

OSCARチップマルチプロセッサ上での MPEG2エンコーディングの並列処理

小高 剛[†] 中野 啓史[†]

木村 啓二^{††} 笠原 博徳[†]

最近の携帯電話, PDA などのモバイル端末では, 静止画像, 動画, 音声処理など様々なマルチメディアアプリケーションの処理が必要となっている. このためマルチメディアアプリケーションを効率良く処理できる低コスト, 低消費電力かつ高性能なプロセッサの開発が望まれている. これらの要求を満たしつつマルチメディアアプリケーションを効率良く処理するプロセッサとして, 複数のプロセッサコアを 1 チップ上に搭載したチップマルチプロセッサアーキテクチャが命令レベル以外の粗粒度タスク並列性, 中粒度ループ並列性など複数レベル並列性も自然に引き出すことができ, 集積度向上に対しスケーラブルな性能向上が得られるアーキテクチャとして注目されている. しかしながら, チップマルチプロセッサアーキテクチャ上で効率の良い処理を行なうには, アプリケーションの特性を解析しプログラムを適切な粒度のタスクに分割し, それらをバランス良く CPU に配置する並列化技術が不可欠である. 本論文では, チップマルチプロセッサ上におけるマルチメディアアプリケーションの一例として, MPEG2 エンコーディングの並列性を抽出しその評価を行なう.

Parallel Processing on MPEG2 Encoding for OSCAR Chip Multiprocessor

TAKESHI KODAKA[†], HIROHUMI NAKANO[†], KEIJI KIMURA^{††} and HIRONORI KASAHARA[†]

Recently, multimedia applications with visual and sound processing are popular on mobile phones and PDAs. To satisfy the needs for efficient multimedia processing, development of low cost, low power consumption and high performance processors for multimedia applications has been expected. Chip multiprocessor architectures which allows us to attain scalability using coarse grain level parallelism and loop level parallelism in addition to instruction level parallelism are attracting much attention. However, to realize efficient processing on chip multiprocessor architectures, parallel processing techniques such as decomposing a program into adequate tasks considering characteristics of a program and assigning these tasks onto processors are essential. This paper describes a parallel processing scheme for MPEG2 encoding for a chip multiprocessor and its performance.

1 はじめに

モバイル端末上での静止画像, 動画, 音声処理などのマルチメディア処理の要求が高まっている. そのため, 各種マルチメディアアプリケーションをモバイル端末上で快適に利用するための, 低消費電力で価格性能比の優れたプロセッサの開発が望まれている.

このようなニーズに対するアプローチとして, 富士通 FR500¹⁾, 日立 SH4²⁾ などのようなマルチメディア用命令セットを追加し処理能力を向上させるものがある. これらのプロセッサでは, スーパースカラアーキテクチャや VLIW アーキテクチャによる命令レベル並列性の利用や, SIMD 命令によるマルチメディア処理で多用される内積演算やベクトル変換演算など同一命令が連続する処理の高速化といった, 主に命令レベル並列性を利用する方式がとられている.

別のアプローチとして, NEC MP98³⁾, 東芝 MeP⁴⁾ のような 1 チップ上にプロセッサコアを複数搭載したチップマルチプロセッサアーキテクチャを用いる方法がとられている. チップマルチプロセッサアーキテクチャは, 従来の SIMD, VLIW などを用いられていた命令レベル並列性に加え, ループイタレーション間の中粒度並列性, 基本ブロック, ループ, サブルーチン間の粗粒度タスク並列性も利用可能であり, 集積度向上に対してスケーラブルな性能向上を目指す上で有望な方式と考えられている. ただし, チップマルチプロセッサ上で効率の良い処理を行なうには, アプリケーションからの並列性抽出やデータ配置の最適化など高度な並列処理の知識, あるいはそれらをサポートする強力なコンパイラが必要となる.

本論文では, チップマルチプロセッサ上におけるマルチメディアアプリケーションの一例として, MPEG2 エンコーディングの並列性の抽出および並列性評価を行なう.

以下, 2 節でマルチグレイン並列処理, 3 節で MPEG2 エンコーディングの並列性の抽出, 4 節で本論文で

[†] 早稲田大学理工学部コンピュータ・ネットワーク工学科
Dept. of Computer Science, Waseda University

^{††} 早稲田大学理工学総合研究センター
Advanced Research Institute for Science and Engineering,
Waseda University

対象とする OSCAR チップマルチプロセッサアーキテクチャについて述べ、5 節で MPEG2 の並列性評価を行なう。

2 マルチグレイン並列処理

ここでは、OSCAR 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)⁵⁾、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⁶⁾ を階層的に組み合わせてプログラム全域に渡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⁵⁾ (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。ここで、BPA は基本的には通常の基本ブロックであるが、並列性抽出のために単一の基本ブロックを複数に分割したり、逆に複数の基本ブロックを融合して一つの BPA を生成する。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) 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により引きだし、その結果をマクロタスクグラフ (MTG) 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複数のプロセッサエレメント (PE) をグループ化したプロセッサグループ (PG) に割り当てる。なお、このグループ化はプログラム中の各部分の並列性に依りソフトウェア的に行なわれる仮想的なものでハードウェア的なグループ化とは異なる。

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) 上で、イタレーションレベル並列実行される。

2.3 近細粒度並列処理⁶⁾

PG に割り当てられた MT が、BPA や中粒度並列処理あるいはループボディ部に粗粒度並列処理を適用できない 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、基本的に BPA 内のステートメント、もしくは IF-THEN-ELSE 等で囲まれた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。その後、近細粒度タスク間のデータ依存を解析してタスクグラフを作成し、このタスクグラフ上のタスクを、データ転送・同期オーバーヘッドを考慮して実行時間を最小化できるように各 PE にスタティックにスケジューリングする。スケジューリング後、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し各 PE 毎に異なるマシンコードを生成する。

3 MPEG2 エンコーディングの並列処理

ここでは、本論文で対象とする MPEG2 エンコーディングアルゴリズムについて述べた後、MPEG2 エンコーディングの並列性について述べる。

本論文では MPEG2 エンコーディングアルゴリズムの参照実装として MediaBench⁷⁾ に収録されている “mpeg2encode” を用いる。また、主なエンコーディングオプションは表 1 に示す MediaBench における評価オプションを用い本論文での使用アルゴリズムとする。ただし、画像サイズは QCIF (176×144) とした。

MPEG2 エンコードのデータ構造は図 1 に示すように階層的である。MPEG ビデオの全シーケンスは、複数の GOP (Group of Picture) で構成され、さらに各 GOP は複数のピクチャ (フレーム) で構

表 1: MPEG2 エンコーディングオプション

# of frames in GOP	15
I/P frame distance	3 (I B1 B2 P...)
Picture type	frame picture
Aspect ratio information	4:3
Frame rate	30 frames/sec.
Bit rate	5000000.0 bits/sec.
Profile	Main
Level	Main
chroma format	4:2:0
video format	NTSC
vbv buffer size	112kbit
progressive sequence	false
intra dc precision	8bit
quantization scale type	non-linear
entropy scan type	Zig-Zag scan
search window size	P:11x11 B1:3x3(forw),7x7(back) B2:7x7(forw),3x3(back)

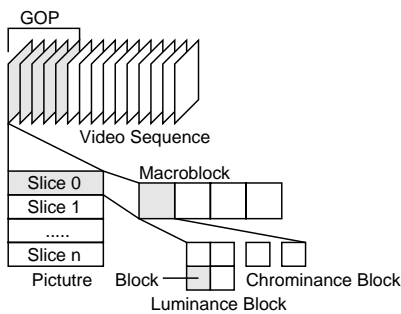


図 1: MPEG2 データ構造

成されている。それぞれのピクチャは、いくつかのマクロブロックから構成されるスライスからなり、さらにマクロブロックは6つのブロック(4つのルミナンスブロックと2つのクロミナンスブロック)から構成される。また、1ブロックは8×8ピクセルで構成される。

MPEG2 エンコードのブロック図を図2に示す。MPEG2 エンコードは、7つのステージからなり動き推定(motion estimation)、動き予測(predict)、DCT 変換構造選択(dct type estimation)、データ変換(transform)、ビットストリーム出力処理(putpict)、逆量子化(iquantize)、逆データ変換(itransform)からなる。各ステージの概要を以下に示す。

動き推定 エンコード対象フレーム画像が参照画像(エンコード対象画像より過去の画像または未来の画像)からどれだけ移動しているかを表す動きベクトルの探索を行なう。また、符合

化の際に前向き予測、後向き予測、双方向予測のどれを利用したかという符合化モードを決定する。

動き予測 動き推定で求めた動きベクトル、符合化モードに基づいて符合化対象ピクチャを生成する。

DCT 変換構造選択 符合化対象ピクチャに対しマクロブロックレベルでフレーム構造 DCT を適用するかフィールド構造 DCT を適用するか決定する。

データ変換 符合化対象ピクチャに DCT 変換構造選択で決定した DCT 変換構造に基づき離散コサイン変換(DCT)を適用する。

ビットストリーム出力 DCTを適用したピクチャをビットストリームとして出力する。ここで、ビットレート制御、量子化係数の決定、量子化、各種ヘッダ送出などを行なう。

逆量子化 量子化適用後のピクチャを逆量子化を行なって復元する。

逆データ変換 逆量子化後のピクチャに対し逆 DCT を適用しピクチャを復号化する。

各ステージ毎にピクチャ全体を構成するマクロブロックに対して処理を行なう。

“mpeg2encode”における処理時間割合を Sun Ultra80 (UltraSPARC-II 450MHz ×4 ただしシングルプロセッサで処理, L2 キャッシュ4M バイト)上でコンパイルオプション“-pg -O2”にてビルドしたバイナリを、gprof を用いて計測したプロファイル結果を図3に示す。プロファイル結果より、MPEG2 エンコーディングのほとんどは動き推定(motion estimation)であることがわかる。

MPEG2 エンコードの並列性抽出に際し、各ステージにおける利用データ範囲に注目すると、まず、ビットストリーム出力では、出力データがビット単位のデータとなり逐次出力が必要となるため並列処理の適用が難しい。しかし、その他の動き推定、動き予測、DCT 変換構造選択、データ変換、逆量子化、及び逆データ変換では処理対象がマクロブロックであり、扱うデータ範囲がマクロブロックに限られているため各マクロブロックレベル処理の間ではデータ依存が発生しない。そのため、基本的にはマ

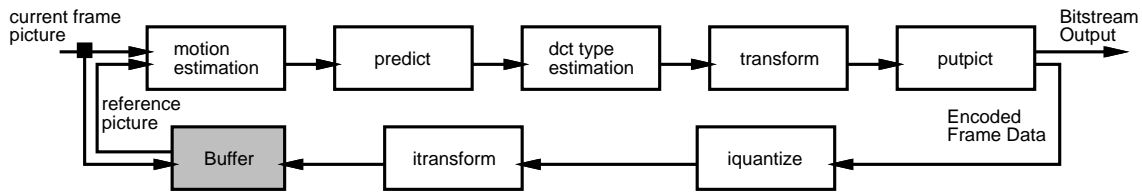


図 2: MPEG2 エンコード ブロック図

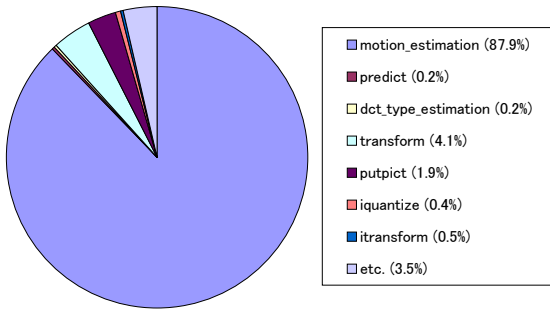


図 3: MPEG2 処理時間割合

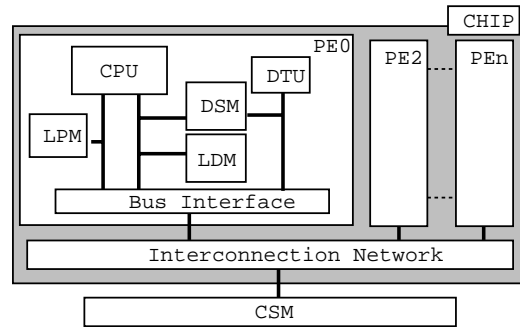


図 4: OSCAR CMP アーキテクチャ

クロックレベルでのループ並列処理可能なプログラムとなっている。本論文では、各ステージの処理がマクロブロックレベル処理になるようにループを分割しマクロブロックレベル処理を粗粒度タスクとして定義する。このように分割を行なった粗粒度タスク間ではデータ依存が発生しないため粗粒度タスク間で並列処理可能となる。

4 OSCAR チップマルチプロセッサアーキテクチャ⁶⁾

ここでは、OSCAR チップマルチプロセッサ (CMP) アーキテクチャおよびそのプロセッサコアアーキテクチャについて述べる。

4.1 メモリアーキテクチャ

OSCAR CMP のネットワークおよびメモリアーキテクチャは、図 4 に示すように CPU、データ転送を CPU の処理とオーバーラップして行なえるデータ転送ユニット (DTU)、各々の CPU で実行するプログラムを格納するローカルプログラムメモリ (LPM)、PE 固有のデータを保持するローカルデー

タメモリ (LDM)、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の分散共有メモリ (DSM) を持つプロセッサエレメント (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、DTU についてはオーバーラップデータ転送スケジューリングアルゴリズムが未実装のため利用していない。また、本評価では PE 間相互結合網として 3 本バスを利用する。

本論文では各メモリサイズ、アクセスレイテンシは LDM の容量は 256K バイトとしアクセスレイテンシは 1 クロック、同様に DSM の容量は 16K バイトで自 PE 内のローカルアクセスには 1 クロック、他 PE へのリモートアクセスには 4 クロックかかるとし、チップ外 CSM は十分な容量が確保されているものとしアクセスレイテンシは 20 クロックとした。

OSCAR CMP では、これら 4 種類のメモリに対し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行なうことができる。

各 PE が持つ CPU は、SPARC V9 規格に準拠したプロセッサである Sun Microsystems 社の UltraSPARC II[®] のパイプライン構成をベースとし、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プロセッサである。

今回の評価で用いる OSCAR CMP のプロセッサコアは、整数演算ユニット (IEU) を 1 本、ロードストアユニット (LSU) を 1 本、浮動小数点ユニット (FPU) を 1 本持つシングルイシューのシンプルな構成とした。

5 性能評価

ここでは、MPEG2 エンコーディングの各処理に対して 3 節で述べた並列性の抽出を行ない、OSCAR チップマルチプロセッサ (CMP) 上で評価した結果について述べる。なお、性能評価にはクロックレベルシミュレータを用いた。

評価に用いるプログラムは、MediaBench に収録されている MPEG2 エンコーディングプログラム “mpeg2encode” を参照実装としたプログラムを用い、MPEG2 エンコーディング処理の中心部分である動き推定、動き予測、DCT 変換構造選択、データ変換、逆量子化、逆データ変換を評価対象とする。各ステージのシーケンシャルソースプログラムを作成し、ループ分割および粗粒度タスク並列性抽出を OSCAR マルチグレイン自動並列化コンパイラを用いて行ない、マクロタスクのスタティックタスクスケジューリングを適用して OSCAR CMP 用バイナリコードを生成した。ただし、データローカリティ最適化、およびデータ転送ユニットによるデータ転送オーバーヘッド隠蔽技術は現時点でマルチメディアアプリケーションへの適用ができていないため本評価では利用しない。入力画像は、MediaBench で用いられる入力画像 (compo.tar.gz 中の rec*. [YUV]) をシミュレーション時間短縮のために 176×144 ピクセルに縮小した画像を用いるが、評価対象とする各ステージへの入力データは内部データで MediaBench より提供されていないので “mpeg2encode” を改編しプログラム中から抽出したデータを用いる。

OSCAR CMP 上での評価結果を逐次実行時間すなわち 1PG1PE 時の処理時間に対する速度向上率として図 5 に示す。図中横軸は左から動き推定 (motion estimation)、動き予測 (predict)、DCT 変換構造選択 (dct type estimation)、データ変換 (transform)、逆量子化 (iquantize)、逆データ変換 (itransform) の各ステージを示す。各処理の棒グラフは左から逐次実行の “1PG1PE”、2 プロセッサを利用した粗粒度タスク並列処理の “2PG1PE”、

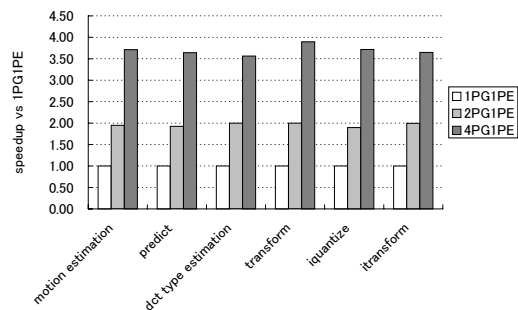


図 5: MPEG2 エンコーディング評価結果

4 プロセッサを利用した粗粒度タスク並列処理の “4PG1PE” である。

まず、使用プロセッサ数が 2 の場合の “2PG1PE” では逐次実行時間に対してそれぞれ動き推定で 1.95 倍、動き予測で 1.93 倍、DCT 変換構造選択では 2.00 倍、データ変換で 2.00 倍、逆量子化で 1.90 倍、逆データ変換で 2.00 倍の速度向上率が得られほぼリニアな性能向上が見られた。この結果は MPEG2 エンコードではマクロブロックレベルの粗粒度タスク並列性が高いことを示している。しかし、マクロブロックレベルの粗粒度タスク間でのデータ依存関係がすべて同じにもかかわらず、動き推定、動き予測、逆量子化では他のステージほど高い速度向上率が得られていない。この理由を図 6 の動き補償の実行トレースを用いて説明する。図 6 は、PG0、PG1 の各 PG における粗粒度タスク (マクロタスク (MT)) の実行トレースを示しており、横軸が時間軸、“MTn” が MT の実行、網かけが idle の状態を表している。動き推定、動き予測、逆量子化では入力データが条件の評価対象となっている条件分岐がある。また、分岐先ではその方向により演算内容が異なるため演算コストの不均衡が生じ、図 6 のように各 PG 間で実行時間が偏りこのような結果となった。

次に、使用プロセッサ数が 4 の場合の “4PG1PE” では逐次実行時間に対してそれぞれ動き推定で 3.71 倍、動き予測で 3.64 倍、DCT 変換構造選択で 3.56 倍、データ変換で 3.90 倍、逆量子化で 3.72 倍、逆データ変換で 3.65 倍の速度向上率が得られ、高い粗粒度タスク並列処理性能を得た。しかし、以下の点を改善することによりさらに性能向上可能である。本論文の入力画像は、QCIF (176×144) としたため、本評価ではマクロブロックレベル処理の粗粒度タスク数は 110 個となる。そのため、使用プロセッ

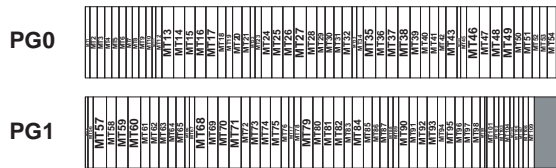


図 6: 動き補償の実行トレース

サ数が 4 では各プロセッサで処理を行なうタスク数が各 PG で 27, 27, 28, 28 と均等ではなくなってしまふ．そのため負荷がプロセッサ間で不均衡となりパフォーマンスが下がっている．

このような負荷の不均衡を改善するアプローチとしてマルチグレイン並列処理が挙げられる．マルチグレイン並列処理を用いることにより粗粒度タスクの内側でステートメントレベルの近細粒度並列処理などの並列処理が階層的に行なわれるようになり，その結果粗粒度タスクの負荷が PG 内 PE にて分散されるため負荷バランスが向上する．以上が，MPEG2 エンコードの各ステージに対する並列性の評価である．今後，このマルチグレイン並列処理を行なう予定である．

6 まとめ

本論文では，MPEG2 エンコーディングの粗粒度タスク並列性の評価を行なった．その結果 OSCAR チップマルチプロセッサアーキテクチャ上にて逐次実行時間に対して 4 プロセッサを用いた粗粒度並列処理において動き推定で 3.71 倍，動き予測で 3.64 倍，DCT 変換構造選択で 3.56 倍，データ変換で 3.90 倍，逆量子化で 3.72 倍，逆データ変換で 3.65 倍の速度向上率が得られ，MPEG2 エンコーディングの並列性が確認できた．

今後の課題として，MPEG2 エンコードへのマクロブロック処理間の粗粒度並列性とマクロブロック処理内での近細粒度並列性の利用したマルチグレイン並列性の適用，データローカライズ手法の適用，及びその評価が挙げられる．

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」及び早稲田大学理工総研プロジェクト研究「自動並列化協調型チップマルチプロセッサ」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宮田操氏（STARC），高橋宏政氏（富士通研），高山秀一氏（松下），安川英樹氏（東芝），倉田隆弘氏（ソニー）に感謝致します．

参考文献

- [1] A. Suga and K. Matsunami: Introducing the FR500 Embedded Microprocessor, *IEEE MICRO*, Vol. 20, No. 4 (2000).
- [2] F. Arakawa, O. Nishi, K. Uchiyama and N. Nakagawa: SH4 RISC Multimedia Microprocessor, *IEEE MICRO*, Vol. 18, No. 2 (1998).
- [3] M. Edahiro, S. Matsushita, M. Yamashita and N. Nishi: A Single-Chip Multiprocessor for Smart Terminals, *IEEE MICRO*, Vol. 20, No. 4 (2000).
- [4] 道中秀治, ほか: A single-Chip MPEG-2 Codec Based on Customizable Media Microprocessor, ICD2002-20 (2002).
- [5] H. Kasahara, M. Obata and K. Ishizaka: Automatic Coarse Grain Task Parallel Processing on SMP using OpenMP, *Proc. 12th Workshop on Languages and Compilers for Parallel Computing* (2000).
- [6] 木村啓二, 加藤孝幸, 笠原博徳: 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*, Vol. 42, No. 4 (2001).
- [7] C. Lee, M. Potkonjak and W. H. Mangione-Smith: MediaBench: A Tool for Evaluating and Synthesizing Multimedia and Communications Systems, *30th International Symposium on Microarchitecture (MICRO-30)* (1997).
- [8] Sun Microelectronics: *UltraSPARCTM User's Manual* (1997).