e-Science Journal
東京工業大学 学術国際情報センター
Vol. 3

TSUBAME ESJ.

TSUBAME 2.0始まる
TSUBAME 1.0から2.0への長い道のり（後編）

星間水素原子ガス乱流のGPU計算
― TSUBAME1.2の120 GPUで11.5 TFlops ―

高速フーリエ変換とGPU

TSUBAME 2.0始まる

TSUBAME 1.0から2.0への長い道のり(後編)

松岡 聡*

*東京工業大学　学術国際情報センター

TSUBAME2.0は我が国初のマルチペタフロップスのスーパコンピュータであり、GPUの積極的な活用、高度にスケーラブルで高バンド幅なノードやネットワークのデザイン、さらにSSD等の新世代のI/Oデバイスの活用など、種々の新ハードウェアやソフトウェアの技術を採用している。TSUBAME2.0は実運用を11月初旬に開始した;本稿後篇では、その直前に行われた種々のベンチマークを取り上げる。特に、Linpackに代表される密行列系の計算主体のベンチマークと、ASUCA気象計算(流体)に代表される疎行列系のメモリ・ネットワークバンド幅が主体のベンチマークはマシンの全く異なる側面の限界を探る。
幸いにもTSUBAMEは両方で記録的な性能を達成した:2010年11月のTop500では世界4位にランクされ、またGreen500においては「世界一電力効率が良い運用スパコン」であることが認定された。
また、ASUCAも気象計算としては世界記録を達成した。これらはTSUBAME2.0のスパコンとしての単なる性能だけではなく技術的先進性をも示すものである。

TSUBAME2.0の産声 1

TSUBAME2.0にとっての最初の試練はLinpackやその他のマシン全体を使用するベンチマークであった。実はスーパーコンピュータにとって稼働初期の段階で大規模ベンチマークを数々行うことは、以下の理由によって大変重要な事なのである。

(1)TSUBAME2.0などの大規模スパコンを構成するコンポーネンツ数は通常のPCと比較すると数千倍にものぼる。単純に計算を司るCPU/GPUのソケット数でも、パソコンは1-2個であるのに対し、TSUBAME2.0は7000以上である。メモリもパソコンの数ギガバイトに対し、TSUBAME2.0のCPUとGPUの全てのメモリを合算すると約100テラバイトと、数万倍である。これらが高負荷時にきちんと長時間同時に動作することを確認するのは、安定運用には大変重要である。マシンの故障率はコンポーネンツ数にほぼ比例するので、例えば一万倍のコンポーネンツ数ならば、約10年に一度しか故障しないパソコンと同じ信頼性ならば、一日に約三回は障害が発生することになる。

(2)同様に、果たして種々の設計性能値がきちんと満たされているか、というのは単純な稼働性にも勝り重要である。複雑なスパコンの場合、特定の条件であるコンポーネンツが動作はするが、性能値を満たしていない場合、そこが速度的律速となって大規模並列アプリケーション全体の性能を大幅に引き下げる可能性がある。例えばTSUBAMEのネットワークを構成するInfinibandは幾つかの速度規格があり、下位互換性を保証する為に自動的に相互のネットワークの端点の速度に合わせるようになっている。しかしながら、何かの異常事態が一瞬生じ、両方の端点が低速の規格で合意してしまうと、通信は問題なく起こるが、速度が数分の一になってしまう。このような事態が起こらない・あるいは起きても検出されてすぐに本来の速度に修復する、ことを確認する必要がある。

(3)また、大規模なスパコンの運用には多数のセンサーによる監視や、ソフトウェアによる資源配分などが必要になるが、多くの場合アルゴリズムの複雑性により、小規模マシンにおいては起こらない現象が起こる。例えば、ジョブのスケジューリングにおいて、そのアルゴリズムの一部にジョブ数nに対しO(n^2)の複雑性がたまたまあった場合、ジョブ数が100倍になれば一万倍のオーバーヘッドが顕在化する。

無論計算機科学的にも設計したスパコンの性能がどこまで出るのか、逆に何がオーバーヘッドになっているのか、を探るのは学術的に重要である。より重要なのは、大規模なアプリケーションを通常運用では困難なマシン全体を使う環境で動作させ、科学的な成果を得ることであるのは言うまでもない。

そのような様々な目的をもって、TSUBAME2.0の運用開始の11月1日直前の10月後半、マシンがその産声を上げた直後に幾つかの大規模ベンチマークが行われた。

(a)Linpack---有名なスパコンの性能リストであるTop500[1]において用いられるベンチマークである。基本的には大規模な密行列のLU分解を行う。計算の複雑性はn×nの行列に対し、計算の複雑さは$2/3\ n^3 + O(n^2)$であるので、マシンのメモリに入り切る、なるべく大きな行列にする方が通信オーバーヘッド等が隠れて効率が良くなる。結果としてトップレベルのスパコンではn=数百万となり、結果的に演算用のCPU/GPUを通常のアプリケーションでは生じない超高負荷をかけて長時間計算を行うこととなる。密行列系における直接法による演算であるので、反復計算と異なりO($10^{19\sim20}$)回の総演算数のうち一つでもエラーがあると最後の残差による検算において全体がエラーとなり、結果として無効となる。逆にネットワークに対する負荷はO(n^2)なので、計算負荷と比較して相対的には低く、スパコンに「見合った」高速ネットワークならばそのオーバーヘッドは10%以下程度である。メモリバンド幅も比較的要求されない。一方Gigabit Ethernetなどの「遅い」ネットワークの場合はネットワークのオーバーヘッドは計算を圧倒する。

LU分解を行う実際のソフトウェアは規定されていないものの、多くのスパコンでは大規模並列Linpackを行うHPL(High Performance Linpack) [2]を用いる。しかしながらTSUBAME2.0では、ヘテロジニアス(性質や性能の違うCPUやGPUなどを用いる)版のLinpackアルゴリズムをHPLに適用し改造した二つの特別バージョンを用いた。一つはLinux OS上で松岡研究室がTSUBAME1以前から

研究を続けきたHeterogeneous HPL[3]であり、もう一つは米マイクロソフト社との共同研究で開発されたWindows HPC用のLinpackである。二つはヘテロジニアス性に対応する為には異なるアプローチをとっており、どちらがより優位になるかがGPUコンピューティングの進化にとって意味のある結果となることが期待された。

また、スパコンの電力性能をランキングするGreen500[4]の為に、正確な電力計測も同時に行い、グリーンスパコンとしての研究成果の実証を目指した。

(b) GPU版ASUCA---ASUCAは気象庁が次世代の予報を実現するために開発を続けている超並列計算機用の気象コードである。東工大GSICの青木研究室は、ASUCAの高性能・マルチGPU版を開発することに成功した[5][6]。主な計算カーネルは差分法による移流計算であり、Linpackと異なり非常に高いメモリバンド幅やネットワークバンド幅を要求する。この種類のアプリケーションにおいて高性能を得るのは、従来は地球シミュレータ等のスパコン専用アーキテクチャのベクトル計算機の独壇場であったが、前編で記したとおりGPUは汎用性アーキテクチャなるも通常のCPUと異なり高いメモリバンド幅とベクトル処理機能を備えており、TSUBAME2.0に移植されれば世界トップレベルの気象コードの実行性能が期待された。実際、TSUBAME2.0は地球シミュレータの約6倍のメモリバンド幅を持ち、ASUCAの新世代の解法と共に記録的な解像度とサイズの気象シミュレーションがリアルタイムに可能となることがTSUBAME1.2における結果と性能モデリングから期待された。

(c) その他、調達の性能要件を満たすべく種々のベンチマークや、種々の基本的な性能確認が行われた。

TSUBAME2.0 の Linpack --- 世界トップのグリーンスパコンへ 2

TSUBAME2.0の初期配備および初期テストを完了した10月中旬からベンチマークが開始された。まずは (a) の二つのLinpackが開始された。初期テストでは生じない高負荷のLinpackテストにより、種々の(予測された)細かい障害が明らかになり、それらを逐一解消していった。先に述べたように、このような高負荷稼働の安定化は実運用のために非常に重要である。その後、交代でLinuxおよびWindowsHPCにてのベンチマークが行われた。両者とも非常に性能が接近していたが、最終的にはLinux上の松岡研究室のHeterogeneous Linpackが僅かに上回った(図1)。なお、異なる条件下ではマイクロソフトのアルゴリズムが上回る可能性は十分あったことは付け加えておく。

結果として、Top500用のLinpack計測では1.192ペタフロップスの性能を得た。これは理論ピーク性能の約52%であり、通常のCPUにおける70-90%と比較すると低い。しかしながら、これによって「GPUの実行性能はCPUマシンよい低い」と結論づけるのは大きな誤りであり、この性能差は実は別な理由による:

1. まず、現状のNVIDIA Fermi GPUは幾つかの人工的な性能的なボトルネックがあり、Linpackの実行時に主要な計算カーネルとなる密行列の積(所謂Level 3 BLAS) の実行性能が理論値と比較して70-75%程度となる。このボトルネックはグラフィックスの実行時や、あるい

```
--------------------------------------------------------------------------------
- The matrix A is randomly generated for each test.
- The following scaled residual check will be computed:
      ||Ax-b||_oo / ( eps * ( || x ||_oo * || A ||_oo + || b ||_oo ) * N )
- The relative machine precision (eps) is taken to be               1.110223e-16
- Computational tests pass if scaled residuals are less than                16.0
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR15R2R16     2490368  1024    59    69            8639.84              1.192e+06
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0008911 ..... PASSED
================================================================================
Finished      1 tests with the following results:
              1 tests completed and passed residual checks,
              0 tests completed and failed residual checks,
              0 tests skipped because of illegal input values.
--------------------------------------------------------------------------------
End of Tests.
================================================================================
```

図1 TSUBAME2.0のLinpack実行の様子
約250万2の行列のLU分解を行い、ほぼ2.4時間で計算を終え、
2010年11月のTop500において世界4位となる平均1.192ペタフロップスの性能を得ている。
残差計算で検算を行い、結果が規定誤差以下になっていることに注意されたい。

図2 2010年11月Green500における
「世界一電力効率が良い運用スパコン」特別賞の証書

TSUBAME2.0 世界ランキング スパコン2大リスト（2010年11月発表時）

The Top 500
ベンチマーク絶対性能、ペタフロップス

1位 ： 2.566 ： 中国防衛大 Tianhe 1-A（11）
2位 ： 1.758 ： 米国オークリッジ国立研究所
Cray Jaguar（81）
3位 ： 1.271 ： 中国深圳国立スパコンセンター
Dawning Nebulae（13）
4位 ： 1.192 ： 日本 東工大 /HP/NEC TSUBAME2.0（2）
5位 ： 1.054 ： 米国ローレンスバークレー国立研究所
Cray Hopper（30）
6位 ： 1.050 ： 仏CEA国立研究所 Bull Bullx（97）
7位 ： 1.042 ： 米国オークリッジ国立研究所
IBM Roadrunner（16）
33位 ： 0.1914 ： 日本原子力研究開発機構 / 富士通（95）
（日本2位）
（緑字はGreen500ランク）

The Green 500
ベンチマーク電力性能、メガフロップス/W

1位 ：1684.20 ： 米国 IBM研究所 BlueGene/Q
プロトタイプ（116）
2位 ： 958.35 ： 日本 東工大 /HP/NEC TSUBAME2.0（4）
3位 ： 933.06 ： 米国 NCSA Hybrid Cluster実験機（403）
4位 ： 828.67 ： 日本 理研　京（170）
5-7位 ： 773.38 ： ドイツ ユーリッヒ大等
IBM QPACE SFB TR（207-209）
10位 ： 636.36 ： 日本 環境研（102）
（日本3位）
2位＋ ： 1448.03 ： 日本国立天文台 Grape-DR（383）
（12月末に追加）
（赤字はTop500ランク）

図3 TSUBAME2.0の2010年11月発表時のTop500,
Green500リストとの順位
（2010年末に改定版のリストが出され、国立天文台の
Grape-DRが2+位：1448.03、Top500 383位である
ことが発表された。しかしながら、TSUBAME2.0の
「運用スパコン世界一」のステータスに変化はない。）

はGPUがベクトル計算機として期待される高バンド幅プロセッサとしての実行時には顕在化しない。一方、通常のCPUの効率は近年の研究とアーキテクチャの進歩により90%以上となることが知られており、ここで15-20%の差が生じる。この効率の差は本質的ではなく、次世代のGPUのアーキテクチャおよびアルゴリズムの進歩により解消されることが期待される。

2. Heterogeneous Linpackにおいては、CPUはBLASの計算に寄与していない。しかしながら、Top500の結果においては使用されなくてもCPUの理論性能を加算する必要がある。TSUBAME2.0のそれぞれの計算ノードの2CPU（Intel Xeon Westmere 2.93Ghz）および3GPU（NVIDIA Tesla M2050）において、全体のピーク性能におけるCPUの寄与率は8%以上であり、それが損失となる。（無論、GPUやネットワークとの通信に参加しないCPUが寄与するアルゴリズムもその進歩により考えられる。実際 TSUBAME1.2の際にはそのようなアルゴリズムが用いられていた[3]。）
3. 今回用いたHeterogeneous LinpackではGPUを行列積計算エンジンとして用いており、CPUのメモリに在する部分行列をGPUにストリーム的に転送し、行列積を計算後ストリーム的に送り返している。通常のアプリケーションでは大きさnの行列に対し行列積の計算コストは$O(n^3)$、転送のオーバーヘッドは$O(n^2)$なので、nを十分大きくすれば転送コストは十分隠れるが、HPLにおいては負荷分散の為にn=数百～千程度の小行列を用いており、無視できないオーバーヘッドが生じる。

これらの理由により、合算で30%程度のオーバーヘッドがHPLでは生じる。これはハードウェアアーキテクチャとソフトウェア・アルゴリズムの進歩により十分解決できるものであり、我々のを含み今後の研究開発活動が期待される。

2010年11月に発表されたTop500リストにおいて、TSUBAME2.0は世界4位にランクされた。これはTSUBAME1における7位より高く、かつ日本の2位のマシンの性能と比較して約6倍の性能を示した。また、Top500の性能とその際の消費電力をベースにランク付けされるGreen500においては、実行時に1243.80KWを記録し、指標として958.35 Flops/Wを得て、2010年11月の発表時に世界2位、かつ実運用のスパコンとしては世界最高位であることが認定された（図2）。

この結果は技術的には大きな意味を持つ。現行の規則では、Top500とGreen500で同時に高位ランクとなることは困難である。実際、Top500の上位ランクは実運用のスパコンで占められており、一方Green500の上位ランクは小規模かつプロトタイプ・あるいは特殊アプリケーション分野の特殊なマシンである（図3）。唯一TSUBAME2.0だけがTop500とGreen500の世界5位以内にランキングされているが、Top500のトップマシンがGreen500の上位にランクされるのは以下の理由で困難である：

（1）Top500の上位マシンは数十～数百億円の価値がある運用スパコンであり、結果として運用には必要だが、性能電力比を悪化させる種々の要因を装備として抱える。例えば、運用スパコンは汎用の

計算に必要な大きなメモリ量が必要だが、これはTop500の性能向上にはあまり寄与しない。しかしながら、DRAMの電力消費量がスパコンに占める割合はかなりで、数十%に達する。逆に性能電力比を上昇させたければ、なるべく少ないメモリのマシンが有利となり、（Green500の上位マシンはことごとくそうであるが）汎用のスパコンとしては問題が大きい。

(2)Top500の上位ランクの獲得はセンターや国家の威信をかけたものであり、性能やランキングを少しでも上げる事はその他の全ての目的を無視しても行われることである。結果として、絶対性能を得るためにはチューニングでも性能電力比を犠牲にする。一方、Green500で上位ランクを得るには、単純にTop500に掲載されれば良い；よって、絶対性能を犠牲にし、Top500のランクが大幅に下がっても問題ない。これらの相反する要求を満たすのは困難である。

(3)Top500の上位マシンのLinpackの行列サイズは上記のようにn=数百万に達し、プロセッサ数も十万規模となる。一方アルゴリズムとマシンの特性により、Linpackは小型のマシンの方が効率が高いことが示される。よって、大型のマシンはLinpack性能を得るのは本質的に不利となる；例えば、単一ノードの実行と比較すると5-10%のペナルティがあることが我々を含む幾つかの研究で明らかになっている[3]

これらのペナルティにも関わらずランクインしたことが大きく評価されたのが2010年11月において「世界一電力効率が良い運用スパコン」特別賞が与えられた要因であり、その研究的バックグラウンドとしてはJST-CREST のUltra Low Power HPC（ULP-HPC）など、の継続的な基礎研究の成果であると言えよう。

3 実アプリケーションにおける性能・メモリバンド幅 --- GPU版ASUCAによる世界最高性能の高バンド幅気象アプリケーション

Top500におけるはスパコンの性能の重要な性能指標であるが、メモリバンド幅やネットワークバンド幅が支配的なアプリケーションに対してはあまり有効な指標とならない。流体構造系に代表される多くの実アプリケーションにおいては、それらの実行時においてどの程度のメモリ・ネットワークバンド幅が達成されるかが実性能に対して支配的であり、結果としてマシン全体が達成しうる合算バンド幅が非常に重要となる。近年のスパコンアーキテクチャでは、計算性能に対する両バンド幅の比率は物理的な制約により低下の一方である。また歴史的にはCrayのXシリーズ、NECのSXなどの所謂「ベクトルスーパーコンピュータ」は、大規模並列計算が困難だった時代に、これらのアプリケーションにおいて高性能を得るために設計された高バンド幅アーキテクチャであり、絶対ピーク性能に対して高い計算効率を誇った。

ここで指摘すべき技術的な要点は、「マシンのピーク性能に対する効率」は基本的にはこれらのバンド幅重視のマシンにおいてはミスリーディングな指標であることにある。十分な計算性能が確保されており、それが基本的に律速になっていないスパコンにおいては、絶対的なメモリ・ネットワークバンド幅と、それをどの程度実アプリケーションで活用できているかが基本的な支配項目である。つまり、どの程度のピーク性能があろうと基本的には全く関係なく、重要なのは絶対的なバンド幅とその利用効率のみなのである。むしろ、SXなどのクラシックなベクトルアーキテクチャでは密行列系の計算重視のアプリケーションに対してはそのバンド幅に見合う計算性能が確保できず、その性能が低く抑えられてしまう。しかしながら、絶対ピーク性能に対しては、その律速がどこにあれ（元々絶対性能が低い為）見せかけの計算効率が良くなる。

ここでの問題はそれによってマシンの優劣を論ずる間違った議論がしばしば展開されることであるが、上記の理由によってそれは正しくない。前編にあったように、GPUはソケットあたり・及び消費電力あたり計算性能・メモリバンド幅とも高い：倍精度演算において、同一ノードで比較すると演算性能は7倍以上であり、メモリバンド幅も同様に6-7である。つまり、CPUとの比較ではTSUBAMEは同世代のx86プロセッサのクラスタから鑑みれば3万ソケットのマシンに相当し、CPUコア数換算では20万コア近くに達する。これはオークリッジ国立研究所の、2010年6月まで世界一位であった、20万以上ものx86 CPUコアで構成されているCray Jaguar スパコンと計算速度・メモリバンド幅両方で同規模とみなせるが、実際はJaguarはTSUBAME2.0のCPUと比較してさらにメモリの実行バンド幅が遅い旧世代のCPUを用いているので、むしろ不利となる。

ASUCAのベンチマークはこれらの技術項目を確認するのに重要であった。特に、気象計算に代表される差分法による移流計算はメモリバンド幅が支配要素となるので、(1)十分に絶対バンド幅を得られるか（2）CPUと比較して上記のメモリバンド幅6-7倍程度の速度差を反映した性能差が得られるか、また(3)気象アプリケーションの大規模実行において、Jaguarとの性能比較はどうか、が重要なポイントとなった。ASUCAアプリケーションの詳細は前号の[5]に譲るが、結果としてGPU版ASUCAは最大3990GPUで実行され[6]、非常に良好な弱スケーリング性（ノード数の増加に比例して問題を増大したときの性能向上比）を示し、単精度で145Teraflops、倍精度でも76.1TeraFlopsを記録した（図4）。今までの世界記録はWRFというASUCAに似たアルゴリズムがベースの米国の先進的気象アプリケーションをJaguarで動作させたときに達成された約50Teraflops（倍精度）であり、それを大幅に上回ったこととなる。また、ソケットあたりの倍精度演算の性能比もほぼ6倍と、CPUとのメモリバンド幅の差の理論値をほぼ踏襲し、設計値の正しさを示した（図5）。

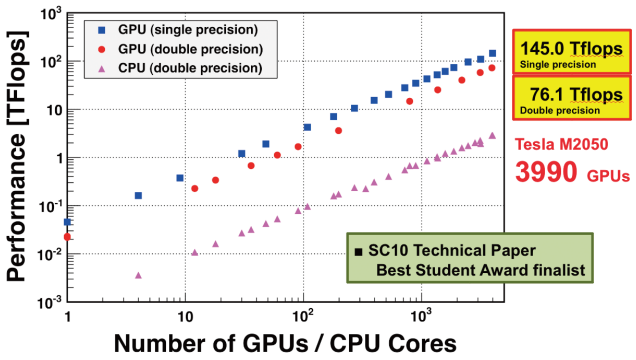

図4 ASUCAのベンチマーク。
弱スケーリングでは3990GPUまで
ほぼ線形に性能向上している。

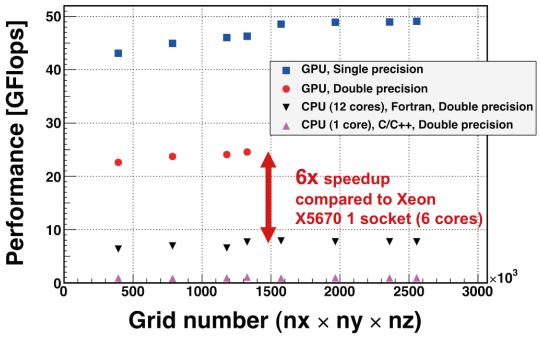

図5 ASUCAのベンチマーク。
ソケットあたりの倍精度時の性能差はほぼ6倍と、
理論値の差通りとなっている。

終わりに：今後のペタスケールアプリケーションへの期待 4

種々のベンチマークおよびマシンの初期デバッグが終了後、TSUBAME2.0は予定通り11月1日から運用が開始された。すでに多くのユーザーが連日種々のアプリケーションを動かしており、通常の最大アロケーションである5000CPU、1200GPU級の大規模アプリケーションの実行も珍しくない。また、超高速I/Oを行えるSSDやLUSTRE並列ファイルシステムの特性を生かしたデータ中心のアプリケーションも見られ、TSUBAME1.2時代から明らかな飛躍が見られており、ユーザーの評判も大変良い。

しかしながら、TSUBAME2.0の真の価値を生かすには、通常のスパコンには見られない特質をハード・ソフト共に有効に活用しなくてはならない。高メモリバンド幅を達成するにはGPUの利用は必須であるが、大規模アプリとして通信が絡むと、MPIとCUDA、CPUとGPUのハイブリッドプログラミングが必要となる。密行列系でもGPUはプログラムの書き方に性能がセンシティブなので、チューニングを注意深く行なわなくてはならない。幸いネットワークはフルバイセクションなので比較的最高性能が出やすいが、I/Oに関してはLustreのストライプサイズを最適に選択したり、局所I/Oを行うSSDとの有効な使い分けが必要となる場合もある。

全体を鑑みると、TSUBAME2.0のような超メニーコア・マルチスレッド・ベクトルの大量のコア群に小規模な低レーテンシスカラーコアが接続されたアーキテクチャは今後ポストペタやエクサフロップスの主流となるであろう。無論将来的にはそれらが一つのチップ内で高バンド幅に接続され、かつメモリは共有されるであろう。現状のGPUプログラムはPCI-e接続のハード上・ソフト上の区別（例えばCUDAにおけるホスト対デバイスの分離されたプログラミングとメモリ転送の必要性）を抱えているが、今回取り上げたベンチマークや、今後TSUBAME2.0で予定されている更なる大規模アプリケーションの実行およびベンチマークはそれらの可能性を先取りし、単なるFLOPSの尺度で測るのではない、真のペタフロップスアプリケーションとして種々の記録を達成し、その本来の科学的成果もあいまって世の中をリードしていくことが大いに期待される。また、4年後に来たる数十ペタフロップスのTSUBAME3.0や、他機関に入るであろう中間的なスパコンの設計に大いに役立つであろう。

参考文献

[1] The Top500 Supercomputing Sites http://www.top500.org/
[2] A. Petitet, R. C. Whaley, J. Dongarra, A. Cleary. "HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers v.2.0", http://www.netlib.org/benchmark/hpl/
[3] Endo, T.; Nukada, A.; Matsuoka, S.; Maruyama, N.Linpack evaluation on a supercomputer with heterogeneous accelerators, Proc. IEEE Parallel & Distributed Processing (IPDPS) 2010, the IEEE Press, Apr. 2010, pp.1-8.
[4] The Green 500 http://www.green500.org/
[5] 下川辺、青木「次世代気象モデルのフルGPU計算」TSUBAME E-Scinece Jounal, vol. 2, 東京工業大学学術国際情報センター、2010年11月、pp.9-13.
[6] T. Shimokawabe, T. Aoki, C. Muroi, J. Ishida, K. Kawano, T. Endo, A. Nukada, N. Maruyama, and S. Matsuoka. "An 80-Fold Speedup, 15.0 TFlops, Full GPU Acceleration of Non-Hydrostatic Weather Model ASUCA Production Code", Proc. 2010 ACM/IEEE Supercomputing, The ACM Press, Nov. 2010, pp.1-11.

星間水素原子ガス乱流のGPU計算

― TSUBAME1.2の120GPUで11.5 TFlops ―

村主 崇行*
* 京都大学　次世代研究者育成センター

銀河において、星間水素原子ガス乱流の駆動源となっている
水素原子ガスの相転移にもとづく熱的不安定を研究するために、3次元流体シミュレーションを行った。
シミュレーション結果を可視化するとともに、塊検出・スペクトル解析を行い、超音速・圧縮性乱流と、標準的な非圧縮性乱流が階層構造をなす星間乱流のありさまを明らかにした。この階層構造を発見できるだけの高解像度シミュレーションが可能になったのは、TSUBAMEのようなGPU並列計算機がもたらした膨大な演算力のおかげといえる。

はじめに 1

僕の仕事を説明する、お気に入りの造語がある。「安楽椅子宇宙飛行士」だ。これまで人類が探査機を送り込んだのは、せいぜい太陽系の内側まで。となりの恒星ですら、現場に行って証拠を集めることは、しばらくできそうにない。この制約を超えて宇宙のことを知るには、地球周辺というたった一点での観測から得られる証拠と、科学の法則をもとに慎重に推理を組み立てていく。法則をもとにいかにゆたかな構造が生まれ、それがさまざまな制約のもとでどのように観えるのかを知る上で、コンピューターシミュレーションは大いに力になる。たゆまぬ計算能力の進歩が可能にした、最近の高解像度・三次元シミュレーションのムービーは、安楽椅子に居ながら宇宙を旅している気分にしてくれる。

今回僕がTSUBAMEを利用して取り組んだのは、恒星の質量を決めるメカニズムの研究だ。銀河系には1cm^{3}あたり水素原子約1個というとても低密度の星間ガスがあり、これがある大きさの塊に分裂し収縮することで恒星が生まれる。もしこの大きさが1桁小さければ、恒星の核融合反応は始まらず、暗い星ばかりになる。もし1桁大きければ、星はあっという間に核融合しつくしてブラックホールばかりになってしまうだろう。何故星間ガスは、銀河に輝く星々が満ち、とある惑星には生命が誕生するような、ちょうどのサイズに分裂するのだろうか。それを決めているのが、水素原子ガスの熱的不安定にともなう乱流---今回の研究テーマだ。

星間水素原子ガスは、わずかに含んでいるCO分子などが輻射を放つことで冷却されたり、まわりの恒星から温められたりするため、現在の銀河の環境では2つの安定相を持つ。水には気体（水蒸気）と液体の2つの相があり、冬の寒い日に息を吐くと、水蒸気が無数の水滴にかわるように、星間水素原子ガスも、超新星爆発などの衝撃をきっかけとして、温かく低密度の相から冷たく高密度の塊を何千個も、一気に形成する。このダイナミックな現象を通じて、それぞれの塊ががやがて恒星となるときの明るさ、色、寿命が決まっていくのだ。むろん、このあと星が生まれ、惑星系を持つまでにはもっと沢山の過程がある。謎も多い。宇宙が相手の推理はとうぶん終わりそうにない。

学際大規模情報基盤共同利用について 2

まず、東京工業大学の計算機を利用することができたいきさつに触れたい。学際大規模情報基盤共同利用・共同研究拠点という、スーパーコンピュータを有する全国の8大学がつくるネットワーク型拠点がある。私はその公募型共同研究という枠組に、平成21年度の試行段階からお世話になって、東京工業大学のTSUBAMEを利用してきた。

公募型共同研究には、私と、長崎大学 先端計算研究センター超並列部門長　准教授の濱田剛の共同研究として申し込んだ。濱田さんのグループは長崎大学でGPUクラスタDEGIMA（DEstination of GPU Intensive Machines）の開発・運用を推進している。また京都大学にも、村主が中心となって構築した小規模なGPUクラスタ TenGU（Tenmon GPU Cluster）がある。学際型共同研究として異分野の多くの研究者と共同してコード開発・研究に当たれたことはたいへん有意義な経験であった。特性の異なる複数の計算機を研究の段階に応じ使い分けられたことも大変便利であった。また東京工業大学のみなさんや、東京大学のJHPCN担当者様方も親切で、技術的サポートや事務的サポートを的確かつ迅速にいただくことができた。

星間乱流のシミュレーション 3

本研究に応募するに当たっていくつかテーマを用意していたが、本稿で紹介するのはこのうち「分子雲形成領域における、2相水素ガスの乱流状態の数値シミュレーションと解析」で、はじめにの項でも述べたように、多岐にわたる宇宙物理学のテーマのなかでも、星間物質の研究に貢献するものだ。

銀河において恒星・惑星系が生まれる場である星間物質で観測されている乱流は圧縮性、非等方的であり、加えて輻射平衡を考慮した水素原子ガスの状態方程式は理想気体とは大きく異なり、100倍程度密度の違う2相が共存する乱流状態である（Field et al, 1969[1]）。にも

星間水素原子ガス乱流のGPU計算
― TSUBAME1.2の120 GPUで11.5 TFlops ―

かかわらず観測される乱流の速度スペクトルは、一様、非圧縮、等方を仮定するKolmogorovスペクトル（速度場の冪指数α_v=11/3）とよく一致している。ただしKolmogorovからの有意なずれ（α_v=3.87±0.11）を観測した報告もある（Chepurnov et.al. 2010[2]）。この乱流の解明は星間現象を理解する上でのグランドチャレンジの1つといえ、これまでにも1000^3規模の解像度のシミュレーションが行われているが、計算機の能力のため、追跡できる時間が限られていた。GPUの高い演算能力を活用することで、同じ解像度のまま、長時間にわたって乱流をシミュレートできるようになり、平衡状態のより詳しい情報を引き出せるようになる。

シミュレートすべき式は、以下の加熱・冷却項を含んだ流体力学の基礎方程式、Navier-Stokes方程式である。

$$\frac{\partial \rho}{\partial t} + \nabla \bullet (\rho \mathbf{u}) = 0$$
$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \bullet (\rho \mathbf{u}\tilde{\mathbf{u}}) + \nabla P = 0$$
$$\frac{\partial E}{\partial t} + \nabla \bullet ((E+P)\mathbf{u}) = \Gamma(\rho,T) - \Lambda(\rho,T)$$

ここで、加熱項Γと冷却項Λは、星間環境で考慮すべき様々な加熱・冷却過程（e.g. Koyama & Inutsuka 2000[3]）をもとに、Inoue & Inutsuka（2008）[4]が提案した次のフィッティング式をもちいる。

$$\Gamma(\rho,T) = 6.802\times10^{-2}\rho$$
$$\Lambda(\rho,T) = \rho^2\left[3.871\times10^{5}\exp\left(\frac{-19.25}{T+0.1626}\right) + 4.25\times10^{-2}\sqrt{T}\exp\left(\frac{-1.496\times10^{-2}}{T}\right)\right]$$

定数がいくつか出てきたが、星間物質のスケールで典型的な物理量を使って無次元化をおこなっている。つまり、

密度： $\rho_w = 1.211\times10^{-21}$ [kg m^{-3}]

圧力： $p_w = 4.832\times10^{-8}$ [Pa]

長さ： $l_w = 1\mathrm{parsec} = 3.086\times10^{16}$ [m]

これは、このガスの音速が約 秒速10kmであることを意味する。これは地球を脱出するロケットの速さよりも速い！しかし、この音速をもってしても、1パーセク（約3.62光年）を横切るのには十万年を要する。この研究で想定した計算空間（図1）は一辺20パーセクであり、乱流の平衡状態を観察するため、音波が8往復する程度の時間をシミュレートした。じつに1600万年間に相当するシミュレーションを行ったことになる。

開発した流体コードの特性を列挙すれば、MPI GPU Full-Godunov 2nd order MUSCL 3-dimensional uniform mesh Navier-Stokes equations solverとなろうか。Godunov法とは古くロシア人数学者Godunov（1959）[5]によって提案された手法で、またvan Leer（1979）[6]らの努力によって確立されたその高次精度版がMUSCL法である。そのアイデアは、ある平面を境界に左右の物理量が一定という初期条件（Riemann問題）は解析的に解くことができるので、各メッシュ境界でこの解をあてはめてやろうというものである。超音速流に特有の衝撃波面をシャープに捉えるには欠かせない処方である。そのかわり演算数は1メッシュ、1ステップあたり約3000flopも必要だが、そこはGPUの性能が活かせるというものだ。

むろんこのコードはMPI経由で多数のGPUを並列に利用して計算を行うことができる。これに加え、故障率の高いマシンや、キューの実行時間に制限があるマシンでも長時間の計算が行えるよう、シミュレーションの内部状態を適宜出力してそこから復帰できるチェックポインティング

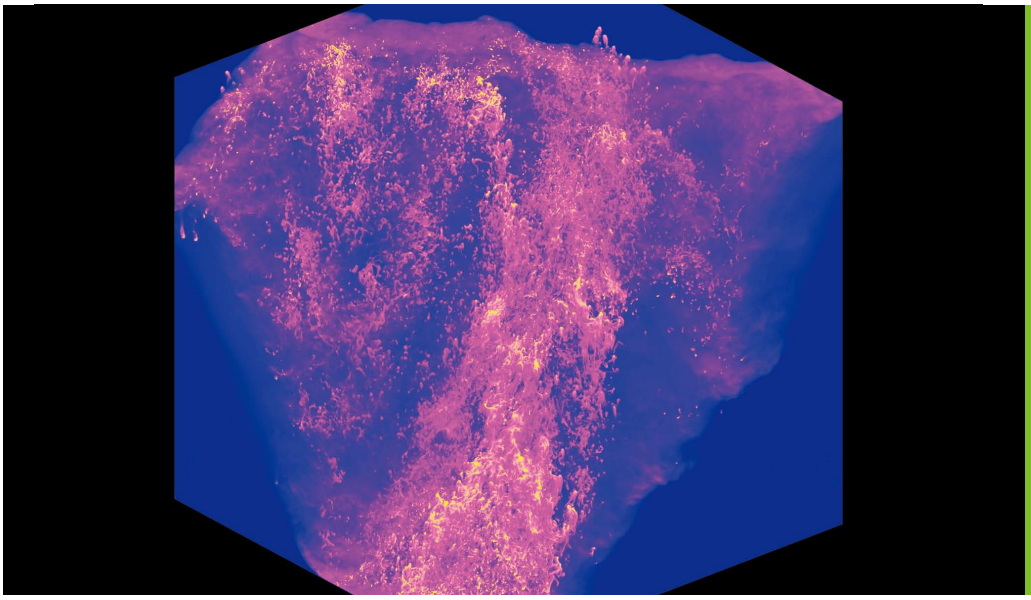

図1
東工大のGPUスパコンTSUBAMEを利用して行った1440^3のシミュレーションの可視化。

機能をつけた。またムービー作成のための不可逆圧縮データ出力機能を、東京工業大学の青木尊之教授のアドバイスも得て実装した。

コード開発やチューニングはおもに長崎大学DEGIMAで行った。DEGIMAでは約800×NVIDIA GT200 GPU型演算器が利用可能であり、うち576個はInfiniBandで連結されていた。単精度ピーク演算性能は514.9TFlop/sであり、最大1769'4720スレッドを同時実行可能であった。また総ビデオメモリは約460GB、総ビデオメモリ帯域幅は64.454TB/sであった。DEGIMAでは、上記約1280^3の解像度のシミュレーションにおいて40.91Tflopsの実効性能を達成した。

つぎに東工大のGPUスパコンTSUBAMEを使って長時間計算を進めた。期間は7/6-7/9、7/20-7/23、8/20-8/24の3回、のべ10日であった。利用したhpc1tes2 キューでは、120個のNVIDIA GT200 GPU型演算器が利用可能であり、単精度ピーク演算性能は124.2TFlop/s、また最大122'880スレッドを同時実行可能であった。総ビデオメモリは480GB、ビデオメモリ帯域幅は合計 1.224TB/sであった。また利用負担金は13万2千円であった。これにより1440^3の解像度で、約8 sound crossing timeの計算ができ、乱流スペクトルを解析する上で十分な量のデータが得られた。また約200枚の可視化用のデータを得た。また実効演算性能は11.5Tflopsであった。

解析と可視化 4

本研究のシミュレーションではこれまでにない大解像度を達成したことから、その出力データ量も膨大なものとなり、その解析および可視化のためのあらたなツール群を開発したことを述べておきたい。例えば1つのスナップショットのサイズは約60GBであり、典型的なパソコンのメモリに収まらない。そこでデータの2次元断面程度のメモリしか消費せず、かつディスクアクセスを最小限に抑えたまま、スペクトル解析や塊検出、可視化などを行うアルゴリズムが必要であった。

例えば、塊検出というデータ解析では、ある密度の閾値を決め、それより密度の大きいセルの連結成分をすべて検出し、おのおの高密度塊（Clump）とその周辺を取り巻くガス（Veil）について密度、音速、相対速度などの物理量を求め、その統計解析を行う必要がある。連結成分の検出には、素朴に考えると3次元データへのランダムアクセスが必要に思える。我々は以上の解析を、（データの2次元断面）×（定数倍）しかメモリに持たずに、全データを2回シーケンシャルアクセスしただけで行うアルゴリズムを開発した。これにより、パソコンでも、解析対象データをディスクに置いたまま、実用上問題のない時間で解析ができるようになった。

この解析の結果、Cold Neutral Medium（CNM:冷たい高密度の星間ガス塊）の速度は確かにCNMの音速より速い。しかし、CNMは周囲を不安定相の気体に取り巻かれている。その音速よりはCNMの速度は遅い、という状況が分かった。塊の運動は本質的にこの不安定相の気体内部での亜音速乱流なので、塊の挙動は非圧縮Kolmogorov乱流で説明できるのではないか、との示唆を得た（図2）。

この部分を検証するために、TSUBAMEで行った1440^3の解像度のシミュレーションのスペクトル解析を行った。これに際しても、膨大な3次元データのフーリエ解析を直接行う困難を避けるため、次のようなアルゴリズムを開発した。つまりまずディスクに保存されたスナップショットを1回シーケンシャルアクセスし、その間に各方向への2次元投影を作成する。次にその投影された速度を2次元フーリエ変換する。最後に2次元のスペクトルから、天文観測で用いられている手法を模して3次元スペクトルを再構築する。このアルゴリズムを採用することで、スペクトル解析の所要時間を短縮できた。また、地球という一点から、天を2次元球面として観測せざるを得ないという制限のある天文観測と比較・検証する上でも有意義なデータを得た。

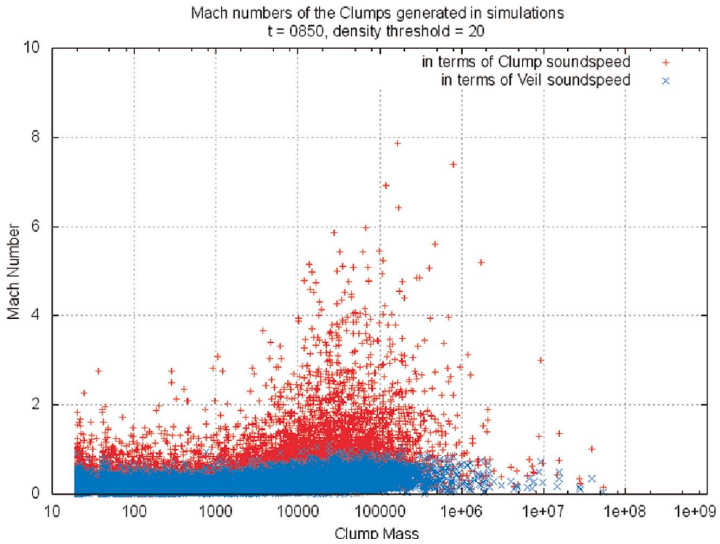

図2
塊検出解析の結果。
検出された塊の質量を横軸に、
検出された塊（Clump）の、
外部（Veil）に対する速度を
プロットしたもの。Clumpの速度は、
Clump自身の音速を基準にして測ると
超音速運動になっているが、
Veilの音速を基準にすればあくまで
亜音速運動（Mach数<1）
であることがわかった。

解析の結果、シミュレートされた乱流の冪は、超音速乱流領域、Kolmogorov様領域、数値的散逸領域の3重構造を示した（図3）。全体としての冪はChepurnov et.al. 2010の速度・密度スペクトル（α_v=3.87±0.11）（α_ε=3.0±0.1）と誤差の範囲内で一致した。超音速領域は比較的softer, Kolmogorov様領域は比較的harderである。これらのことから、Chepurnov et.al.のような非Kolmogorov的観測は超音速領域とKolmogorov領域の重ね合わせとして説明できるかもしれない、という示唆を得た。

このデータの可視化にあたっても、ハードディスクに置いたまま、1回あるいは数回のシーケンシャルアクセスのみで、流体速度、密度、化学状態、温度など、さまざまな物理量を可視化できるプログラムを開発した。これにより、熱不安定性を引き起こしている衝撃波面が激しくゆらぎ、強い噴出流が間欠的に突きあげる乱流のありさまが明らかとなった（図1）。

また、大規模データの可視化を得意とする京都大学小山田研と共同研究により、40面タイルドディスプレイを利用した大画面・高解像度の可視化を行った（図4）。さらにTSUBAMEで実行した計算を、青木尊之教授と協力してムービーにし、TSUBAMEの広報活動等に活用していただいた。

分子雲形成領域における、2相水素ガスの乱流状態の数値シミュレーションについては、その後も現在に至るまで、物理条件を変えた計算や、追加のデータ解析解析を行ったりしながら研究を進めている。

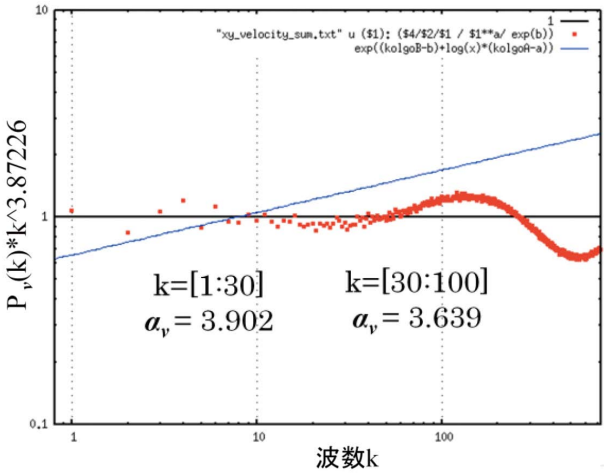

図3 速度スペクトル解析結果とそのフィット。

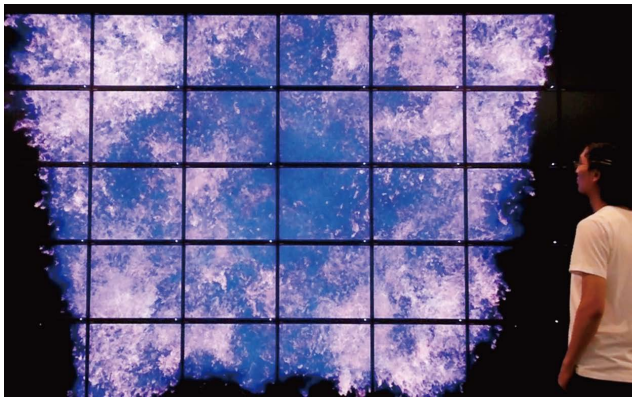

図4 京都大学小山田研と共同で行った、40面ディスプレイによる可視化。

流体から磁気流体へ 5

以上の流体コードの開発・運用経験をふまえ、8月ごろより磁気流体コードを開発開始した。磁気流体力学は電離気体、プラズマの挙動を記述する基本方程式のひとつである。宇宙物理学では降着円盤、諸々のジェット現象、太陽、地球磁気圏まで、さまざまな天体現象の活動に磁場が不可欠な役割を果たしている。また磁気流体力学は地上でのプラズマ制御、たとえば核融合炉の研究などにも使われている。GPGPUを使うことで、いままでにない高解像度・高速の磁気流体力学シミュレーションが可能になれば、これらの学問をも活性化するし、また太陽フレアの予測、人工衛星の防御などを通じて人類の生活にも直接の利益をもたらすだろう。

現在のバージョンは時間・空間二次精度のコードで、磁気流体力学のリーマン・ソルバーとしてHLLDを採用した（Miyoshi & Kusano 2005[7]）。現在は磁気流体コードの開発を続けるかたわら、開発中のコードを用いてテストを行っており、流体力学や磁気流体力学のMHD衝撃波がきちんと解けることを確認している段階である。（図5）

多数GPU向けの流体コードを開発、運用した経験から、素朴なコードを書いていては行数があまりに多くなってしまって大変である一方、C++/CUDAによる抽象化をしようとしても、C++の抽象化能力の枠内では、限界があることもわかってきた。そこで、あくまでも研究の補助的研究としての所要時間にとどめつつ、本研究で必要となるようなコード（とくに、流体コードの開発からまなんだもの）を簡潔に生成できる道具を、との拘束条件のもとで、簡易的なコード・ジェネレータcprbを設計した。（https://github.com/nushio3/cprb）cprbはHaskellで書かれており、ruby風の文法で、C++のコードをメタプログラミングできる。現在のコードはcprbを使って書いているが実に快適である。並列プログラミングのための新しい手法を模索する上での一歩となったと考えている。

僕が磁気流体コードを使ってぜひ解き明かしたいと思っているのは原始惑星系円盤における放電現象。宇宙の雲のつぎは宇宙の雷というわけだ。実は僕は氷ダストどうしの衝突による放電---ようは、地上で起きているのと同じメカニズムの雷が、原始惑星系円盤でも起こりうるのではないか、という研究をしたことがある（Muranushi,2010 [8]）。また原始惑星系円盤において、Resistive MHDの素過程のみから放電を引き起こすメカニズムとしてInutsuka-Sano（2005）[9]が提案されて

いた。この論文においては、当メカニズムはone-zoneモデルで説明されていたが、最近Okuzumiにより、3次元的に不安定を引き起こしうることが予言された。ぜひこれをシミュレートしてみたい。そのためにはResistive磁気流体力学シミュレーションが必要で、ゆくゆくはダストとか化学反応も考慮していく必要がある。3次元、高解像度シミュレーションが必要なので、ぜひこれからも自作のコードでGPU並列計算機の性能を引き出して行きたい。

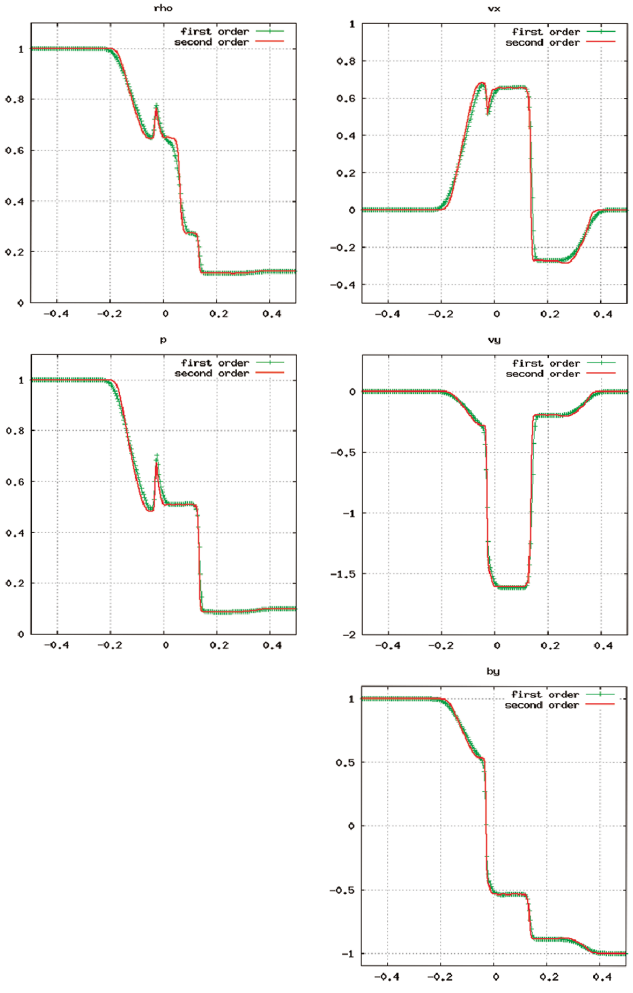

図5 磁気流体コードの衝撃波問題のテスト。
Miyoshi & Kusano 2005[7]のFig. 8. に対応。
$(\rho, p, v_x, v_y, v_z, B_y, B_z) = (1, 1, 0, 0, 0, 1, 0)$を左側、
$(\rho, p, v_x, v_y, v_z, B_y, B_z) = (0.125, 0.1, 0, 0, 0, -1, 0)$を右側、
$B_x = 0.75$を初期条件とし、$t = 0.1$まで積分したもので、
図の解像度は256メッシュ。

謝辞

村主崇行は京都大学次世代研究者育成センターから研究の支援をいただいている。濱田剛さん、青木尊之さんほかコード開発に当たってアドバイスをいただいた方々には深く感謝したい。またこの研究は学際大規模情報基盤共同利用・共同研究拠点にかかわる皆様のご協力のおかげで実現できた。記して謝意を表す。

参考文献

[1] Field G. B., Goldsmith D. W., Habing H. J., (1969), Astrophysical Journal Letters, 155, L149+
[2] Chepurnov A., Lazarian A., (2010), Astrophysical Journal, 710, 853
[3] Koyama, H., & Inutsuka, S. (2000), Astrophysical Journal, 532, 980
[4] Inoue, T. & Innutsuka, S. (2008), Astrophysical Journal, 687:303-310
[5] Godunov, S. K. (1959), Matematicheskii Sbornik, 47, 271-306
[6] van Leer, B. (1979) Journal of Computational Physics 32, 101-136
[7] Miyoshi & Kusano, Journal of Computational Physics 208 (2005) 315–344
[8] Muranushi, T. (2010), Monthly Notices of the Royal Astronomical Society. 401, 2641–2664
[9] Inutsuka & Sano, The Astrophysical Journal, Volume 628, Issue 2, pp. L155-L158 (2005)

高速フーリエ変換とGPU

額田 彰 *
* 東京工業大学　学術国際情報センター

高速フーリエ変換は現在マルチメディア分野から大規模シミュレーション分野に至るまで幅広い分野で利用されており、
常に高速化が求められている代表的な計算の一つである。TSUBAME2.0 では大々的にGPU が導入されている。
CPU で実行する場合と比べてGPU を使うと高速フーリエ変換の計算がどれくらい加速されるのかを紹介する。

1 はじめに

FFT とは何かを正確に理解している人はあまり多くないであろう。今の日本の若い人たちに「FFT って何の略なのか知ってますか？」と尋ねると「Final Fantasy Tactics」と答える人の方が多いような気がする。これはスクウェア社が1997 年に発売したゲームのようで、実際にGoogle でFFT を検索するとこちらの関連するホームページが圧倒的に上位を占めていた。私もFinal Fantasy シリーズはそれなりに遊んできたが、Tactics はやったことがない。もちろん私がここで紹介するFFT はこれのことではなく、高速フーリエ変換（Fast Fourier Transform）のことである。

2 高速フーリエ変換

高速フーリエ変換とは以下の数式で定義されるようなN個の入力データ$X(k)$ からN 個の出力データ$Y(k)$ への数値変換である。

$$Y(k) = \sum_{j=0}^{N-1} X(j) e^{-2\pi ijk/N}$$

何だか難しい数式であるが、それほど難しいことを計算しているわけではない。具体的な用例を挙げると、物理空間や時間方向のN個のサンプル点から各周波数の成分がどれくらいかを求めるというものである。より身近な例ではMP3 プレイヤーなどに使われている技術がある。CD などに収録されている音楽データを圧縮する時に、どの周波数成分（＝音の高さ）が大きいのかを調べている。人の耳は強い周波数成分があるとその周りの弱めの音が聞き取りにくいという性質を利用して、それならなくても同じではないか？ということでばっさりカットすることでデータサイズを圧縮している。このようなマルチメディア系の小規模な用途だけでなくスーパーコンピュータで計算するような分子シミュレーションなどの大規模なアプリケーションまでFFT は非常に幅広い分野で利用されている。

N点FFT を上に挙げた式通りに計算すると$O(N^2)$ の演算が必要になる。$N = 512$ の場合およそ2,097,152 回の浮動小数点数演算が行われることになる。しかし上の式で$Y(k)$ を全て計算する場合には途中結果で共通する部分があるため実際に必要な計算量はNに大きく依存するが、一般に$O(N \log N)$になる。$N = 512$ の場合でおよそ24, 192 回だけで計算することができ、定義式通りに計算する場合の90 分の1 くらいである。

このように浮動小数点数演算の数を最小化してしまうと、特に現在の高性能なCPUでは演算数よりも、計算の前にデータをメモリから読み込む操作と計算の後にデータをメモリに書き込む操作が実行時間のほとんどを占めるようになる。このためFFT の計算は一般的にはメモリアクセスがボトルネックと言われている。TSUBAME 2.0 に搭載されているCPU（Westmere-EP, 2.93GHz, 6コア）のメモリバンド幅（メモリ転送速度）は理論最大性能で32GB/s であるが、これでもFFT の計算には十分ではない。

TSUBAME 2.0 の計算ノードには2 個のCPUが搭載されているが、それに加えて3 個のGPUが搭載されている。スーパーコンピュータにはものすごく大きなディスプレイ画面が必要なのでビデオカードもたくさんある・・・・というわけではない。これらのGPU はCPU と同じように計算資源として用意されているのである。3Dゲームなどに欠かせないGPU であるが、中で何をやっているのかについてはあまり知られていない。綺麗な画面を表示するために実はものすごく沢山の演算が行われている。画面描画処理は普通の計算と違って待ってはくれない。決められた時間内に1 枚の画面の計算を終えなければならない。このためGPU は非常に多くの演算を同時に実行できるようになっている。

このような高性能でしかも安価な計算資源を画像処理以外の用途にも使おうという動きは何年も前からあった。GPU を利用した汎用計算ということでGeneral Purpose computing on Graphics Processing Units を略してGPGPUと呼ばれている。初期は目的の計算を画像処理プログラムとして実行するというもので画像処理の枠組みに無理やり組み込んでいたために効率があまり出ていなかった。最近になってGPUベンダもGPGPU を意識したGPU を設計するようになった。GPUによる高速な動画エンコードのような使い方も既に

今では珍しくない。GPU用のプログラム開発も以前と比べると格段に簡単になり、研究者やソフトウェア開発者を中心にGPUの利用が普及して来た。特にNVIDIA社のCUDAという統合開発環境はGPUプログラムの開発にC言語を拡張したプログラミング言語を採用しており現在のGPGPUの主流になっている。CUDAはNVIDIA社のGPU専用の開発環境であるが、各社製GPUやマルチコアCPUを対象とした共通のプログラム言語としてOpenCLというものも策定されている。

グラフィックス処理は通常あまり高い精度を必要としないので単精度の浮動小数点数演算を用いている。このためCPUで動かしていたプログラムをGPUに移植する場合には精度に関する問題が度々議論になっていた。最近のGPUはGPGPU用途のために倍精度演算もサポートするようになったため、この精度問題は解決されている。

GPUの長所は高い演算性能だけではない。大量のデータをチップ内に転送するための高いメモリバンド幅も備えている点が従来の演算アクセラレータと大きく異なる点である。GPU（TeslaM2050）のメモリバンド幅は約150GB/sとCPUの4倍以上になる。そこでGPUでFFTの計算をすればCPUの時の4倍以上の性能が出せるのではないだろうか？

CPU と GPU を使った FFT の性能 3

それでは早速性能を見てみよう。図1はTSUBAME 2.0に搭載されるCPUまたはGPUでFFTを計算した場合の性能である。性能を示す単位としてGFLOPS（ギガフロップス）を用いている。これは1秒あたりに何ギガ（10億）回の浮動小数点数演算を実行することができたかを示している。それぞれのFFTライブラリで実際に何回の演算を実行したかは不明なのでFFTでは慣例としてN点FFTの演算数を$5N\log N$として扱う。

まずCPUを使用するFFTライブラリとしてはFFTWライブラリ3.1.2とIntel MKLライブラリ10.2.5がある。どちらもTSUBAME 2.0環境に用意されている。マルチスレッドに対応しており、1個のCPUの6コア全てを使用した場合の性能を表示している。次にGPUを使用するFFTライブラリとしてNVIDIA社が提供するCUFFTライブラリ3.1の性能を表示している。このライブラリはCUDAツールキット3.1に付属するものである。そして最後のNukadaFFTは名前から推測できるかと思うが私が開発しているFFTライブラリである。

FFTWライブラリとMKLライブラリの性能はほぼ同じである。細かい実装は異なるのであるが、最終的にはメモリへのアクセス速度で性能が決まっている。

CUFFTライブラリとNukadaFFTライブラリの性能には大きな差がある。CUFFTライブラリは単精度の性能はとても高いのであるが、倍精度の方はまだ十分な最適化が行われていないようである。

メモリバンド幅の理論値ではGPUはCPUの4.6倍くらいである。ところがFFTの計算性能では7倍もの性能差がある。これは理論値と実際の転送速度の間に大きな差があるからである。GPUのメモリバンド幅は約150GB/sであるが、FFTの計算をしているときの転送レートは82GB/s程度しかない。このように理論最大性能より低くなっている理由は幾つかある。Tesla M2050のメモリはエラー訂正機能を持っている。メモリへ読み書きする際にデータ本体に加えてエラー訂正用の符号を転送しているためメモリバンド幅を余計に消費してしまう。メモリへの書き込み操作は読み込み操作と比べると少し遅くなるという性質がある。FFTの計算の場合にはメモリアクセスの50%が書き込み操作になるのでメモリアクセス効率がかなり低下してしまう。また単純にメモリをコピーするだけの場合と比べ

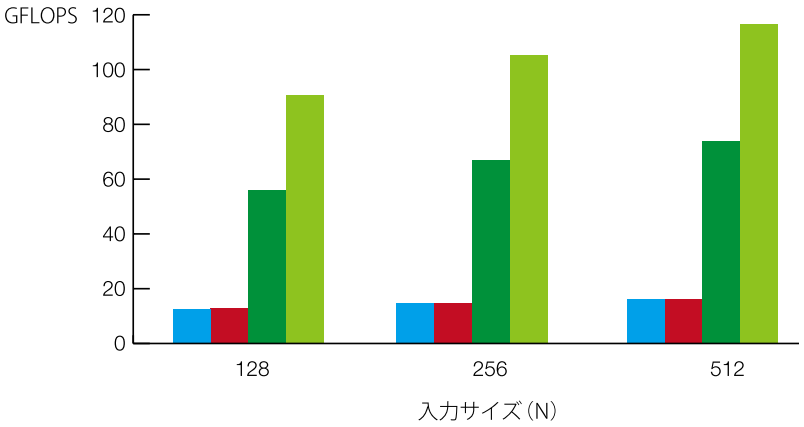

図1
このグラフは128点FFT,256点FFT,512点FFTを65,536組、倍精度で計算する場合の性能(GFLOPS）である。
Intel MKLライブラリ10.2.5とFFTWライブラリ3.1.2は6コアのCPU(Westmere-EP, 2.93GHz）で、NVIDIA CUFFTライブラリ3.2とNukadaFFTライブラリ1.0はGPU(NVIDIA Tesla M2050）を使用している。

るとFFT の計算を行っているとどうしてもメモリアクセス効率が低下してしまう。CPU の場合も同様である。単純なメモリコピーを行うときでもその転送スピードは20GB/s 以下になるが、FFT の計算をしているときは12GB/s にも満たないのである。

4 自動チューニング

GPU 製品の種類はとても多い。CUDA に対応するNVIDIA社のGPUはGeForce シリーズ、Quadro シリーズ、Tesla シリーズ、ION シリーズやモバイルGPUを含めると既に100種を超えている。主要なCUDA 対応GPU の仕様を表1に記載した。GPUの性能は年々向上し続けており、プロセッサコア数や演算性能、メモリバンド幅などはGPU のモデルによって様々である。GeForce シリーズはコンシューマ向けのゲーム用途のGPUである。一方Teslaシリーズはハイパフォーマンス用途を前提としていて、倍精度演算性能を強化しているし、メモリ容量も大きく、メモリのエラー訂正機能も搭載している。もちろんTSUBAME 2.0 に搭載されているのはこのTesla シリーズの方である。

FFT の計算を実行するGPU プログラムは一つだけではない。高い性能を実現するためには入力サイズに対応したプログラムを用意する必要があるし、同じ入力サイズの場合でも色々な最適化パラメータがあり、それらのパラメータの組み合わせの数だけのプログラムのバリエーションが考えられる。ではその中でどれが一番速く計算できるのであろうか？実はGPU のモデルによって異なる。

私がNukadaFFT ライブラリの主要な部分を開発していた頃はまだTesla M2050 もGeForceGTX 480 も存在してなかったため、主にGeForce GTX 280 を使っていた。それなのにTesla M2050 のような新しいGPU でも高い性能を実現できているのはなぜだろうか？これは自動チューニング機能を実装しているからなのである。ここで用いている自動チューニング手法は難しいものではない。考えられる全てのプログラムのバリエーションを自動的に生成して、それらを片っ端から実行してみて一番速かったものを選んでいるだけである。

この自動チューニングにかかる時間は入力サイズや使用するGPUやCPU などに依存するが、大抵の場合1 分以内に完了する。一般的なFFT の利用方法としては同じ入力サイズでのFFT の計算を何回も繰り返すので初回のこの自動チューニングにかかる時間は問題にならないだろう。また自動チューニングの結果はファイルに保存されるので次回実行時からは自動チューニングが必要ない。

GPU を使う場合に自動チューニングが必要という訳ではない。ライブラリを提供するような場合を除けば利用するGPU のモデルはかなり限られるだろう。FFT のように入力データのサイズに大きく依存するのであれば自動化した方が効率がよいが大抵は手作業で最適化を行うことも可能である。

5 おわりに

GPU によるFFT の計算の現状を紹介した。TSUBAME 2.0 に搭載されているCPUとGPUで比較した場合、GPU を使うことによって約7倍の速度向上が得られている。これはFFT に限ったことではない。他にも流体計算などのメモリアクセスがボトルネックになる計算にも高いメモリバンド幅を持つGPU は適している。もちろんGPU は演算量が多い計算も得意である。今後も色々なアプリケーションがGPU 化されていくことであろう。

NukadaFFT ライブラリは今でも頻繁にアップデートが繰り返されている。最新版は以下のURL から入手可能であるので興味がある方は是非試していただきたい。

http://matsu-www.is.titech.ac.jp/~nukada/nufft/

表1 主要なCUDA 対応GPU の仕様。

	GeForce 8800 GTX	GeForce GTX 280	GeForce GTX 480	Tesla M2050
発売時期	2006 年11 月	2008 年6 月	2010 年4 月	2010 年7 月
プロセッサコア数	128	240	480	448
メモリ容量	768MB	1024MB	1536MB	3072MB
倍精度演算性能	非サポート	77.8GFLOPS	168.1GFLOPS	515.0GFLOPS
メモリバンド幅	86.4GB/s	141.7GB/s	177.4GB/s	150.2GB/s

謝辞

本研究の一部は科学技術振興機構（JST）の戦略的創造研究推進事業（CREST）『ULP-HPC:次世代テクノロジのモデル化・最適化による超低消費電力ハイパフォーマンスコンピューティング』、Microsoft Technical Computing Initiative "HPC-GPGPU: Large-Scale Commodity Accelerated Clusters and its Application to Advanced Structural Proteomics"、NVIDIA CUDA Center of Excellence Program、及び科学研究費補助金若手研究(A)22680002 によるものである。

参考文献

[1] 青木尊之, 額田彰. 『はじめてのCUDAプログラミング』, 工学社, 東京, 2009 年11 月.

[2] Akira Nukada and Satoshi Matsuoka.Auto-Tuning 3-D FFT Library for CUDA GPUs. In Proceedings of the ACM/IEEE conference on Supercomputing (SC09), Portland, ACM, Page 1-10, November 2009.

● TSUBAME e-Science Journal No.3
2011年2月25日　東京工業大学 学術国際情報センター発行 ©
ISSN 2185-6028
デザイン・レイアウト：キックアンドパンチ
編集：　TSUBAME e-Science Journal 編集室
青木尊之　渡邊寿雄　関嶋政和
ピパットポンサー・ティラポン　深山史子
住所：　〒 152-8550 東京都目黒区大岡山 2-12-1-E2-1
電話：　03-5734-2087　FAX：03-5734-3198
E-mail：tsubame_j@sim.gsic.titech.ac.jp
URL：　http://www.gsic.titech.ac.jp/

TSUBAME

TSUBAME 共同利用サービス

『みんなのスパコン』TSUBAMEは、
当初は主に東工大学内の研究・教育のために利用されておりましたが、
平成21年7月よりTSUBAME 共同利用サービスを開始し、
学術・産業・社会へと広く貢献しております。

課題公募する利用区分とカテゴリー

共同利用サービスには、「学術利用」、「産業利用」、「社会貢献利用」の3つの利用区分があり、さらに「成果公開」と「成果非公開」のカテゴリーがあります。現在は随時申請を受け付けており、申請課題は厳正な審査の下、採択の可否を決定します。採択課題の利用期間は当該年度末までです。

TSUBAME 共同利用とは…

東工大学内のみならず、より多くの方にTSUBAMEサービスを提供

他大学や公的研究機関の研究者の**学術利用**［有償利用］

民間企業の方の**産業利用**［有償・無償利用］

その他の組織による社会的貢献のための**社会貢献利用**［有償利用］

共同利用にて提供する計算資源

共同利用サービスの利用区分・カテゴリー別の利用課金表を下に示しました。TSUBAME 2.0における計算機資源の割振りは口数を単位としており、1口は標準1ノード（12CPUコア, 3GPU, 55.82GBメモリ搭載）の3000時間分（≒約4ヵ月）相当の計算機資源です。この計算機資源は、1000CPUコアを1日半とか、100GPUを3.75日といった利用も可能です。

利用区分	利用者	制度や利用規定等	カテゴリー	利用課金
学術利用	他大学または研究機関等	共同利用の利用規定に基づく	成果公開	1口：100,000円
産業利用	民間企業を中心としたグループ	「先端研究施設共用促進事業」に基づく	成果公開	トライアルユース（無償利用）
				1口：100,000円
			成果非公開	1口：400,000円
社会貢献利用	非営利団体、公共団体等	共同利用の利用規定に基づく	成果公開	1口：100,000円
			成果非公開	1口：400,000円

産業利用トライアルユース制度（先端研究施設共用促進事業）

共同利用サービスの「産業利用」は、東京工業大学学術国際情報センターが実施する文部科学省先端研究施設共用促進補助事業を兼ねております。その中のトライアルユース制度では、初めてTSUBAMEを利用する民間企業の方に限り、無償での利用（1利用期間は最長1年間、2回まで）が可能です。この制度でスパコンTSUBAMEの敷居を下げることで、より多くの方にスパコンの魅力を体験していただいております。

お問い合わせ

●東京工業大学　学術国際情報センター　共同利用推進室
●e-mail　tsubame@gsic.titech.ac.jp　Tel. 03-5734-2085　Fax. 03-5734-3198

詳しくは http://www.gsic.titech.ac.jp/tsubame/をご覧ください。

GSIC
Global Scientific Information and Computing Center