

並列処理技術の真の普及に向けて



研究室紹介

伊野文彦*

Towards Real Popularization of Parallel Processing Techniques

Key Words : High-performance computing, GPU Computing, biomedical simulation

はじめに

本稿では、大阪大学大学院情報科学研究科コンピュータサイエンス専攻並列処理工学講座ならびにその研究内容を紹介します。本講座は、2002年4月の大学院情報科学研究科の創設とともに、大学院基礎工学研究科情報数理系専攻からの改組・再編により設置されています。設置当初の教員は、萩原兼一教授（現・大阪大学名誉教授）、藤本典幸助教授（現・大阪府立大学教授）ならびに伊野文彦助手の3名でした。萩原教授の定年退職を受けて、2017年8月に筆者が着任し、現在は教授・伊野文彦および准教授・置田真生の2名で講座を運営しています。

改組・再編前の1994年7月の大学院基礎工学研究科での発足以来、本講座は一貫して並列処理に関わる研究、すなわち多数の計算機を同時に駆使することにより、規模の大きな計算の高速化を目指す研究を推進しています。以降では、その研究成果として、(1) Graphics Processing Unit (GPU) コンピューティングによる応用の高速化、(2) 遊休GPUを活用するためのシステムソフトウェアおよび(3) 方程式からの全自動シミュレーション技術を紹介します。

GPU コンピューティングによる高速化

昨今の深層学習技術や仮想通貨取引の普及とともに

に、GPUは汎用の計算加速器としての地位を確立したと言っても過言ではありません。学術的のみならず、商業的にもGPUは歴代の並列計算機としてもっとも成功したハードウェアと言えるでしょう。その加速の源泉は、数千個の計算コアだけでなく、それらにデータを迅速に供給できる高速なメモリにあります。例えば、NVIDIA社のTesla V100 GPUは、2.5次元積層技術を用いたHBM2メモリにより、900 GB/sもの高いメモリ帯域幅を提供していて、同等のCPUと比べて7倍もの帯域幅を実現しています。これらハードウェアの持つ高いピーク性能だけでなく、その性能を高い効率で引き出すための並列プログラム開発環境として、Compute Unified Device Architecture (CUDA) が提供されています。CUDAプログラムは、複数のGPUスレッドを各計算コア上で並列実行し、長いメモリ遅延を暗黙的に隠蔽することで、高い実効効率を達成します。

本講座では、GPUの持つ高い実効性能に着目し、その応用先をコンピュータグラフィックスの外へ広げることが目的として(図1)、2004年頃からLU分解や画像の位置合わせの高速化に取り組みました。開発した高速化手法は、対象の問題をGPU本来の3次元描画問題に帰着し、OpenGLなどのグラフィックス処理系を用いて実装されています。さらに、2006年におけるCUDAの発表以降、GPUによる加速が容易ではない問題として、整数演算を主とする生体医工学分野への応用 [1] あるいはGPUの不得意とする不規則性の強い問題の高速化 [2] に取り組んでいます。

さらに、産学連携に関してもGPUによる加速を特長とする製品開発に貢献しています。なかでも株式会社島津製作所との共同研究として、医用ならびに工業用Computed Tomography (CT) 撮像装置のための再構成処理の高速化に取り組んだことは、



* Fumihiko INO

1975年2月生まれ
大阪大学 大学院基礎工学研究科 博士
後期課程中退 (2002年)
現在、大阪大学大学院情報科学研究科
コンピュータサイエンス専攻 教授
博士 (情報科学)
TEL : 06-6879-4350
FAX : 06-6879-4354
E-mail : ino@ist.osaka-u.ac.jp

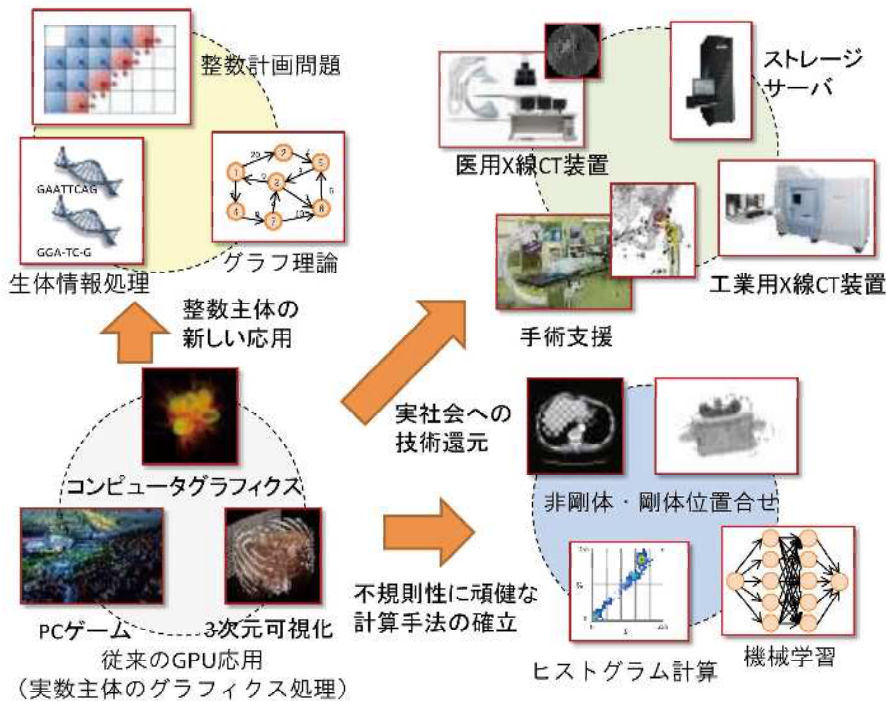


図1 GPUコンピューティングによる応用の高速化 (CT撮像装置の画像は株式会社 島津製作所の提供)

GPU技術の製品化に寄与できただけでなく、引用数の多い国際論文発表につながった点で、産学双方にとってよい事例になりました。また、半導体企業との共同研究では、GPUによる高速なシミュレーションが半導体の微細化を促進しただけでなく、共同研究に従事していた学生が連携先の企業へ就職できた点で、優秀な学生の育成とその社会への輩出を本務とする高等教育機関としてうまく機能できた事例になりました。

遊休GPUのためのシステムソフトウェア

GPUを装備する計算サーバは、今やAmazon EC2などの多くのデータセンタにおいて採用されています。しかし、GPUの大半は計算サーバというよりはデスクトップPCの一部として社会に流通しています。それらの遊休時間を遠隔から活用し、科学計算を高速化するためのサイクル共有システム [1] を開発しています (図2)。このシステムには2種類のユーザ、すなわちGPUを提供するホストならびにGPUを利用するゲストが存在します。ホストとゲストに対し、GPU上で公平なマルチタスク実行を実現する必要があります。

最新のGPUは、ハードウェアによるプリエンブション機能を提供していますが、科学計算の高速化

と同時に滑らかな描画を実現できるほど十分なものではありません。そこで、GPUの計算負荷に応じて、ゲスト側プログラムが計算資源をホスト側プログラムに譲る仕組みを開発しています。このような仕組みのもと、遺伝子配列の相同性検索や分散深層学習の高速化を、LAN環境で相互接続されたデスクトップPC群上で実現しています。

このシステムで特徴的なことは、数百ミリ秒程度の極めて短い遊休時間を活用している点です。遊休時間の長さは冪分布にしたがうことが経験的に知られていて、一瞬とはいえ、かなりの頻度で潜在的に発生する遊休時間を蓄積していけば、多数の計算コ

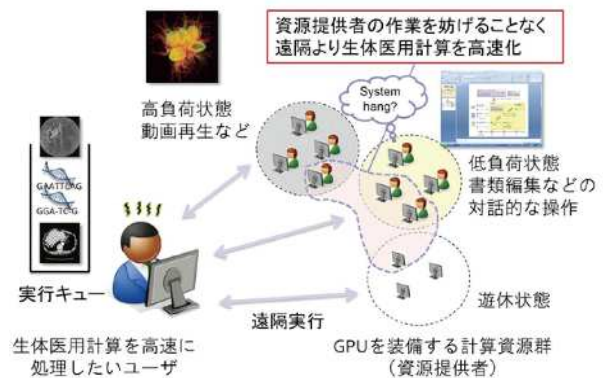


図2 遊休GPUを活用するサイクル共有システム

アを活用するシステムには大きな原動力となります。計算コアの数が増大していくメニーコア・プロセッサの時代において、一瞬の遊休サイクルを活用することの重要性を、具体的な応用とともに提示しています。

方程式からの全自動並列シミュレーション

心室筋細胞の膜電位ダイナミクスなど、生体機能の時間的な振る舞いを解析するためのシミュレータ Flint [3] を開発しています。Flint の特長は、生体機能の原理を数学的に記述する常微分方程式や関数のみを入力として、GPU、メニーコア CPU およびベクトル計算機といった多様な並列計算機向けの並列プログラムを自動生成できる点です。また、常微分方程式の数値解法としてオイラー法やルンゲ・クッタ法などを指定できます。生体機能の解明に関心を持つ研究者は、一切のプログラムを記述することなく、数学的な記述からシミュレーション結果を迅速に得ることができ、その分析に専念できます。

Flint のシミュレート対象は、任意の常微分方程式で表される系、すなわち非線形システムです。例えば、図3に示す視神経ネットワークの振る舞いをシミュレートすることで、大脳視覚野の動態を分析し (図4)、生体が持つ視覚情報処理機能の解明に

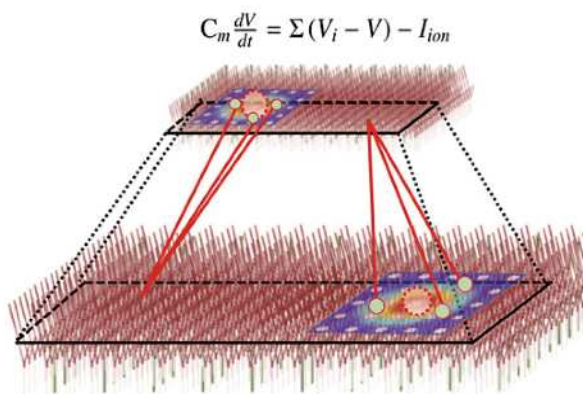


図3 視神経ネットワークモデル

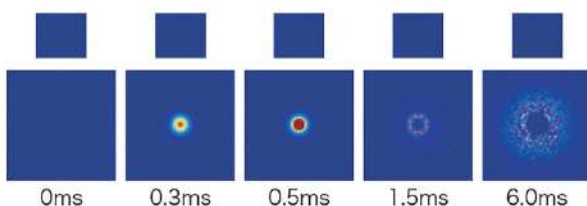


図4 視神経モデルの膜電位シミュレーション

貢献します。このネットワークモデルは数千万個におよぶ連立方程式で構成されており、生体医用工学における研究を加速するためにはシミュレーションの高速化が不可欠です。非線形システムを対象とする Flint は、生体医用工学だけでなくロボット工学などの多様な分野へ応用できます。

おわりに

1980年代に貿易摩擦の問題に巻き込まれた日本の高性能ベクトル計算機は、社会的に大きな注目を集めました。しかし、当時のベクトル計算機は極めて高価であり、誰しもが自由に利用できるものではありませんでした。それから40年が経過し、ベクトル計算機の子孫ともみなせる GPU は、極めて強力な計算性能を低コストで実現し、幅広い分野に浸透しています。ただし、高速化のための並列プログラミングの敷居は依然として高く、現状ではそれぞれの分野において必要とされる関数群を、一部の有能な開発者が実装し、ライブラリとして提供しているのが典型的な利用形態です。この形態は、応用先のアルゴリズムが枯れて固定されている場合にはうまくいき、行列計算のための cuBLAS や深層学習のための cuDNN がドメイン特化ライブラリとして著名です。しかし、並列化対象の逐次アルゴリズムが発展途上にある場合、逐次アルゴリズムの開発者自身が並列化を施す必要があります。そのような観点では、並列処理技術は十分に普及したとは言えません。並列処理技術の真の普及に向けて、産学連携を含め、技術的な貢献を続けて参ります。

また、社会への人材輩出を司る高等教育機関としては、講座内部の国際化を推し進め、国際的な素養を持つ優秀な人材を育成して参ります。計算量あるいはデータ量が多くて実用化を妨げている、あるいは長い計算時間が研究の迅速な展開を妨げているなど、お困りのことがございましたら、ぜひお声がけ頂ければ幸いです。

参考文献

- [1] F. Ino, Y. Munekawa, and K. Hagihara, "Sequence Homology Search Using Fine Grained Cycle Sharing of Idle GPUs," IEEE Trans. Parallel and Distributed Systems, 23(4):751–759, April 2012.

- [2] Y. Mitani, F. Ino, and K. Hagihara, “Parallelizing Exact and Approximate String Matching via Inclusive Scan on a GPU,” *IEEE Trans. Parallel and Distributed Systems*, 28(7):1989–2002, July 2017.
- [3] T. Okuyama, M. Okita, and et al., “Accelerating ODE-based Simulation of General and Heterogeneous Biophysical Models using a GPU,” *IEEE Trans. Parallel and Distributed Systems*, 25(8):1966–1975, August 2014.

