

# GPUを用いた行列演算ソフトウェアの開発

土居 主尚<sup>1,a)</sup> 坂部 啓

**概要:** 大規模疫学データを解析する状況で数万人のデータに対し統計解析モデルを用いる場合、一般に条件分岐が少なく大規模なデータに対し同じ計算を行う。我々は画像処理を専門とする GPU(Graphics processing unit) がこのような計算に適していることに着目し、GPU による汎用の数値計算 (General-purpose computing on graphics processing units, GPGPU) を用いた統計解析ソフトウェアの開発に着手した。このソフトウェアは統計解析パッケージとして有名な R のライブラリとして実装した。既存の R プログラムも最低限の書き換えで本ソフトウェアの恩恵を受けられるよう、R の機能である演算子のオーバーライドを利用し、既存の関数を本ソフトウェアの関数に置き換える機能を持たせた。またデータを GPU が使用するビデオメモリ領域に転送するオーバーヘッドを削減するため、一度ビデオメモリに転送した後はそのまま保持できるようにした。またコンピュータのメインメモリに比べ容量が小さなビデオメモリを有効に活用するため、ビデオメモリでしばらく使われなかったデータを自動的にメインメモリに戻すことでビデオメモリの利用効率改善を図っている。

## 1. はじめに

近年、統計解析の分野では従来よりも大規模なデータを扱ったり、またベイズ統計学のように反復計算を前提とした手法が頻繁に用いられるようになってきている。このため一つの計算に数日かかることもあり、計算の高速化が望まれている。しかし、計算を行っている CPU の動作速度は近年伸び悩んでおり、CPU を使う限りは著しい計算の高速化は困難である。一方、グラフィック描画のための専用計算機である GPU(Graphics processing unit) は近年の三次元グラフィックス性能の向上と共にその性能を向上させている。GPU は CPU が有している条件分岐などのオーバーヘッドを隠蔽するための機構を省略し、その分のリソースを演算プロセッサに割り当てることにより浮動小数点演算性能を高めている。また、グラフィック描画のために多量のデータに対し同一の処理を行うよう多数のプロセッサを集積し、これに適したメモリ構造としている。GPU はこのような特徴を有しているため、グラフィック描画と同様に大量のデータに対し同一の処理を行う状況では高いパフォーマンスを発揮する。そのため、汎用の数値演算に GPU を用いる GPGPU (General-purpose computing on graphics processing units) が近年注目を浴びている。

本研究では、統計解析の分野で用いられる手法の多くが

GPU に適した多量のデータに対し同一の処理を行うものであることに着目し、統計解析への GPGPU の適用を検討した。適用対象ソフトウェアとして、統計パッケージ「R」を選択した。R は解析の現場で頻繁に用いられるが、それは R が無償で利用できるだけでなく、開発者とユーザーが比較的近く、多くの解析手順が利用可能となっているからである。R では自分自身が書いた R のコードをパッケージと呼ばれる拡張ソフトウェアとして提供することで、世界中の R ユーザーが容易に利用できるようになっており、2014 年 4 月 14 日現在で 5,000 を越えるパッケージが利用可能となっている。しかしながら、R がデフォルトで用いる計算用のスレッド数は 1 であり、マルチコア CPU の残りのコアや、汎用計算に利用可能な GPU は利用しない。また R のパッケージとして、GPU を利用するものもいくつか存在するが、実装されている関数は多くなく、また Windows<sup>©</sup> の R から容易に利用可能なものは存在しない。そこで本研究では GPU を計算に用いる Windows 向け R パッケージの開発を試みた。

## 2. 本パッケージの特徴

現状で本パッケージは開発中であるが、以下の特徴を備えるものが完成する見込みである。

### 2.1 Windows 版 R から利用可能

Windows は多くのパーソナルコンピュータ上で動作する OS であり、その上で容易に利用可能となることにより、

<sup>1</sup> 独立行政法人放射線医学総合研究所  
千葉県千葉市稲毛区穴川 4-9-1

<sup>a)</sup> kztkdi@gmail.com

多くのユーザーが利用可能となる。

## 2.2 データ転送の最小化

GPGPUにて計算をする際、メインメモリにあるデータをGPUから扱えるビデオメモリに転送が必要であるが、この転送のための所要時間の占める割合が比較的大きいことが知られている。そこで、計算のために一度ビデオメモリに転送されたデータは、ビデオメモリ上に留め、次に利用される際には転送することなくそのまま計算に利用可能とした。そのために、Rのオブジェクトの外部ポインタという仕組みを利用した。

## 2.3 既存のRコードの最小限の変更

Rには演算子のオーバーライドという機能があり、既存の関数を上書きすることが可能である。そこで、Rの既存の関数をGPUを用いるコードで上書きすることにより、既存のRのコードを変更箇所を最小限に留めて、高速化することが可能である。

## 2.4 ビデオメモリの有効活用

ビデオメモリ上にあるデータは転送のコストなく利用可能である一方で、その容量には限りがある。そこでビデオメモリの容量が不足した際や、不足が予想されそうな際、最も長い間参照されなかったデータをメインメモリに戻す機能を実装した。

## 2.5 マルチGPUへの対応

しばしば複数のGPUを搭載可能なコンピュータも存在する。そのようなコンピュータで、GPUの搭載個数に応じてパフォーマンスが向上すれば理想的である。しかしながら、他のGPUに搭載されたビデオメモリへのアクセスには通常データ転送が必要となる。従ってデータを分割して転送し、別のGPUで計算を行った後、その結果を再度転送したのでは、大きなパフォーマンスの向上は望めない。そこで本パッケージではデータを分割しての複数GPUへの対応はせず、一連の計算処理の中でメモリ転送が比較的少なくなるレベルでの分割を検討している。

## 3. おわりに

現状では、このパッケージは開発中である。研究報告会では、今後の方向性などについて意見を頂ければ幸いである。