

データストリーム処理におけるGPUタスク並列を用いた スケーラブルな異常検知

Scalable Anomaly Detection on Data Stream Processing with GPU Task Parallelism

上野 晃司¹⁾, 鈴木 豊太郎²⁾

Koji Ueno and Toyotaro Suzumura

1) 東京工業大学 大学院情報理工学研究科 (〒152-8550 東京都目黒区大岡山2-12-1)

2) IBM東京基礎研究所 (〒242-8502 神奈川県大和市下鶴間1623-14)

Key Words : Stream Computing, Task Parallelism, SVD, Anomaly Detection

1. 背景

データストリーム処理[1]とは、継続して入力されるデータを、蓄積せずにリアルタイム処理するという、データ処理方式の1つである。異常検知とは、センサーなどから取得したデータから、異常を発見することであり、異常をなるべく早く検知したいという要求から、リアルタイム処理を行うデータストリーム処理と親和性が高い。異常検知を行う方法として、様々なアルゴリズムが考えられているが、本研究では、変換点検知アルゴリズムSST[2](Singular Spectrum Transformation)とその近似アルゴリズムIKA-SST[2](Implicit Krylov Approximation based SST)を取り上げる。SSTは、特異値分解(SVD, Singular Value Decomposition)を利用したアルゴリズムであり、IKA-SSTはSSTの近似を高速に求めるアルゴリズムである。本研究の目的は、より多くのセンサーデータを処理するため、GPUを使って変化点検知を高速化することである。

SSTの計算量の大部分を占めるSVDを、GPUで計算する場合、CULA[3]の利用が考えられるが、CULAは、行列サイズが1000以下の小さい行列に対しては、効率よく計算することができない。小さい行列では、並列可能なデータが足りないため、GPUの性能を引き出すことができないからである。SSTで必要になるSVDの行列サイズは、500以下で十分な場合が多いので、小さい行列に対して高速化できないことが問題となる。

2. GPUタスク並列

小さい行列の計算で、GPUの性能を引き出すには、複数の行列の計算を並列化することが必要である。CUBLASやCULAなどは、1つの行列内の並列性のみを使って、GPUに計算させているので、これをシングルタスクと呼ぶことにする。これに対し我々は、複数の行列の計算を並列化することである、GPUタスク並列を提案する。我々の提案する手法は、CUDAブロックの独立性を利

用したものである。また、発表ではFermiコアで利用可能なconcurrent kernelを使った方法との比較についても言及する。

3. 変換点検知に適用

SSTとIKA-SSTをGPUタスク並列で実装し、性能評価を行った。SSTへの適用では、複数の行列のSVDをGPUタスク並列で計算するプログラムを実装した。図1は、CPU(AMD Phenom X4 9850) 1コアに対するGPUの高速化である。

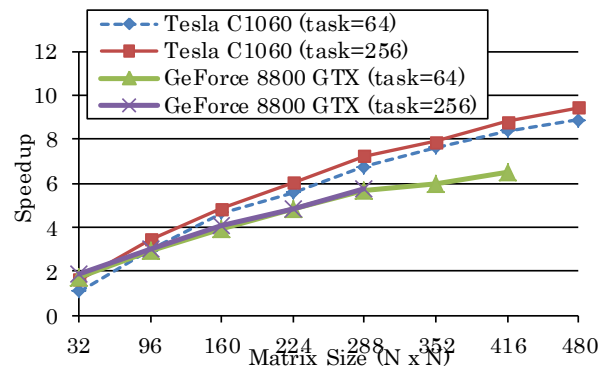


図1 SVDの高速化 ("task"は並列に計算する行列の数)

IKA-SSTへの適用では、複数のセンサーデータの変化度スコアをGPUタスク並列で計算するプログラムを実装した。これにより、CPU 1コアに対して、10~20倍程度の高速化が得られた。

参考文献

- [1] Bugra Gedik, etc. SPADE: The System S Declarative Stream Processing Engine SIGMOD 2008.
- [2] Tsuyoshi Ide, Koji Tsuda. Change-point detection using Krylov subspace learning. Proceedings of 2007 SIAM International Conference on Data Mining (SDM2007), pp.515-520, Minneapolis, Minnesota, USA, April, 2007.
- [3] CULA. CULA Programmers Guild R10-1 <http://www.culatools.com/>.