

電気・電子情報工学専攻	学籍番号	M123209	指導教員氏名	市川 周一 藤枝 直輝
申請者氏名	板垣 佑哉			

論文要旨 (修士)

論文題目	GMP Library における多倍長整数乗算の高速化手法
------	-------------------------------

現在の数値計算では、CPUがハードウェアで処理可能なIEEE754浮動小数点数が多く用いられる。しかし、円周率など大きい桁数を要求する計算や精度保証が必要な計算など、より高い精度が必要な場合、多倍長計算が用いられる。現状のシステムにおけるCPU等のハードウェア構成では、多倍長計算はソフトウェアで実装され、IEEE754浮動小数点演算に比べ計算時間が長くなる。

多倍長計算を高速、簡易に実行するための多倍長計算ライブラリは多く開発されている。その中でもGNUが提供するGMP Libraryは比較的高速で広く用いられている。四則演算は基本演算として多く実行されるが、加減算に比べ、乗除算は計算時間が大きくなることが多い。CPUのマルチコア化とともに並列化による高速化が進んでいるため、OpenMPを用いたGMPの多倍長整数乗算の高速化手法を検討した。

多倍長計算ライブラリでは4倍長、8倍長など固定桁数を用いる場合もあるが、本論文では計算に必要な任意の桁数を扱う多倍長計算を対象とする。多倍長乗算のアルゴリズムのひとつとしてKaratsuba法が挙げられる。基本のアルゴリズムに比べ、乗算回数は4分の3に減少する。

GMPのmpz_mul関数は可変長整数型 (mpz_t) を扱う多倍長整数乗算を行う関数である。mpz_tは整数型64 bitの配列へのポインタと配列サイズをもち、必要に応じて配列を再割り当てすることで多倍長数を実現している。この時の配列サイズをsizeと定義する。mpz_mulでは被乗数と乗数のsizeによって乗算の手順は異なる。本論文では、乗数のsize = 1 の時 (case 1) と、被乗数と乗数のsizeが同じ時 (case 2) の乗算に対して高速化を行った。

本論文の提案手法では、mpz_mul関数における計算手順case 1, 2において並列化を行う。case 1では、被乗数を分割し、分割した乗算に対して並列化を行った。この時の並列度は2, 4, 8とした。case 2では、最初にKaratsuba乗算を行い、依存性のない乗算に対して並列化を行った。この時の並列度は3, 9とした。

評価方法として、case 1, 2それぞれの乗算方法に対してサンプルプログラムを作成し実行時間を測定した。case 1の時には、10進数 n 桁 \times 64 bit (size = 1) の乗算をcase 2の時には、10進数 n 桁 \times 10進数 n 桁の乗算を行う。それぞれ、最低1秒間ループさせたときの乗算の実行回数および実行時間から1秒当たりの実行回数 (OPS: Operations Per Second) を算出し、この処理を3回実行した時の中間値を測定した。評価の結果、case 1では並列度8において、 $n = 3 \times 10^7$ の時に元のライブラリと比較して、7.727倍の高速化を示した。case 2では並列度3において、 $n = 5 \times 10^5$ の時に元のライブラリと比較して、1.699倍の高速化を示した。

また、case 1の乗算を含むアプリケーションとして、Machinの公式を使用した π の計算を行うプログラムを作成した。これについて評価を行った結果、測定した範囲での提案手法の実行時間は改良前の1.1倍~2.0倍であった。今後の課題として、mpz_mul関数の並列化適用箇所を増加を検討する必要がある。