C-032

FPGA アレイ Cube を用いた文字列編集距離の性能評価
Exploratory Performance Evaluation of Edit-Distance on Cube

吉見 真聡 * 三木 光範 * 天野 英晴 † オスカー メンサー ‡
Masato YOSHIMI Mitsunori MIKI Hideharu AMANO Oskar MENCER

1. はじめに

科学計算に要するデータ量や演算量は現在でも増大を続けており，それに伴う電力消費は莫大なものとなってきている．近年の Top500 ランキングでは，PC クラスタがその大部分を占めており，広い汎用性を持つ高性能計算システムとして運用されている．しかしその一方で，実行するアプリケーションをある程度制限して，高い演算性能を維持したまま運用コストを大幅に削減しようとする，専用ハードウェアを使用した計算システムの研究も盛んになってきている．なかでも FPGA は，ハードウェア構造を柔軟に構成できる性質から，科学計算用のアクセラレータとしての可能性が広く検討されてきた．

本研究報告では，FPGA アレイ Cube を対象に文字列編集距離計算のアプリケーションを実装し，その性能について検討する．この例は入出力データに比して演算量が多く，豊富な並列性を持つため，計算システムの性能評価の目的で広く使用されるアプリケーションである．評価結果は，近年のマルチコアプロセッサを用いたマルチスレッド実行の性能と比較する．

2. Cube

2008 年に開発された Cube は，一次元接続された多数の FPGA による並列計算プラットフォームである [1]．図 1 にその構造を示すように，Cube は FPGA が 8 × 8 の正方形状に配置されたボードが最大 8 枚された計算システムである．Cube の各 FPGA 間，ボード間は 64 ビットの信号線で一次元に接続され，最大 512 個の FPGA で構成される．FPGA アレイを通過する際に計算が行われるので，実行対象のアルゴリズムがパイプライン，シストリックアレイで実行できる場合には，高い性能が期待できる．Cube は強力なエネルギー性能比が得られるシステムであるものの，現在はアプリケーション実装例が多いとは言えず，性能評価や高速実行に関する検討は初期段階にある．

3. 文字列編集距離

本研究報告で Cube の対象アプリケーションとして選択した編集距離 (レーベンシュタイン距離) の計算は，2 つの文字列の距離を求めるアルゴリズムである．一方の文字列に対し，文字の (1) 挿入，(2) 削除，(3) 置換の手順を繰り返し，他方の文字列に変換するために要する手順の最小回数が編集距離となる．編集距離は，画像認識の際のパターンマッチング処理や，遺伝子やアミノ酸配列を対象としたデータマイニング，文字の綴り間違いの検出・修正など，様々なアプリケーションにおいて基本的なアルゴリズムとして利用されている．

*同志社大学理工学部
†慶應義塾大学理工学部
‡インペリアルカレッジ・ロンドン

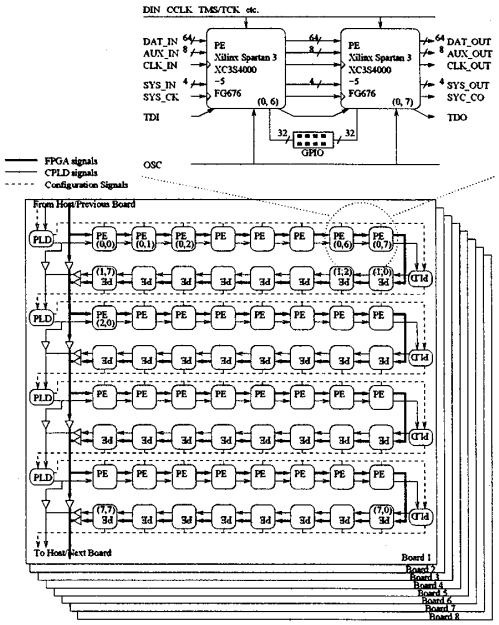

図 1: Cube のアーキテクチャ

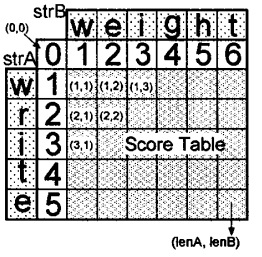

図 2: アルゴリズムの概要

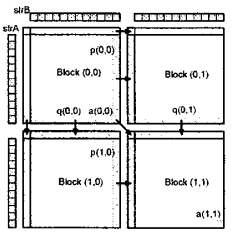

図 3: 編集距離計算のブロック化手法

また，様々なレベルで高い並列性を持つアルゴリズムとしても知られており，Cell/B.E. や GPU などのマルチコアを対象に実装の高速性を競うコンテストが開催されるなど，ベンチマークアプリケーションとしても利用されている [2][3]．図 2 にアルゴリズムの概要を示す．編集距離は文字列 `strA` から `strB` に変換するために要する処理回数を数えることで求める．これは，スコアテーブル c の要素に処理回数を書き込んでいくことで求められる．各文字列長が `lenA` および `lenB` のとき，テーブルの大きさは $(\texttt{lenA}+1)\times(\texttt{lenB}+1)$ である．テーブルの要素 $c(i,0)$ および $c(0,j)$ は，i および j で初期化される．要素 $c(i,j)$ の値は，隣接要素の値 $c(i-1,j-1)$，$c(i,j-1)$，$c(i-1,j)$ を用いて以下の手順で求められる．(1) `strA` の $i-1$ 番目の文字と `strB` の $j-1$ 番目の文字を比較し，

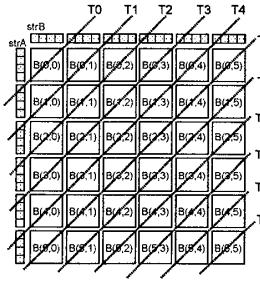

図 4: 並列実行アルゴリズムの概要

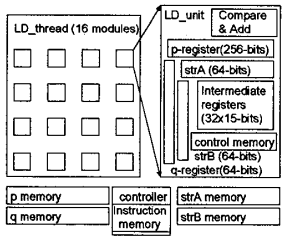

図 5: 編集距離計算のブロック計算モジュール

表 1: C プログラム実行環境

CPU	Intel Core 2 Quad Q6600 @ 2.4GHz
RAM	4.0 GByte
OS	GNU/Linux 2.6.23 X86_64
Compiler	gcc-4.1.2(-O3 -lpthread)

等しければ 1，そうでなければ 0 を一時変数 a に代入する．(2)$c(i-1,j-1)+a$，$c(i,j-1)+1$，$c(i-1,j)+1$ のうち，最小値を $c(i,j)$ に格納する．(1)-(2) の手順を繰り返してスコアテーブルを埋めていき，$c(\mathtt{lenA},\mathtt{lenB})$ の値が編集距離になる．

4. 設計

4.1 編集距離の並列計算法

図 3 に示すように，編集距離計算のスコアテーブルは複数の計算ブロックに分割してそれぞれ並列に計算を進めることができる．計算ブロック (i,j) は，strA と strB の部分文字列，中間データ $p(i,j-1)$ と $q(i-1,j)$，およびブロックの左上端の要素の値 $a(i-1,j-1)$ の 5 つが入力として与えられ，計算の結果，$p(i,j)$，$q(i,j)$，$a(i,j)$ を隣接ブロックの計算のために出力する．図 4 に破線で示すように，並列実行可能なブロックが存在するので，複数の演算リソースを使用して高速化を図ることができる．並列計算可能なブロック数は，計算の進展にともなって増減する．

4.2 Cube における計算手法

Cube を対象とした編集距離計算の実装では，各 FPGA に図 3 に示した 1 つのブロック計算モジュールを構成し，図 6 の手順で分散処理を行う．

ブロック計算モジュールの構成を図 5 に示す．各 FPGA は 8×8 文字の編集距離計算を行う LD_unit モジュールを 16 個持ち，合計で 128 × 128 文字の計算を行う．Cube は 100[MHz] で動作し，512 個の FPGA を用いて最大 64k × 64k 文字を 1 ブロックとした並列計算を実行する．

5. 評価

実装したハードウェアモジュールについて評価するため，FPGA 単体，Cube ボード 1 枚 (FPGA64 個) および 8 枚 (FPGA512 個) で編集距離計算を実行した場合の性能を求めた．また，Cube の性能と比較するため，pthread ライブラリを使用してマルチスレッド実行に対応した C プログラムを実装した．このプログラムを表 1 に示した環境で，4096 × 4096 文字を 1 ブロックとして 1 スレッド，8 スレッドで並列実行した場合の計算性能と比較した．その結果を図 7 に示す．図 7 に示したように，Cube では FPGA 数に従って性能が向上していることが明らかになった．また，C プログラムでもプロセッサのコア数に従って計算時間が向上している．一方で，両者は問題の文字列長が長くなった場合の性能向上が得られなくなっているため，モジュールのパイプライン効率等の検討を行い，より高いスケーラビリティを得る必要があると言える．

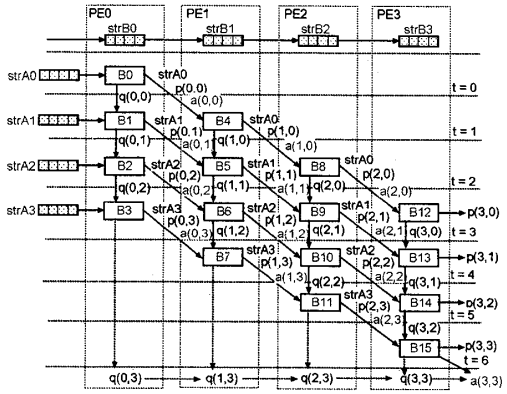

図 6: Cube 上の計算手順とデータフロー

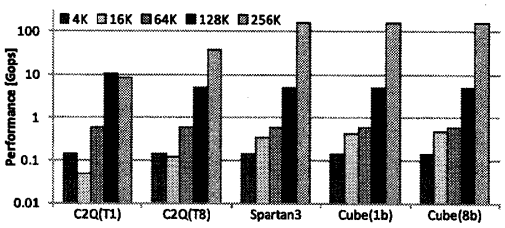

図 7: 計算性能の比較

6. まとめと今後の展開

本研究を通して，単純な編集距離計算において，Cube がマルチコアプロセッサでのマルチスレッド実行と比較して約 300 倍程度の計算能力を示すことが確認された．今後は，Cell/B.E. や GPU での実装との性能比較や，電力消費量に注目した評価を通して，編集距離計をはじめとしたストリームアプリケーションにおける Cube の利点を示すとともに，ハードウェア実装の検討を通して，より問題サイズに対してスケーラビリティを得られるハードウェア構造について検討する予定である．

参考文献

[1] Mencer, O., Tsoi, K. H., Craimer, S., Todman, T., Luk, W., Wong, M. Y. and Leong, P. H. W.: CUBE: A 512-FPGA CLUSTER, *Proc. IEEE Southern Programmable Logic Conference (SPL'09)* (2009).

[2] CellChallenge2009 実行委員会:http://www.hpcc.jp/sacsis/2009/cell/.

[3] GPUChallenge2009 実行委員会:http://www.hpcc.jp/sacsis/2009/gpu/.