

科学研究費助成事業 研究成果報告書

平成 27 年 6 月 4 日現在

機関番号：32612

研究種目：基盤研究(B)

研究期間：2012～2014

課題番号：24300112

研究課題名(和文)超高速空間モデルシミュレータの実装

研究課題名(英文)Design and Implementation of high-performance spatial model simulator

研究代表者

舟橋 啓 (Funahashi, Akira)

慶應義塾大学・理工学部・准教授

研究者番号：70324548

交付決定額(研究期間全体)：(直接経費) 13,800,000円

研究成果の概要(和文)：細胞・組織レベルにおける空間モデルシミュレーションは数時間から数日と計算時間がかかり、高速化の必要性が高まっていた。本研究課題では、GPU(Graphics Processing Unit)を利用することで、細胞・組織レベルにおける空間モデルシミュレーションを高速化するソフトウェア基盤を構築した。空間モデルシミュレーションは偏微分方程式ベースの拡散方程式、移流方程式、反応拡散方程式の3種類を対象とし、高速化を行った。本研究課題の実装により、移流方程式では52倍、反応方程式では64倍、拡散方程式では63倍の高速化を達成した。

研究成果の概要(英文)：On a spatial model simulation based on PDEs, the simulation space is discretized by grid and simulators will compute on each grid sequentially which will increase the simulation time enormously depending on the number of grids. In order to solve this problem, we have applied parallelization of this sequential numerical integration on GPGPU (General Purpose computing on GPU). In this research, we parallelized a CPU-based SBML spatial model simulator by GPGPU. We implemented numerical integration of advection, reaction and diffusion equation with NVIDIA CUDA. For the evaluation of the CPU application, we used Intel Xeon X5687 and for the evaluation of GPU case, we used Tesla K40. As a result, we achieved 52x performance improvement in advection equation, 64x in reaction equation, 63x in diffusion equation for 512x512 grids.

研究分野：システム生物学

キーワード：情報工学 並列処理 GPGPU 空間モデル 偏微分方程式 数値積分 SBML

1. 研究開始当初の背景

近年、測定技術の発展により、細胞内における分子局在や濃度勾配などの定量的な解析が可能となった。これに伴い、分子の空間的な偏りが細胞のダイナミクスに与える影響の解析が盛んに行われている。このような生命システムのダイナミクスを理解する手法の一つとして、空間情報を考慮した数理モデルによる空間モデルシミュレーションが挙げられる。空間モデルシミュレーションでは偏微分方程式モデルを扱う際、シミュレーション空間を格子状に離散化し、各格子で数値計算を行う。その際、個々の格子について逐次的に計算を行うため、計算時間が膨大になるという問題点が存在する。

この問題を解決するためには計算機による演算処理高速化の必要があり、その手法として PC クラスタをはじめとする並列計算機はスーパーコンピュータや専用ハードウェアに比べ低コストで高い処理能力を得られることから広く用いられている。しかし、PC クラスタはトータルとしての発熱量が多く、空調が整った専用のサーバ・ルーム内に設置する必要があり、省エネルギー、省スペース性の面において問題がある。また、PC クラスタ全体の性能向上が必要となった場合、構成要素である PC をすべて入れ替える必要があり、性能向上を達成するために必要となる人的、金銭的コストが高い。さらに、PC クラスタの運用は計算機に関する高度の知識を必要とし、誰にでも容易に取り扱えるものではない。また、構成する部品点数も多いことから維持・管理に要するコストが高い、等の問題点がある。

一方で、PC 上で描画を担当する GPU の性能向上はめざましく、GPGPU(General Purpose computation on GPU)と呼ばれる科学技術計算を高速な GPU 上で行う技術が脚光を浴びており、バイオイメージング、分子動力学計算、信号処理といった分野において GPU を科学計算分野へ利用しようとする動きが急速に広がりつつある。GPGPU は PC クラスタなど他の並列計算の手法よりも価格性能比、電力性能比などの面において優れているため、個人レベル、研究室レベルでの高速なシミュレーション環境の構築が容易である。

2. 研究の目的

本研究課題では、GPGPU による低コスト且つ超高速空間モデルシミュレーション環境の構築を目指した。具体的には、CPU 上で逐次的に数値計算を行う空間モデルシミュレータを GPGPU によって並列化し、高速化を試みた。

3. 研究の方法

(1) 偏微分方程式ソルバの理解と GPU による並列化の検討

偏微分方程式ソルバ (シミュレーションエンジン) の理解を進め、並列化による高速化の可能性について十分に検討を行った。図 1 に細胞・組織レベルの空間モデルシミュレーションで対象となる空間モデルの概念図、及び主要構成要素を示す (図 1 は細胞内での空間モデルの例を示す)。

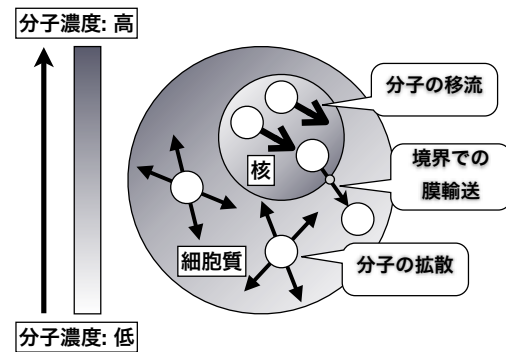


図 1: 空間モデルの概念図

空間モデルには

- ・反応区画の形状・位置関係
- ・分子の空間分布
- ・分子の移流・拡散・境界での輸送
- ・分子間の反応に関する情報

が含まれており、各区画には区画内に存在する各分子のダイナミクスが偏微分方程式、及び常微分方程式で記述される。空間モデルシミュレータは上記空間モデルより必要な情報を抽出し、分子濃度の時空間発展に関する数値積分を行う必要がある。そのため、偏微分方程式で記述された空間モデルの数値解析を行う際に必要となるのが

1. 空間の離散化

2. 時間積分

の 2 点である。本研究課題では上記 2 点のアルゴリズムについて GPU を用いた高速化を行った。検討対象とするのは空間の離散化に関しては有限差分法と有限体積法を、時間積分に関してはオイラー法、ルンゲクッタ法の数値積分アルゴリズムとした。

(2) 数値積分の並列化

数値積分の並列化には

- (a) 粗粒度並列化
- (b) 細粒度並列化

の 2 通りのアプローチが考えられる。(a) 粗粒度並列化は一般的な PC クラスタ上で実装されている並列化アルゴリズムであり、各プロセッサ (CPU) に異なる演算を行わせる方式である。数値積分の場合、複数回実行する必要があるシミュレーションを各 CPU が分担して実行する方式が一般的である。粗粒度並列処理は文字通り並列化の粒度が粗いため、並列化が容易である半面、パフォーマンス向上に限界がある。一方、(b) 細粒度並列化は一つのタスクを細かく分割し、各プロセッサが分割された小問題を担当する方式である。

細粒度並列処理の簡単な例として、図 2 に CPU, GPU の演算ユニットのイメージ図を示す。GPU は多数の演算ユニット(図中では人形で示されている)を持ち、各演算ユニットが個別の演算を実行できる。

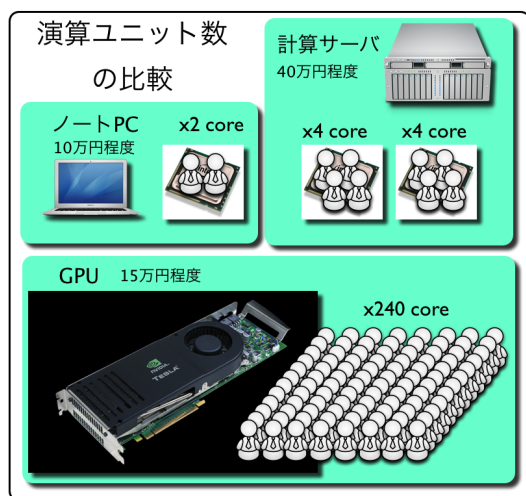


図 2: CPU, GPU の演算ユニット数の比較。GPU は多数の演算ユニットを持つ。

粗粒度並列処理では、各演算ユニットが独立の演算を行う一方、細粒度並列処理では一つの大きな問題(シミュレーション)を細かいパーツに分割し、各演算ユニットは個々のパーツの演算を行う。細粒度並列処理では一つの大きな問題を複数の演算ユニットが協力して解くことが可能であるため、本研究課題に最適なアプローチである。細粒度並列処理は高度な並列アルゴリズムが必要となるが、大幅な性能向上を達成する可能性を秘める。本研究課題では高速シミュレーション処理を GPU 上で達成するため、粗粒度並列化と細粒度並列化を組み合わせたハイブリッドアプローチを検討した。オイラー法、ルンゲクッタ法等、各数値積分アルゴリズムによって並列化のアプローチは異なるため、個別の細粒度並列化アルゴリズムを構築した。なお、シミュレータは NVIDIA 社が提供する C 言語による統合開発環境である CUDA (Compute Unified Device Architecture) を用いて実装を行った。

(3) 標準モデル記述言語への対応

本研究課題で開発した空間モデルシミュレータが広く普及するためには空間モデルの構築が容易であること、及び他の空間モデルシミュレータで作成した空間モデルを読み込むことが可能であることが非常に重要な要素である。特に後者に対応することはソフトウェアの普及を左右する要因であるため、本研究課題では標準モデル記述言語である SBML (Systems Biology Markup Language) への対応を行うことが必須であると考えた。申請者は SBML の空間モデル拡張を行なっ

り、既に CPU レベルのシミュレータにて SBML の空間モデルの読み込みを達成している。本研究課題ではこの成果を活用し、空間モデルシミュレータの SBML 対応を行った。SBML では空間モデルの記述に 3 種類の方法が提案されており、本研究課題では空間モデルシミュレーションで広く利用されている Analytic Volume (数式により表現されたシミュレーション空間)と Sampled Volume (顕微鏡画像から構築されたシミュレーション空間)の対応を行った。Sampled Volume の空間モデル構築のため、細胞の 3 次元構造を蛍光顕微鏡画像にて取得する実験系を構築した。図 3 に蛍光顕微鏡画像から構築した細胞の 3 次元空間モデルを示す。

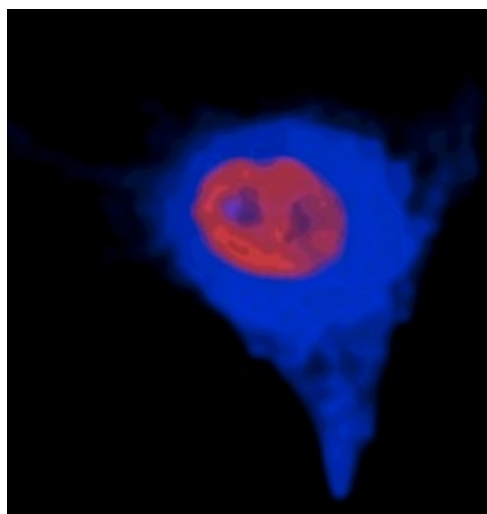


図 3: 3 次元画像再構成により構築した空間モデル

(4) 効率的なメモリアクセスアルゴリズムの構築

CUDA で GPU のパフォーマンスを十分に引き出すには、GPU コンピューティングに適したプログラミングが不可欠である。例えば、CUDA のメモリ階層とその特性を十分に理解した上でアルゴリズムの構築を進める必要がある。図 4 は GPU の構造と内部メモリの模式図である。

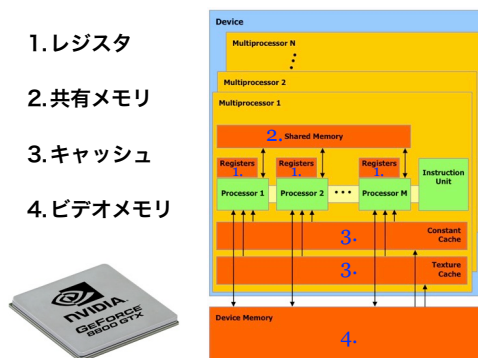


図 4: GPU の構造と内部メモリ

GPU は内部的に並列計算機と同等の機能を有しており、並列に演算を実行することが可能である (図 4 Processor 部分)。内部メモリはアクセス速度の順に (a)レジスタ、(b)共有メモリ、(c)キャッシュ、(d)ビデオメモリと分かれており、同様の順番でメモリサイズが大きくなる。シミュレーション対象となるモデルが巨大になることもあるため、モデル内のすべての演算対象を一番サイズが大きい (d) ビデオメモリに置くことが安全だと考えられるが、その場合アクセス速度が遅いため全体のパフォーマンス向上が望めない。そのため、極めて高速に動作する (a)レジスタ、(b)共有メモリを有効活用することを検討した。具体的には、シミュレーションを行う前に対象となるモデルを分割し、パラメータを (a) のレジスタに、必要となる生化学反応方程式、分子濃度を (b) の共有メモリに配置した。

4. 研究成果

(1) 拡散方程式の高速化

図 5 に拡散方程式のシミュレーションを CPU (Intel Xeon X5687, 3.60 GHz)、GPU (NVIDIA Tesla K40) 上で計算した実行時間とその内訳を示す。グラフ中の領域は左から順に「拡散方程式の数値積分」「境界条件の計算」「計算結果の更新」「メモリ確保」に費やした時間を示す。CPU 実装では拡散反応の数値積分に大部分の計算時間を費やしていたが、GPU 実装では効率的に並列化を行い、計算時間の削減が行われていることがわかる。

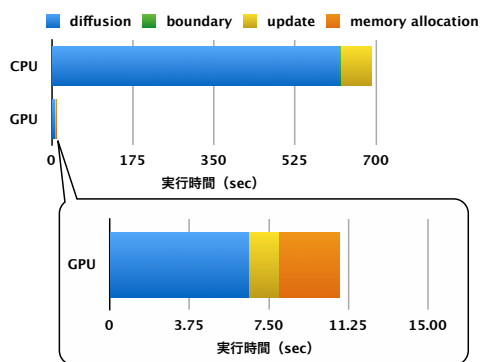


図 5: CPU/GPU での実行時間とその内訳

当研究課題の実装により、拡散方程式のシミュレーションを最大で 63 倍高速化することに成功した。

(2) 移流方程式の高速化

図 6 に移流方程式のシミュレーションを CPU (Intel Xeon X5687, 3.60 GHz)、GPU (NVIDIA Tesla K40) 上で計算した実行時間とその内訳を示す。移流方程式の GPU 実装では、実行時間の大部分をメモリ確保に占められているが、これは GPU を使用開始する際に必要となるオーバーヘッドの 2 秒間であり、むしろ移流方程式の数値積分、及び値の更新に関しての並列化が効果的に行えていることを表している。

当研究課題の実装により、移流方程式のシミュレーションを最大で 52 倍高速化することに成功した。

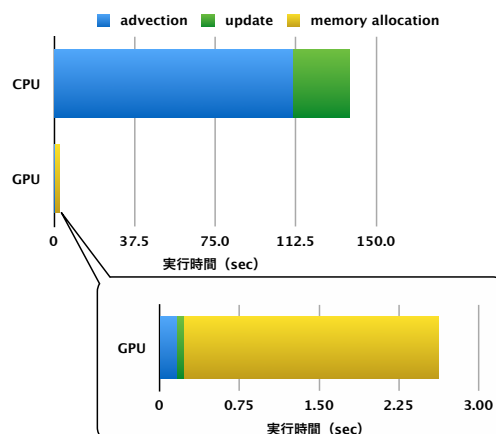


図 6: CPU/GPU での実行時間とその内訳

(3) 反応方程式の高速化

図 7 に反応方程式のシミュレーションを CPU (Intel Xeon X5687, 3.60 GHz)、GPU (NVIDIA Tesla K40) 上で計算した実行時間とその内訳を示す。

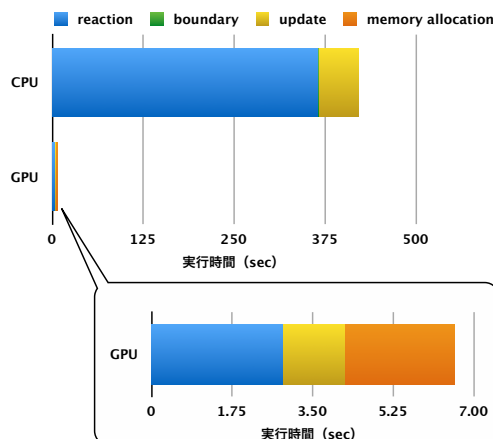


図 7: CPU/GPU での実行時間とその内訳

拡散方程式、移流方程式と同様、高速化の効果が得られた。当研究課題の実装により、反応方程式のシミュレーションを最大で 64 倍高速化することに成功した。

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文] (計 3 件)

1. K Sumiyoshi, H Hirata, N Hiroi, A Funahashi, “Acceleration of discrete stochastic biochemical simulation using GPGPU”, *Frontiers in Physiology*, 査読有, 6, 2015.
doi: 10.3389/fphys.2015.00042
2. Y Nakai, M Ozeki, T Hiraiwa, R Tanimoto, A Funahashi, N Hiroi, et al.
“High-speed microscopy with an electrically tunable lens to image the

dynamics of in vivo molecular complexes” , Review of Scientific Instruments, 査読有, 86, 2015.
doi: 10.1063/1.4905330

3. R Keller, A Dorr, A Tabira, A Funahashi, MJ Ziller, R Adams, N Rodriguez, NL Novere, N Hiroi, H Planatscher, A Zell, A Drager, “The systems biology simulation core algorithm” , BMC Systems Biology, 査読有, 7, 2013
doi: 10.1186/1752-0509-7-55

[学会発表] (計 16 件)

1. A Funahashi, “Computational Platform for Systems Biology” , Dagstuhl Seminar on Multiscale Spatial Computational Systems Biology, 23rd-28th Dec. 2014, Wadern (Germany)
2. Y Nakai, N Tamura, T Hiraiwa, T Okuhara, VM Draviam, A Funahashi, N Hiroi, “Development of High-speed 3D imaging system with electrically tunable lenses for deeper probing of subcellular structures” , International Workshop on Quantitative Biology 2013, 25th Nov. 2013, Icho-kaikan(Osaka, Suita).
3. T Matsui, N Hiroi, A Funahashi, “Implementation of a spatial model simulator and its SBML support” , HARMONY 2012, 21st-25th May 2012, Maastricht (The Netherlands)

[図書] (計 0 件)

[その他]

ホームページ等

<http://fun.bio.keio.ac.jp/>

6. 研究組織

(1) 研究代表者

舟橋 啓 (FUNAHASHI, Akira)
慶應義塾大学・理工学部・生命情報学科・
准教授
研究者番号 : 70324548

(2) 連携研究者

広井 賀子 (HIROI, Noriko)
慶應義塾大学・理工学部・生命情報学科・
専任講師
研究者番号 : 20548408