科学研究費助成事業　　研究成果報告書

平成 ２６ 年　 ６ 月　 ９ 日現在

機関番号：１４４０１
研究種目：基盤研究(B)
研究期間：2011～2013
課題番号：２３３００００７

研究課題名（和文）相互依存関係を持つ異種タスクの同時処理に関するＧＰＧＰＵによる高速化の研究

研究課題名（英文）A study on GPGPU acceleration of simultaneous processing heterogeneous tasks with mutual dependence relation

研究代表者
萩原　兼一（Hagihara, Kenichi）

大阪大学・情報科学研究科・教授

研究者番号：００１３３１４０

交付決定額（研究期間全体）：（直接経費）　15,800,000円、（間接経費）　4,740,000円

研究成果の概要（和文）：汎用的な生物物理モデルは数式と常微分方程式の集合である．変数はイオン濃度などの物理量を表す．生物物理シミュレーションは変数の時間変移を計算する．Flintは，大規模で多種多様な生物物理モデルのシミュレータである．様々なモデルを処理するための数式の中間コード表現を用いている．本研究ではGPUを用いてFlintを高速化する二つの方法を提案する．第１の方法はインタプリタ方式であり，中間コードを自動的に並列化する．第２の方法は中間コードからソースコードに変換する．ソースコード量が少なくなり，そのコードのコンパイル時間が短くなる．シミュレーションコード実行性能は，後者の前者の2.4倍である．

研究成果の概要（英文）：A general biophysical model here is a set of mathematical expressions (functions) and ordinary differential equations with variables that represent ion concentration in cells, electrical current of neurons and so on. The biophysical simulation calculates time variation of these variables. Flint is a simulator that numerically integrates heterogeneous biophysical models.
We propose two acceleration methods for Flint using a graphics processing unit (GPU). The first method interprets multiple bytecodes in parallel on the GPU. It automatically parallelizes the simulation using a level scheduling algorithm. The second method translates a model into a source code through the internal bytecode, which speeds up the compilation of the generated source codes, because the code size is diminished because of bytecode unification. The latter achieves a performance of 2.4 times higher than that of the former.

研究分野：総合領域

科研費の分科・細目：情報学・ソフトウェア

キーワード：GPUコンピューティング　汎用生体シミュレーション　PHML　無閉路有向グラフ　計算依存グラフ　高速化

１．研究開始当初の背景

GPGPU(General Purpose computations on GPU)とは，コンピュータの画面描画部品である **GPU**(Graphics Processing Unit)の高い演算能力を，画面描画ではなく数値計算等の汎用(General Purpose)処理に適用するプログラミング技法であり，2002 年ごろから研究が始まった．当時の GPU は，高速に画面を描画するために百台程度の要素プロセッサ SP を並列に動作させている．当初，GPGPU は，GPU の画面描画の副作用を利用した計算であったのでGPGPUと呼ばれていたが，最近は GPU そのものが描画部品ではなく並列処理に特化したプロセッサとして設計され，約 500 台の SP をもち，中には描画能力のないものもある．その意味では GPU とは言えなくなったものもある．そして 2006 年後半に **CUDA**(Compute Unified Device Architecture)と呼ばれる GPU プログラムの統合開発環境も提供されてからは **GPU コンピューティング**とも呼ばれるようになった．

GPGPU は，応用問題に内在する計算構造と GPU のアーキテクチャとの適合度により，その高速化の効果は異なる．データ並列なタスクのように，うまく適合する場合は，GPU １台で CPU 数十台に匹敵する能力を発揮する．したがって，GPGPU は，高性能計算（**HPC**: High Performance Computing)分野で有効な技術として定着し，現在，次に示すように世界中で競争中である．HPC に関する世界で最も有名な会議 **SC**（http://sc14.supercomputing.org/）でも，2005 年から GPGPU に関する発表が多く，多数の参加者を集めて熱心に討論されている．また，2009 年から SC の GPU 版とも言える **GTC**(GPU Technology Conference; http://www.nvidia.com/object/gpu_technology_ conference.html)が始まり，非常に多くの研究が発表されている．さらに，国内では東京工業大学が中心になりGPGPU研究会が開催され，GPU に関する研究発表や講習会が開催されている．

申請当時，世界中で競われている研究は，GPU アーキテクチャに適合する応用課題選びと，その GPU プログラム（カーネルと呼ばれる）の開発・高速化がほとんどである．カーネルは応用課題個別に設計・実装する必要があり，任意に計算式が与えられたときに，それを自動的に GPU で計算できる状況ではない．このような昨今の潮流から，GPGPU の適用範囲は限定的なものとなっている．本研究代表者は，HPC 分野における GPGPU の効果をさらに高めるためには，2 に示すような一見GPGPUには向かない応用課題への展開に取り組むことが不可欠であると考えている．

２．研究の目的

効率のよい GPU プログラムは，次の項目 A)～C)を満たすように実装することが主流である．

A)応用課題の処理を，単一タスクの **SIMT**（Single Instruction Multiple Threads）型として構成する．SIMT とは，プログラムカウンタを共有するスレッド（プログラム）の集合が共通にアクセス可能なメモリ空間を用いて並列実行する形態のプログラムである．このタスクを，GPU のカーネルプログラムとして実装する．GPU では，多数のデータ各々に専用のスレッドを設定し，それらの多数のスレッドが同時にデータ処理している．

B)GPU は，要素プロセッサ SP が２階層の共通にアクセスできるメモリ（共有メモリ SM とデバイスメモリ DM）もつ．メモリアクセスに関して，SM はバンクコンフリクトが起きないように，DM はコアレス参照と呼ばれるある条件を満たすように，カーネルを設計する．そのようにすれば，多数の SP がメモリに同時アクセスでき．並列性の高い GPU プログラムとなり，実行効率がよい．

C)ホストプロセッサに当たる CPU と GPU 間のデータ転送能力が低いので，CPU と GPU 間のデータ送受量は，必要最小限とする．

項目 A)を満たす中でさらに C)を満たす応用課題をうまく選び，B)の工夫を凝らしたカーネルを作成することが世界中で競われている研究内容である．この研究は当然続けるべきであり，それによって GPGPU の応用範囲を広げることができる．しかし，研究代表者は，GPGPU の応用範囲を格段に広げるには次の研究を推進すべきと考える．

本基盤研究では，以上の背景を踏まえ，次の 1)および 2)の「一見 GPGPU には向かない応用課題」を対象とし，効率のよい GPGPU 解法を探求するという，挑戦的課題を研究目的とする．

1)概念的に，処理するタスクが単一ではなく，異種多数のタスクを同時に処理する．

2)既存研究のように，特定の処理タスクに対するカーネルを設計・実装するのではなく，コンパイラのようにユーザが任意に定義した多数のタスク（計算式）を GPU で実行できるようにする．

具体的には，先ず次世代スーパーコンピュータの重要課題でもある，生体機能シミュレーションを具体的例とし，その実行時間を CPU で実行するよりも数倍～10 倍程度高速化する GPGPU 処理を具体目標として研究する．生体機能シミュレーションでは，同時に働いている複数臓器の機能をシミュレーションする必要がある．それらは異種類かつ多数のタスクを同時に処理することになり，本研究のよい題材である．任意に出現する各方程式を解くカーネルを，その都度開発することは事実上不可能であり，これを効率よく GPGPU 化する方法を研究する．本基盤研究は，研究代表者らが得た GPGPU に関する多

くの知見と生体シミュレーションの研究を融合して発展させるものである．

３．研究の方法

Flint は入力の PHML で記述したモデルからシミュ レーションコードを生成する．記述言語 PHML は代数関数，常微分 方程式（ODE：Ordinary Diferential Equation）および遅延微分方程式(DDE:Delay Diferential Equation)により生体機能を表現する．Flint はこれらの微分方程式の初期値問題を，オイラー法またはルンゲ=クッタ法を用いて 時間発展問題として解く．以降では，代数関数，ODE および DDE をまとめて数式と記述する．

入力された PHML モデルから数式および変数の情報を抽出し，数式を頂点，数式間の依存関係を有向辺とする依存グラフを作成する（図１(a)）．次に，依存グラフをレベルスケジューリングアルゴリズムにより数式計算の全体集合をレベルにより分類する．レベル k の計算はレベル k-1 の計算結果に依存するので，レベルの順に数式計算を実行できる．また，同一レベルの数式計算の集合は依存関係がないので並列計算可能となる（図１(b)）．各レベルの計算をフェーズと呼ぶ．計算の少ないフェーズは，その前後どちらかのフェーズにマージする（図１(c)）．次に似た計算をまとめる（図２）．さらに，divergent branch を回避するために冗長なスレッドを追加する（図３）．

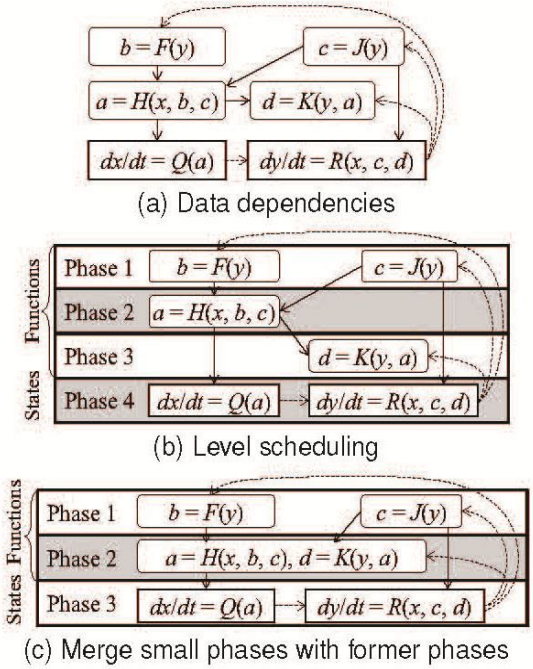

図１．データ依存（実線）にもとずく計算式の評価順を決定.

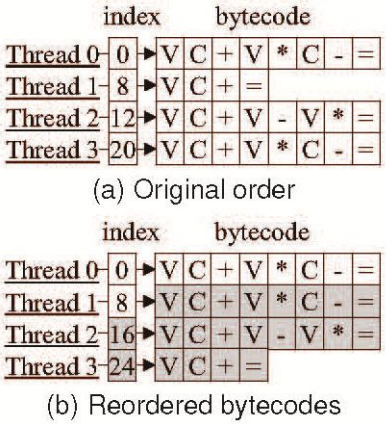

図２．中間コードの類似性による分類.

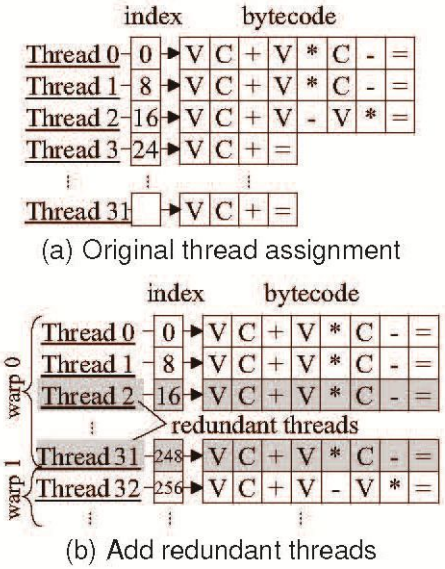

図３．Divergent branch 削減のために冗長スレッドの追加.

４．研究成果

表１にインタプリタ(IS)方式とトランスレータ(TS)方式による生体モデルの計算時間をまとめる．計測に用いた CPU は Intel Core i7 930（2.8GHz 4 コア）で主記憶容量は 12GB である．GPU は NVIDIA Tesla C2070 である．CPU-1 は CPU の１コアでの実行を，CPU-8 は CPU の同時マルチスレッディング（4 コアで 8 スレッド）での実行を表す．

Luo-Rudy は心細胞を記述した PHML モデル，LR-Ring は Luo-Rudy 細胞を 80 個リング状に接続したモデル，LR-x は Luo-Rudy 細胞を x 個直線状に結合したモデルである．Wang および Rybak は異なる網結合の神経モデルを表す．Mix は異種なモデルを表すために，Wang，Rybak，LR-100 を単純に接続した仮想的なモデルである．

すべての計測は，1000 ミリ秒の期間（1 ステップが 0.01 ミリ秒で 10 万ステップ）のシ

ミュレーション時間を測った．

数千の数式からなる大規模なモデルでは，IS 方式は 1 台の CPU による計算より 2.3〜37 倍高速である．一方，39 個の式からなる小規模な Luo-Rudy は GPU の方が 1/7 の遅さである．

TS 方式は，比較的規模の大きい Rybak，Mix，LR-1000〜LR-5000 のモデルに関しては IS より高速であるが，他のモデルでは 2〜3 秒遅い．これは TS 方式の場合，ソースコードの生成およびコンパイルの時間を含むからである．

表１．各種モデルの計算時間

Model	Interpreter		Translator		
	CPU-1	GPU	CPU-1	CPU-8	GPU
Luo-Rudy	1.03	7.04	0.169	0.596	6.88
Wang	40.0	6.00	2.73	1.77	7.02
Rybak	44.1	18.0	2.50	2.39	7.91
LR-Ring	71.9	8.10	4.86	3.31	8.14
LR-100	90.4	7.45	6.08	3.56	7.57
Mix	186	20.4	11.3	6.76	9.58
LR-1000	1570	18.2	64.2	19.5	11.4
LR-2000	3390	33.8	134	37.5	20.0
LR-3000	5250	50.2	204	55.8	26.4
LR-4000	7120	66.8	279	75.5	30.8
LR-5000	8990	83.6	355	95.0	35.4

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 11 件）

① Tomohiro Okuyama, Masao Okita, Takeshi Abe, Yoshiyuki Asai, Hiroaki Kitano, Taishin Nomura, and Kenichi Hagihara. Accelerating ODE-based Simulation of General and Heterogeneous Biophysical Models using a GPU. IEEE Transactions on Parallel and Distributed Systems, 査読有, 201, 印刷中.
DOI: 10.1109/TPDS.2013.198

② Fumihiko Ino, Yosuke Oka, Kenichi Hagihara, A Fine Grained Cycle Sharing System with Cooperative Multitasking on GPUs, International Journal of Networking and Computing, 査読有, 2014, 印刷中

③ Yoshiyuki Asai, Takeshi Abe, Hideki Oka, Masao Okita, Kenichi Hagihara, Samik Ghosh, Yukiko Matsuoka, Yoshihisa Kurachi, Taishin Nomura, and Hiroaki Kitano. A Versatile Platform for Multilevel Modeling of Physiological Systems: SBML-PHML Hybrid Modeling and Simulation. Advanced Biomedical Engineering, Vol. 3, pp. 50-58, 査読有, 2014.
DOI:http://dx.doi.org/10.14326/abe.3.50

④ Kei Ikeda, Fumihiko Ino, Kenichi Hagihara, Efficient Acceleration of Mutual Information Computation for Nonrigid Registration Using CUDA. IEEE Journal of Biomedical and Health Informatics, 査読有, Vol.18, No.3, 2014, pp.956-968
DOI:10.1109/JBHI.2014.2310745

⑤ Yuki Sugimoto, Fumihiko Ino, Kenichi Hagihara, Improving Cache Locality for GPU-based Volume Rendering, Parallel Computing, 査読有, Vol.40, No.5/6, 2014, pp.59–69
DOI:10.1016/j.parco.2014.03.013

⑥ Fumihiko Ino, Kentaro Shigeoka, Tomohiro Okuyama, Masaya Motokubota, Kenichi Hagihara, A Parallel Scheme for Accelerating Parameter Sweep Applications on a GPU, Concurrency and Computation: Practice and Experience, 査読有, Vol.26, No.2, 2014, pp.516–531
DOI:10.1002/cpe.3016

⑦ Fumihiko Ino, Shinta Nakagawa, Kenichi Hagihara, GPU-Chariot: A Programming Framework for Stream Applications Running on Multi-GPU Systems, IEICE Transactions on Information and Systems, 査読有, Vol.E96-D, No.12, 2013, pp. 2604-2616
DOI: 10.1587/transinf.E96.D.2604

⑧ 伊野文彦, 萩原兼一, GPU アクセラレータとその研究動向, Medical Imaging Technology, 査読無, Vol.31, No.3, 2013, pp.147–152
DOI:10.11409/mit.31.147

⑨ Fumihiko Ino, Yuma Munekawa, Kenichi Hagihara, Sequence Homology Search Using Fine Grained Cycle Sharing of Idle GPUs, IEEE Trans. Parallel and Distributed Systems, 査読有, Vol.23, No.4, 2012, pp.751–759
DOI:10.1109/TPDS.2011.239

⑩ Tomohiro Okuyama, Fumihiko Ino, Kenichi Hagihara, A Task Parallel Algorithm for Finding All-Pairs Shortest Paths Using the GPU, International Journal of High Performance Computing and Networking, 査読有, Vol.7, No.2, 2012, pp.87-98
DOI:10.1504/IJHPCN.2012.046384

⑪ Fumihiko Ino, Akihiro Ogita, Kentaro Oita, Kenichi Hagihara, Cooperative Multitasking for GPU-Accelerated Grid Systems, Concurrency and Computation: Practice and Experience, 査読有, Vol.24, No.1, 2012, pp.96–107
DOI:10.1002/cpe.1722

〔学会発表〕（計 13 件）

① 三谷康晃，伊野文彦，萩原兼一,GPUにおいて動的グラフを高速処理するためのフレームワークの検討，第 14 回ハイパフォーマンスコンピューティングと計算科学シンポジウム，2014/1/7-8，一橋大学一橋講堂（東京都）

② Fumihiko Ino, The Past, Present, and Future of GPU-Accelerated Grid Computing, 1st International Symposium on Computing and Networking (CANDAR 2013), 2013/12/4–6, Matsuyama, Japan

③ 重岡謙太朗，伊野文彦，萩原兼一，GPUを用いた分枝限定法におけるメモリ参照効率を高めるための配列パッキング手法，情報処理学会ハイパフォーマンスコンピューティング研究会，2013/9/30–1，沖縄産業支援センター（沖縄市）

④ Yoshiyuki Asai, Takeshi Abe, Hideki Oka, Masao Okita, Tomohiro Okuyama, Kenichi Hagihara, Yoshihisa Kurachi, and Hiroaki Kitano. PhysioDesigner and Flint: a platform for multilevel modeling of physiological systems and simulation.the 5th International Conference on Computational Bioengineering (ICCB 2013), Leuven, Belgium, 2013/9/11. 1 pages.

⑤ 南翔太，伊野文彦，萩原兼一，GPU サイクル共有システムのための遊休時間予測手法の比較．第 12 回情報科学技術フォーラム，2013/9/4–6，鳥取大学（鳥取市）

⑥ Yoshiyuki Asai, Takeshi Abe, Hideki Oka, Masao Okita, Tomohiro Okuyama, Kenichi Hagihara, Samik Ghosh, Yukiko Matsuoka, and Hiroaki Kitano. A versatile platform for multilevel modeling of physiological systems: Template/instance framework for large-scale modeling and simulation. 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'13), pp.5529-5532, 2013/7/6, Osaka, Japan,

⑦ Tomohiro Okuyama, Masao Okita, Takeshi Abe, Yoshiyuki Asai, Taishin Nomura, Hiroaki Kitano, and Kenichi Hagihara. Accelerating General and Heterogeneous Biophysical Simulations Using the GPU. the 4th GPU Technology Conference (GTC 2013), 2013/3/18–21, San Jose, CA, USA

⑧ Fumihiko Ino, Kenichi Hagihara, Fine-Grained Cycle Sharing of Idle GPUs for Homology Search, 4th GPU Technology Conference (GTC 2013), 2013/3/18–21, San Jose, CA, USA

⑨ 岡陽介，伊野文彦，萩原兼一，協調マルチタスキングを用いて短い遊休時間を活用する GPU グリッドシステムの提案，情報処理学会ハイパフォーマンスコンピューティング研究会，2013/2/21–22，清風荘（あわら市）

⑩ Yoshiyuki Asai, Takeshi Abe, Masao Okita, Tomohiro Okuyama, Nobukazu Yoshioka, Shigetoshi Yokoyama, Masaru Nagaku, Kenichi Hagihara, and Hiroaki Kitano. Multilevel modeling of Physiological Systems and Simulation Platform: PhysioDesigner, Flint and Flint K3 service. the 12th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT 2012), pp. 215-219, Izmir, Turkey. Presented at the 3rd Workshop on High Speed Network and Computing Environments (HSNCE 2012) 2012/7/17.

⑪ Muhammad Ismail Faruqi, Fumihiko Ino, and Kenichi Hagihara, Acceleration of Variance of Color Differences-Based Demosaicing Using CUDA, 10th International Conference on High Performance Computing and Simulation (HPCS 2012), 2012/7/2-6, Madrid, Spain

⑫ 伊野文彦，遊休 GPU を用いて科学計算を高速化する GPU グリッドについて，平成 23 年度情報処理学会関西支部支部大会，2011/9/22，大阪大学（大阪府）

⑬ 岡陽介，伊野文彦，萩原兼一，GPU 向けマルチタスキングにおけるオーバヘッド削減の検討，第 9 回先進的計算基盤システムシンポジウム，2011/5/26，秋葉原コンベンションホール（東京都）

〔その他〕
ホームページ等
http://www-hagi.ist.osaka-u.ac.jp/

６．研究組織

(1)研究代表者
萩原　兼一（KENICHI HAGIHARA）
大阪大学・大学院情報科学研究科・教授
研究者番号：００１３３１４０

(2)研究分担者
伊野　文彦（FUMIHIKO INO）
大阪大学・大学院情報科学研究科・准教授
研究者番号：９０３４６１７２

研究分担者
置田　真生（MASAO OKITA）
大阪大学・大学院情報科学研究科・助教
研究者番号：５０５６３９８８