

H.264 デコーダにおける 2 パス限定投機方式の適用検討

川上憂騎[†] 大津金光[†] 横田隆史[†] 馬場敬信[†]

[†]宇都宮大学工学部情報工学科

1 はじめに

動画像圧縮符号化の標準規格である H.264 は従来方式 (MPEG-2) に比べ 2 倍以上の高い圧縮率を達成することが可能であり、ワンセグ放送、携帯端末など様々な分野で普及しているが、デコード処理に非常に多くの演算量が必要であり、デコーダの高速化に対する需要が高まっている。この課題に対して、スレッドレベル並列性および命令レベル並列性による高速化のアプローチがなされているが、H.264 のエントロピー符号化方式である CABAC (Context-based Adaptive Binary Arithmetic Coding) を一例として、高速化の効果が十分に得られない場合が存在する。

一方、半導体の集積技術の向上により計算機システムにおける利用可能なハードウェア資源は増大しているが、消費電力の増加などの問題からクロック速度の向上が飽和している状況であり、クロック速度の向上とは異なる手法による高速化手法が求められている。この課題に対して、我々の研究室ではプログラム中の実行経路 (パス) を投機の対象とする 2 パス限定投機方式を提案した [1]。2 パス限定投機方式は実行頻度が高いループ中のパスのうち、上位 2 つのパスに限定してパスベース投機的マルチスレッド実行するが、投機が失敗した場合はオーバーヘッドが増加する。このような背景から、本研究では H.264 デコーダプログラムを対象として 2 パス限定投機方式を適用し、その有効性を検証する。本稿では、H.264 デコーダプログラムの挙動を解析して実行されるプログラムパスの実行頻度を評価することで、投機成功率の観点から 2 パス限定投機方式の適用可能性を明らかにする。

2 2 パス限定投機方式

2 パス限定投機方式 [1] はプログラムの実行パスのうち実行頻度の高い上位 2 つのパスが支配的であることに着目し、プログラムの実行時に実行されるパスを予測して予測パスに特化して無駄な処理を削減した最適化コードを実行することで、スレッド間並列性やスレッド自身の実行性能の向上を目指したパスベース投機的マルチスレッド実行方式である。2 パス限定投機方式はプログラムの制御フローに沿ってプログラムをスレッド分割し、それぞれを投機的に並列実行を行うモデルである。先行スレッドが次のスレッドを順次起動することで複数スレッドを起動し、スレッドレベル並列性を活用する。投機実行が失敗した場合は一旦回復処理を行い、もう一方のパスを投機実行する。再度投機が失敗した場合は非投機のプログラムコード

を実行する。二度の投機処理を行うこの方式を我々は DoubleSpeculation と呼んでいる。

2 パス限定投機方式はプログラム中のホットループを対象とし、ループのイテレーション単位で投機的マルチスレッド実行することから、性能向上は投機の成功率に大きく依存する。そのため、プログラム中のホットループのうち実行頻度の高い上位 2 つのパスの全体に占める実行割合がどの程度になるかを明らかにすることが重要となる。

3 評価

以下の手順により、H.264 デコーダプログラムを対象として、実行頻度の高い上位 2 つのパスが全体に占める割合を明らかにする。

1. プログラムの実行トレースを取得する
2. 実行トレース情報を元にホットループを検出する
3. 選出したホットループ中のパスを集計する

3.1 評価環境

対象となるデコーダプログラムは東芝のモバイルマルチメディアシステム向けのスケラブルマルチコアプロセッサである Venezia [2] 上で動作する H.264 デコーダプログラムとし、評価の際は Venezia 実機のシミュレーション環境である Venezia シミュレータ上で実行した。また、関数の実行サイクル数などのプロファイル情報の取得にはトレースベースプロファイラを用いた。

3.2 評価用入力映像

評価に用いる映像は標準画像の Football, Susie, Tempete の 3 つを用いた。表 1 に入力映像データの符号化条件を示す。

表 1: 符号化条件

プロファイル	High
レベル	1.0
ビットレート	30kbps
解像度	176 × 144
フレームレート	15fps
フレーム数	10frame
GOP 構造	IBPBP
I フレーム間隔	5frame
走査方式	Progressive
探索アルゴリズム	Uneven Multi-Hexagon
探索範囲	[-16,15]
Sequence	Football, Susie, Tempete

3.3 対象関数とループの選別

2 パス限定投機方式の適用の有効性を評価する対象となる関数とループの検出方法について述べる。評価対象となる関数はプログラム実行全体で多くのサイク

Application Study of Two-Path Limited Speculation Method using H.264 Decoder

[†]Yuki Kawakami, Kanemitsu Ootsu, Takashi Yokota and Takanobu Baba

Department of Information Science, Faculty of Engineering, Utsunomiya University (†)

ルを占めるホットな関数とし、総実行サイクル数の大きい上位の関数から関数内にループを含んでいる関数を選択した。また、既に十分最適化処理が施されていると考えられるため、Venezia の SIMD 命令や VLIW 命令を含んだ関数については評価対象から除外した。表 2~4 は 3 種類の入力映像 (Football, Susie, Tempete) をそれぞれデコードした際の実行サイクル数が多い順に上位 5 つの関数である。ただし、関数 StageMotionVectorPrediction.execute 内には 6 つのループが存在するが、今回はそのうち最もイテレーション数が多いループを対象とした。表より分かる通り、順位は変動するものの、いずれの入力映像の場合も同じ関数が頻繁に実行されている。以後、これらの関数に含まれるループをそれぞれをループ A~E と呼ぶものとする。

表 2: 入力映像 Football 時の実行サイクル上位関数

関数名	総実行サイクル数	ループ
_xmemset	1031411	A
BitstreamReader_skip	919758	B
RenormD	600026	C
StageMotionVectorPrediction.execute	530772	D
mv_store_8x8	448299	E

表 3: 入力映像 Susie 時の実行サイクル上位関数

関数名	総実行サイクル数	ループ
_xmemset	903631	A
BitstreamReader_skip	666174	B
StageMotionVectorPrediction.execute	509993	D
mv_store_8x8	461400	E
RenormD	409263	C

表 4: 入力映像 Tempete 時の実行サイクル上位関数

関数名	総実行サイクル数	ループ
_xmemset	1013906	A
BitstreamReader_skip	800376	B
StageMotionVectorPrediction.execute	518168	D
RenormD	508513	C
mv_store_8x8	453306	E

3.4 実行パスの集計結果

前節にて選出した関数内のループに対してパス集計を行った。パスの集計を行った結果を図 1~3 に示す。3 つの評価用映像をデコーダプログラムで処理した結果、対象ループのパスには顕著な偏りがあることがわかる。関数 _xmemset, StageMotionVectorPrediction.execute, RenormD, mv_store_8x8 内の各ループ (すなわち、ループ B 以外のループ) では #1 path が実行パスの 100% を占めている。関数 BitstreamReader_skip のループ (ループ B) は #1 path が 90% 以上の比率を占め、#2 path が 8.60%~9.10% を占めている。この結果から、BitstreamReader_skip を除く 4 つの関数は、関数内の対象ループを投機実行すると #1 path の投機実行で投機が成功し、方式による投機実行が効果的であることがわかる。また、BitstreamReader_skip についても DoubleSpeculation 方式により、1 度目の投機実行が失敗した場合に 2 度目の投機が成功する確率は 99.96% となり、2 パス限定投機方式の適用により高速化が期待できる。

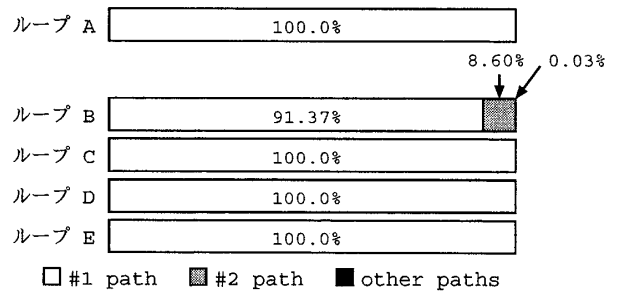


図 1: 入力映像 Football のパス集計結果

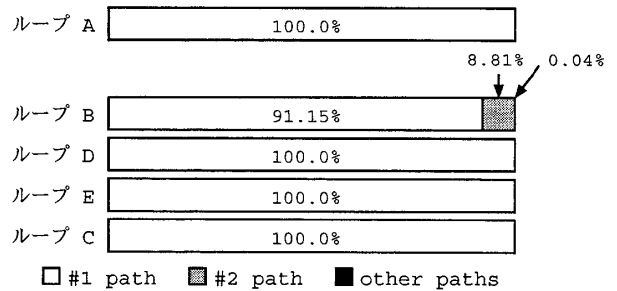


図 2: 入力映像 Susie のパス集計結果

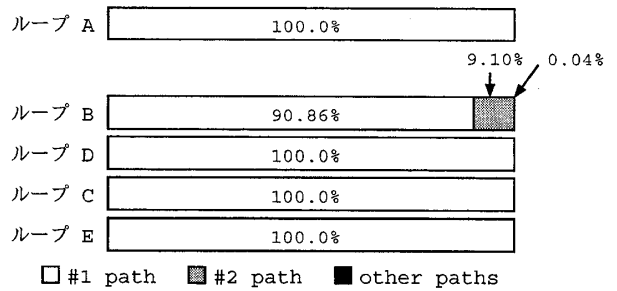


図 3: 入力映像 Tempete のパス集計結果

4 おわりに

H.264 デコーダプログラムを用いた評価の結果、投機実行成功率の観点から 2 パス限定投機方式の適用により高速化が期待できることを明らかにした。今後は実際に H.264 デコーダプログラムに対して 2 パス限定投機方式を適用し、性能評価を行う予定である。

謝辞

本研究は、一部日本学術振興会科学研究費補助金 (基盤研究 (C)20500047, 同 (C)21500049, 同 (C)21500050) および宇都宮大学重点推進研究プロジェクトの援助による。また (株) 東芝セミコンダクター社より Venezia 開発環境ならびに H.264 コードを提供いただき実施しました。

参考文献

- [1] 横田隆史, 斎藤盛幸, 大津金光, 古川文人, 馬場敬信, “2 パス限定投機方式の提案”, 情報処理学会論文誌: コンピューティングシステム, Vol.46, No.SIG 16(ACS-12), pp.1-13, 2005
- [2] T. Miyamori “Venezia: a scalable multicore subsystem for multimedia applications”, in Proceedings of the 8th International Forum on Application-Specific Multi-Processor SoC, Aachen, Germany, June 2008.