

マルチコア上でのマルチメディアアプリケーションの自動並列化

宮本孝道[†] 浅香沙織[†] 鎌倉信仁[†]
山内宏真[†] 間瀬正啓[†] 白子準[†]
中野啓史[†] 木村啓二[†] 笠原博徳[†]

半導体集積度向上に伴う消費電力の増大，プロセッサ動作クロック周波数向上の鈍化，ハードウェア・ソフトウェア開発期間の増大といった問題に対処すべく，一つのチップ上に複数のプロセッサコアを集積するマルチコアプロセッサが注目を集めている．また，携帯電話，デジタルTV，ゲーム等の情報家電上ではコーデック，グラフィックス等のマルチメディア処理の高速化が非常に重要となっている．本稿ではこのようなマルチメディアアプリケーションのマルチコアプロセッサ上での並列化について述べる．特に動画像処理の例としてMPEG2エンコード，MPEG2デコード，音声処理の例としてMP3エンコード，静止画処理の例としてJPEG2000エンコードのOSCARマルチグレイン自動並列化コンパイラを用いたマルチコア上での並列処理について述べる．各マルチメディア処理の並列処理をOSCARマルチコアプロセッサ，IBM p5 550Q Power5+ 2 コア集積マルチコアプロセッサベース SMP サーバ上で評価を行った．OSCARマルチコアプロセッサ上では，逐次処理に対して，提案するローカルメモリ最適化，データ転送最適化を伴う並列化手法により4プロセッサでの並列処理では，MPEG2エンコードで3.81倍，MPEG2デコードで3.04倍，MP3エンコードで3.09倍，JPEG 2000エンコードで3.79倍の速度向上が得られた．IBM p5 550Q上では，逐次処理に対してOSCARコンパイラによる8プロセッサでの並列処理では，MPEG2エンコードで5.19倍，MPEG2デコードで5.12倍，MP3エンコードで3.69倍，JPEG 2000エンコードで4.32倍の速度向上が得られた．

Automatic Parallelization for Multimedia Applications on Multicore Processors

TAKAMICHI MIYAMOTO[†] , SAORI ASAKA[†] , NOBUHITO KAMAKURA[†] ,
HIROMASA YAMAUCHI[†] , MASAYOSHI MASE[†] , JUN SHIRAKO[†] ,
HIROFUMI NAKANO[†] , KEIJI KIMURA[†] and HIRONORI KASAHARA[†]

Multicore processors have attracted much attention to handle the increase of power consumption along with the increase of integration degree of semiconductor devices, the slowdown of improvement of processor clocks, and the increase of hardware/software developing period. Also, speeding up multimedia applications is required with the progress of the consumer electronics like mobile phones, digital TV and games. This paper describes parallelization methods of multimedia applications on the multicore processors. Especially in this paper, MPEG2 encoding and MPEG2 decoding are selected as examples of video sequence processing, MP3 encoding is selected as an example of audio processing, JPEG 2000 encoding is selected as an example of picture processing. OSCAR multigrain parallelizing compiler automatically parallelizes these media applications. This paper evaluates parallel processing performances of these multimedia applications on the OSCAR multicore processor, and the IBM p5 550Q Power5+ 8 processors SMP server. On the OSCAR multicore processor, the parallel execution with the proposed method of managing local memory and optimizing data transfer using 4 processors, gives us 3.81 times speedup for MPEG2 encoding, 3.04 times speedup for MPEG2 decoding, 3.09 times speedup for MP3 encoding, 3.79 times speedup for JPEG 2000 encoding against the sequential execution. On the IBM p5 550Q Power5+ 8 processors server, the parallel execution using 8 processors gives us 5.19 times speedup for MPEG2 encoding, 5.12 times speedup for MPEG2 decoding, 3.69 times speedup for MP3 encoding, 4.32 times speedup for JPEG 2000 encoding against the sequential execution.

1 はじめに

従来，マイクロプロセッサの性能向上の牽引力になっていた命令レベル並列性の利用と周波数の向上は半導体集積度の向上と共に，並列性抽出の限界，消費電力の増大等が顕在化し，今後の性能向上が難しくなっている．これらの問題に対処するためマルチコアプロセッサが注目を集めている．SCE/IBM/東芝の Cell¹⁾，NEC/ARM

の MPCore，MP211²⁾，富士通 FR-V³⁾，パナソニック Uniphier，ルネサステクノロジ SH-X3⁴⁾ といった情報家電向け組込み用マルチコアや，PC，サーバ向けのインテル Dual コア Xeon⁵⁾ や Core 2 Duo，AMD の Dual/Quad コア Opteron⁶⁾，ワークステーション，ハイエンドサーバ用に開発された Sun SPARC T1，T2，そして IBM Power4，5，5+⁷⁾ などが例として挙げられる．

マルチコアプロセッサでは複数のプロセッサコアを一つのチップ上に集積することにより，プロセッサコア間で命令レベル並列性よりも並列性の大きいループレベル，タスクレベルのより粒度の粗い並列処理の実現が可能となる．また，各プロセッサコアをコンパイラ，OSとも協調し低周波数低電圧で動作させ，適切に並列処理することで，高性能化，低消費電力化が実現可能なアーキテクチャ

[†]早稲田大学理工学術院コンピュータ・ネットワーク工学科
〒169-8555 東京都新宿区大久保 3-4-1 Tel: 03-5286-3371

[†]Department of Computer Science, School of Science and Engineering, Waseda University 3-4-1 Ohkubo, Shinjuku-ku, Tokyo, Japan 169-8555 Tel: +81-3-5286-3371

としても期待されている。一方で、マルチコアでも従来より問題となっていたメモリウォールの問題は残り、キャッシュやローカルメモリ等のチップ内の近接メモリの有効利用が必要である。特にローカルメモリを持つアーキテクチャではローカルメモリへのデータ配置とメモリ間のデータ転送命令を明示的に行い、プログラムの並列性を十分引き出すためにはデータ転送オーバーヘッドの隠蔽が重要となる。データ転送最適化手法として Direct Memory Access Controller(DMAC) を用いたデータブロード・ポストストア (PLPS) 手法^{8)~11)}によりプログラムの演算処理とデータ転送のオーバーラップによるデータ転送オーバーヘッド隠蔽手法が実現されている。従来、DMACの利用手法として、マルチメディアプロセッサに対してのプログラミングとして off-chip memory から on-chip memory への転送¹²⁾を行う手法がよく用いられている。筆者等は従来より自動マルチグレイン並列化コンパイラとの協調動作により実効性能が高く価格性能比の良いコンピュータシステムの実現を目指す OSCAR マルチコアアーキテクチャを提案している^{13),14)}。この OSCAR マルチコアアーキテクチャは、全てのプロセッサコアがアクセスできるオンチップあるいはオフチップ共有メモリ (CSM) の他に、プロセッサコアのプライベートデータを格納するローカルデータメモリ (LDM) とプロセッサコア間の同期やデータ転送に使用する 2 ポートメモリ構成の分散共有メモリ (DSM)、そして、プロセッサコアと非同期に動作可能なデータ転送ユニット (DTU) を持つ。

増大するデータを処理するマルチメディア処理において命令レベルよりさらに粒度の大きな並列処理粒度を利用するためには、適切なデータ分割、プロセッサ近傍メモリへのデータ配置、メモリ間のデータ転送最適化を適用することによる処理の高速化が重要となる。本稿では特に、動画像処理として MPEG2 エンコード、MPEG2 デコード、音声処理として MP3 エンコード、静止画処理として JPEG 2000 エンコードを対象とした並列化を行う。

本稿ではマルチメディア処理のマルチコアプロセッサ上での並列処理を、OSCAR マルチグレイン自動並列化コンパイラにより、プログラムを大域的に解析し、各マルチメディア処理でのワーキングセットをローカルメモリ上へ配置し、プログラム全体のデータ転送をプロセッサコアと非同期に動作可能な DTU を用いた隠蔽を行い、OSCAR マルチコアプロセッサ上において評価を行う。また、IBM p5 550Q Power5+ 2 コア集積マルチコアプロセッサ搭載 SMP サーバ上で評価を行う。

本稿の構成を以下に示す。第 2 章では対象とするマルチコアアーキテクチャについて述べる。第 3 章では本稿で扱う粗粒度タスク並列処理手法について述べる。第 4 章では粗粒度タスク並列処理を用いたマルチメディアアプリケーション毎の並列化手法について述べる。第 5 章ではマルチメディアアプリケーションの OSCAR マルチコア上、IBM p5 550Q 上での性能評価結果について述べる。第 6 章で本稿のまとめを述べる。

2 対象マルチコアアーキテクチャ

本章では、今回マルチメディア処理を行う対象とする OSCAR マルチコアアーキテクチャと市販マルチコアベース SMP マシン IBM p5 550Q について簡単に説明する。

2.1 OSCAR マルチコアアーキテクチャ

OSCAR マルチコアアーキテクチャは自動マルチグレイン並列化コンパイラとの協調動作により、実効性能が高く価格性能比の良いコンピュータシステムの実現を目指したアーキテクチャである。

OSCAR マルチコアアーキテクチャを図 1 に示す。OSCAR マルチコアは 1 つのチップ上に複数のプロセッサエレメント (PE) を持つ。各 PE は単純な一命令発行の in-order プロセッサコア、プロセッサプライベートなデータを保持する 1 ポートのローカルデータメモリ (LDM)、共有データや同期変数を保持する 2 ポートの分散共有メ

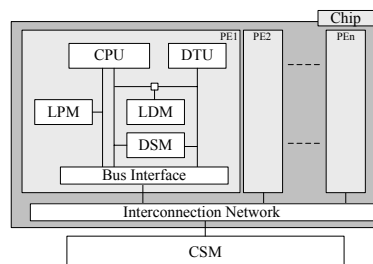


図 1: OSCAR マルチコアアーキテクチャ

モリ (DSM)、プログラムコードを保持するローカルプログラムメモリ (LPM)、そして CPU と非同期にバースト転送が可能なデータ転送ユニット (DTU) を持つ。チップ上の全ての PE はバスやクロスバといった Interconnection Network によってオンチップあるいはオフチップ集中共有メモリ (CSM) に接続されるが本稿ではオフチップ CSM を仮定し評価を行う。今回、プロセッサコアの周波数は組み込み用途を想定した 400MHz とし、各メモリへのアクセスレイテンシを CSM は 24 クロック、LDM は 1 クロック、ローカルの DSM は 1 クロック、リモートの DSM は 4 クロック、そして LPM は 1 クロックと設定した。チップ内のメモリレイテンシの算出には ITRS 20003¹⁵⁾ および CACTI¹⁶⁾ を、チップ外のレイテンシには Elpida Memory 社のデータシート^{17),18)} をそれぞれ利用した。

LDM または DSM サイズは、今回評価に使用したメディアアプリケーションのデータワーキングセットが載るサイズである 256Kbyte とした。

2.1.1 データ転送ユニット (DTU)

今回の評価で仮定する OSCAR マルチコア上のデータ転送ユニット (DTU) について説明する。OSCAR DTU では連続領域転送、転送元、転送先でストライド長が異なるストライド転送、SCATTER、GATHER 転送が可能であり DTU 命令はコンパイラが自動的に生成する。DTU の起動には二種類の方法がある。一つはコンパイラが生成した上述のパラメータをローカルメモリ上に設定し、実行時に転送パラメータの先頭アドレスを DTU に通知し、DTU を駆動する方法である。このとき、複数のパラメータをローカルメモリ上の連続する領域に設定しておけば、パラメータチェーンが形成され、CPU による一度の駆動で複数の領域を転送することが可能となっている。もう一つは CPU が転送パラメータ値を直接 DTU のレジスタに設定し、駆動する方法である。

今回、DTU のバースト幅は 64byte と設定した。

2.2 IBM p5 550Q SMP サーバ

IBM p5 550Q は Power5+ 2 コア集積マルチコアプロセッサを 4 つ搭載した 8 プロセッサ SMP サーバであり、1.5GHz の周波数、32KB の L1-D キャッシュ、64KB の L1-I キャッシュを持つ Power5+ を 2 コア集積チップに対して 1.9MB の L2 キャッシュ、36MB の L3 キャッシュを搭載している。提案する並列化方式がローカルメモリと共有メモリを持つ OSCAR マルチコアアーキテクチャのみならず、通常の SMP アーキテクチャでも有効であることを示すために本システムを用いた。

3 粗粒度タスク並列処理

粗粒度タスク並列処理とは、ソースプログラムを基本ブロックあるいは基本ブロックを融合・分割した形である疑似代入文ブロック (BPA)、ループの一般形である繰り返しブロック (RB)、サブルーチンブロック (SB) の 3 種類のマクロタスク (MT) に分割し、そのマクロタスクを複数のプロセッサエレメント (PE) から構成されるプロセッサグループ (PG) に割り当てて実行することにより、マクロタスク間の並列性を利用する並列処理手法である。

3.1 マクロタスクの生成

粗粒度タスク並列処理では、まずソースプログラムを BPA, RB, SB の 3 種類のマクロタスクに分割する。ループ並列処理不可能な実行時間の大きい RB や、インライン展開を効果的に適用できない SB に対しては、その内部を階層的に粗粒度タスクに分割して並列処理を行う。

3.2 マクロフローグラフ (MFG) の生成

マクロタスクの生成後、マクロタスク間のコントロールフローとデータ依存を解析し、その結果を表す図 2(a) に示すようなマクロフローグラフ (MFG) を生成する。

図 2(a) の各ノードはマクロタスクを表し、実線エッジはデータ依存を、点線エッジはコントロールフローを表す。また、ノード内の小円は条件分岐を表す。MFG ではエッジの矢印は省略されているが、エッジの方向は下向を仮定している。

3.3 マクロタスクグラフ (MTG) の生成

MFG はマクロタスク間のコントロールフローとデータ依存は表すが、並列性は表していない。並列性を抽出するためには、コントロールフローとデータ依存の両方を考慮した最早実行可能条件解析をマクロフローグラフに対して行う。マクロタスクの最早実行可能条件とは、コントロール依存とデータ依存を考慮したそのマクロタスクが最も早い時点で実行可能になる条件である。

マクロタスクの最早実行可能条件は図 2(b) に示すようなマクロタスクグラフ (MTG) で表される。

MFG と同様に、MTG におけるノードはマクロタスクを表し、ノード内の小円はマクロタスク内の条件分岐を表している。実線のエッジはデータ依存を表し、点線のエッジは拡張されたコントロール依存を表す。拡張されたコントロール依存とは、通常のコントロール依存だけでなく、データ依存とコントロール依存を複合的に満足させるため先行ノードが実行されないことを確定する条件分岐を含んでいる。

また、エッジを束ねるアークには 2 つの種類がある。実線アークはアークによって束ねられたエッジが AND 関係にあることを、点線アークは束ねられたエッジが OR 関係にあることを示している。

MTG においてはエッジの矢印は省略されているが、下向きが想定されている。また、矢印を持つエッジはオリジナルのコントロールフローを表す。

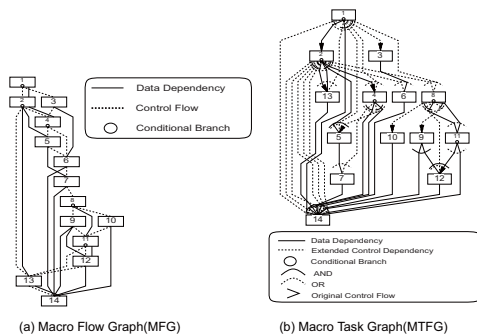


図 2: マクロフローグラフとマクロタスクグラフ

3.4 粗粒度タスクスケジューリング手法

粗粒度タスク並列処理では、生成されたマクロタスクはプロセッサグループ (PG) に割り当てられて実行される。PG にマクロタスクを割り当てるスケジューリング手法として、コンパイル時に割り当てを決めるスタティックスケジューリングと実行時に割り当てを決めるダイナミックスケジューリングがあり、マクロタスクグラフの形状、実行時不確実性などを元に選択される。スタティックスケジューリングは、マクロタスクグラフがデータ依存エッジのみを持つ場合に適用され、コンパイラがコンパイル時にマクロタスクの PG への割り当てを決定する

方式である。スタティックスケジューリングでは、実行時スケジューリングオーバーヘッドを無くし、データ転送と同期のオーバーヘッドを最小化することが可能である。

ここでは粗粒度タスクスケジューリング手法として、図 3 に示す Load2 と Store1 のようにタスク処理とデータ転送のオーバーラップを効果的に行うために、データ転送をデータの生成時点から使用時点までの間でインターコネクションネットワークの占有状態、対象メモリの利用状態、使用 DTU の利用状態等を考慮してタスク処理とオーバーラップ可能な最も早い時点においてデータ転送を挿入するデータプレロード・ポストストア手法を ETF/CP (Earliest Task First/ Critical Path) アルゴリズムに付加した ETF/CP with PLPS を用いる。また、図 4(a) で網掛けされた MT のようなデータ依存を持つ複数ループをキャッシュあるいはローカルメモリサイズにおさまるように、図 4(b) のように整合して分割を行うデータローカライゼーション手法¹⁹⁾ の適用が可能なループ集合に対しては、同一のデータにアクセスする分割後の MT 集合であるデータローカライザブルグループ (DLG) を同一のプロセッサへ優先的に連続でタスクを割り当て、キャッシュあるいはローカルメモリを介したデータの授受によりデータ転送の最小化を行う。得られたタスクスケジューリング結果に対して、各 MT でアクセスする配列データ、あるいは、DLG 単位でアクセスする配列データに対してローカルメモリへの配置、残存するデータ転送を隠蔽するためにプレロード・ポストストア手法を用いてデータ転送の駆動タイミングを決定する。これらのタスクスケジューリング、ローカルメモリへのデータ配置、データ転送オーバーラップを 2 プロセッサ、1 本バスにおける適用例を図 5 に示す。

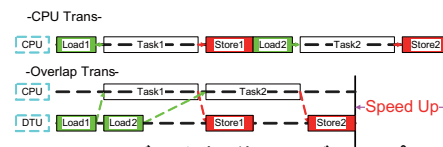


図 3: データ転送オーバーラップ

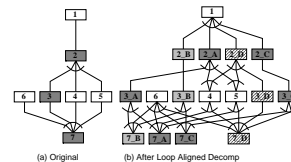


図 4: ループ整合分割の例

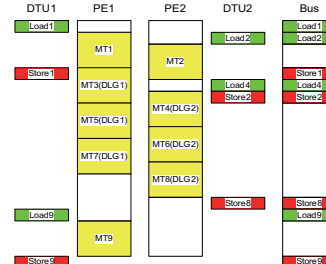


図 5: 提案手法のスケジューリング結果

4 マルチメディアアプリケーションの並列化手法

情報家電上でのコーデック等のマルチメディア処理が重要になり、アルゴリズムの複雑化から高い処理性能が要求される。従来、本要求に応えるために、DSP²⁰⁾ やマルチメディアプロセッサ²¹⁾ の利用や、Intel SSE²²⁾、Sun VIS²³⁾、MIPS MDMX²⁴⁾ のようなマルチメディア拡張命令セットの追加が挙げられる。本稿ではマルチメディア処理において動画処理の例として MPEG2 エン

コード, MPEG2 デコード, 音声処理の例として MP3 エンコード, 静止画処理の例として JPEG2000 エンコードを選択した。以下に各マルチメディアアプリケーションの並列化手法について述べる。各マルチメディアアプリケーションにおいて, 並列性の抽出を行った MTG に対して, 第 3 章で提案した粗粒度タスクスケジューリング手法によりタスクスケジューリング, ローカルメモリ管理, メモリ間データ転送のオーバラップを行った。

4.1 MPEG2 エンコード

MPEG2 のデータ構造を図 6 に, MPEG2 エンコードの処理ブロック図を図 7 に示す。MPEG2 エンコード処理は, 動き推定, 動き予測, DCT モード選択, データ変換, ビットストリーム出力, 逆量子化, 逆データ変換の 7 つのステージからなる。MPEG2 エンコードではマクロブロックレベルの並列性を利用する²⁵⁾。ただし, ビットストリーム出力ステージにおいてはマクロブロック間でのデータ依存が存在するため, マクロブロック毎に逐次で処理を行う必要がある。MPEG2 エンコード処理における並列性抽出後の MTG を図 8 に示す。

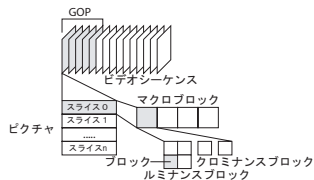


図 6: MPEG2 のデータ構造

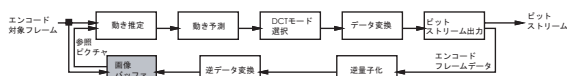


図 7: MPEG2 エンコードの処理ブロック図

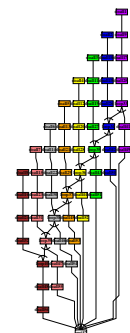


図 8: 並列性抽出後の MPEG2 エンコードの MTG 例

4.2 MPEG2 デコード

MPEG2 デコードのブロック図を図 9 に示す。MPEG2 デコード処理は, 可変長復号化, 逆量子化, 逆量子化後の各係数値の制限処理, 逆離散コサイン変換, 動き補償予測, 足し合わせ処理の 6 つのステージからなる。MPEG2 デコード処理ではスライスレベルの並列性とスライス処理内部でのマクロブロックレベルの並列性を利用し, スライスに対する可変長復号化処理中のスライスヘッダの検出処理を分割するプレスキャンニング手法を適用した²⁶⁾。プレスキャンニングではビットストリームを先頭から走査するためにスライス毎に逐次で処理を行う必要がある。MPEG2 デコーダにおける並列性抽出後のマクロタスクグラフを図 10 に示す。

4.3 MP3 エンコード

MP3 のデータ構造を図 11 に, MP3 エンコードのブロック図を図 12 に示す。MP3 エンコード処理は, サブバンド分析, 心理聴覚分析, 変形離散コサイン変換, 非

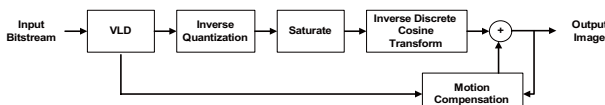


図 9: MPEG2 デコードの処理ブロック図

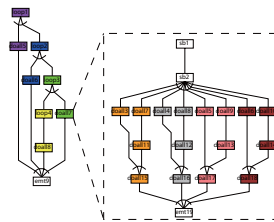


図 10: 並列性抽出後の MPEG2 デコードの MTG 例

線形量子化, ハフマン符号化の 5 つのステージからなる。MP3 エンコード処理ではフレームレベルの並列性を利用する。ただし, 心理聴覚分析, 変形離散コサイン変換においてフレーム間でのデータ依存が存在するため, フレーム毎に逐次で処理を行う必要がある。MP3 エンコードにおける並列性抽出後のマクロタスクグラフを図 13 に示す。

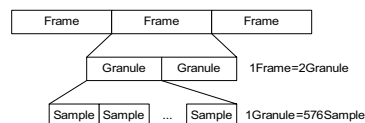


図 11: MP3 のデータ構造

4.4 JPEG 2000 エンコード

JPEG 2000 のデータ構造を図 14 に, JPEG 2000 エンコードのブロック図を図 15 に示す。JPEG 2000 エンコード処理は, DC レベル変換, 離散ウェーブレット変換, スカラ量子化, EBCOT (Embedded Block Coding with Optimized Truncation) 符号化の 4 つのステージからなる。JPEG 2000 エンコードでは DC レベル変換, 離散ウェーブレット変換では画像の縦方向, 横方向の並列性を利用し, スカラ量子化, EBCOT では 64x64 ピクセルのコードブロック単位の並列性を利用する。ただし, 離散ウェーブレット変換においては縦方向, 横方向の処理間や各レベルの処理間では転置転送により, データ転送が発生する。ここで, スカラ量子化は各サブバンドに対して行う処理であるがコードブロックがサブバンド境界を越えて生成されないためコードブロック単位で処理を行うことが可能となる。JPEG 2000 エンコードにおける並列性抽出後のマクロタスクグラフを図 16 に示す。

5 性能評価

本章では第 4 章で述べたマルチメディアアプリケーションの性能評価について述べる。

5.1 評価アプリケーション

評価アプリケーションとして第 4 章で示した 4 つのメディアアプリケーションを対象とした。表 1 に評価を行った各メディアアプリケーションの参照元プログラム, 特徴的な処理パラメータを示す。記述していない処理パラメータは各参照プログラムにおけるデフォルトパラメータを使用した。

5.2 OSCAR マルチコア上での性能評価

本評価はクロックレベルの詳細なシミュレータを用いて行った。

各評価はローカルメモリ上への配列のデータ配置を行い, CPU が CSM と LDM 間のデータ転送を行った場合と, DTU により CSM と LDM (DSM) 間, PE 間のデータ転送を CPU の演算処理とオーバラップさせた場合を比

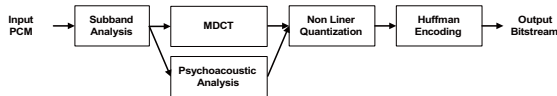


図 12: MP3 エンコードの処理ブロック図

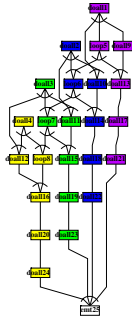


図 13: 並列性抽出後の MP3 エンコードの MTG 例

較した．評価に使用した入力データを表 2 に，評価結果を図 17 に示す．図中の横軸には評価アプリケーション，プロセッサ数を示し，縦軸には 1 プロセッサで CPU で転送を行った場合の実行時間に対する速度向上率を示す．CPU でデータ転送を行った場合の逐次処理に対して，CPU でデータ転送を行った 4 プロセッサでの並列処理では MPEG2 エンコードで 3.44 倍，MPEG2 デコードで 2.45 倍，MP3 エンコードで 3.01 倍，JPEG 2000 エンコードで 3.43 倍，DTU を用いた 4 プロセッサでの並列処理では MPEG2 エンコードで 3.81 倍，MPEG2 デコードで 3.04 倍，MP3 エンコードで 3.09 倍，JPEG 2000 エンコードで 3.79 倍の速度向上が得られた．この結果より，DTU を用いたデータ転送オーバーラップの効果として MPEG2 デコードにおいて 4 プロセッサで最大 24% の速度向上が得られることが確認できた．

5.3 IBM p5 550Q 上での性能評価

評価に使用した入力データを表 3 に，IBM XL C コンパイラ version 8.0 / XL Fortran コンパイラ version 10.1 と OSCAR コンパイラの評価結果を図 18 に示す．図中の横軸には評価アプリケーション，プロセッサ数を示し，縦軸には XLC/XLF コンパイラによる 1 プロセッサの実行時間に対する速度向上率を示す．本評価においては並列処理性能を評価するために，実行環境に著しく依存する I/O 処理の時間を除外し，演算処理部分のみを評価の対象とした．XLC/XLF コンパイラによる逐次処理に対して，OSCAR コンパイラによる 8 プロセッサでの並列処理では，MPEG2 エンコードで 5.19 倍，MPEG2 デコードで 5.12 倍，MP3 エンコードで 3.69 倍，JPEG 2000 エンコードで 4.32 倍の速度向上が得られ，本並列化手法がマルチコアプロセッサを搭載した SMP サーバ上においても有効であることが確認できた．

6 まとめ

本稿では，マルチメディアアプリケーションの例として動画処理である MPEG2 エンコード，MPEG2 デコード，音声処理である MP3 エンコード，静止画処理である JPEG 2000 エンコードに対する粗粒度タスク並列処理手法を提案し，OSCAR マルチコア上，IBM p5 550Q

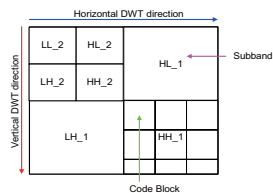


図 14: JPEG2000 のデータ構造

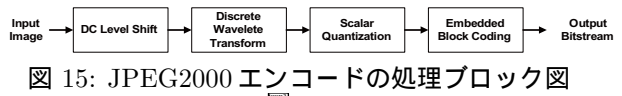


図 15: JPEG2000 エンコードの処理ブロック図

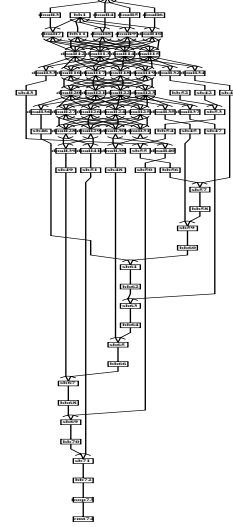


図 16: 並列性抽出後の JPEG2000 エンコードの MTG 例

Power5+ 2 コア集積マルチコアプロセッサ搭載 SMP サーバ上で評価を行った．OSCAR マルチコア上では，逐次処理に対して 4 プロセッサで MPEG2 エンコードで 3.81 倍，MPEG2 デコードで 3.04 倍，MP3 エンコードで 2.94 倍，JPEG 2000 エンコードで 3.79 倍の速度向上が得られた．IBM p5 550Q Power5+ 2 コア集積マルチコアプロセッサ搭載 SMP サーバ上では，逐次処理に対して OSCAR コンパイラによる 8 プロセッサでの並列処理では，MPEG2 エンコードで 5.19 倍，MPEG2 デコードで 5.12 倍，MP3 エンコードで 3.69 倍，JPEG 2000 エンコードで 4.32 倍の速度向上が得られた．

本研究の一部は NEDO “リアルタイム情報家電用マルチコア技術”，NEDO “先進ヘテロジニアスマルチプロセッサ技術” 及び STARC (半導体理工学研究センター) “並列化コンパイラ協調型チップマルチプロセッサ技術” の支援により行われた．

参考文献

- [1] Pham, D. et al.: The Design and Implementation of a First-Generation CELL Processor, *In Proceeding of the IEEE International Solid-State Circuits Conference* (2005).
- [2] Cornish, J.: Balanced Energy Optimization, *International Symposium on Low Power Electronics and Design* (2004).

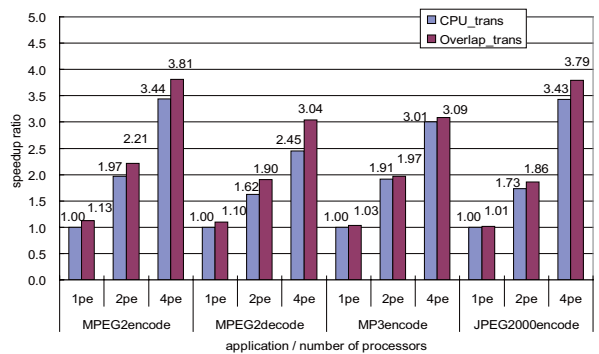


図 17: OSCAR マルチコア上での性能評価

表 1: 評価メディアアプリケーション仕様

MPEG2 エンコード	
参照元プログラム	MediaBench ²⁷⁾ mpeg2encode
評価パラメータ	Frames in GOP : 15 I/P frame distance : 3 Picture Type : Frame Picture Bitrate : 5Mbps
MPEG2 デコード	
参照元プログラム	MediaBench mpeg2decode
MP3 エンコード	
参照元プログラム	UZURA MP3 encoder ²⁸⁾
評価パラメータ	Bitrate : 128kbps CBR Compression : lossy
JPEG2000 エンコード	
参照元プログラム	JJ2000 ²⁹⁾
評価パラメータ	Bitrate : 100bpp WaveletLevel : 3 Compression : lossy

表 2: OSCAR マルチコア上での評価入力データ

アプリケーション	入力データ
MPEG2encode	Sub-QCIF 4frames (I/P/B/B) 4:2:0 format
MPEG2decode	Sub-QCIF 4frames encoded with 5Mbps Bitrate
MP3encode	44.1kHz Stereo PCM 4frames
JPEG2000encode	128x128 pixel

- [3] Suga, A. and Matsunami, K.: Introducing the FR 500 embedded microprocessor, Vol. 20, pp. 21–27 (2000).
- [4] Kamei, T.: SH-X3 : An Enhanced SuperH Core for Low-power MP Systems, *Fall Microprocessor Forum 2006* (2006).
- [5] Intel: <http://www.intel.com/multi-core/>.
- [6] Keltcher, C. N., McGrath, K., Ahmed, A. and Conway, P.: The AMD Opteron processor for multiprocessor servers, *IEEE Micro*, Vol. 23, pp. 66–76 (2003).
- [7] Clabes, J. et al.: Design and implementation of the POWER5 microprocessor, *IEEE ISSCC*, pp. 55–57 (2004).
- [8] 藤原, 白鳥, 鈴木, 笠原: データプレロードおよびポストストアを考慮したマルチプロセッサスケジューリングアルゴリズム, *電子情報通信学会論文誌 (D-I)*, Vol. J75-D-I, No. 8, pp. 495–503 (1988).
- [9] 藤本, 橋本, 笠原: データ転送と処理のオーバーラップを用いたデータ転送最小化自動並列化コンパイラ, *電気学会情報処理研究会資料*, No. IP-96-24 (1996).
- [10] 木村, 古郷, 尾形, 笠原, 橋本: 処理とデータ転送のオーバーラッピングを考慮したダイナミックスケジューリングアルゴリズム, *CPSY97* (1997).
- [11] 宮本, 中川, 浅野, 内藤, 仁藤, 中野, 木村, 笠原: マルチコアプロセッサ上での粗粒度タスク並列処理におけるデータ転送オーバーラップ, *情報処理学会研究報告 2006-ARC-167(HOKKE-2006)*, pp. 55–60 (2006).
- [12] D Kim, R Managuli, Y. K.: Data cache and direct memory access in programming mediaprocessors, pp. 33–42 (2001).
- [13] 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処理, *情報処理学会論文誌*, Vol. 40, No. 5 (1999).
- [14] Kimura, K., Wada, Y., Nakano, H., Kodaka, T., Shirako, J., Ishizaka, K. and Kasahara, H.: Multigrain Parallel Processing on Compiler Cooperative Chip

表 3: IBM p5 550Q 上での評価入力データ

アプリケーション	入力データ
MPEG2encode	SIF 300frames 4:2:0 format
MPEG2decode	SIF 300frames encoded with 5Mbps Bitrate
MP3encode	44.1kHz Stereo PCM 253frames
JPEG2000encode	1920x1080 pixel

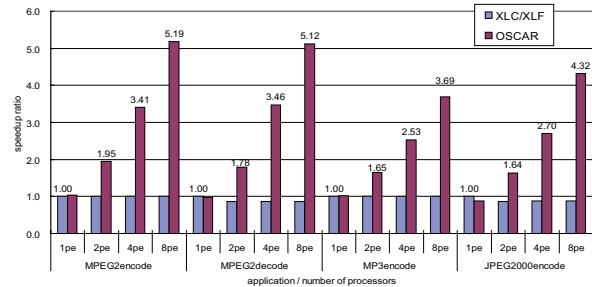


図 18: IBM p5 550Q 上での性能評価

Multiprocessor, *Proc. of 9th Workshop on Interaction between Compilers and Computer Architectures (INTERACT-9)* (2005).

- [15] ITRS: International Technology Roadmap for Semiconductors 2003 Executive Summary (2003).
- [16] Wilton, S. and Jouppi, N.: CACTI: An enhanced cache access and cycle time model, *IEEE Journal of Solid-State Circuits*, Vol. 31, No. 5, pp. 677–688 (1996).
- [17] ELPIDA MEMORY, INC.: *PRELIMINARY DATA SHEET 512bits DDR SDRAM EDD 5104 ABTA, EDD 5108 ABTA* (2003).
- [18] ELPIDA MEMORY, INC.: *PRELIMINARY DATA SHEET 256bits DDR2 SDRAM EDE 2504 AASE, EDE 2508 AASE, EDE 2516 AASE* (2003).
- [19] 石坂, 中野, 八木, 小幡, 笠原: 共有メモリマルチプロセッサ上でのキャッシュ最適化を考慮した粗粒度タスク並列処理, *情報処理学会論文誌*, Vol. 43, No. 4 (2002).
- [20] Eiji Iwata and et.al.: A 2.2GOPS Video DSP with 2-RISC MIMD, 6-PE SIMD Architecture for Real-Time MPEG2 Video Coding/Decoding, *IEEE International Solid State Circuits Conference Digest of Technical Papers*, pp. 258–259 (1997).
- [21] Yong Yao: Chromatic's Mpaact 2 Boosts 3D, *Microprocessor Report*, Vol. 10 (1996).
- [22] S. K. Raman, V. Pentkovki and J.Keshava: Implementing Streaming SIMD Extensions on the Pentium III Processor, Vol. 20, No. 4 (2000).
- [23] March Tremblay and et.al.: VIS Speeds New Media Processing, *IEEE Micro*, pp. 10–20.
- [24] Earl Killian: Extending the MIPS Instruction Set for Digital Media and Emerging Applications, *Microprocessor Forum 96* (1996).
- [25] 小高剛, 中野啓史, 木村啓二, 笠原博徳: チップマルチプロセッサ上での MPEG2 エンコードの並列処理, *情報処理学会論文誌*, Vol. 46, No. 9 (2005).
- [26] Iwata, E. and Olukotun, K.: Exploiting Coarse-Grain Parallelism in the MPEG-2 Algorithm, *Technical Report CSL-TR-98-771* (1998).
- [27] 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).
- [28] UZURA3:MPEG1/LayerIII Encoder in FORTRAN90. http://members.at.infoseek.co.jp/kitaurawa/index_e.html.
- [29] R Grosbois, D Santa Cruz, J. A. B. B. D. B. F. H. G. M. and Onno, P.: <http://jj2000.epfl.ch/>.