

科学研究費助成事業 研究成果報告書

平成 26 年 5 月 15 日現在

機関番号：14401

研究種目：若手研究(B)

研究期間：2011～2013

課題番号：23700057

研究課題名(和文) GPUグリッドのための細粒度サイクル共有技術の理論構築と応用

研究課題名(英文) Theoretical construction and application of fine-grained cycle sharing technique for GPU grid

研究代表者

伊野 文彦 (Ino, Fumihiko)

大阪大学・情報科学研究科・准教授

研究者番号：90346172

交付決定額(研究期間全体)：(直接経費) 3,300,000円、(間接経費) 990,000円

研究成果の概要(和文)：本研究では、グラフィクスハードウェアGPU(Graphics Processing Unit)向けのマルチタスク機構をもとに、ネットワーク上の計算資源を共有する計算グリッドシステムのための細粒度サイクル共有技術を開発した。本技術をバイオ情報学における相同性検索問題に応用し、電子文書を編集しながらも科学計算を高速化できることを示した。また、GPUの遊休サイクルモデルを構築し、遊休時間長の分布に基づいて計算資源を選択する手法を開発した。

研究成果の概要(英文)：We developed a fine-grained cycle sharing technique for computational grid systems, which share computational resources on the network. Our technique is based on a multitasking mechanism for graphics processing units (GPUs). We applied our technique to a bioinformatics problem, called homology search, and demonstrated that scientific applications can be accelerated while editing electronic documents. We also constructed an idle cycle model for GPUs and developed a method that selects computational resources according to the distribution of idle period lengths.

研究分野：総合領域

科研費の分科・細目：情報学・計算機システム・ネットワーク

キーワード：高性能計算 GPGPU マルチタスク グリッド計算

1. 研究開始当初の背景

GPU (Graphics Processing Unit) とは、PC を構成する部品の 1 つである。その目的は、画面の描画を高速化することにある。近年、PC ゲームなどの娯楽分野において、現実性の高い画像を実時間で生成したいという要望があり、GPU は CPU と比べて高い性能を実現している。CPU の性能向上が、クロック周波数の物理的な限界に起因して停滞するにつれ、GPU に関する研究は世界中で盛んに行われている。GPU は、描画処理の加速器に留まらず、C 言語の拡張 CUDA (Compute Unified Device Architecture) で記述されたプログラムを高速に実行できる道具として定着している。

しかし、現在の GPU アーキテクチャはハードウェアによるプリエンブションを提供していない。したがって、単一のアプリケーションを高速化することは得意だが、同時に複数のアプリケーションを処理することは不得手である。この性能特性は、複数のユーザがネットワーク上の遊休計算資源を共有するグリッドシステムにおいて問題となる。この問題を避けるために、従来のグリッドシステムは、スクリーンセーバを用いて排他的に遊休 GPU を活用する形態を採用してきた。つまり、スクリーンセーバが活性化したときのみ、その計算資源が遊休であるとみなし、遠隔から科学計算を実行してきた。この排他的な制御により、科学計算が資源提供者の操作を妨げないようにしている。

2. 研究の目的

本研究は、もともと GPU の機能が画面描画にあることに着目し、グラフィクス技術と密に連携し、それを動力源とする計算基盤の構築を目的とする。特に、グラフィクス技術と高性能計算技術の双方を必要とする、滑らかな描画と高速な科学計算の両立を目指し、そのための基礎理論を展開する。

これらの実現により、これまでスクリーンセーバを用いて活用されてきた、分単位の長い遊休時間だけでなく、秒単位の短い遊休時間を活用でき、複数のユーザが GPU を効率よく共有できる。例えば、計算資源の提供者が電子文書を編集しながらも、バックグラウンドジョブとして科学計算の高速化に貢献できる。この基礎理論に加え、バイオ情報学や医用画像工学への応用を、本計算基盤上で加速することにより、その有用性を示す。

3. 研究の方法

本研究では、細粒度サイクル共有技術の基礎研究から応用へと展開するために、GPU 向けマルチタスク機構をもとに、グリッド環境において細粒度サイクル共有を実現するための課題 (1) および (2) を解決する。さらに、バイオ情報学などにおける応用へ展開するための研究基盤を確立するために、課題 (3) に取り組む。

(1) 細粒度スケジューリング機構の開発

性能保証：これまで開発してきたマルチタスク機構をもとに、画面描画の滑らかさと科学計算の性能を制御し、滑らかさを保証するためのスケジューリング機構を開発する。

資源監視：スクリーンセーバに基づく従来システムを拡張し、GPU やキーボード・マウスなどの操作状況に関してネットワーク上の PC を遠隔監視し、それらの履歴をデータベースに蓄積できるシステムを開発する。GPU・CPU の計算負荷、キーボード・マウス操作の有無、描画レートなどの履歴を少なくとも 1 分間隔でデータベースに蓄積する。

資源選択：画面描画および科学計算の相性に応じて、計算を適切な計算資源に割り当てるための資源選択アルゴリズムを開発する。

で取得した履歴を用い、アルゴリズムの優位性をシミュレーションにより確認する。

(2) 遊休サイクルモデルの確立

遊休モデル：GPU を活用するグリッドシステムの実行効率を高めるためには、長い遊休時間だけでなく、秒単位の短い遊休時間を活用する必要がある。日常業務のために使用される PC を対象として、その遊休時間の長さや分布を含む履歴を確率論によりモデル化し、遊休サイクルモデルを確立する。

設計モデル：システムの目標性能に対し、どの程度の GPU をいかに集約すればよいかを見積もるための設計理論を確立する。また、分単位の遊休時間を活用する従来システムや計算資源を専有するクラスタに対し、本システムの性能を明らかにする。

(3) システム統合および実証実験

有用性評価：(1) と (2) の成果を統合し、企業などにおいて文書作成や表計算などを処理しながらも、バイオ情報学などにおける実用的な応用を高速化できることを示す。

4. 研究成果

(1) 細粒度スケジューリング機構の開発

性能保証：画面の描画レート f と CUDA プログラムを入力とし、 f を下回らない範囲で CUDA プログラムの性能を最大化する細粒度スケジューリング機構を開発した。具体的には、科学計算の性能と描画の滑らかさがトレードオフの関係にあること(図 1)に着目し、科学計算の仕事を分割することにより、その実行時間を制御する。CUDA の実行モデルは、互いにデータ依存のない計算の集合(スレッドブロック)を前提にしているため、ブロック単位での分割が可能である。さらに、画面描画および科学計算の処理が公平に GPU 上で実行されるよう、科学計算の仕事を周期 f の時間間隔で処理する(図 2)。これにより科学計算および描画処理を交互に実行できる。これらは描画アプリケーション側のコード修正を必要としないため、グリッドのような多様かつ大量の PC 環境に適応できる。

科学計算として行列積や相同性検索の問題を取り上げ、開発したスケジューリング機構を4種類のGPUアーキテクチャ(G80, Tesla, Fermi および Kepler)上で評価した。結果、毎秒 f 回の描画を実現するためには、1回あたりの仕事を $1/f$ 秒程度で完了できるように、仕事を分割すればよいことが分かった(図3)。したがって、計算資源の提供者があらかじめ最小の描画レートを申告しておけば、その申告値を下回らないようなマルチタスク実行が可能である。また、電子文書などの編集を滞らせることなく科学計算を実行するためには、 $f=10$ 程度で実用に耐えることが分かった。さらに、 $f=10$ 程度であれば、分割に起因するオーバーヘッドが無視できるほど小さいことを確認した。

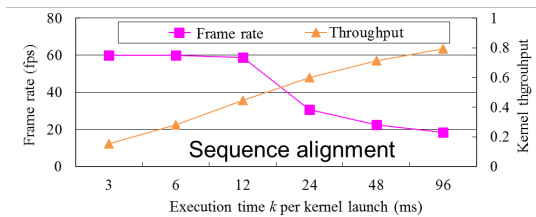


図1 描画レートと相同性検索性能のトレードオフ関係

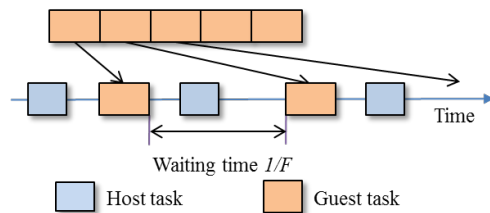


図2 マルチタスク実行の様子

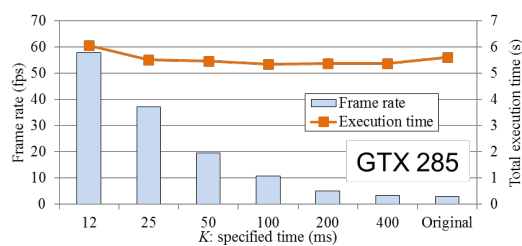


図3 仕事の分割と描画レートの関係

資源監視：計算資源の遊休状態を監視するために、キーボード・マウスのイベントを検出する監視プログラムを開発した。これらのイベントに加え、監視プログラムはビデオメモリの使用量ならびにCPUの使用率をもとに、遊休状態の計算資源を検出する(図4)。これにより、スクリーンセーバに基づく従来システムが検出できなかった、数秒程度の短い遊休時間を低オーバーヘッドで検出できることを確認した。

さらに、グリッドシステムの実行効率を高

めるために、計算資源における遊休・繁忙状態の定義を見直し、3種類の状態(完全遊休、部分遊休および繁忙)を導入した(図5)。これに伴い、遊休の度合い(完全もしくは部分のいずれか)を見積もるための手法を開発した。この手法は、科学計算を処理する直前に、直ちに制御を戻す空閑数を呼び出し、その遅延の長さをもとに遊休の度合いを識別する。このように、マウス・キーボードのイベントを監視していないため、資源提供者の対話的な操作に遮られることなく、遊休状態のGPUを検出できる。計測結果が、あらかじめ測定しておいた遅延よりも短い場合、その計算資源を部分遊休とみなす。この場合、述べた通り、科学計算を周期 f で処理し、毎秒 f 回の描画を実現する。一方、そうでなければ完全遊休とみなし、そのような周期を導入することなく、科学計算を連続処理することにより実行効率の向上を図る。

開発した監視プログラムを用い、研究室における6名の被験者を対象に半年以上に渡る履歴を取得した。なお、本研究の目的は滑らかな描画と高速な科学計算の両立であるため、履歴に残す対象は平日の10時から18時とした。これにより、計算資源が対話的に操作されることのない時間帯(深夜など)、すなわち従来システムでうまく対処できる時間帯を除外した。

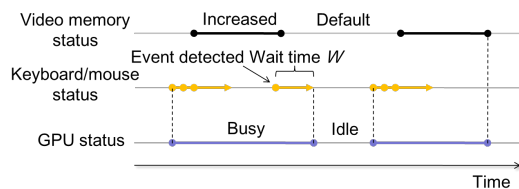


図4 遊休計算資源の検出手法

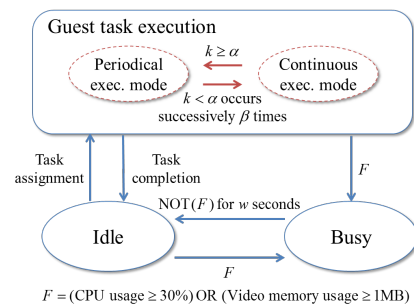


図5 完全遊休と部分遊休を区別するマルチタスク手法

資源選択：で取得した履歴を分析した結果、遊休時間の長さの分布は冪分布にしたがうことが分かった(図6)。すなわち、遊休時間が短くなればなるほど、その検出数は爆発的に増大する。したがって、GPUを共有するシステムでは、頻繁だが短い遊休サイクルを検出することが重要であり、計算の投入時だけでなく、将来に渡って安定して遊休となる資源を選択すべきである。

この安定性に着目する資源選択アルゴリズムを開発した。このアルゴリズムは、現在の遊休時間の長さをもとに計算資源を降順に整列し、先頭から順に仕事を割り当てる。例えば、過去1秒に渡り遊休状態を維持している計算資源Aと過去5秒に渡り遊休状態を維持している計算資源Bがある場合、後者に仕事を割り当てる。実際に、取得した履歴によれば、次の5秒後も遊休状態を維持する確率は、前者が17%であるのに対し、後者は40%に達する(図6)。したがって、5秒以下で完了する科学計算であれば、前者の83%が実行に失敗してしまう。本アルゴリズムは、そのような失敗を軽減できる。

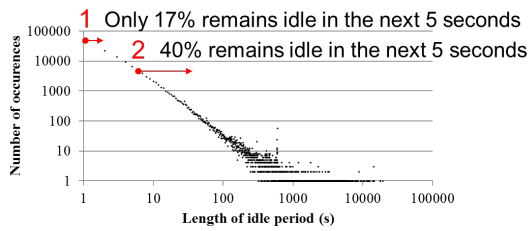


図6 遊休時間の長さの分布

(2) 遊休サイクルモデルの確立

遊休モデル：で定義した遊休状態にしたがい、GPUを対象とする遊休モデルを構築した。具体的には、計算資源の状態間を遷移する確率過程を履歴から推定し、その予測精度を検証した。構築した遊休モデルは、将来における計算資源の状態が現在の状態およびその経過時間に依存するとみなし、確率過程として準マルコフ過程を採用する。この遊休モデルを、自己回帰モデルや直前の平均に基づく手法と比較した。有意水準5%のt検定によれば、構築した遊休モデルは半数のユーザにおいて統計的に予測精度が優れていたことが分かった。ただし、予測精度は20~45%にとどまり(図7)、実用に向けてさらなる改善が必要である。

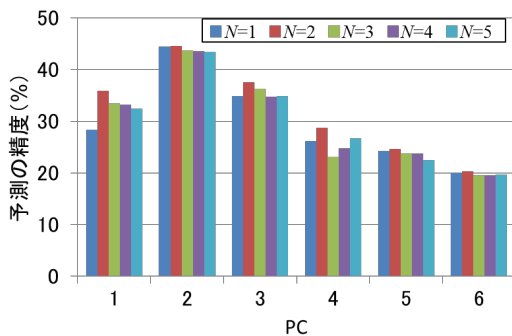


図7 準マルコフ過程に基づく遊休時間の予測精度

設計モデル：後述する実証実験の結果をもとに、GPUを活用するグリッドシステムの性能を見積もる手法を検討した。まず、遊休

の度合いを区別しない場合(参照)、計算資源の稼働時間のおよそ55%程度は遊休とみなせることが分かった(図8)。ただし、この比率は計算資源を提供するユーザの操作状況に依存する。また、相同性検索問題に限れば、P台の専有GPUクラスタに相当する性能を引き出すためには、2P台の共有GPUを用意すればよいことが分かった。

次に、遊休の度合いを区別する場合、一般的な編集作業や開発作業を主な用途とするPCにおいては、稼働時間のおよそ99%を遊休とみなせることが分かった。つまり、資源提供者がPCを対話的に操作しているような状況において、GPUが繁忙になることは稀であり、仮に繁忙に陥ったとしても一瞬で部分遊休もしくは完全遊休に戻ることが分かった。したがって、P台のGPUを持つグリッドシステムの性能は、同数のGPUクラスタとほぼ同等とみなせる。

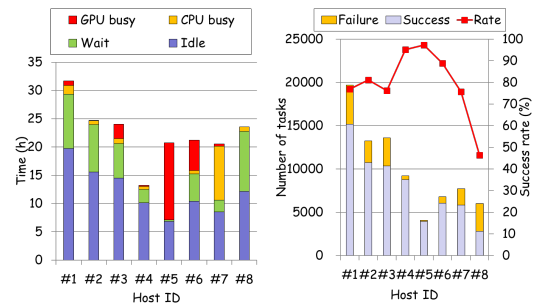


図8 各計算資源における状態とジョブ実行の内訳

(3) システム統合および実証実験

有用性評価：グリッドが得意とする組合せ爆発型の応用として、バイオ情報学における相同性検索問題を取り上げ、研究室のローカルエリアネットワーク環境において、システムの実証実験を実施した。相同性検索問題は、検索対象となる生物学データベースをあらかじめ計算資源側に転送しておけば、マスター・ワーカー方式による効率のよい並列化を実現できる(図9)。したがって、遊休状態に遷移した計算資源は、資源管理サーバに検索ジョブを要求しさえすれば、直ちに検索ジョブを処理できる。データベースとして、タンパク質のアミノ酸配列を含むUniProtKB/Swiss-Protを用い、検索ジョブは長さ63~511からなる8種のアミノ酸配列を用意した。

まず、相同性検索のプログラムをマルチタスク実行できるように、で述べた仕事の分割を施した。この際、検索性能を高めるために、ディスクからのデータ読み出し、CPUからGPUへのデータ転送、GPU上の計算、およびGPUからCPUへの結果転送をパイプライン実行できるようにCUDAプログラムを記述した。次に、任意の検索ジョブに対して、 $1/f=100$ ミリ秒程度で1回あたりの実行を完

了できるように、性能モデルを構築した。この性能モデルは、相同性検索の実行時間が全体の計算量ならびにスレッド間の同期オーバーヘッドで近似できることに基づく。この性能モデルのパラメータを最小二乗法により推定することにより、任意の検索配列長に対してその95%の実行時間を96~111ミリ秒に収めた(図10)。目標の1/f=100ミリ秒に対して、実行時間の誤差はやや大きい、滑らかな描画レート1/fを実現する用途に対しては十分であることを確認した。

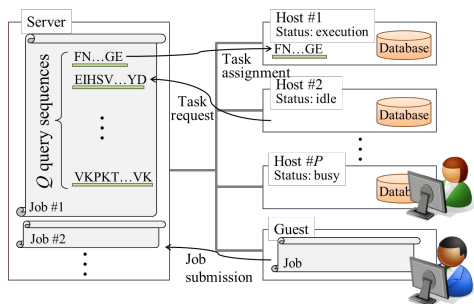


図9 遊休資源を活用する相同性検索システム

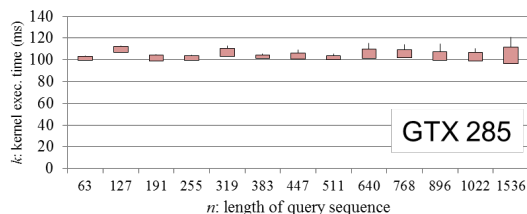


図10 検索配列長ごとの実行時間の分布

次に、8台のPCを用い、一ヶ月にわたり相同性検索システムを運用した。8台のうち、1台は世代の古いGPUとし、残りは同一世代である。その結果、8名の被験者はバックグラウンドジョブの存在に気づくことなく、日常業務を遂行できることが分かった。また、世代の古いGPUではジョブ1個あたりの実行時間がおよそ18秒に達し、計算資源が頻りに繁忙状態に陥ることで、多くのジョブ実行に失敗していた。したがって、GPUアーキテクチャの世代に合わせて1回あたりのジョブ実行を5秒程度に収めることがシステムの実行効率を高めるために大切である。

さらに、スクリーンサーバに基づく従来システムや計算資源を専有するクラスタと比較するために、運用時に取得した履歴をもとに、シミュレーションを実施した。結果、相同性検索問題に対しては、本システムは60Gセル/秒(図11)、従来システムは30Gセル/秒(図12)、専有クラスタは120Gセル/秒(図13)の計算スループットを達成できることが分かった。したがって、分単位の遊休時間に加えて、秒単位の遊休時間を活用することにより、システムの性能を倍増できることが分かった。

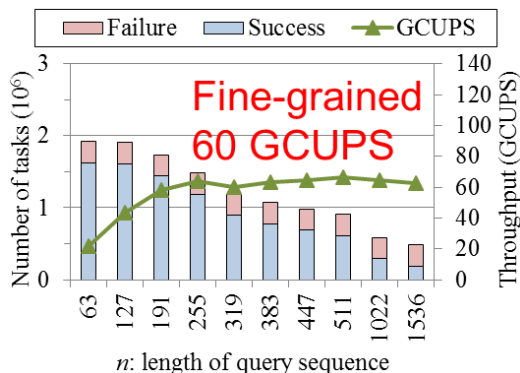


図11 本システムによる計算性能

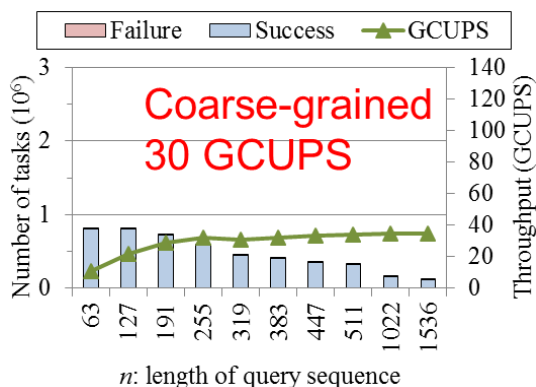


図12 従来システムによる計算性能

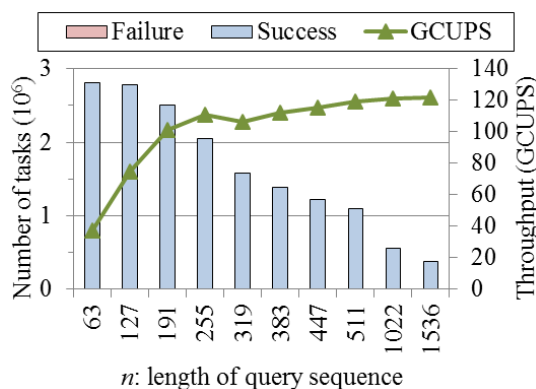


図13 専有GPUクラスタによる計算性能

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)

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

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

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

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

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

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

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

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

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12件)

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)

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

重岡謙太郎, 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)

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)

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

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利用する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)

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

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)

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/~ino/>

6. 研究組織

(1) 研究代表者

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