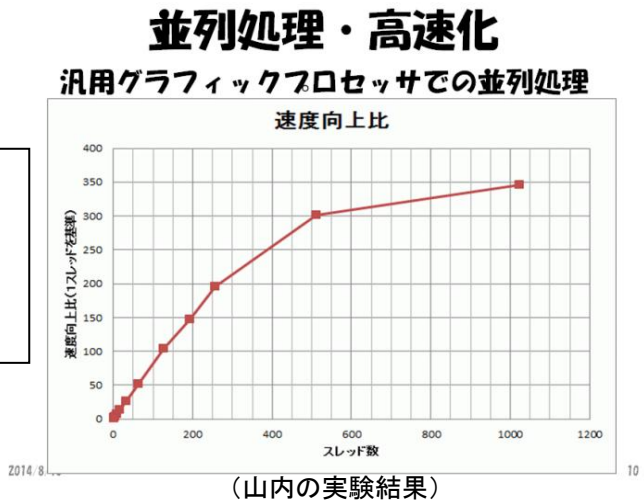
第７回　コンピュータ科学基礎論　資料　　2017-11-17 山内

今回は、汎用並列ハードウェアの限界について、いくつかの問題を考えてみたい。

(1) GPGPU のコアをどうやったら使いきれるか、考えてみて欲しい。 基本的にコア数程度の並列性が出せるかどうか、という点が、疑問である。

Device 0: "GeForce GTX TITAN Z"	
CUDA Driver Version / Runtime Version	6.5 / 6.5
CUDA Capability Major/Minor version number:	3.5
Total amount of global memory:	6143 MBytes
(15) Multiprocessors, (192) CUDA Cores/MP:	2880 CUDA Cores
GPU Clock rate:	876 MHz (0.88 GHz)



（山内の実験結果）

(2) GPGPU では、倍精度浮動小数計算資源が少ない。これを回避する方法を考えてみてほしい。
一般に、数値計算プログラムはほぼデフォルトで、倍精度浮動小数を使うだろう。 単精度では有効桁数が不足することが多い(7 桁？)し、いちいち考えるのが面倒ということもある。 今通常はすべて倍精度で計算している演算を、単精度で済むものをきちんと選んで単精度で計算するとしたら、どういうやり方が考えられるか？
（注： たとえばグラフィックスの処理では、一部を全面的に単精度で済むと考えられ、だからこそ GPU が単精度浮動小数でしか計算しないようになっている。）
更には、現在は倍精度でなければ結果の精度の保証ができない計算を、何らかの工夫で単精度で済ませる手立てはあるだろうか？

(3) NVIDIA のアーキテクチャでは、GPU 用のメモリと「ホスト」(計算機本体)のメインメモリは別々に持っており、その GPU 側で計算するためにはデータを転送しなければならない。 これに対してどういう対応が考えられるだろうか？
現在は、基本的(原始的)なプログラミング環境(CUDA)ではプログラマが陽に指示して転送することになっており、プログラミング上の手間は面倒であるが、プログラマが性能上のチューニングをすることが可能である。 他方、一部のプログラミング言語・環境では転送を自動化している。 どう考えたらよいだろうか？

＜最終レポート＞

この授業の評価の対象として、1つレポートを書いてください。分量は問いませんが、A4で7～10枚程度を期待しています。授業の内容を参考にして、自分でいろいろと考えたり調べたりしてみて下さい。 テーマは次のものとします。

「1つのアルゴリズムを取り上げて、それをどのようにして並列化するのがよいか、考えてみよ」

取り上げるアルゴリズムは、自分の研究の中で使われるものでもよいですし、他のアルゴリズムを考えてもよいでしょう。なるべく、身近なもの、自分が使ったことがあるものについて、考えてみてください。 見当たらない場合は、「遺伝子のアラインメントのアルゴリズムの並列化」「動的プログラミング Dynamic Programming の並列化」（同じものですが） を考えてみてください。データ依存性があるので並列化が困難な例ですが、足し算の例と同様に工夫する価値があります。

考えてほしいポイントは、最終的には「並列化によって十分な時間短縮・性能向上が達成できるか」ですし、「それを阻むものは何か」でしょう。 並列度が得られないこともあるでしょうし、直列部分が多いこともあるでしょう。また、GPGPU のような NUMA の環境ではデータ転送が大きくなるために時間短縮できないということもあるでしょう。 様々な側面を考えてみてください。（締切り： 12 月 01 日(金)午後 3 時、 提出先： 山内の部屋 4541 のポストへ）