

低遅延ライブビデオストリーミングシステムにおける並列化に関する研究

A Study of Parallel Processing in Low Delay Live Video Streaming System

80815334 遠藤健太 (Kenta Endoh) Supervisor: 矢向高弘 (Takahiro Yakoh)

1. 序論

プロセッサの発熱やクロック数の限界などの理由から単一コアの性能向上が困難になった結果、2000年代前半に複数のコアを搭載したマルチコア・プロセッサが登場した。マルチコア・プロセッサは複数のコアで処理を並列化することで処理性能を向上させている。現在、市販PCとして販売されている製品のほとんどがマルチコア・プロセッサを搭載している。これに伴い、プログラムを組む上でも処理性能向上のために、並列化を意識したプログラミングが必要となった。

本研究室では、低遅延のライブビデオストリーミングシステムに関する研究を行ってきた。ライブビデオストリーミングとは、カメラから取得した映像をリアルタイムで処理し、配信を行うシステムであり、一般的に遅延を小さく抑える必要がある。このシステムでは、触覚情報を利用した遠隔操作など遅延要求の厳しい環境での映像による操作補助利用を想定しており、より高速な処理が求められる。よって、並列化処理による高速化に対応することは必須である。そこで、本論文では低遅延ライブビデオストリーミングシステムにおいて並列化処理を実装するための設計手法について述べる。

2. 低遅延ライブビデオストリーミングシステム

このシステムでは図1のような処理を行っている。

送信側は、取得した映像フレームを均一な複数のブロック領域に分割し、そのブロック領域の重要度に応じて送信優先度を付加する。優先度決定後、優先度が高い順にブロックにエンコードを施し、逐次送信を行う。このとき、エンコードにはフレーム間差分をとらないJPEGを利用する。また、優先度決定により重要度が低いブロックに関しては、送信する必要がないという考えのもとに送信を行わずに処理コストを減らす。受信側は、届いたブロックデータを順次デコードして位置情報をもとにデータを表示バッファに溜め込み、更新時間になると表示を行う。

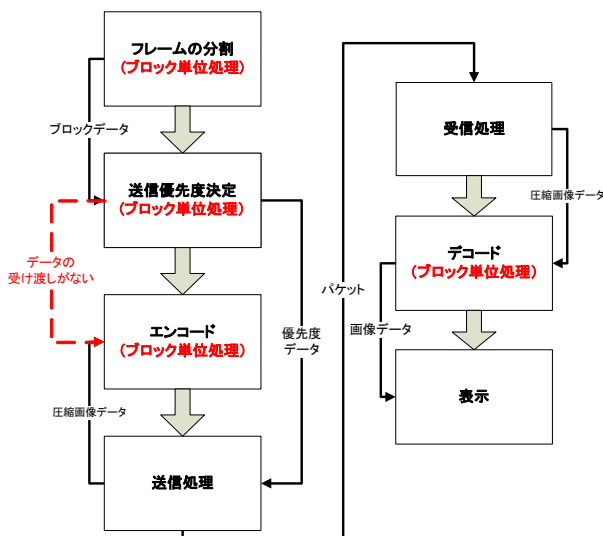


図1. 低遅延ライブビデオストリーミングシステム概要

このシステムの特徴はブロックそれぞれを個別にデコードできる点である。もし更新時間までにフレームデータすべてが届かなかったとしても受信ブロックだけでデコード・表示が可能である。また、優先度送信により重要領域が先に受信されるため、更新表示される可能性が高くなるという利点もある。

3. 並列化設計と解決すべき問題点

プログラムにおいて並列処理を実行するために必要な機構をスレッドという。スレッドは1プロセス内に複数の流れを生成することで、同時に処理を行うことを可能にする。スレッドを複数生成して、独立した処理の流れが並列実行されるようなプログラムをマルチスレッドプログラムという。

マルチスレッドプログラムにおいて並列化を行う際にはスレッドごとの処理コストが同等になるように分配を行う必要がある。そして、代表的な処理並列化手法としては、**データ並列化**、**タスク並列化**が挙げられる。データ並列化は処理するデータをスレッド数に応じて分離し、各データを独立して処理する手法である。殆どの場合、同様のデータに分割することができれば処理コストも均一になることが多く、本システムにおけるブロック単位の処理とも極めて親和性が高い。一方、タスクの並列化はデータではなくデータの処理を分配する手法である。並列化するタスクはデータの受け渡しなど時間的な前後関係が必要な処理は並列することができない。その点において、エンコードと送信優先度決定の処理は並列化可能である。この2点を考慮し並列化モデルを構築すると、ブロック単位の**データ並列化のみのモデル**およびエンコードと送信優先度決定処理の並列処理も行う**データ並列化・タスク並列化併用のモデル**の2つのモデルが考えられる。だが、これらのモデルで高い並列化性能を得るには、解決すべき問題が存在する。それは**スレッド数**、**ブロックサイズ**、**優先度決定手法**など構成要素の最適化である。特に、ブロックサイズや優先度決定手法は、処理遅延と映像品質のトレードオフを抱えており、単純に最適化することはできない。

4. チューニングのための指標作成

トレードオフを抱える要素のチューニングを行うには、映像遅延と映像品質を包括的に考慮した性能評価指標が必要である。そこで、まず任意のタスクを決め、映像遅延と映像品質がそれぞれ操作性にどう影響するかを計測し、その操作性によってシステムのトレードオフ要素を評価する評価指標を作成した。映像遅延および映像品質が操作性に与えた結果が図2の2つのグラフである。双方とも縦軸はタスク完了1回あたりにかかった時間(TCT)を表している。計測されたプロットをもとに2次関数で近似曲線を描いた。近似曲線として2次関数を選択したのは、x軸の値が0のときy軸のTCTが正の値をとり、かつシンプルに近似できるからである。

この2つ近似曲線より得られた映像遅延がないときのTCTの式、映像劣化がないときのTCTの式から映像遅延および映像品質が存在する際のTCTの式を算出することができる。これをもとにしてTCTによる当操作性曲線を描いたのが図3である。これが操作性による性能評価指標である。

列化モデルにおいて、同じ操作性指標によって性能評価を行った。
この結果、最適な並列化システムの構築を行うことができた。

6. 結論

本論文では、遅延時間及び映像品質が操作性に与える影響を正規化した評価指標のもとに、システムの構成要素のチューニングし、如何にして最適化された並列化システムを構築するか、その手段を順に述べた。

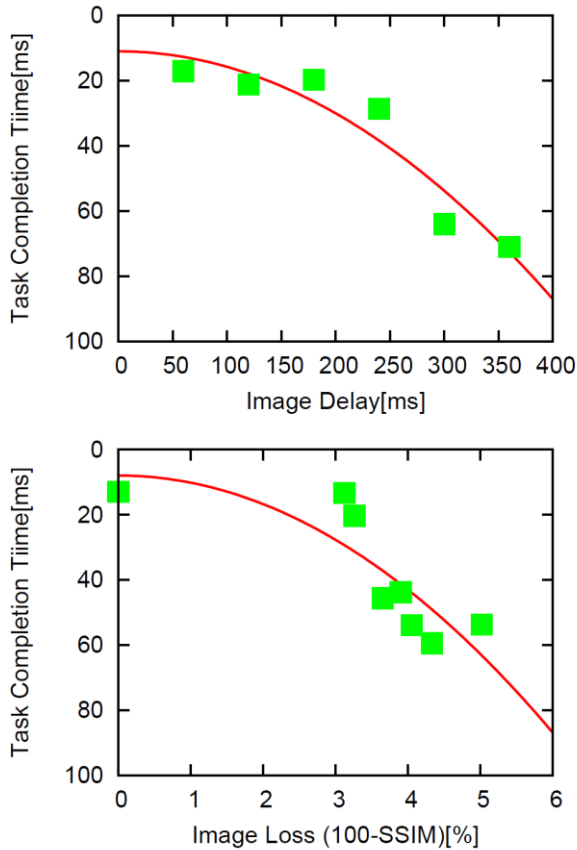


図2. 遅延および映像品質と性能評価値の関係

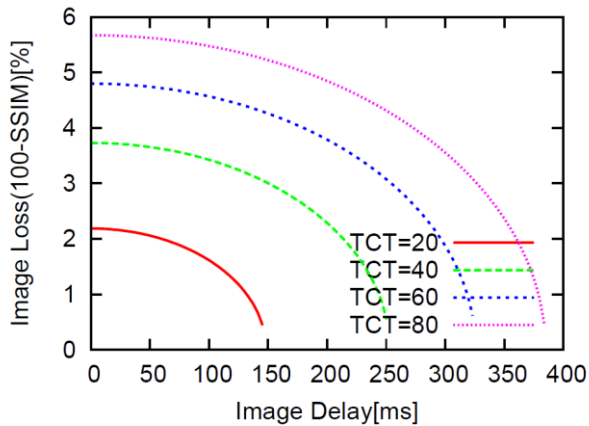


図3. タスク完了時間(TCT)評価による等操作性曲線

5. 並列化システムのチューニング

この評価指標をもとに要素のチューニングを行った。このTCTの値が最小になるような要素の値を選択すればよいのだが、それぞれの構成要素の値は互いに影響を与えているため、本来個別に要素をチューニングしていくことはできない。よって、近似的に最適化として、システム全体への影響が大きい構成要素からチューニングを行っていく。

優先度の決定方法はタスク並列化を使用するモデルにおいて、エンコードと優先度決定処理それぞれに割り振るスレッド数を決定するうえで重要となるため、あらかじめチューニングを行った。これにより最適なスレッド数と優先度決定手法を決定することができ、その後、ブロックサイズなど他の要素のチューニングを行った。

そして最後に、すべての構成要素のチューニングを行った2つの並