

GPU プログラムの進化的計算によるパラメータチューニング手法の提案

吉見真聡[†] 廣安知之^{††} 三木光範[†]

1. はじめに

本研究では、プログラムのパラメータチューニング手法として進化的計算を使用する方法を提案する。並列性が高く、広く知られている Smith-Waterman アルゴリズムを対象に、マルチコアプロセッサと GPU でパラメータによる計算時間の評価を提示し、進化的計算によるパラメータチューニングが有効であることを確認する。

2. 進化的計算を用いたパラメータチューニング手法

一般的にアルゴリズムの実装は、並列計算を行う部分問題のサイズをパラメータライズしていると考えられる。また、マルチコアプロセッサで同時に実行するスレッド数や、GPU のカーネル起動時に指定する Block と Thread の値のように、プログラムが実行されるプラットフォームに関するパラメータも存在する。これらのパラメータはプログラムの運用前に実行と評価を繰り返して調整されると考えられるが、進化的計算を利用することにより、パラメータ調整に要するコストを削減する。

3. Smith-Waterman アルゴリズム

Smith-Waterman アルゴリズム (以下、SW 法) は、2 本の文字列から類似部分文字列を抽出するアルゴリズムである。SW 法は、図 1 に示すように、文字列 2 本の類似度が格納されるスコアテーブルにスコアを計算していく。SW 法は高い並列性を持つことが知られており、図 2 に破線で示す対角線方向のブロックは並列に計算できる。ブロックはスコアテーブルを分割した部分テーブルであり、個々の計算コアが担う部分計算の単位である。GPU で SW 法を高速に実行する様々な実装が試みられており、マイクロプロセッサと比べて優れた性能が報告されている¹⁾。

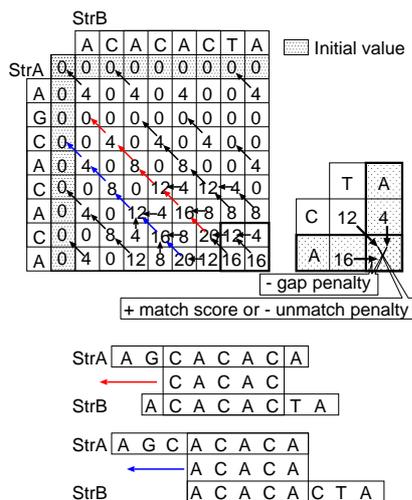


図 1 Smith-Waterman アルゴリズム

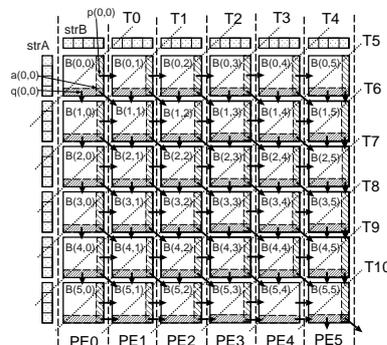


図 2 SW 法の並列性とデータフロー

4. SW 法の並列実装

本研究の有効性を確かめるために、図 3 にその概要を示すような並列プログラムを実装した。マスタスレッドはスコアテーブルを保持し、部分ブロックをジョブキューに格納する、SW スレッドはジョブキューから取り出した情報を元に部分ブロックのスコアを計算する。SW 法を、マイクロプロセッサおよび GPU で計算するプログラムをそれぞれ実装した。前者は C 言語と pthread ライブラリを用い、スレッド数と部分ブ

[†] 同志社大学 理工学部
^{††} 同志社大学 生命医科学部

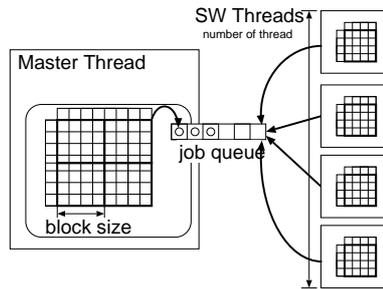


図 3 SW 法の並列実装

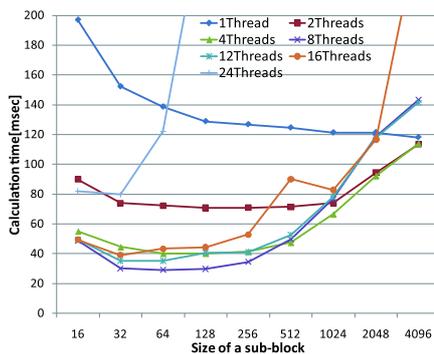


図 4 Calculation time versus number of thread with various block size for logical 8 cores (physical 4 cores) Sandy-Bridge processor

ロックのサイズをパラメータとして指定できる．後者は CUDA のプログラミングモデル上で，1つの部分ブロックで計算するサイズを Thread で指定し，部分ブロック数だけ Block で指定することができるようになっている．

5. 評価

4節で述べた SW 法のプログラムを表 1 および表 2 に示す環境で実行した．その結果を図 4, 図 5 および図 6 に示す．これらの図に示したように，部分ブロックが小さい場合と大きい場合に計算時間が長くなる傾向があり，パラメータチューニングが有効に働く可能性が確認されたと言える．GPU はブロックの切り替えに要する時間がブロックの計算時間に比べて大きいことが図 6 に現れていると考えられる．C2050 の傾向が異なる理由は，fermi 世代の GPU が持つハードウェアキャッシュが原因であると考えられる．

表 1 マルチコア CPU 環境

CPU	Opteron 2423HE (6 cores×2) 2.0GHz	Core i7-2600 4 cores 3.40GHz
RAM	12 GB	16 GB
OS	Linux 2.6.18 X86_64	
Compiler	gcc-4.3.2 (-O3 -lpthread)	

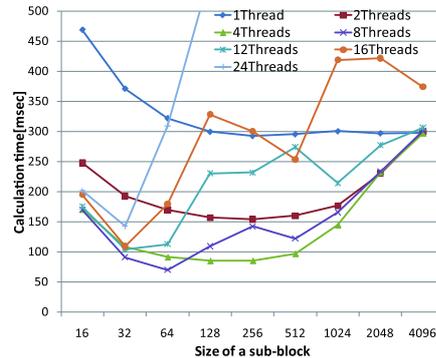


図 5 Calculation time versus number of thread with various block size for 12 cores (two six-cores) Opteron processor

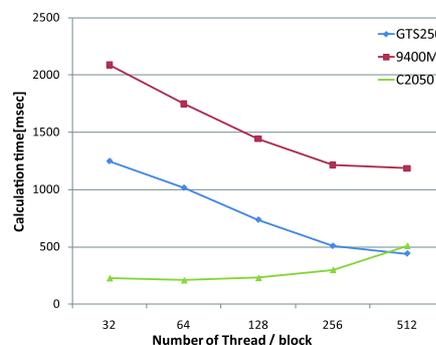


図 6 Calculation time versus number of thread (block size) on GPUs

6. まとめ

本研究では進化的計算を用いたパラメータチューニングによる GPU プログラムの最適化を目標に，SW 法を用いてパラメータが計算時間に及ぼす影響について調査した．実験の結果から，パラメータチューニングが有効に働く可能性が確認された．

参考文献

- 1) Dohi, K. and *et.al.*: Highly efficient mapping of the Smith-Waterman algorithm on CUDA-compatible GPUs, *ASAP'10*, pp.29–36 (2010).

表 2 GPU 環境

GPU	C2050	GTS250	9400M
cores	448	128	48
RAM	3.0 GB	512MB	256MB
Clock rate	1.15GHz	1.84GHz	0.95GHz
CPU	Xeon W3530 4 cores×2 2.80GHz	Core i7-2600 4cores 3.40GHz	C2D SU9400 2 cores 1.40GHz
OS	Linux 2.6.18 X86_64		
Compiler	gcc-4.3.2 (-O3 -lpthread)		
nvcc	3.1	3.2	3.2