

*GPGPU*による 魔方陣の解法プログラムの実装

電気通信大学 情報理工学部

情報・通信工学科 佐藤研究室

齊藤 友輝

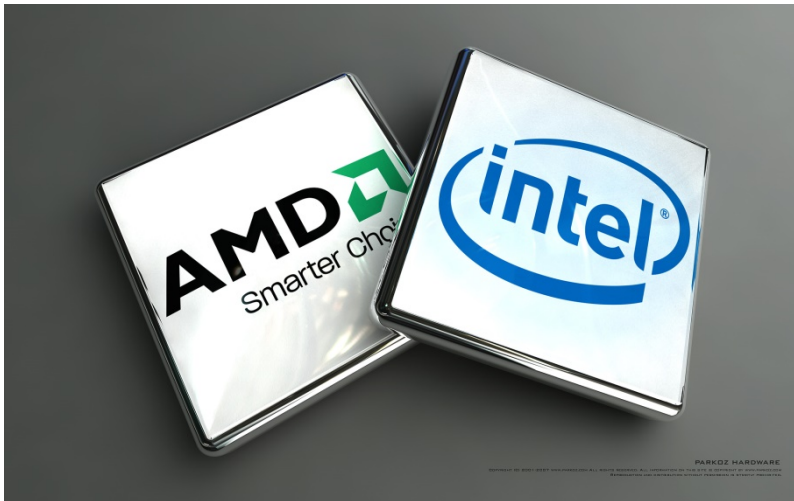
目次

➤ GPGPU

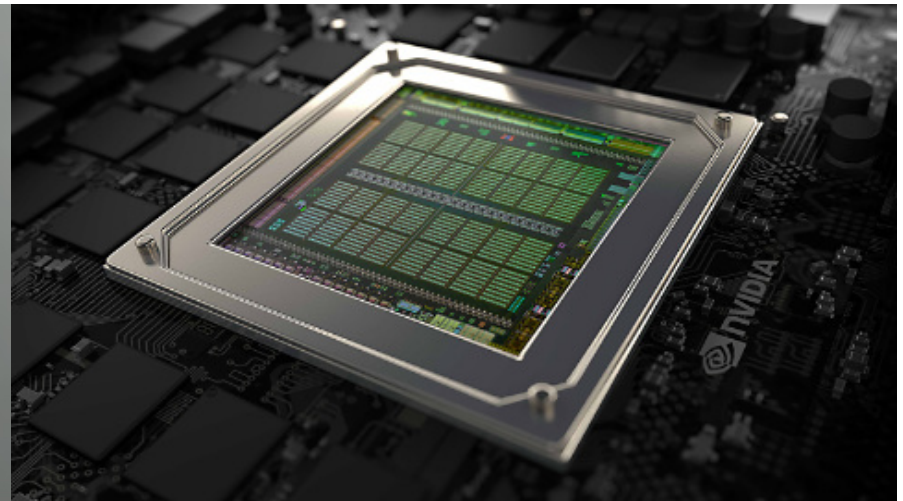
- 魔方陣
- 4次魔方陣の実装方法
- 実装結果
- 5次魔方陣の実装の検討

GPGPUとは

- 汎用プロセッサ：様々な処理に対応できるように設計されたプロセッサ
 - GPU：並列計算に特化して設計されたプロセッサ
- GPGPU (General-purpose computing on graphics processing units)
GPUを画像処理以外の目的に応用する技術



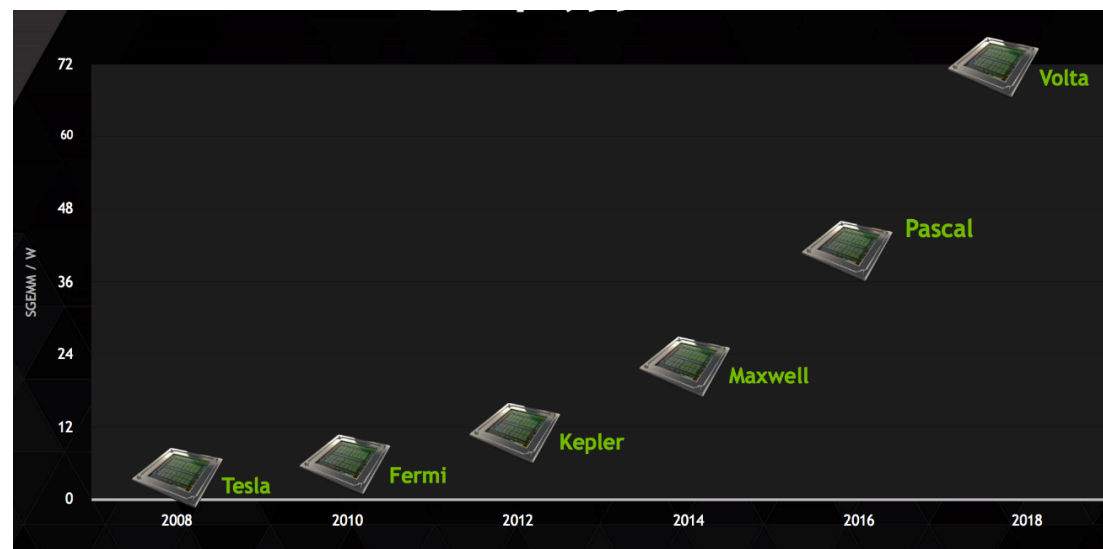
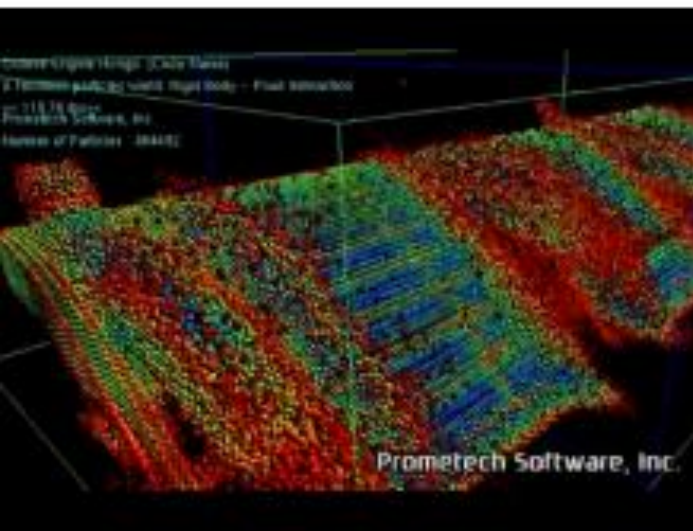
CPU



GPU

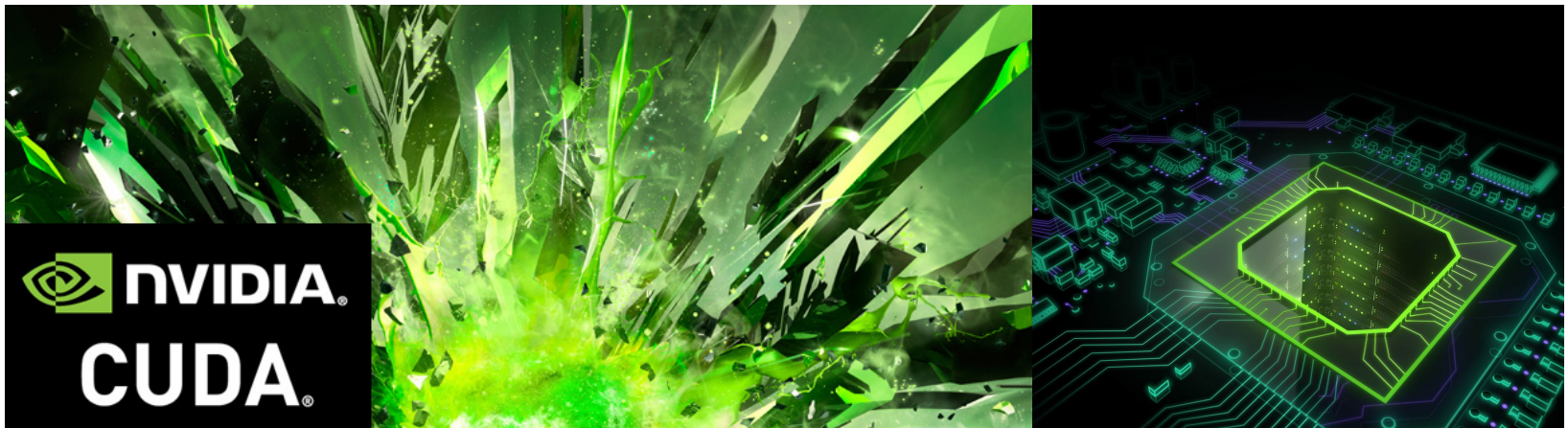
注目されるGPGPU

- 様々な分野へGPGPUの応用が進んでいる
 - シミュレーション（物理・流体・天体）
 - 最適化問題
 - 暗号計算
- アーキテクチャの刷新も頻繁に行われている



CUDA

- CUDA (Compute Unified Device Architecture)
: NVIDIA社が提供するGPU向け統合開発環境
 - NVIDIA社はCUDAを無償で提供
 - NVIDIA社製のGPUを購入するとGPGPUの開発環境が整う



目次

- GPGPU

- 魔方陣

- 4次魔方陣の実装方法

- 実装結果

- 5次魔方陣の実装の検討

魔方陣

- 正方形の方陣（マス目）に、縦・横・斜めの和が同じになるよう数字を置いたもの
- 今回は1からマス目の総数までの数字を全て使うものを扱う

2	9	4
7	5	3
6	1	8

解は
1つ

9	6	3	16
4	15	10	5
14	1	8	11
7	12	13	2

解は
880通り

1	7	13	19	25
18	24	5	6	12
10	11	17	23	4
22	3	9	15	16
14	20	21	2	8

解は
2億7530万5224通り

目次

- GPGPU
- 魔方陣
- 4次魔方陣の実装方法
- 実装結果
- 5次魔方陣の実装の検討

4次魔方陣の解法アルゴリズム①

- 酒井正彦教授の用いた制約条件を4次魔方陣に適用
 - 回転・対称による解の重複防止
 - 行・列の入れ替えによる解パターンの生成
- 枝刈り法による解空間の限定
 - 魔方陣の性質を利用して、
埋めるべき値を一意に決定してゆく
 - 探索空間が総当りの 10^{-5} 以下に

9	6	3	16
4	15	10	5
14	1	8	11
7	12	13	2

4次魔方陣の解法アルゴリズム②

- 限定した解空間をバックトラッキングで探索
 - 魔方陣の条件や, 解の等価構造を制約条件にする
- 解の候補パターンはモジュロ演算で生成
 - GPGPUで生成するプロセスごとに固有の値を算出
 - 固有の値からモジュロ演算でいくつかの整数を生成

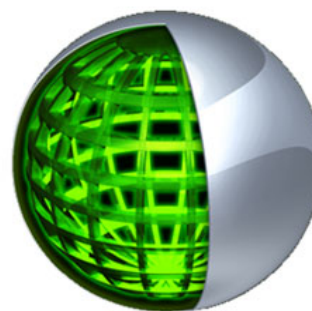
実験・開発環境



Windows 8.1



統合



NVIDIA®
Nsight™

使用したハードウェア

GeForce GTX 980Ti



計算コア数 2816
メモリ量 6GB

GeForce GTX 750Ti



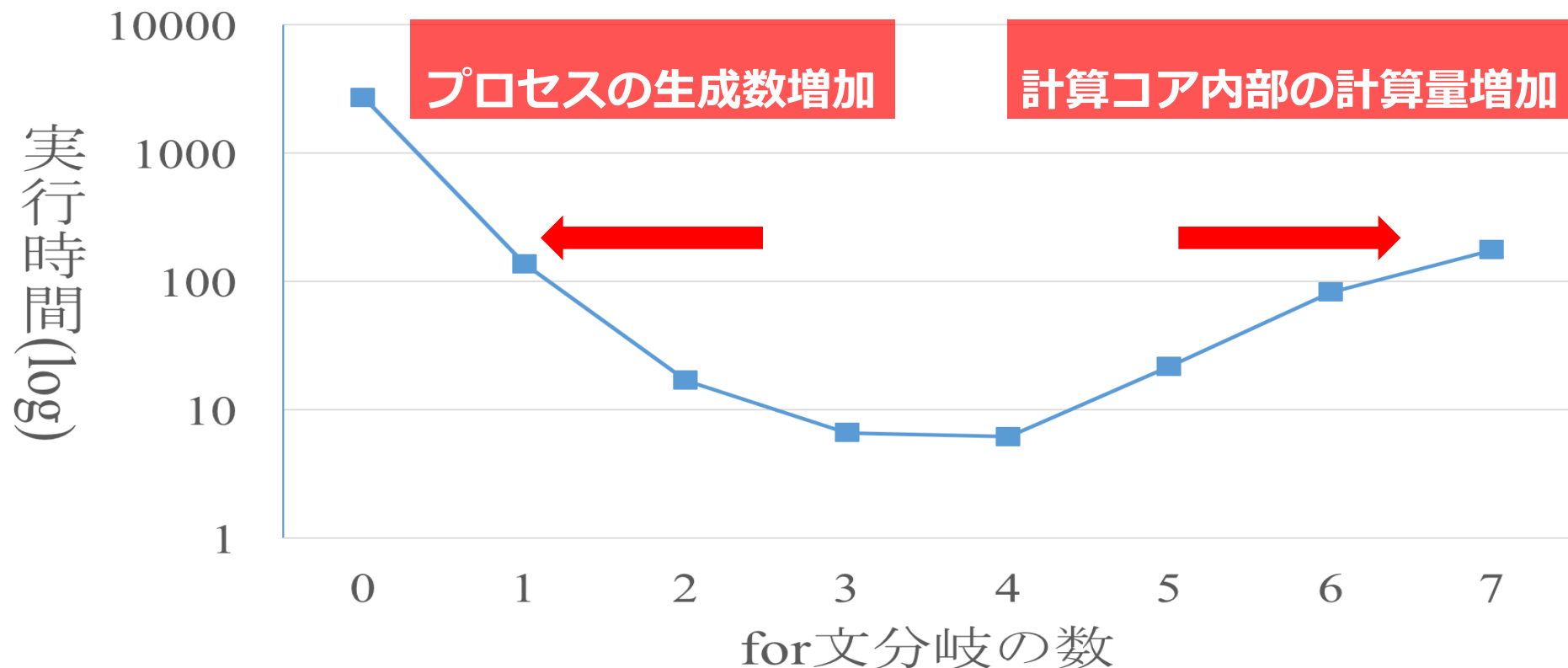
計算コア数 640
メモリ量 2GB

目次

- GPGPU
- 魔方陣
- 4次魔方陣の実装方法
- 実装結果
- 5次魔方陣の実装の検討

最適な条件を探す

- 実装の方法を変えて計算の粒度の調整を行う
 - プロセスの生成数を増やす方向（並列度を上げる）
 - 計算コア内部での計算量を増やす方向（並列度を落とす）

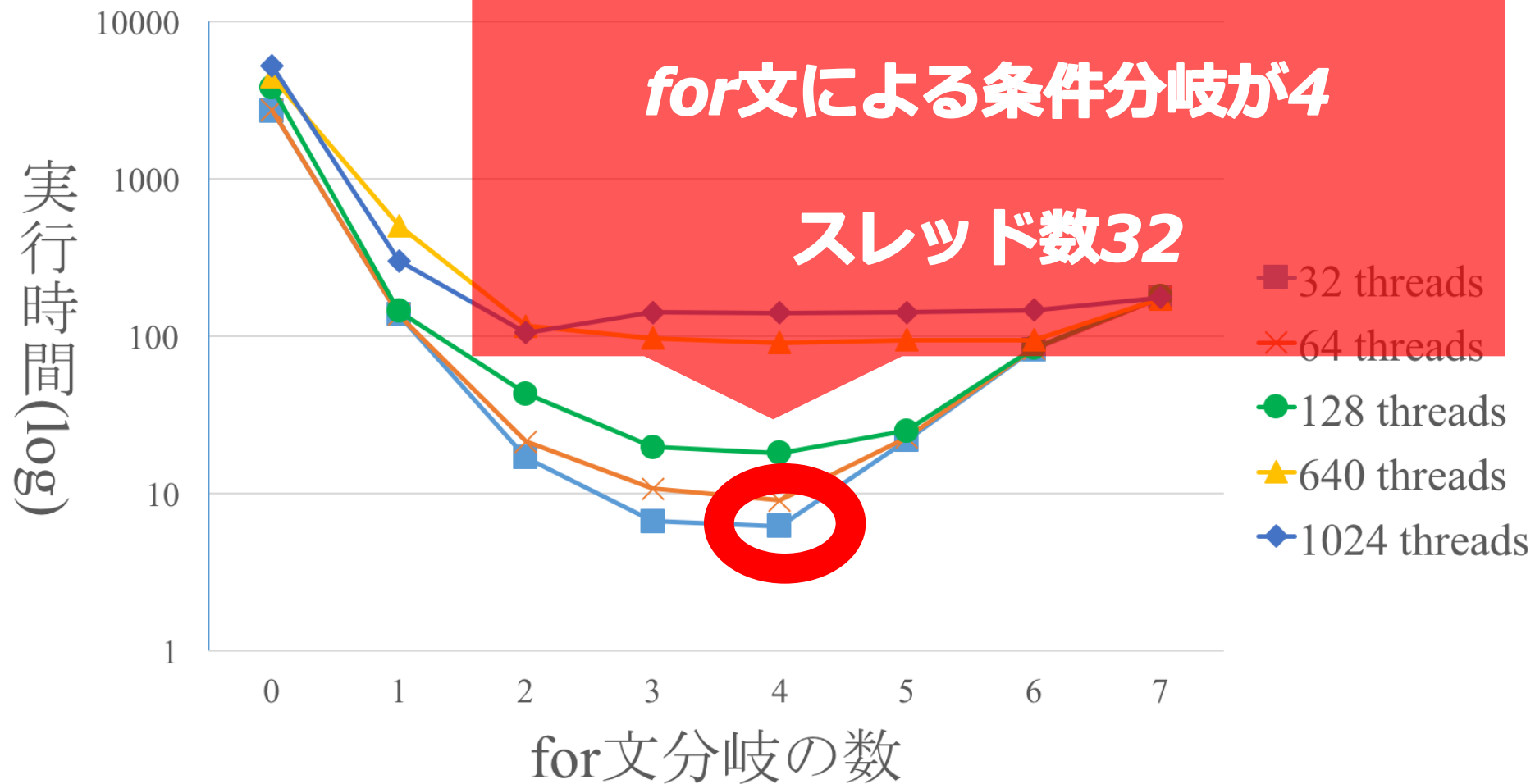


GeForce GTX 980Tiへの実装

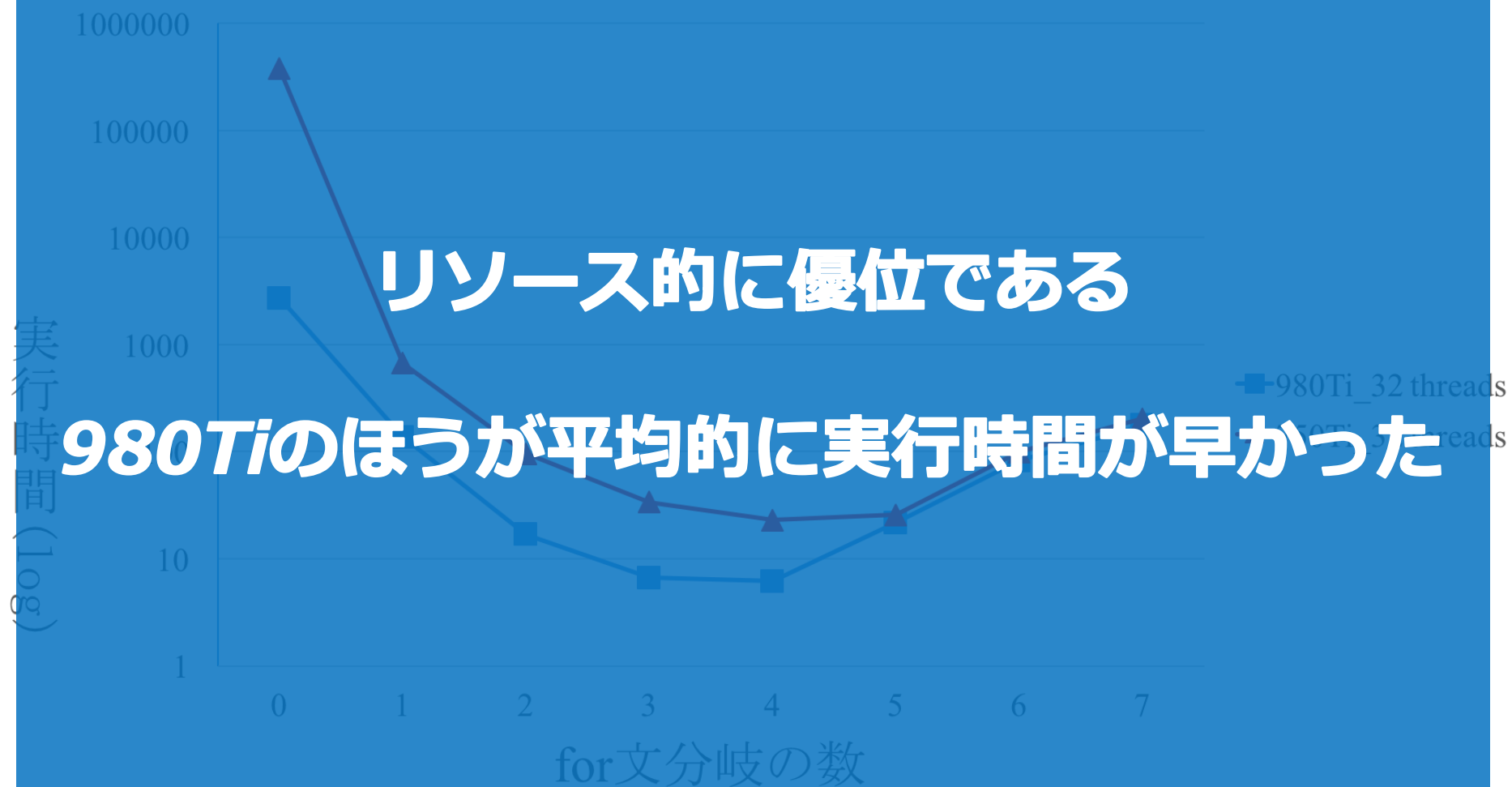
最適条件

for文による条件分岐が4

スレッド数32



GeForce GTX 980Ti GTX 750Tiへの実装



汎用プロセッサでの実装と比較

- C++で4次魔方陣の解法プログラムを実装
 - CUDAと同じ制約条件を用いて実装
 - モジユロ演算による解パターンの生成
- 要件
 - Windows 8.1 64bit
 - Intel core i5 4590
 - Memory DDR3 16GB

実行したプログラム	実行時間 <i>Elapsed Time</i> (s)
4次魔方陣C++	0.011
4次魔方陣CUDA	0.006

目次

- GPGPU
- 魔方陣
- 4次魔方陣の実装方法
- 実装結果
- 5次魔方陣の実装の検討

5次魔方陣の実装

- 酒井正彦教授の用いた制約条件を利用
 - 回転・対称による解の重複防止
 - 行・列の入れ替えによる解パターンの生成
- 枝刈り法による解空間の限定
 - 魔方陣の性質を利用して、埋めるべき値を一意に決定

1	7	13	19	25
18	24	5	6	12
10	11	17	23	4
22	3	9	15	16
14	20	21	2	8

実装結果

- 5次魔方陣の解法プログラムの実装は成功していない
 - GPU側に送る命令が重すぎるのか、GPUのドライバを巻き込んでプログラムが異常終了してしまう
 - プロファイリングソフトで検証しようとする、リソース不足でプロファイリングソフトが停止してしまう

まとめ

- 4次魔方陣のGPGPUによる実装が出来た
- C++による汎用プロセッサ上での実装よりも速い処理速度を観測できた
- 5次魔方陣など、大きな値や大量なメモリを扱う問題への対策
- オーバーフローしてしまわないようなデータの流し方, 計算の回し方の検討

- 以上