









- Step0. Flag を 0 とする 要素数 N とする
  - Step0-2. データを分割する
- Step1. 以下の操作をデバイスで行う
  - Step1-1. 奇数と奇数+1 について比較をする
  - Step1-2. 比較の結果が  $data[i] < data[i+1]$  のとき交換する
  - Step1-3. 交換が起こった場合のみ flag を 1 にする
- Step2. 以下の操作をデバイスで行う
  - Step2-1. 偶数と偶数+1 について比較をする
  - Step2-2. 比較の結果が  $data[i] < data[i+1]$  のとき交換する
  - Step2-3. 交換が起こった場合のみ flag を 1 にする
- Step3. Step1 と Step2 を交換が行われなくなるまで繰り返し, 交換が行われなくなればデバイスの処理を終了する
- Step4. 分割されたデータ列についてホストでマージソートを行い結合する
- Step5. 終了する

図 4.1.1 ハイブリッドアルゴリズム

#### 4. 実験

##### (1) 実験環境

実験環境は, Windows7 (32b)で CPU は Intel Core i5-760(2.8GHz), GPGPU として GeForce GT 220 を用いて, プログラム環境は CUDA3.2 による Visual studio8.0 の VisualC++を用いた.

スレッド数は 50, ブロック数は「データ数/スレッド数/2」である. 実験データは, レコード数  $N=1,000$  から 1,000,000 の降順のデータを用意した.

表 4.2.1 並列化した奇偶転置ソートの動作例

グラフィックカード	GeForce GT 220
CPU	Intel Core i5-760 2.8GHz
メインメモリ(GB)	4
CUDA	Version 3.2
CUDA コア数	48
グラフィッククロック(MHz)	625
メモリクロック(MHz)	1360

##### (2) 実験結果と考察

大規模データサイズ対応のためホストとデバイスと連携化(マージソート)で 1,000,000 レコードに対応した.

ホスト単体での処理に比べて 1,000,000 レコードの場合には 61 倍高速化が確認できた. レコード数 1,000 から 40,000 までのレコードではホスト単体とハイブリッド化したものを比較するとホスト単体で行ったほうが早いという結果が得られた. これはハイブリッド化したものは処理を行うために準備する時間がかかるためだと考えられる. 図 4.3 を用いてホスト単体の処理とハイブリッド化した処理の O を説明する.

ホスト単体のグラフの関数は  $y = 6E-11x^2 + 7E-08x - 0.0119$  を示している. グラフの関数の信頼性は  $R^2=1$  より信頼できる. グラフの関数よりホスト単体は  $O(N^2)$ であることが確認できる.

ハイブリッド化したソート結果のグラフの関数は  $y=8E-07x+0.0609$  を示している. このグラフの関数の信頼性は  $R^2=0.9996$  より先ほどのホスト単体のグラフの関数と同様に信頼できる結果である. グラフの関数よりハイブリッド化を行った場合の処理は  $O(N)$ で最大レコード数の 1,000,000 まで処理を行うことができているという結果が得られた.

表 4.2.2 ホスト単体とホストとデバイスを用いたソートの実行時間比較

	HOST	Hybrid	HOST/Hybris
1000	0	0.0597	0.00
2000	0	0.0634	0.00
5000	0.0029	0.0638	0.05
10000	0.0077	0.0686	0.11
20000	0.0243	0.0804	0.30
40000	0.0808	0.0954	0.85
80000	0.296	0.1282	2.31
100000	0.4735	0.1445	3.28
200000	2.2569	0.2336	9.66
500000	13.8782	0.4731	29.33
800000	35.3168	0.7502	47.08
1000000	55.3022	0.8988	61.53

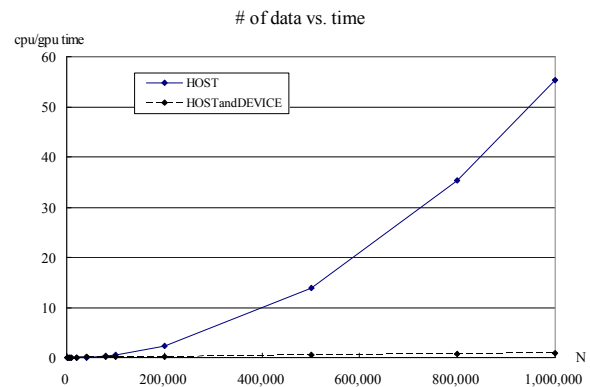


図 4.2.2 ホスト単体とホストとデバイスを用いたソートの実行時間比較

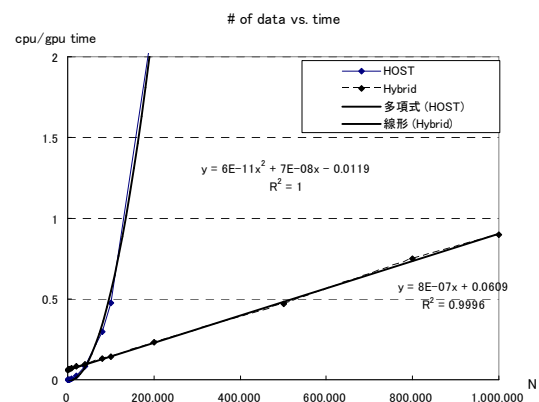


図 4.3 ホスト単体とホストとデバイスを用いたソートの実行時間比較

## 5. まとめ

奇偶ソートでGPGPUによる並列ソートの実装した。扱えるデータは 20,000 と限られてしまう。単一スレッドに比べて多数のスレッドを用いて並列化をした場合にはレコード数 5,000 で最大 13 倍の高速化が確認された。

レコード数 5,000 まではおよそ理想としていた  $O(N)$  になっていることが確認できる。しかし、レコード数が増えるにしたがって、 $O(N)$  から外れるという結果になった。

その際、規模が大きなレコードの高速化が課題となる。また扱えるレコード数は 20,000 と限られているので大規模データへの拡張が課題となる。

そこで提案したのが、並列化した奇偶ソートを高速化するために Global メモリよりも高速な Shared メモリを利用する方法である。その際 Shared メモリでは同一ブロック内のスレッドからしかアクセスできないことが問題になってくる。そのため分割することによって Shared メモリを効率よく利用できるように考慮した。ホストで分割を行い、デバイスで分割されたデータをそれぞれ別のブロックで並列に奇偶転置ソートを行う方法を提案した。デバイスで分割したデータを処理した後、ホストでマージソートを行い、データを目的とした順番に並べた。

大規模データサイズ対応のため、ホストとデバイスと連携化(マージソート)で 1,000,000 レコードに対応した。ホスト単体での処理に比べて 1,000,000 レコードの場合には 61 倍高速化が確認できた。ハイブリッド化を行った場合の処理は  $O(N)$  で最大レコード数の 1,000,000 まで処理を行うことができているという結果が得られた。

## 謝辞

本研究を科学研究費補助金(課題番号 22500049)として助成いただいた独立行政法人日本学術振興会に感謝いたします。

## 参考文献

- [B1]Odd-even Transposition Sort  
<http://ww2.cs.mu.oz.au/498/notes/node34.html>
- [B2]Nvidia CUDA3.2 マニュアル
- [B3]全世界のデータ量 | 最新の関心ニュース  
[http://gigazine.net/news/20110213\\_295exabytes/](http://gigazine.net/news/20110213_295exabytes/)
- [B4]大菊祥子, 中井駿介, 豊永昌彦, “GP-GPU による並列ソートアルゴリズムの評価,” 1-27, 平成 23 年度電気関係学会四国支部連合大会講演論文集 (2011 年 9 月 23 日阿南高専).
- [B5]日本経済新聞記事「クラウドデータセンタ, 磁気テープ記録容量を 44 倍に」, 2010 年 1 月 22 日.