論文の内容の要旨

論文題目　FAST FOURIER TRANSFORM USING GPU

（GPU を利用した高速フーリエ変換）

氏名　　額田　彰

高速フーリエ変換(FFT)は現在様々なアプリケーションで用いられている最も重要な計算の一つである。FFT の高速化は直接これらの多くのアプリケーションの実行時間短縮に繋がるため大きな意義がある。現在様々な CPU アーキテクチャが乱立しているが、各アーキテクチャに対応した多くの高速化手法などが提案されており、また高度に最適化された FFT ライブラリも多く提供されている。

FFT の計算はメモリアクセスの比重が高く、その高速化には高いメモリバンド幅を持つ非常に高価な計算機システムが必要になる。そこで近年注目されているのが GPU による汎用計算(GPGPU)である。多数のコアを搭載することによる高い浮動小数演算性能と多数のメモリコントローラによる高いメモリバンド幅によって多くのアプリケーションが GPU により高速化を実現している。特に高いメモリバンド幅は FFT の計算に有効である。

しかしながら GPU により FFT の計算を高速化することは容易ではない。CPU と GPU のアーキテクチャの違いが大きく、また GPU ではプログラミングの制約も多いため、既存の CPU 用の手法では GPU の性能を引き出すことができない。CPU とは異なり多数のプロセッサコアを搭載する GPU の場合、従来のマルチスレッド型手法では GPU の計算資源をほとんど活用することができないために性能が低い。GPU メモリは高いバンド幅を持つが、連続アクセスや局所性の高いメモリアクセスパターンに最適化されているため、従来の多次元 FFT アルゴリズムにおける転置処理のメモリアクセス効率が著しく低い。また FFT の計算は入力サイズに大きく依存するため、それぞれの入力サイズに対してコードのチューニングを行うことは非現実的である。GPU アーキテクチャは 2 のべき乗サイズの FFT

計算の効率がよいが、それ以外では計算資源の無駄が生じる。FFT 計算を複数 GPU で行うこともある。この場合、クラスタ上での並列 FFT 計算と同様に GPU メモリ間での全対全通信が必要となり、特に GPU 内での計算が GPU により高速化されているため全対全通信が占める時間の割合が 8 割以上と非常に大きい。

本論文では、このような GPU を利用した FFT 計算に関わる課題に対して以下のような手法を提案し、高い性能を実現した。(1) GPU の多数のコアや shared memory を介したスレッド間データ交換やハードウェアによる同期などの機能を活用する細粒度並列な FFT アルゴリズムを提案した。(2) GPU で効率よく計算するため、転置の代わりにブロック化した multi-row FFT を拡張したカーネルを用いる多次元 FFT アルゴリズムを提案した。(3) 任意の入力サイズに対応するため、入力サイズの因数分解方法、スレッド数、shared memory で生じるバンクコンフリクトを回避するためのパディングの自動挿入などのパラメータについて網羅的探索により最適なものを決定する自動チューニング手法を提案し、多くの入力サイズにおいて CUFFT ライブラリの何倍もの性能を達成した。(4) 2 のべき乗以外のサイズに対しても、GPU の計算資源を無駄なく利用することができる Warp 単位のスケジューリング手法を提案した。(5) 複数 GPU を搭載するシステムにおいて、P2P 機能を活用したスケジューリングの最適化手法を提案し、複数 GPU を利用した高速化を実現した。(6) 複数ノードのシステムにおいても GPU 間の全対全通信性能の向上及び安定化のため、低レベルの IBverbs API を利用して小さいメッセージの送受信時のオーバヘッド削減、複数の RDMA 通信の同時実行によるネットワーク競合時のペナルティ軽減、複数の InfiniBand レールへの動的な RDMA 転送割り当てによりネットワーク競合を最小化、などの最適化手法を提案し、その結果ノード数が多い場合にも高い性能を確保し、最大 256 ノード(768GPU)で 4.8TFLOPS の性能を達成した。

以上のように、高速フーリエ変換の計算を GPU によって高速化するために必要な様々なアルゴリズムを提案した。様々な入力サイズや、様々な世代の GPU、シングル GPU 構成だけでなく GPU メモリから溢れるデータサイズ、単一ノードの複数 GPU を使う場合、大規模な GPU クラスタまであらゆる環境に対応可能であることを示した。