

平成25年度 情報工学コース卒業研究報告要旨

枝廣 研究室	氏 名	奈 加 大 樹
卒業研究題目	マルチ GPU を用いた C 言語と MATLAB プログラムの高速化	
<p>近年、コンピュータの中央処理装置である Central Processing Unit (CPU) のクロック周波数の向上による高性能化は消費電力、発熱量の問題から限界を迎えつつある。そこでさらなる高性能化を図る手段として、マルチコア技術やメニーコア技術が注目されている。1つのチップに多数のプロセッサコアを搭載し、プログラムの並列処理を行うことによって、クロック周波数を向上させなくても CPU の高性能化を実現できる技術である。現在、マルチコア CPU で並列処理できる限度は高々数十スレッドであるが、メニーコア技術では数千以上のスレッドを並列処理させることが可能となる。</p> <p>その中で注目されているのが、Graphics Processing Unit (GPU) である。GPU は一般的に画像処理に特化した補助演算装置であり、数百から数千のコアが並列に動作し、高速かつ繊細な画像処理を実現することが可能となる。この GPU の高い演算能力を汎用的な数値計算に利用する技術が General-Purpose computing on Graphics Processing Units (GPGPU) である。GPGPU は並列度の高いプログラムに高い能力を発揮し、現在では物理シミュレーションや信号解析、画像認識などの分野で広く使われるようになってきている。これらのプログラムでは GPGPU による高速化が実現できるが、大規模な並列度を扱うプログラムでは GPU のコア数を遥かに超えたスレッドを生成する必要があり、実行待ちスレッドによる遅延が大きくなってしまいうという問題を抱えている。</p> <p>本研究では、GPGPU を用いたプログラムのさらなる高速化について、複数の GPU にスレッドを分担し実行待ちスレッドを減らすマルチ GPU 処理を実装し、評価を行った。評価に用いたモデルは、C 言語と MATLAB の画像認識プログラムである。MATLAB は高い数値計算機能を備えた技術計算言語であり、多くの画像認識プログラム等が MATLAB によって書かれている。NVIDIA 社が提供する GPGPU の統合開発環境である CUDA は C 言語をベースとしており、MATLAB に直接実装することはできない。しかし、プログラムの全てを C 言語に変換しようとするプログラミングの生産性の面で大きな問題がある。そこで本研究では MATLAB で GPGPU を行うためのフレームワークについても紹介する。</p> <p>マルチ GPU 処理の実装では、POSIX 仕様におけるスレッド処理の標準である pthread を用いた。GPU の数だけ pthread を生成し、各 pthread にそれぞれ GPU を割り当て、各 GPU に担当範囲を並列計算させる処理を実装した。マルチ GPU 処理を実装したプログラムの実行時間を GPU が 1 枚から 4 枚の場合でそれぞれ計測したところ、全てのプログラムで GPU の数を増やすほど実行時間が高速化することが確認できた。また、マルチ GPU 処理は並列度が大規模なプログラムであるほど高い能力を発揮することを示した。</p> <p>今後の課題としては、プログラミングの生産性の向上、デバイスメモリの確保と開放の並列化が挙げられる。GPGPU は元々プログラミングの生産性の面で問題を抱えており、マルチ GPU 処理ではプログラムの構造がより複雑なものとなるため、プログラマの負担は非常に大きなものとなる。これを改善する手段としては、GPU 処理、マルチ GPU 処理を定式化し、半自動的に実装を行うことができるようなツールの開発などが考えられる。</p>		