GTC Workshop Japan 2011 (2011 年 7 月 22 日)

最長共通部分列を求めるビット並列アルゴリズムの CUDA による高速化

GPU Computing for the Longest Common Subsequence with Bit-Parallelism

藤本　典幸, 河南　克也
Noriyuki Fujimoto and Katsuya Kawanami

大阪府立大学　大学院理学系研究科　情報数理科学専攻（〒 599-8531 大阪府堺市中区学園町 1-1）

Key Words: *CUDA, bit-parallelism, dynamic programming*

1　はじめに

2 つの文字列の類似度を示す指標の 1 つに最長共通部分列（Longest Common Subsequence，以下 LCS）[2] がある．ここで文字列 S に含まれる各文字が S に出現する順序で文字列 T 中に出現するとき，S は T の部分列であると言い，2 つの文字列 S_1 と S_2 の両方の部分列である文字列のうち，最長のものを S_1 と S_2 の LCS と言う．例えば文字列 abcdefghij と cfilorux の LCS は cfi，文字列 abcde と baexd の LCS は ad, ae, bd, be である．LCS は遺伝子配列の比較や文字列のあいまい検索，スペルチェックなど，様々なことに応用できる．

2 つの文字列の LCS の 1 つを時間計算量 $O(mn)$ と空間計算量 $O(m+n)$ で計算する逐次アルゴリズムが Hirschberg によって提案されている [3]．このアルゴリズムは再帰的であり，再帰の度に LCS の長さ（Length of LCS，以下 LLCS）を計算するが，この LLCS 計算が Hirschberg のアルゴリズムを支配している．ビット並列計算を用いて LLCS を計算する Crochemore らのアルゴリズム [1] を適用すると，空間計算量は $O(m+n)$ のままで，Hirschberg のアルゴリズムの時間計算量を $O(\lceil m/w \rceil n)$（w は計算機のワードサイズ）に改善できる．しかし遺伝子配列等の比較では長さ 100 万文字以上の文字列を扱うので，更なる高速化が必要である．そこで我々は GPU を用いて LCS を更に高速に計算する方法を研究している [4].

2　提案手法

本発表では Crochemore らのビット並列アルゴリズムを用いて改善した Hirschberg の CPU 用 LCS アルゴリズムを，GPU を用いて高速化する方法を提案する．Crochemore らのアルゴリズムは多倍長のビット単位論理演算の他に，逐次性が強い多倍長の算術加算を含んでいる．ビット単位論理演算は GPU 並列化が容易であるが，算術加算は GPU での実装に工夫が必要である．本研究においては，スレッドブロック間においてはウェーブフロント型の並列化を，ブロック内スレッド間においては Sklansky の全加算器並列化法 [6] を用いることによりこの問題を解決した．

3　評価実験

本発表で提案する手法に基いて CUDA でビット並列アルゴリズムを実装し，長さ 27 万文字から 1677 万文字の文字列に対して，2.93GHz Intel Core i3 530 CPU の 1 コア（SSE 命令は用いていない）と GeForce 8800 GTX，GTX 285，GTX 480 GPU を用いて評価を行った．なお，OS は 64bit Windows 7 Professional，コンパイラは Visual Studio 2008 Professional（コンパイルオプションは Release モードのデフォルト設定）を用いた．使用した CUDA のバージョンは 3.1，グラフィクスドライバのバージョンは 266.58 である．その結果，CPU 上でのビット並列アルゴリズムに対しては最大 12.77 倍，Hirschberg の CPU 用 LCS アルゴリズムに対しては最大 76.5 倍高速であった．また，Kloetzli らの GPU を用いた既存アルゴリズム [5] に対しては 10.9 倍から 18.1 倍高速であった．

参考文献

[1] M. Crochemore, C. S. Iliopoulos, Y. J. Pinzon, and J. F. Reid: A Fast and Practical Bit-Vector Algorithm for the Longest Common Subsequence Problem, Information Processing Letters, Vol.80, No.6, pp.279–285 (2001)

[2] D. Gusfield: Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology, Cambridge University Press (1997)

[3] D. S. Hirschberg: A Linear Space Algorithm for Computing Maximal Common Subsequences, Communications of the ACM, Vol.18, No.6, pp.341–343 (1975)

[4] 河南克也, 藤本典幸: GPU を使用したビット並列アルゴリズムに基く最長共通部分列の導出, 先進的計算基盤システムシンポジウム（SACSIS）, pp.365–372 (2011)

[5] J. Kloetzli, B. Strege, J. Decker, and M. Olano: Parallel Longest Common Subsequence Using Graphics Hardware, proc. of the 8th Eurographics Symposium on Parallel Graphics and Visualization (EGPGV) (2008)

[6] J. Sklansky: Conditional-Sum AdditionLogic, IRE Transactions on Electronic Computers, EC-9, pp.226-231 (1960)