるダイナミックスケジューリング時のタスク割当におけるスケジューリ

ング情報の通知等に使用する。DSM の容量は 1PE あたり 16k バイトとし、自 PE からのアクセスレイテンシは 1 クロック、他 PE からのアクセスレイテンシは 4 クロックとする。さらに、チップ外部には集中共有メモリ (CSM) が接続され、各 PE で共有されるデータを格納する。この CSM のアクセスレイテンシは、20 クロックとする。

OSCAR 型 SCM では、これら 4 種類のメモリに対しコンパイラが最適なデータ配置を行なうことにより効率の良い並列処理を実現することを目指している。

3.2 プロセッサコアアーキテクチャ

今回の評価では、各 PE が持つ CPU は、SPARC V9 規格に準拠したスーパースカラプロセッサ Sun Microsystems 社の UltraSPARC II²⁰⁾ 命令セットをベースとし、バリア同期機構等用の特殊レジスタや特殊レジスタを操作するための命令を付加したアーキテクチャを用いている。

また、今回の評価における OSCAR 型 SCM では、プロセッサコアとして整数演算ユニット (IEU)、ロードストアユニット (LSU)、浮動小数点ユニット (FPU) を各 1 本持つ、UltraSPARC II パイプライン構成の 1 命令 in-order 発行のシンプルなアーキテクチャを用いている。

また、SCM とスーパースカラとの比較のため UltraSPARC II 相当のプロセッサコア (US-II) を用いたアーキテクチャについても評価を行っており、表1のようにプロセッサコアは、IEU を 2 本、LSU を 1 本、FPU を 2 本持つ、同時最大発行命令数 4in-order 発行で、命令バッファエントリ数は 12 としている。なお、シンプルなプロセッサを 4 基持った OSCAR 型 SCM と UltraSPARC II 相当のプロセッサコア 1 基持った OSCAR 型スーパースカラプロセッサのトランジスタ数はほぼ同一であると考えられる。

4 性能評価

ここでは、実アプリケーションに対してマルチグレイン並列処理を適用して評価した結果について述べる。1 命令発行の PE を複数搭載した SCM アーキテクチャと複数命令発行のスーパースカラプロセッサコアをクロックレベルの精密なシミュレータを用

表 2: プログラムの詳細

プログラム名	目的	データセット	入力の変更点
mgrid	Multi-grid 法	test	配列サイズ LMT を 7 → 3 ループ回転数 40 → 4
swim	shallow water equation の求解	test	配列サイズ M,N を 512 → 64 ループ回転数 ITMAX, MPRINT 10 → 2
apsi	気象予測	train	NUMBER OF TIME STEP を 720 → 3
hydro2d	Navie Stokes 方程式	test	パラメータ MP を 402 → 52 NP を 160 → 20
fpppp	量子化学計算	ref	NATOMS を 30 → 2

表 1: プロセッサコア仕様

	Simple	US-II Type
パイプライン段数	9	
同時命令発行数	1	4
IEU	1	2
FPU	1	2
LSU	1	1
命令発行タイプ	in-order	

いて評価を行う。

4.1 評価対象プログラム

今回の性能評価には, SPEC2000fp の mgrid, swim, SPEC95fp の apsi, hydro2d, fpppp のソースコードを用いた. mgrid は 3 次元空間の Multi-grid 法による求解プログラム, swim は Shallow water equation の求解プログラム, hydro2d は流体力学 Navie Stokes 方程式の求解プログラム, これらのプログラムの実行時間の大半は多重ループを処理するサブルーチンであり, ループ並列性が利用できる. apsi は気象予測のプログラムで Doall ループが並ぶプログラムであるが, ループ回転数及びループボディは小さいのでループ並列性の利用は難しい. fpppp は, 量子化学計算のプログラムで, 最も実行時間の大きなサブルーチンは巨大な基本ブロックを持つプログラムである.

今回の評価では, シミュレート時間短縮のために, 各プログラムのデータサイズ及びループ回転数を

表 2 のように縮小している.

また, fpppp では, OSCAR コンパイラにおけるスカラリネーミングにバグがあるため今回は手動でスカラリネーミングを行った.

4.2 評価結果

図 2 は PE 数を 1, 2, 4 とした場合, 及び Ultra-SPARC II 相当のスーパースカラプロセッサの場合での単一のシングルイシュープロセッサによる実行時間に対する速度向上率を示している. 今回の評価では変数, 配列などは集中共有メモリ (CSM) に置かれ, データローカライゼーションによるローカルメモリの活用はしていない.

mgrid, swim, hydro2d といったループ並列性がメインのアプリケーションでは, OSCAR 型スーパースカラプロセッサ (図 2 中 4issue x 1) では, 1.03 倍, 1.16 倍, 1.12 倍程度の速度向上率であったのに対して, OSCAR 型 SCM では, swim で, 2 プロセッサ時 1.98 倍, 4 プロセッサ時 3.85 倍の性能が得られ, 同様に mgrid で 2 プロセッサ時 1.89 倍, 4 プロセッサ時 3.37 倍, hydro2d で 2 プロセッサ時 1.50 倍, 4 プロセッサ時 2 倍の性能向上が得られた.

また, 従来のマルチプロセッサでは並列性の抽出が困難とされてきた²¹⁾ apsi, fpppp においてもマルチグレイン並列処理を適用することで性能向上が得られた.

apsi は Doall ループが並ぶプログラムであるが, 各々のループは回転数及び, ループボディが小さいため, 従来のマルチプロセッサシステムや他のシングルチップマルチプロセッサでは, 効率の良い並列

処理は得られていなかった。スーパースカラプロセッサでは、1.04 倍の性能向上に対して、OSCAR 型 SCM では同期及びデータ転送オーバーヘッド軽減により、2 プロセッサで 1.28 倍、4 プロセッサで 1.45 倍の性能が得られた。

fpppp では、スーパースカラプロセッサでは 1.03 倍の性能向上に留まっている。だが、fpppp の実行時間の大半を占めるサブルーチン TWLDRV 及び FPPPP にはループ並列性がないが近細粒度タスク（ステートメント）間には並列性がある。したがって OSCAR 型 SCM 上では、近細粒度並列性を利用することによって、2 プロセッサで 1.59 倍、4 プロセッサで 2.34 倍の性能向上が得られた。

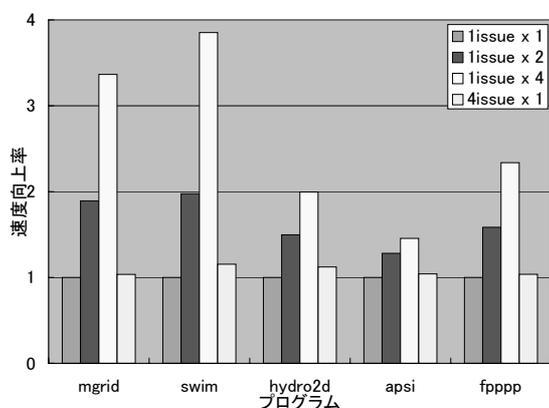


図 2: PE 数の違いによる速度向上率

5 まとめ

本論では、OSCAR 型シングルチップマルチプロセッサシステム上で SPEC2000fp の mgrid, swim, SPEC95fp の hydro2d, apsi, fpppp を用いたマルチグレイン並列処理の予備的な性能評価を行った。その結果、ループ並列性が多い mgrid, swim, hydro2d では、4issue のスーパースカラプロセッサはシングルイシュー単一プロセッサに対する速度向上が平均 1.10 倍の性能向上であったのに対して、4PE SCM では平均 3.07 倍の速度向上が得られた。

現在の SMP サーバでは並列性を有効に利用できない SPEC95fp の apsi, fpppp は、スーパースカラプロセッサではシングルイシュー単一プロセッサに対する速度向上が 1.04 倍、1.03 倍であったに対し、4PE SCM では、それぞれ、1.46 倍、2.34 倍の性能向

上を得ることができた。

これらの予備性能評価の結果より、OSCAR 型シングルチップマルチプロセッサ上でのマルチグレイン並列処理により、SPEC2000fp, SPEC95fp の 5 本のプログラムにおいて、同程度のトランジスタを用いるスーパースカラプロセッサと比べ、高い速度向上を達成できることが確かめられた。

今後の課題として、今回の評価では利用できなかった階層的並列性の利用、キャッシュなどローカルメモリの有効活用、タスクとデータ転送のオーバーラップ処理によるデータ転送オーバーヘッドの隠蔽などが挙げられる。

謝辞

本研究の一部は、STARC「自動並列化コンパイラ協調型シングルチップマルチプロセッサの研究」及び経済産業省ミレニアムプロジェクト「アドバンスト並列化コンパイラ」により行われた。本論文作成にあたり、有益なコメントをいただいた、STARC 平田雅規氏、宮田操氏、高橋宏政氏（富士通）、浅野滋博氏（東芝）、安川英樹氏（東芝）、倉田隆弘氏（ソニー）、高山秀一氏（松下）に感謝致します。

参考文献

- [1] Sun Microsystems Inc. :MAJC Home Page. <http://www.sun.com/microelectronics/MAJC/>. Sun Microsystems, Inc., 2000.
- [2] M. Edahiro, S. Matsushita, M. Yamashita, and N. Nishi. A single-chip multiprocessor for smart terminals. *IEEE MICRO*, Jul. 2000.
- [3] K.Okulotun, L.Hammond, and M.willey. Improving the performance of speculatively parallel applications on the hydra cmp. In *Proc. of the 1999 ACM Int'l Conf on Supercomputing*, June 1999.
- [4] J.G.Steffan and T.C.Mowry. The Potential for Using Thread-Level Data Speculation to Facilitate Automatic Parallelization. In *Proc. of the 4th Int'l Conf. on High-Performance Computer Architecture (HPCA-4)*, Feb 1998.

- [5] R.Barua, W.Lee, S.Amarasinghe, and A.Agarwal. Maps:a compiler-managed memory system for raw machines. In *Proc. of ISCA-26*, June 1999.
- [6] 木村, 尾形, 岡本, 笠原. シングルチップマルチプロセッサ上での近細粒度並列. *情報処理学会論文誌*, 40(5), May. 1999.
- [7] Joel M. Tendler, Steve Dodson, Steve Fields, Hung Le, and Balaram Sinharoy. *POWER4 System Microarchitecture*, October 2001.
- [8] L. Hammond, B. Hubbert, M. Siu, M. K. Prabhu, M. Chen, and K. Olukotun. The stanford HYDRA CMP. *IEEE MICRO*, 19(2), 1999.
- [9] T.N.Vijaykumar and G.S.Sohi. Task Selection for a Multiscalar Processor. In *31th Int'l Conf. on Microarchitecture (MICRO-31)*, Nov-Dec 1998.
- [10] J.-Y. Tsai, Z. Jiang, E. Ness, and P.-C. Yew. Performance study of a concurrent multi-threaded processor. In *Proc.4th Int'l Conf. on HPCA-4*, Fec 1998.
- [11] 笠原. マルチプロセッサシステム上での近細粒度並列処理. *情報処理*, 37(7):651-661, Jul 1996.
- [12] Padua and Wolfe. Advanced compiler optimization for super computers. *C.ACM*, 29(12):1184-1201, 1996.
- [13] 笠原, 合田, 吉田, 岡本, 本多. Fortran マクロデータフロー処理のマクロタスク生成手法. *信学論*, J75-D-I(8):511-525, 1992.
- [14] 本多, 合田, 岡本, 笠原. Fortran プログラム粗粒度タスクの oscar における並列実行方式. *信学論 (D-I)*, J75-D-I(8):526-535, 1992.
- [15] Kasahara, Honda, and Narita. A multigrain parallelizing compilation scheme for oscar. In *Proc.4th Workshop on Lang. And Compilers for Parallel Computing*, Aug 1991.
- [16] 笠原, 尾形等. マルチグレイン並列化コンパイラとそのアーキテクチャ支援. *信学技報*, IDC98-10, CPSY98-10, FTS98-10, pages 71-76, 1998.
- [17] H. Kasahara, M. Obata, and K. Ishizaka. Automatic coarse grain task parallel processing on smp using openmp. *Proc. of 13th International Workshop on Languages and Compilers for Parallel Computing (LCPC'00)*, Aug. 2000.
- [18] 笠原. 並列処理技術. コロナ社, 1991.
- [19] 木村, 加藤, 笠原. 近細粒度並列処理用シングルチップマルチプロセッサにおけるプロセッサコアの評価. *情報処理学会論文誌*, 42(4), Apr. 2001.
- [20] Sun Microelectronics. *UltraSPARCTM User's Manual*, Jul. 1997.
- [21] M.W. Hall, J.M. Anderson, S.P. Amarasinghe, B.R. Murphy S.-W Liao, E. Bugnion, and M. S.Lam. Maximizing multiprocessor performance with the suif compiler. *IEEE Computer*, pages 84-89, Dec. 1996.