

令和 6 年 6 月 14 日現在

機関番号：12601

研究種目：基盤研究(A) (一般)

研究期間：2020～2022

課題番号：20H00580

研究課題名(和文) 余剰コアを活用する高性能計算・データ解析支援

研究課題名(英文) High-performance computing and data analysis support leveraging unused cores

研究代表者

埴 敏博 (Hanawa, Toshihiro)

東京大学・情報基盤センター・教授

研究者番号：30308283

交付決定額(研究期間全体)：(直接経費) 34,700,000円

研究成果の概要(和文)：本研究では、高性能計算の性能向上に直接寄与しない「余剰コア」に対し、CPUで実行している主計算を支援する役割を与えることで、システム全体の性能改善や、電力制御、プロファイリング機能などの付加機能を低オーバーヘッドで実現することを目的とし、ユーザレベルでそのような支援機能を実現するフレームワーク“UTHelper”を検討した。
その結果、ユーザプログラムに手を加えず実行中プロファイリングや並列度変更の実現、余剰コアを用いたIn Situ解析、動的コア割当を用いた負荷分散均衡化による格子H行列演算の高速化、余剰コアを用いたGPU間通信高速化、時空間ブロッキングによる遊休演算器の活用などを実現した。

研究成果の学術的意義や社会的意義

近年の高性能計算システムにおいてCPUのコア数が増加し続けているが、全コアを使うのが最善だと誤解されることも多く、実際には消費電力や放熱、メモリバンド幅等の観点でコアをあえて使用せず残した方がよい場合がある。つまり「余剰コア」が生じる。本研究では、いくつかの実アプリケーションに対し、これらの余剰コアを補助的に用い、主計算を支援することによって、全体として性能向上を実現できることや、性能を落とさずに付加機能を実現できることを実例として示すことができた。

研究成果の概要(英文)：This research aims to improve the overall system performance and realize additional functions such as power control and profiling functions with low overhead by giving "extra cores" that do not directly contribute to the performance improvement of high-performance computation a role in supporting the main computation running on the CPU. We studied "UTHelper," a framework to realize such support functions at the user level.
As a result, we realized profiling and parallelism change during execution without modifying the user program, in situ analysis using extra cores, load balancing using dynamic core allocation to speed up lattice H-matrix operations, inter-GPU communication using extra cores, and utilization of idle arithmetic units through time-space blocking.

研究分野：高性能計算

キーワード：高性能計算 メニーコア 余剰コア MPI GPU OpenMP 非同期IO

1. 研究開始当初の背景

高性能計算(HPC: High Performance Computing), 高性能データ解析 (HPDA: High Performance Data Analysis)を支えるスーパーコンピュータにおいて, システムが利用できる消費電力は限界を迎えている. 今後さらなる性能向上を実現するためには, 電力当たり性能に優れたプロセッサを導入する必要がある. 半導体プロセス技術の進歩のスピードが鈍るに従って, 電力当たり性能の最大化を指向した結果, クロック周波数はほぼ一定のままであり, スレッド当たり性能も微増にとどまっている一方で, 論理コア数が飛躍的に増加している.

研究開始当初においても, CPUには50を超えるコア数が搭載され始め, ノードあたり100コアを超えるシステムが当然になりつつあり, メニーコア時代になったと言える. しかしながら, プロセッサは熱設計消費電力(Thermal Design Power:TDP)を超えて動作できないため, 全体のコアの動作状況に応じて動的にクロックを上下させることによって全体をTDPに収めるように調整している. 全コアを使えば最大性能が得られるわけではなく, 同時に多数のコアの演算器を高い頻度で動作させると, 却って性能低下が生じる結果となることもある. プロセッサ全体のメモリバンド幅が限られることから, メモリアクセスが支配的なケースでは使用コア数を減らした方が良い結果になることもある. GPUのようなアクセラレータ搭載型のスパコンにおいても, アクセラレータ使用中には, CPUのコアは, データ転送をオーバラップ処理する程度でアクセラレータの処理が終わるのを待っている状況も多い.

このように, メニーコア時代には, コアが多数あっても直接的に性能には寄与しない, あるいは同時に使用すると却って性能を低下させてしまう「余剰コア」が多数存在することになる.

2. 研究の目的

本研究ではこれらの「余剰コア」に対して, 主計算を支援する役割を与え, 主計算の性能向上, 電力効率の向上, 低オーバヘッドで付加機能の提供を実現することを目的とする. そのような機能を提供するフレームワークとしてユーザレベルで支援機能を実現するライブラリとツール群などからなる「UTHelper」を開発する.

UTHelperでは, (1) 余剰コアによる実行中モニタリング (In Situ プロファイリング), (2) キャッシュプリフェッチによる性能向上, (3) 電力・性能バランスの向上, (4) 通信, ファイルIOの隠蔽, (5) In Situ データ解析, などを容易に実現することを目的とする.

このような種々の支援機構を統一的に扱うフレームワークはこれまで存在しておらず, 統一的なフレームワークを開発することに独自性がある. また本研究提案は, メニーコアプロセッサで構成されるスパコンシステムの潜在能力を引き出すべく, 実行中に (In Situ) プロファイルを取得しながら動的最適化を行い, 様々な付帯処理を効率よく行おうという点で独創的である.

本研究成果は, 各種スパコンで用いることが可能なように, オープンソースソフトウェアのUTHelperパッケージとして整備する. これによって柔軟かつ高度な性能最適化が実現でき, 次世代スパコンシステムにおける高性能計算・データ解析のさらなる性能向上, 生産性の向上に資する.

3. 研究の方法

余剰コアを活用するUTHelperにより, これまで使われてこなかった休眠状態の余剰コアを適切に使用して性能向上が実現できることを示す.

本研究では, まず余剰コアが活用できるケースについて調査を行う. 主計算においてメモリバンド幅が律速している状態では, 余剰コア側でメモリ操作を行うと主計算のバンド幅に悪影響を与えると考えられる. 但し, 先にメモリ要求を発行しキャッシュにプリフェッチができていれば, キャッシュミスによるメモリアクセスはキャッシュヒットに変わるため, メモリバンド幅を節約できる可能性がある. GPUなどのアクセラレータについても, 余剰コアを使ってメモリ転送を管理することで, 主計算とアクセラレータとの協調計算が容易になる.

UTHelperでは, 動的に計算コア群から不要な余剰コアを切り出してヘルパースレッドとして使用したり, 元通り計算に復帰させるなど, 柔軟な制御を行うため, ユーザレベルで, 通常のOpenMPの枠組みの中で作られる必要がある. 実際にヘルパースレッドが動作する際には, コア間のキャッシュ共有関係を元にしたスレッドの再割付が必要になる. あらかじめキャッシュ構成などのハードウェア情報を取得し, 適切なコアに割付し直すような機構を設計・実装する.

余剰コアにおいては, 実際のコアの動作状況をリアルタイムにモニタする, In Situ プロファイリングを実行する. CPUに備わっているパフォーマンスカウンタを使って, キャッシュのヒット率や命令実行数などの情報もリアルタイムに取得できる. CPU, DRAMの電力を取得, 電力制限を行うインタフェースが備わっている. これらのインタフェースから取得した情報を動的に解析しながら, CPU全体のコア動作を制御する. また, In-Situ データ解析とそれに伴うファイルIO機能, 非同期通信の実現, 通信隠蔽やメモリ等の管理機能を検討する.

4. 研究成果

(1) 余剰コアの活用に向けた実行中プロファイリング手法の検討

「余剰コア」に対し、主計算を支援する役割を与えることで、システム全体の性能改善や、電力効率を高めるための省電力制御、付加機能を低オーバーヘッドで実現することが可能ではないかと考え、ユーザレベルでこのような支援機能の実現を検討してきた。そのためには、実行時のプロファイルを行いながら、適切な並列度に変更する機能が必要になる。そこで、並列度の動的制御、動的なプロファイリングを行う機構を試作し、このような機構が実現可能であることを示し、また実ベンチマークに対して、提案するプロファイリング手法を適用することで余剰コアの存在が認識可能であることを確認し、妥当な性能分析が可能であることを示した。実際には、System Tap を用いた動的なプロファイリングと並列度変更を実現した。さらに NAS Parallel Benchmarks の FT に適用した結果、プロファイルによって得た実行クロック数の値から余剰コアの発生を確認できた(工藤 & 埜 2020)。この結果を後で述べる動的コアバインディング手法(DCB)に取り入れ、一体で開発することによって一層実用性を高めることができるようになった。なお、本成果は2021年度コンピュータサイエンス領域奨励賞を受賞した。

(2) 余剰コアを用いた In-situ 解析の実現

宇宙物理分野における実アプリケーションを例題として、余剰コアを用いた In-situ 解析の実現に必要な機能を実装した。銀河衝突シミュレーションにおいては、銀河どうしの衝突・合体過程を重力多体シミュレーションを用いて精密に計算しておき、計算途中に出力しておいたスナップショットデータを事後的に解析することで、数値的に得られた構造と実際に観測されている構造とを比較するというワークフローが一般的である。余剰コアを用いた In-situ 解析が実現できれば、大量のスナップショットデータを保持する必要もなくなり、より効率的に研究を遂行できるようになる。我々は GPU 向けに最適化された重力ツリーコード GOTHIC (Miki & Umemura 2017; Miki 2019) においては、GPU の処理を監督する 1 コア以外の CPU コアが余剰コアとなっていることに着目し、この余剰コアを活用してデータ解析を実行した。この際に OpenMP の task 指示文を用いることで、プログラムの構造を大きく変えることなく In-situ 解析を実現できたため、他のアプリケーションにおいても同様に適用可能な技術であると考えられる。本機能の実現により、シミュレーションとデータ解析全体の実行時間は最短で元の 54%になるまで高速化された(赤沢 et al. 2022)。さらに、データ解析部分を分割し、逐次実行で十分な処理に 1 コアのみ割り当てることでコア割り当ての最適化、シミュレーションとソートと解析とをオーバーラップする 3 ステージパイプライン処理を OpenMP task 指示文のみで実現し、シミュレーションと解析のみのオーバーラップの場合と比較して実行時間が 30%削減、元の場合と比べて 55%削減された(Wu & Hanawa 2024)。

またここで開発された技術を応用して、温度、消費電力や動作周波数といった GPU 状態を主計算の処理と同期した上で監視する機能についても実現した(三木 & 埜 2023)。この監視機能の実現によって、サーマルスロットリングや供給電力不足による動作周波数低下といった性能低下要因の分析ができるようになり、今後の多様なコード開発への貢献が期待される。

(3) 動的コア割当を用いた負荷分散均衡化による格子H行列演算の高速化

負荷分散不均衡は、超並列処理において並列化されたアプリケーションの性能を低下させる大きな問題である。分散メモリ並列計算環境では MPI/OpenMP のハイブリッド実装が広く用いられているが、効果的な並列化のためには MPI プロセス間や OpenMP スレッド (コア) 間での負荷分散を維持する必要がある。一方で、超並列アプリケーションにおいて、効率的な通信パターンの構築も重要な問題である。MPI プロセス間の負荷分散均衡と効率的な通信パターンの構築は両立が難しい場合も多い。格子H行列ライブラリ HACApK では、分散メモリ並列計算環境で格子H行列・ベクトル積に際して、MPI プロセス間の通信量がプロセス数に依存せず一定になるアルゴリズムを採用しており、その代償として MPI プロセス間で一定程度の負荷分散不均衡を許容していた。

本研究では、格子H行列・ベクトル積計算の高速化を実現すべく、MPI プロセスへの動的コアバインディング (DCB) 手法を提案した。DCB 手法では、各プロセスに不等数のコアを割当て、MPI プロセス間の負荷不均衡を OpenMP スレッドレベルで緩和する。並列性能を向上させるだけでなく、計算時間を増やすことなく使用コア数を減らすことで消費電力を削減する。ノード間の負荷分散は DCB では対応できないが、プロセスをノードにマッピングすることでこの問題を解決する方法についても検討した。格子H行列・ベクトル積計算に DCB を適用した数値実験の結果では、図 1 に示す通り、DCB を適用しない場合に比べて大幅な計算時間の短縮に成功した。

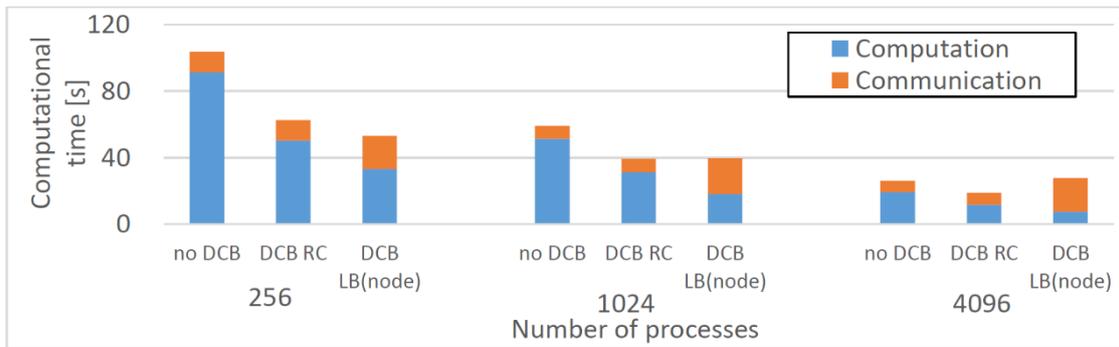


図 1 格子H行列・ベクトル積の計算時間(Kawai et al., 2023)

(4) 余剰コアを用いた GPU 間通信高速化

複数 GPU によるステンシル計算では、GPU 間の通信がアプリケーション性能低下につながる。本研究課題では、GPU 計算中に CPU の余剰コアを用いて GPU 間通信を隠蔽する手法を開発した。計算領域を複数 GPU へどのように分割して割り当てるかが実行性能に大きく影響することがわかり、領域の分割方法を変えながら、性能向上に対する通信隠蔽手法の有効性について検証を行った。東京大学の Wisteria/BDEC-01 Aquarius に搭載された 32 台の NVIDIA A100 を利用した $704 \times 704 \times 2560$ の計算領域に対する格子ボルツマン法による流体計算では、y 軸と z 軸を 1×32 に分割した場合に比べ、 4×8 に分割することで 1.07 倍高速化することがわかった。さらに、これらに対して、余剰コアであった CPU を用いて GPU 間の通信を隠蔽する機構を開発し、導入したところ、 1×32 および 4×8 に分割した計算はそれぞれ 1.20 倍と 1.18 倍の高速化を達成した。このように CPU の余剰コアを用いた GPU 間通信の隠蔽が性能向上に有効であることを示した。

これと合わせて、GPU 計算中に CPU の余剰コアを用いて非同期にファイル入出力を行い、アプリケーション全体を高速化する手法を検証した。従来用いられていた CUDA や OpenACC による GPU 計算の実装では、CPU 側で非同期処理を行うことためには、CPU 側の処理実装にも多くの開発が必要であった。本研究課題では、CPU 処理も GPU 計算も同じ枠組みで扱える OpenMP Offload を用いることで、従来よりも簡便に非同期処理を実現した。まず、計算処理全体を CPU においてスレッド並列化し、GPU 計算を管理するスレッドとファイル入出力処理を行うスレッドに分ける。双方のスレッドがメモリへアクセスする箇所では omp barrier を用いて同期し制御する。図 2 に CUDA による従来実装と OpenMP Offload を用いた実装による測定結果を示す。1 台の NVIDIA A100 を用いて $512 \times 256 \times 128$ の計算領域に対する格子ボルツマン法による流体計算で検証した。ファイル入出力処理を同期で行うと 112.5 秒の実行時間がかかるところ、GPU 計算と CPU によるファイル出力処理を非同期に行うことで 61.3 秒まで短縮することが可能となった。これはそもそもファイル入出力を行わない実装による実行時間とほぼ同じであり、CPU の余剰コアを活用することでファイル入出力を GPU 計算で完全に隠蔽することを実現した。また、簡便な仕組みにもかかわらず、余剰コアを効率的に用いたことで、従来の CUDA による同期ファイル出力(同期)を伴う計算よりも高速な実行を実現した。

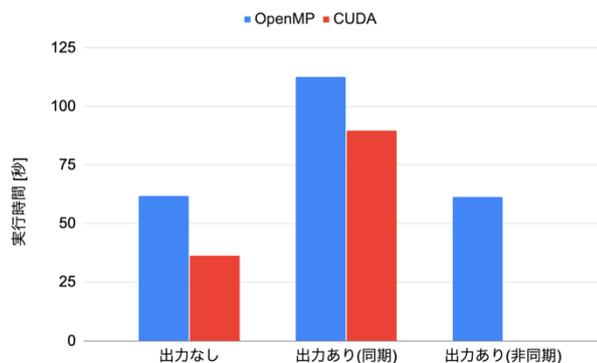


図 2 CUDA および OpenMP Offload で実装されたファイル出力なし、同期による出力あり、非同期による出力ありの流体シミュレーションの実行時間の比較

(5) 時空間ブロッキングによる遊休演算器の活用

本研究では、流体シミュレーションに頻出するステンシル計算と呼ばれる計算カーネルにおいて有効な、時空間ブロッキングと呼ばれる高速化手法に着目し、最新プロセッサにおける評価及び有効な手法の開発を行なった。時空間ブロッキングは、依存性があるために単純には並列化できない時間ループをブロックに区切り、計算の依存性を解決しながら、ブロック内の計算をキャッシュ内で完結することで、メモリへのアクセス頻度を下げ、高速化する手法である。この手法は、計算の依存関係を解消するために冗長な計算を行うため、演算コアへの負荷は増加する。よって、ステンシル計算がメモリ性能に律速される場合において、時空間ブロッキングを適用することで、十全に利用できていなかった演算コアの利用効率を高めることができる。

本研究では拡散方程式のカーネルで7点ステンシル計算と呼ばれる、最も基礎的な3次元のステンシル計算を対象とし、時空間ブロッキングを含む最適化を適用し、Sapphire Rapids, A64FX, Intel Xeon CascadeLakeなどの最新のプロセッサを用いて評価した(星野 & 埜 2021)。

図3図1に結果を示す。グラフ右端のTBが時空間ブロッキングの適用結果であり、DDRメモリを搭載する従来型のCPUであるCascadeLakeで非常に有効であるものの、Sapphire Rapidsでは有効ではなかった。時空間ブロッキングはメインメモリへの負荷を抑え、キャッシュの負荷を増加させる手法であり、今回適用した時空間ブロッキング手法は、ラストレベルの共有キャッシュを利用する手法であった。(星野 et al. 2024)におけるベンチマークでは、Sapphire Rapidsはラストレベルキャッシュが律速原因となるため、効果が得られなかったと考えられる。またA64FXにおいても時空間ブロッキングは有効ではなかったが、こちらはプロファイリングの結果を見るに、時空間ブロッキングで冗長な計算が増えたことによって、キャッシュではなくコアの負担増によって性能が低下したと考えられる。A64FX及びSapphire Rapidsでの高速化のためには、単なるパラメータの調整ではなく、実装方針を切り替える必要があり、性能モデルに基づく自動最適化を今後の課題とする。

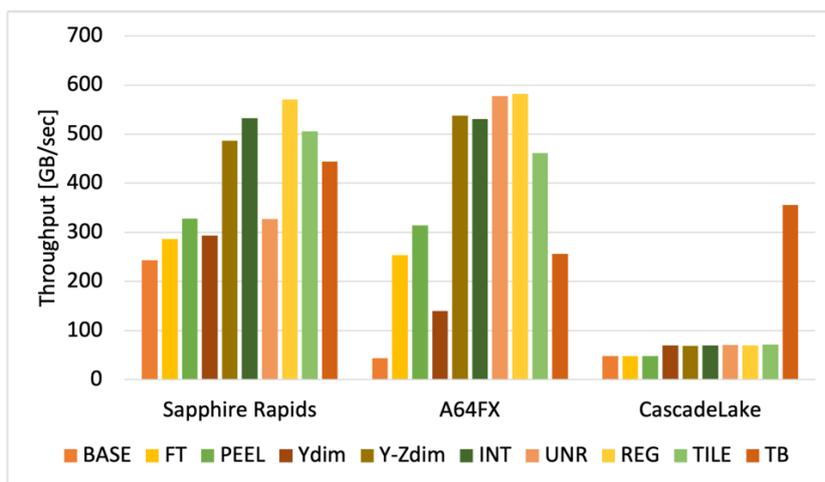


図3 時空間ブロッキング(TB)を含む、一連の最適化を順次適用した際の性能 (Throughput)

5. 主な発表論文等

〔雑誌論文〕 計19件（うち査読付論文 6件 / うち国際共著 0件 / うちオープンアクセス 1件）

1. 著者名 Ida Akihiro	4. 巻 30
2. 論文標題 Solving Block Low-Rank Matrix Eigenvalue Problems	5. 発行年 2022年
3. 雑誌名 Journal of Information Processing	6. 最初と最後の頁 538 ~ 551
掲載論文のDOI (デジタルオブジェクト識別子) 10.2197/ipsjip.30.538	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -

1. 著者名 伊田 明弘	4. 巻 2022-HPC-185 (11)
2. 論文標題 対称ブロック低ランク行列の固有値問題解法	5. 発行年 2022年
3. 雑誌名 情報処理学会研究報告 (ハイパフォーマンスコンピューティング)	6. 最初と最後の頁 1-8
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 Ida Akihiro	4. 巻 60
2. 論文標題 Algebraic Partition Construction Method for Hierarchical Matrices	5. 発行年 2024年
3. 雑誌名 IEEE Transactions on Magnetics	6. 最初と最後の頁 1 ~ 4
掲載論文のDOI (デジタルオブジェクト識別子) 10.1109/TMAG.2023.3305333	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 三木 洋平, 埴 敏博	4. 巻 2023-HPC-190 (28)
2. 論文標題 NVIDIA H100 PCIeおよびAMD MI210におけるN体計算コードの性能評価	5. 発行年 2023年
3. 雑誌名 情報処理学会研究報告 (ハイパフォーマンスコンピューティング)	6. 最初と最後の頁 1-13
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 Kawai Masatoshi、Ida Akihiro、Hanawa Toshihiro、Nakajima Kengo	4. 巻 -
2. 論文標題 Dynamic Core Binding for Load Balancing of Applications Parallelized with MPI/OpenMP	5. 発行年 2023年
3. 雑誌名 Proc. of Intl. Conf. on Computational Sci. 2023	6. 最初と最後の頁 378 ~ 394
掲載論文のDOI (デジタルオブジェクト識別子) 10.1007/978-3-031-36024-4_30	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 Kawai Masatoshi、Ida Akihiro、Hanawa Toshihiro、Hoshino Tetsuya	4. 巻 -
2. 論文標題 Optimize Efficiency of Utilizing Systems by Dynamic Core Binding	5. 発行年 2024年
3. 雑誌名 Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region Workshops	6. 最初と最後の頁 77-82
掲載論文のDOI (デジタルオブジェクト識別子) 10.1145/3636480.3637221	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 Tianya Wu, Toshihiro Hanawa	4. 巻 2024-HPC-193 (29)
2. 論文標題 Pipelined task control using OpenMP	5. 発行年 2024年
3. 雑誌名 情報処理学会研究報告 (ハイパフォーマンスコンピューティング)	6. 最初と最後の頁 1-7
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 Wenrui Mai, Toshihiro Hanawa	4. 巻 2024-HPC-193 (23)
2. 論文標題 Decoupling of Simulation on Heterogeneous Clusters Using WaitIO	5. 発行年 2024年
3. 雑誌名 情報処理学会研究報告 (ハイパフォーマンスコンピューティング)	6. 最初と最後の頁 1-6
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 畠山 昂, 下川辺 隆史	4. 巻 2023-HPC-188 (17)
2. 論文標題 複数GPUでの埋め込み境界-格子ボルツマン法の計算の最適化と性能モデルの構築	5. 発行年 2023年
3. 雑誌名 情報処理学会研究報告 (ハイパフォーマンスコンピューティング)	6. 最初と最後の頁 1-8
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 畠山 昂, 下川辺 隆史	4. 巻 27
2. 論文標題 複数GPUを用いる際の埋め込み境界-格子ボルツマン法の性能向上	5. 発行年 2022年
3. 雑誌名 第27回計算工学講演会論文集	6. 最初と最後の頁 975-980
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 大森 拓郎, 下川辺 隆史, 朝比 祐一	4. 巻 27
2. 論文標題 OpenMP Offloadingを用いたGPUでの格子ボルツマン法実行における性能評価	5. 発行年 2022年
3. 雑誌名 第27回計算工学講演会論文集	6. 最初と最後の頁 981-986
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 佐久間 大我, 下川辺 隆史, 大森 拓郎	4. 巻 28
2. 論文標題 oneAPIを用いた様々なデバイス上でのステンシル計算の実装	5. 発行年 2023年
3. 雑誌名 第28回計算工学講演会	6. 最初と最後の頁 716-718
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 Ziheng Yuan, Takashi Shimokawabe	4. 巻 28
2. 論文標題 Accelerating lattice Boltzmann method simulation with GPU computation using C++ standard language parallelism	5. 発行年 2023年
3. 雑誌名 第28回計算工学講演会	6. 最初と最後の頁 725-729
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 Ozawa So, Ida Akihiro, Hoshino Tetsuya, Ando Ryosuke	4. 巻 232
2. 論文標題 Large-scale earthquake sequence simulations on 3-D non-planar faults using the boundary element method accelerated by lattice H-matrices	5. 発行年 2022年
3. 雑誌名 Geophysical Journal International	6. 最初と最後の頁 1471 ~ 1481
掲載論文のDOI (デジタルオブジェクト識別子) 10.1093/gji/ggac386	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 Hoshino Tetsuya, Ida Akihiro, Hanawa Toshihiro	4. 巻 -
2. 論文標題 Optimizations of H-matrix-vector Multiplication for Modern Multi-core Processors	5. 発行年 2022年
3. 雑誌名 Proc. on 2022 IEEE International Conference on Cluster Computing (CLUSTER)	6. 最初と最後の頁 462-472
掲載論文のDOI (デジタルオブジェクト識別子) 10.1109/CLUSTER51413.2022.00056	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 赤沢 龍哉, 埴 敏博, 三木 洋平	4. 巻 2022-HPC-183
2. 論文標題 余剰コアを活用したOpenMP TaskによるIn-situ解析の実現	5. 発行年 2022年
3. 雑誌名 情報処理学会研究報告ハイパフォーマンスコンピューティング	6. 最初と最後の頁 1-10
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 工藤純, 埴敏博	4. 巻 2020-HPC-177 (7)
2. 論文標題 余剰コアの活用に向けた実行中プロファイリング手法の検討	5. 発行年 2020年
3. 雑誌名 情報処理学会研究報告ハイパフォーマンスコンピューティング	6. 最初と最後の頁 1-8
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 河合 直聡, 伊田 明弘, 中島 研吾	4. 巻 -
2. 論文標題 不均一なコア割付による動的負荷分散手法の検討	5. 発行年 2020年
3. 雑誌名 日本応用数理学会2020年年会講演予稿集	6. 最初と最後の頁 1-2
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 星野哲也, 埴敏博	4. 巻 2021-HPC-178
2. 論文標題 A64FXにおけるテンポラルブロッキングの実装と性能評価	5. 発行年 2021年
3. 雑誌名 情報処理学会研究報告ハイパフォーマンスコンピューティング	6. 最初と最後の頁 1-8
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

〔学会発表〕 計11件 (うち招待講演 2件 / うち国際学会 8件)

1. 発表者名 三木 洋平, 埴 敏博, 河合 直聡, 星野 哲也
2. 発表標題 AMD製GPU・NVIDIA製GPU両対応direct N-body codeの実装と性能評価
3. 学会等名 日本天文学会 2022年秋季年会
4. 発表年 2022年

1. 発表者名 三木 洋平
2. 発表標題 N体シミュレーションコードのGPU実装 --これまでとこれから--
3. 学会等名 シミュレーション天文学のこれまでとこれから -ハードウェア・アプリケーション・サイエンス-
4. 発表年 2023年

1. 発表者名 Takuro Omori, Takashi Shimokawabe
2. 発表標題 Performance Optimization Of Lattice Boltzmann Method On A64FX
3. 学会等名 15th World Congress on Computational Mechanics & 8th Asian Pacific Congress on Computational Mechanics (国際学会)
4. 発表年 2022年

1. 発表者名 Akira Hatakeyama, Takashi Shimokawabe
2. 発表標題 Performance improvement of immersed boundary-lattice Boltzmann method on multiple GPUs
3. 学会等名 15th World Congress on Computational Mechanics & 8th Asian Pacific Congress on Computational Mechanics (国際学会)
4. 発表年 2022年

1. 発表者名 Ziheng Yuan, Takashi Shimokawabe
2. 発表標題 Accelerating lattice Boltzmann method with GPU and C++ standard parallelization
3. 学会等名 10th International Congress on Industrial and Applied Mathematics (国際学会)
4. 発表年 2023年

1. 発表者名 Ziheng Yuan and Takashi Shimokawabe
2. 発表標題 Accelerating Lattice Boltzmann method with C++ standard language parallel algorithm
3. 学会等名 International Conference on High Performance Computing in Asia-Pacific Region (HPCAsia) 2024 (poster) (国際学会)
4. 発表年 2024年

1. 発表者名 Tetsuya Hoshino, Akihiro Ida, Toshihiro Hanawa
2. 発表標題 Optimizations of H-matrix-vector Multiplication for Modern Multi-core Processors
3. 学会等名 10th International Congress on Industrial and Applied Mathematics (国際学会)
4. 発表年 2023年

1. 発表者名 Tetsuya Hoshino, Akihiro Ida, Toshihiro Hanawa
2. 発表標題 Optimizations of H-matrix-vector Multiplication for Modern Multi-core Processors
3. 学会等名 Japan Geoscience Union Meeting 2023 (国際学会)
4. 発表年 2023年

1. 発表者名 Masatoshi Kawai, Akihiro Ida
2. 発表標題 Numerical Evaluation of Dynamic Core Binding Library with H-matrix Application
3. 学会等名 2022 Conference on Advanced Topics and Auto Tuning in High-Performance Scientific Computing (招待講演) (国際学会)
4. 発表年 2022年

1. 発表者名 埴 敏博
2. 発表標題 メニーコアシステムにおける余剰コア有効利用に向けて
3. 学会等名 第28回 自動チューニング研究会オープンアカデミックセッション (ATOS28) (招待講演)
4. 発表年 2022年

1. 発表者名 Akira Hatakeyama, Takashi Shimokawabe
2. 発表標題 Multi-GPU computing of moving boundary flow using lattice Boltzmann method
3. 学会等名 International Conference on High Performance Computing in Asia-Pacific Region (HPCAsia) 2022, poster (国際学会)
4. 発表年 2022年

〔図書〕 計0件

〔産業財産権〕

〔その他〕

-

6. 研究組織

	氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考
研究分担者	下川辺 隆史 (Shimokawabe Takashi) (40636049)	東京大学・情報基盤センター・准教授 (12601)	
研究分担者	星野 哲也 (Hoshino Tetsuya) (40775946)	名古屋大学・情報基盤センター・准教授 (13901)	
研究分担者	三木 洋平 (Miki Yohei) (70734375)	東京大学・情報基盤センター・助教 (12601)	

6. 研究組織（つづき）

	氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考
研究 分 担 者	伊田 明弘 (Ida Akihiro) (80742121)	国立研究開発法人海洋研究開発機構・付加価値情報創生部門 (地球情報科学技術センター)・副主任研究員 (82706)	

	氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考
研究 協 力 者	河合 直聡 (Kawai Masatoshi) (80780791)	名古屋大学・情報基盤センター・特任助教 (13901)	

7. 科研費を使用して開催した国際研究集会

〔国際研究集会〕 計0件

8. 本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関