1ZF-3

GPU配列相同性検索ツールの マルチGPU向け最適化と長断片への対応

坂田　幸佑 † 鈴木　脩司 ††　石田　貴士 ††　秋山 泰 † ††
† 東京工業大学 工学部情報工学科　†† 東京工業大学 大学院情報理工学研究科　計算工学専攻

1　序論

近年、次世代シーケンサーと呼ばれるハイスループットなゲノム解読装置が登場し、大量のDNA断片を短時間で得ることが可能になったが、特にメタゲノム解析ではデータの情報解析が追いつかないという深刻な問題が発生している。これに対して先行研究“GHOSTM”[1]はシングルGPUを用いて処理することで配列の相同性検索の高速化を実現させた。しかし大規模なデータには更なる高速化が必要であり、マルチGPU環境への対応が期待されている。また最新型のシーケンサーから得られる配列断片が長くなりつつあることで、GHOSTMではGPUのメモリ容量の制限内で処理するには設計変更が必要である。そこで本研究では、GHOSTMを更に高速化するために、マルチGPU環境向けの最適化実装を行い、また同時にと長断片への対応に取り組んだ。

2　方法

2.1　GHOSTM

GHOSTMは先行研究で開発された配列相同性検索ツールであり、NVIDIA社が提供するGPGPU向けの総合開発環境であるCUDAを用いて実装されている。また対象とするDB（データベース）とクエリはDNA配列及びタンパク質配列の両者に対応している。相同性検索における手順は、まずDBとクエリにそれぞれ前処理を施し、それらの結果を用いてGPUでSmith-Waterman (SW)アルゴリズムを用いて局所的アラインメントを計算し、そのうちスコアの高いものを出力するというものである。DBとクエリの前処理についてはある文字数で配列を分割し、分割で得られた文字列をインデックスとして記録する。これにより文字列の問い合わせと、局所的アラインメントを計算する候補の探索を円滑に処理することが可能となる。

Optimizing GPU based homology search tool for multi-GPU system
† Kousuke Sakata(sakata.k.th@bi.cs.titech.ac.jp)
†† Shuzi Suzuki(y_suzuki@bi.cs.titech.ac.jp)
†† Takashi Ishida(y_t.ishida@bi.cs.titech.ac.jp)
† †† Yutaka AKIYAMA(akiyama@cs.titech.ac.jp)
Depart of Computer Science,Tokyo Institute of Technology (†)
Graduate School of Information Science and Engineering,Tokyo Institute of Technology (††)
Ookayama 2-12-1-W8-76, Meguro-ku, Tokyo, 152-8550 Japan.

SWアルゴリズムは正確であるが速度が遅いという欠点があるが、GPUで実行することによりの高速化が可能なことが知られている[2]。

2.2　GHOSTMの並列化

本研究で行った並列処理について説明する。本研究はGHOSTMの高速化を図るためにマルチGPU環境への対応を行った。マルチGPU環境に対応するには使用するGPU数と等しいCPUスレッドが必要となるためスレッド化にはPOSIX thread（pthreads）ライブラリを用いた。Pthreadsを用いて並列処理を施した部分は局所的アラインメントの候補探索とそのスコア計算の部分である。

各スレッドに与えるデータは前処理で分割処理を施したクエリデータである。そのため各クエリについての計算の間では通信が必要なく、それぞれが独立に処理をする。そのため、単にクエリの分割によりバッチ処理によるプロセスレベルの並列化が可能であるが、ノード内ではI/Oやメモリ効率、処理の自動化を考えてスレッドレベルでの並列化を目指した。

現段階では実装中であるが、上記のスレッドレベル並列化によるメモリ効率はメモリ空間が大きなGPUについて非常に有効であると考える。例として1GPUで3.1のクエリ100万本のデータを処理する場合約4GBのメモリ空間を要する。これをバッチ処理でプロセスレベルで実行する場合、ホスト側であるCPUは(実行するプロセス数)×4GBのメモリ空間を要することになり、大規模なデータを扱うメタゲノム解析の場合計算機環境によっては懸念すべき問題となる。

3　実験結果

本研究で利用したマルチGPU環境は東京工業大学が保有しているTSUBAME 2.0スーパーコンピュータである。使用したCPUはXeon X5670（2.93GHz）、GPUはNVIDIA Tesla M2050（1.15GHz）であり、ノードあたり12 CPUコア、3 GPUが搭載されている。以下に実験手順と結果について述べる。

3.1　使用データ

本研究では既知タンパク質配列のDBとして京都大学KEGG[3]のWebサイトから2009年4月30日にダウンロードしたタンパク質配列データ（gene.pep）を使用した。このDBの中の配列の本数は約423万

本、全配列の合計長は約15億塩基である。クエリファイルは東工大大学院生命理工学研究科の黒川顕教授のグループから頂いた60塩基対 の 1000 万本の DNA 配列を頂き、計算比較用にランダムに選択した1000、3000、1万、10万、100万本の DNA 断片配列を使用した。

3.2 GHOSTM のオプション

インデックスとして用いる部分文字列の長さ K は K=4 に固定した。それ以外のパラメータは、先行研究 [1] に準拠し、一般的な相同性検索ツール (BLAST) と比較して速度、精度ともに最適であることが確認されている値を用いた。

3.3 BLAST との比較

一般的なツールである BLAST との比較にはランダムなクエリ1万本を用いた。理由は BLAST の処理時間が膨大になるためである。今回使用した BLAST のオプションは以下の通りである。

```
$ blastall -p blastx -G 8 -E 8 -g T -F F -b 10 -v 10 -a 1 -e 50 -M PAM30
```

表1に示すように GHOSTM の方が約 40 倍高速でであった。

3.4 並列数

TSUBAME 2.0 の各ノードは3つの GPU が載っており、スレッドレベルでの並列数は3が上限である。そのため、大規模な並列化を行うにはノード数を増やす必要がある。現段階ではノード内でのマルチ GPU 環境に対応しているが、ノードをまたいだ並列化ではデータ分割による分散実行と MPI ライブラリを用いた同時並列実行とを検討している。

3.5 計算速度比較

速度比較には 3.1 節で述べたランダムなクエリについて行った。それぞれ GHOSTM をシングル GPU で実行した場合と 3GPU で実行した場合を比較した。速度の計測には time コマンドを用いた。結果の表2の値は各処理を5回実行した平均値である。また速度比は 1GPU/3GPU の計算時間の比を計算した値である。

4 結果と考察

クエリの本数が多い場合は速度向上比が約 3 倍となっており、結果から GHOSTM のマルチ GPU 環境への対応ができたといえる。またクエリの本数が少ない場合は実行時間とクエリの本数の間に比例関係がないことがわかる。これは GPU の処理より I/O での処理が全体の計算時間に影響を与えているためである。また配列断片が長くなったことによるメモリ使用量の問題も解決することができた。マルチ GPU ではそれぞれの GPU が独立にメモリを確保しているので N 枚の GPU を使用すれば N 倍のメモリ空間を得ることが可能となる。本研究のツールは長断片であるクエリであったとしてもそれを分割して GPU の数だけ並列処理を実行する。これにより豊富な計算機環境であればメモリ空間の問題を解決することが可能となる。

	計算時間
GHOSTM	202
BLAST	7922

表 1: GHOSTM と BLAST の計算時間（秒）

クエリ本数	1GPU	3GPU	速度向上比
1000 本	43.2	45.3	0.95
3000 本	61.8	49.2	1.26
1 万本	123.7	75.3	1.64
10 万本	938.2	344.5	2.72
100 万本	9155.0	3148.0	2.90

表 2: 各クエリにおける計算時間の比較（秒）

5 結論

本研究では GPU 配列相同性検索ツールのマルチ GPU への対応を提案し，速度向上を確認した。今後の課題としては並列処理におけるメモリ効率を見直し、また更に処理を円滑にするためにクエリの本数から自動的にノード数を割り振り TSUBAME 2.0 上で GHOSTM を利用しやすいように改良を進めていくことである。

参考文献

[1] 鈴木 脩司, 石田 貴士, 秋山 泰, GPU による DNA 断片配列の高速マッピング, 情処研報,2010-BIO-21(30):1-6,(2010)

[2] *Manavski SA, Valle G,CUDA compatible GPU cards as efficient hardware accelerators for Smith- Waterman sequence alignment, BMC Bioinformatics, 9:26,（2008）*

[3] *Kanehisa, M., Goto, S., Furumichi, M., Tanabe, M., and Hirakawa, M,KEGG for representation and analysis of molecular networks involving diseases and drugs, Nucleic Acids Res, 38:355-360,（2010）*