

J_030

超高精細映像用ソフトウェアベースリアルタイムエンコーダの 分散処理実装

Distributed Implementation Design of Full Software Real-time Encoder Applicable for Super High Definition Images

内藤 整†
Sei Naito

小池 淳†
Atsushi Koike

1. まえがき

次世代の映像アプリケーションとして、4K デジタルシネマ[1]など HDTV を上回る解像度を有する超高精細映像の実用化が進展している。超高精細映像を圧縮伝送する技術的な枠組も整理されており、例えば H.264[2]や JPEG2000[3]は仕様の的に 4K デジタルシネマに対応できる。ここで商用サービスにて利用されるビデオエンコーダ装置の実装は、専用 LSI によるハードウェアベースのものが主流であった。ハードウェアエンコーダの欠点として、高解像度対応や符号化機能の増強などを実現する際に、LSI の作り直しを伴い、膨大なコスト負担を強いられる点が挙げられる[4]。一方で、近年の CPU パフォーマンスの向上や PC 間ネットワークスピードの高速化の流れを受け、超高速の計算機システムの開発は、複数台の PC 利用による分散コンピューティングによるものが一般的になりつつある[5]。しかしながら、細かいタイミング制御が要求されるリアルタイムシステムにおいての実用例は少ない。本論文は、分散コンピューティングを駆使したオールソフトウェア処理によるリアルタイムエンコーダの実現を目指し、ポイントとなる要素技術について導入を行うものである。2.では提案するシステムアーキテクチャ、およびエンコード処理を構成するノード配置方式の導入を行う。3.では提案したシステムアーキテクチャについて、PC プラットフォームを利用したパフォーマンス評価を行い、提案技術の有効性を示している。

2. 提案方式

2.1 システム構成

分散コンピュータシステムで必須となる並列プログラミング手法について予備検討を行った。まず並列プログラミングで実用されている代表的なパラダイムとして、Open MP, MPI(Message Passing Interface), Shared Memory が有名である。このうち Open MP は SMP(マルチプロセッサ)環境における CPU 間の並列処理においてのみ適用可能であり、MPI と Shared Memory は PC クラスタなどにおいてノードを跨いだ並列化を実現するものである。ノードとは PC クラスタの構成要素、つまり PC を指す。本研究では、プラットフォームへの依存度を少なくするとともに、扱う画像解像度に対するスケーラビリティをコンピュータリソースの増減により達成するという観点から、CPU 間の並列実装だけでは不十分であると考え、MPI または Shared Memory による CPU 間、およびノード間での並列処理を駆使したシステムアーキテクチャを採用することとした。

複数台の PC により構成される並列処理プラットフォームとしては、汎用の PC クラスタを採用することとした。PC クラスタでは、PC 間は専用の高速インターコネクト回線(Myrinet, Giga-bit Ethernet など)で接続され、CPU およびネットワークの各リソースはすべてエンコード処理に割り当てることが可能である。

次に前述の MPI と Shared Memory の優劣について検証する。まずビデオエンコーダを並列処理により実現する上では、ノード間で大量の画像データを高速に相互参照できることが必須である。比較のため、表 1 に示す PC クラスタ上で HDTV の画像データを 1 フレーム分、ノードを跨いで参照するのに要した時間を測定した結果を表 2 に示す。結果は 100 回試行した際の平均値と最大値を示す。同表よりノード間のデータ参照速度に関して MPI の優位性が認められる。よって以後の検討において並列処理で用いるプロセス間のデータ通信はすべて MPI により実装することとする。

2.2 ビデオ符号化処理の分散処理実装

ビデオ符号化処理の分散処理実装方式を以下に提案する。分散コンピューティングでとられる処理の並列化手法は、データ並列とジョブ並列に大別される。ビデオ符号化に限定すれば、データ並列は画面分割やサブバンド分割による入力データの細分化に相当し、ジョブ並列は一連の符号化処理シーケンスを構成する動き補償予測や DCT といった機能レベルの細分化に相当する。提案手法においては、ビデオ符号化処理をフロントエンド部分と符号化部分に大別した上で、各部においてデータ並列を施す構成とした。

表 1 PC クラスタの諸元

CPU	AMD Opteron 2.6GHz
メモリ	2GB(1CPUあたり)
装置内ノード数	12 CPU
ノード間アクセス	RapidArray インターコネクト[6] 4GB/s

表 2 MPI と Shared Memory の比較

方式	データ参照コスト msec	
	平均	最大
MPI	3.00	3.11
Shared Memory	4.01	4.27

† (株) KDDI 研究所, KDDI R&D Laboratories Inc.

ブロック図を図1に示す。ノードは役割に応じて分類され、符号化ノードとフロントエンドノードの2種類が存在する。このうち符号化ノードは、分割画面に対する符号化処理を担い、フロントエンドノードは入力画像の符号化ノードへの分配、符号化ノードから取得したストリームの多重化といった外部機器への入出力処理を担う。以後、1符号化ノードが担当する画像エリアを単にセグメントと呼ぶ。ここで各符号化ノードが完全に独立して処理を行うと、分割画面単位での符号化処理を行うことに起因して符号化効率を大きく低下させる恐れがあるため、提案方式においては、以下の仕組みを導入することとした。

- A) 動き補償予測効率の低下を避けるため、隣接セグメント間では符号化ノードを跨いだ局所復号画像の参照を可能とする。
- B) フレームを構成するセグメント境界において顕著な画質差が発生するのを防ぐため、フレームの符号化処理に先立ち、フロントエンドノードによってセグメント単位のビット配分処理が施される。セグメント間での画質差を抑えるとともに統計多重による符号化効率の改善効果が見込まれる[7]。

本システムによりリアルタイム処理を実現するためのタイミングチャートを図2に示す。フロントエンドノードではビデオの垂直同期信号をトリガーとして、各符号化ノードに対する画像データの分配、および処理開始指示をMPI通信によって行う。リアルタイム処理に必須の条件として、符号化ノードに与えるセグメントサイズとしては、全ノードにおいて、ビデオ入力、符号化、ストリーム出力にかかるトータルの時間が安定的にフレーム間隔を下回るような値を選択する必要がある。符号化ノードのストリーム出力はフロントエンドノードにて一旦蓄積され、多重化、ヘッダ付与の後、総符号化レートに応じて適切なタイミングでIPパケット出力される。フロントエンドノードでの多重化、およびヘッダ付与に要する処理時間の合計についても同様に、フレーム間隔を下回る必要がある。

3. 性能評価

2.で提案したエンコーダ実装方法を評価するため、一例としてMPEG-2の符号化処理に準じた符号化ノードに搭載するソフトウェアモジュールを試作し、PCクラスタ上への分散配置を行った。ただしエンコーダ実装方法自体は符号化方式に非依存であり、適用対象はMPEG-2に限定されない。符号化ノードに適用する符号化条件を表3に示す。PCクラスタは表1で導入したものと同一のものを使用する。入力映像としては、DCIより提供されている4Kデジタルシネマ(4096画素×2160ライン、24Hz、プログレッシブ)評価用のテストシーケンスSIEM[8]を使用した。SIEMに収録されている画像の一部を図3に示す。前提として入力となる4K画像データは図4に従ってセグメントに分割され、各セグメントが符号化ノードに割り当てられる構成とした。図中sはセグメントの識別に使用するパラメータである。



図3 テストシーケンス

表3 符号化条件

画像フォーマット	2048画素×540ライン/24Hz, 4:2:0, 8bit/コンポーネント
符号化方式	MPEG-2 Main Profile@High Level
GOP	N=12, M=1(Bフレームなし)
MV探索範囲	128画素×64ライン(半画素) ※隣接セグメントの参照にも同探索範囲を適用
レート制御	全画面に対する一括制御

4K resolution picture

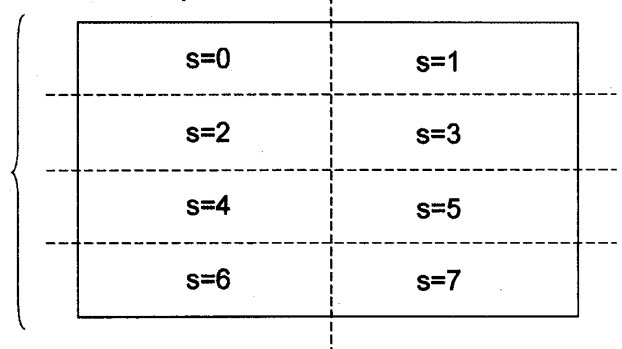


図4 4K画像のセグメント分割

まず符号化ノード単体の性能評価を行う目的から、PCクラスタ中の1ノードを符号化ノード(s=0)に割り当て、符号化処理に要する処理時間の内訳を算出した結果を表4に示す。処理時間は1200フレーム分の符号化処理を行った結果、1フレームあたりの処理時間について求めた平均値と最大値により評価した。以後の符号化実験においてシミュレーション時間は同様に1200フレームとする。結果より、符号化ノード単体の性能として、入力画像のフレーム間隔が約40msecであるのに対し、与えられた画像サイズに対する処理時間は常にこれを下回り、リアルタイム処理を安定的に達成可能なレベルにあるといえる。また処理時間の内訳としては、ピクチャタイプに依らず、符号化処理に要する時間が支配的であり、90%近くを占めている。

次に提案手法のスケラビリティを評価するため、並列動作させる符号化ノードの数がトータルのパフォーマンスに与える影響を観測した。符号化ノードの並列数の上限は4K解像度に相当する8に設定し、また中間的なケースとして4の場合も評価対象に加えることとした。並列数4においては、図4におけるs=0,1,2,3の4セグメントを符号化

対象としてノードに与えた。便宜上の理由から稼働する符号化ノードの数に依らず、フロントエンドノードは PC クラスタ中の特定の 1 ノードにより構成した。符号化ノード数と処理時間の関係を表 5 に示す。処理時間は、1 フレームあたりの平均値と最大値で示す。結果より、1 フレームあたりの処理時間は、符号化ノード数、つまり処理の並列度に依存せずほぼ一定である。これに対しシステムがリアルタイム処理を達成している画面サイズは、符号化ノード数に比例して拡大を遂げており、符号化ノードの増設が直接的にシステムの処理能力向上に寄与していることが読み取れる。

さらに並列符号化処理においては、ノード間に大量かつ頻繁なデータ参照が発生するため、並列数とノード間データ参照量の関係を調べた。ある符号化ノードに着目し、1 フレーム分の符号化処理に起因して、他ノードを相手に送信または受信するデータ量を測定した結果を表 6 に示す。データ量は 1 フレームあたりの平均値で示す。ノード数 8 においては、ノード間で相互参照される動き予測データについて、最大値を評価する目的から $s=2$ に相当する符号化ノードに着目して算出を行った。その他のケースでは $s=0$ にて観測を行った。結果より、I ピクチャの符号化においては、隣接セグメントへの動き予測参照が発生しないため、並列数に関係なく一定のデータ量となっている。一方 P ピクチャの符号化では、並列数を上げるにつれて送信、受信ともにデータ量が増えている。このうちデータ量が格段に大きい受信処理に着目すると、I ピクチャの受信データ量は、フロントエンドノードから取得する入力画像に相当する。よって P ピクチャの受信データ量においても同成分が支配的であり、特定ノードからの受信データ量の最大値は、並列数に依らず一定である。一般に PC クラスタを構成するノード間のインターコネクト回線は、通信スループット確保の観点からすべてのノード間をピアツーピアで直結するため、トータルのデータ量よりも、特定ノードを相手に扱うデータ量を集中させないことが重要である。以上を踏まえ、設計上ノード間通信に求められる回線スループットに対して、並列数の増加が与えるインパクトは少ないといえる。

表 4 符号化ノード単体の処理パフォーマンス

処理項目	処理コスト msec			
	I ピクチャ		P ピクチャ	
	平均	最大	平均	最大
画像入力	1.55	2.57	1.55	1.73
符号化処理	21.55	23.65	21.19	30.89
ストリーム出力	0.09	0.27	0.06	0.25
合計	25.61	27.93	25.63	34.65

表 5 符号化ノード数と処理時間の関係

符号化ノード数	処理コスト msec			
	I ピクチャ		P ピクチャ	
	平均	最大	平均	最大
1	25.61	27.93	25.63	34.65
4	25.72	28.74	26.01	36.24
8	25.15	30.97	26.05	38.92

表 6 ノード間で参照されるデータ量

符号化ノード数	データ量 MByte			
	送信		受信	
	I	P	I	P
1	0.12	0.01	3.11	3.11
4	0.12	0.32	3.11	3.42
8	0.12	0.53	3.11	3.63

4. むすび

超高精細映像への応用を視野に入れ、分散コンピューティングに基づいたオールソフトウェアによるリアルタイムエンコーダの実装アーキテクチャについて導入を行った。具体的には、PC クラスタ上での MPI による並列プログラミングをベースとして、これに適合するビデオ符号化の分散処理実装アーキテクチャを提案した。PC クラスタ上での実験結果から、提案方式の有する高いスケーラビリティを確認するとともに、同手法の応用によって 4K に代表される超高精細映像のリアルタイムエンコーダが実現可能であることを示した。

謝辞

本研究は独立行政法人 情報通信研究機構からの受託研究「ソフトウェア符号化技術に関する研究開発」における研究成果としてまとめられたものである。

参考文献

- [1] Digital Cinema Initiatives, "Digital Cinema System Specification v4.3", Dec, 2004
- [2] Text of ISO/IEC 14996 10 Advanced Video Coding 3rd Edition (ISO/IEC JTC1/SCD29/WG11 and ITU-T SG16 Q.6)
- [3] ISO/IEC JTC1/SC29/WG1 N1890, "JPEG2000 part I Final Draft International Standard", Sept. 2000
- [4] 内藤 整, 小池 淳, "JPEG2000 符号化の並列処理実装に関する一検討", 情処学研報, Vol. 2005, No. 23, March 2005
- [5] <http://phase.hpcc.jp>
- [6] <http://www.cray.com/products/xd1/index.html>
- [7] 酒澤茂之, 滝嶋康弘, 和田正裕, 羽鳥好律, "マルチエンコーダシステムによる動画符号化制御方式", 映情学会誌, Vol. 50, No.7, pp.958-964, July 1996
- [8] <http://www.dcmovies.com/StEMAcce.pdf>

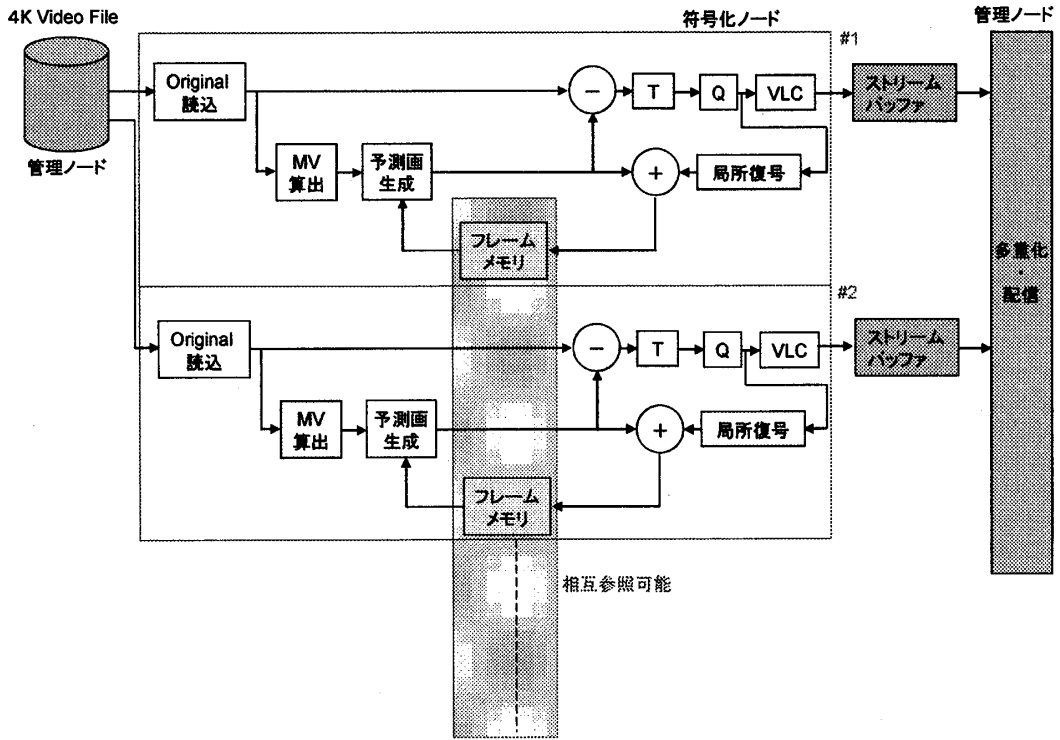


図1 ビデオエンコーダのブロック図

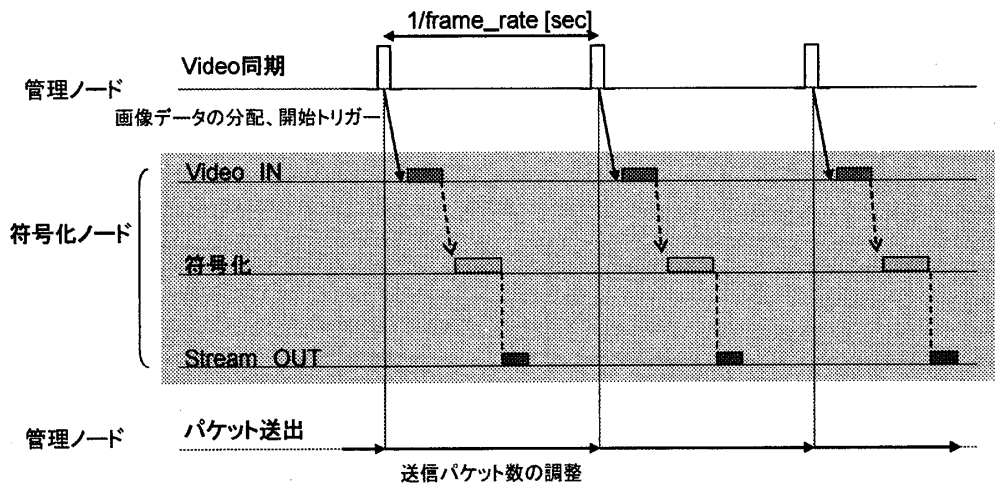


図2 ビデオ符号化処理のタイミングチャート