

令和 4 年 6 月 17 日現在

機関番号：34309

研究種目：基盤研究(C)（一般）

研究期間：2017～2021

課題番号：17K00099

研究課題名（和文）グラフ探索アプリケーションの大規模並列環境での高性能化に向けた並列言語の開発

研究課題名（英文）Development of a Parallel Language for Graph Search Applications in Massively Parallel Environments

研究代表者

平石 拓 (Hiraishi, Tasuku)

京都橘大学・工学部・専任講師

研究者番号：60528222

交付決定額（研究期間全体）：（直接経費） 3,500,000円

研究成果の概要（和文）：実用的なグラフアプリケーションを分散環境で性能向上させるための並列言語に関する研究を、提案しているタスク並列言語Tascellの改善およびそれを用いたアプリケーションプログラムの実装という形で行った。言語処理系の改善としては主に、タスクワーカへの割当て戦略（スティーラ先の選択戦略）の改善や、（MPIによる）計算ノード間の通信機構の実装の改善などを行った。このように改良されたTascellを用いて、主にグラフマイニングと上記の階層型行列生成計算を対象とし、実装方式や性能パラメータのチューニングなども行い、最終的にスパコン等の分散環境上で優れた性能を実現することができた。

研究成果の学術的意義や社会的意義

本研究で対象としている、ワーカ間の知識共有を伴う実用的なグラフ探索・生成アルゴリズムの分散環境での高性能並列化の例は世界的にも少なく、そのような目標を見据えた並列言語のワークスティーラ戦略や知識共有支援機能の研究は、スーパーコンピュータ等の高性能計算環境の応用範囲を広げる意味でも有意義なものである。たとえば、階層型行列の生成処理は既存の実装では効率良い並列化が困難であったものを、本研究で並列化に成功したものである。本研究で開発したアプリケーション以外にも、SATソルバ、ソーシャルネットワークの分析、ゲーム木探索等の応用が考えられる。

研究成果の概要（英文）：We improved our proposed task-parallel language, called Tascell, and implementations of several application programs using Tascell, as a research project to develop a parallel language that can deliver good performance in distributed memory environments for practical graph applications. We improved strategies of task assignment to Tascell workers, namely strategies of how a thief worker chooses a victim worker at a work-steal attempt. We also improved our MPI-based implementation of inter-node communication in Tascell. Using the improved implementation of Tascell, we developed and tuned several task-parallel applications such as graph mining and construction of Hierarchical matrices, and we achieved good performance on distributed memory systems including a supercomputer.

研究分野：プログラミング言語，高性能・並列計算

キーワード：プログラミング言語 並列・分散処理 高性能計算 数値計算 データマイニング

1. 研究開始当初の背景

高性能計算で要求される並列数は、ノードレベル・システムレベルともに増大の一途にある。60以上のコアを有する Xeon Phi 等のメニーコアプロセッサだけでなく、Xeon 等の汎用プロセッサでも 20 コア以上を有するものが登場しており、そのようなプロセッサ数十～数百台による 1,000 並列以上の計算が特別ではなくなってきている。

したがって、アプリケーションの高性能化のためには、こうした大規模並列環境に対応した並列実装が不可欠であり、そのための研究開発が数多く行われている。しかし、充足可能性問題 (SAT) ソルバやグラフマイニング、ゲーム木探索に代表される、探索木の走査により実装されるグラフ探索アプリケーションや、不規則な木構造を生成するようなアプリケーションでは、大規模並列計算による高性能化は困難を極める。

この種の問題の高性能並列化を実現するためには 2 つの大きな課題を解決する必要がある。1 つ目は、分散環境でのコア間の負荷均衡の実現である。探索木の走査の並列化では、各部分タスク (ワーカに割り当てる部分探索木) のサイズの事前見積りが困難なため、タスクを持たないワーカが他のワーカからそのタスクの一部を盗む「ワークスティール」による動的負荷分散により負荷均衡を実現する必要がある。しかし、そのような処理を分散環境で効率良く実現するのは簡単ではなく、また実装に多大なプログラミングの労力を要する。

2 つ目の課題は、ワーカ間の知識共有の効率良い実現である。実用的なグラフ探索アルゴリズムの多くでは、探索中に得た知識をテーブルに蓄え、これを不要な部分探索木の走査を回避することによる探索空間の削減 (いわゆる「枝刈り」) や、より効率的な探索順序への変更のために利用する。この技法を並列探索でも実現するためには、ワーカ間でテーブル情報を効率良く共有する必要がある。

これらの 2 つの課題の解決は難しく、特に分散環境で良好な結果が得られた並列グラフ探索の実装例はほとんど報告されていない。負荷均衡の問題を少ない労力で解決できるプログラミング言語としては Cilk, Cilk Plus, X10 等のタスク並列言語が有名だが、Cilk, Cilk Plus は分散環境には対応しておらず、X10 は分散環境での実行に対応しているが、計算ノード間の自動負荷均衡化は行わない。研究レベルでは計算ノード間の自動負荷均衡にも対応したものもある (SilkRoad や X10/GLB など) が、そのいずれにおいても性能評価で示されているグラフ探索の例はワーカ間の知識共有を行わない単純な (すなわち「実用的と呼べない」) ものみである。

こうした問題に対し、我々は以前よりタスク並列言語 Tascell による実用的なグラフ探索アプリケーションの高性能化の研究を進めている。Tascell は、ワーカは他のワーカからタスク要求を受けない限りは一切タスクを生成せず、要求を受けるとタスク生成に最適な過去の状態を一時的に復元したのちタスク生成する、という独特なワークスティール機構を備え、これにより Cilk 等の他の主要なタスク並列言語を大幅に上回る性能および効率的なノード間の自動負荷均衡を実現している。

その後の研究で、グラフマイニングによる蛋白質の解析の並列化 Tascell を用いて行った。その結果、定期的な差分情報送信による分散環境でのワーカ間の知識共有などの技法により、8 ノード × 16 コアの実行で 1 ノード × 16 コアに対して 1.75 倍の性能向上を達成した。この性能向上率は一見小さいと考えられるかもしれないが、この問題の複雑さとこれまで他の研究でもこの種の計算で分散環境での性能向上がほとんど得られていないことを鑑みれば大きな前進である。とはいえ、複数の性能阻害要因により並列性能が頭打ちになっていることも事実であり、より大規模な並列環境で満足な性能を得るためには、それらの要因の解消が必須である。

2. 研究の目的

本研究は、充足可能性問題 (SAT) ソルバやグラフマイニング、ゲーム木探索のようなグラフ探索の実問題を主なターゲットとして、大規模並列環境 (研究期間内での現実的な目標として数百～千コア程度) に対応したアプリケーションの高性能実装およびそのプログラミングを容易にするためのタスク並列言語処理系の完成を目的とする。

タスク並列言語処理系に関しては、既存の Tascell 処理系をベースとし、以下のような処理系実装の改善および追加機能の開発を行う。

- 通信層の実装改善: ノード間通信は、現状は TCP/IP 実装を基本としている。MPI 版の試験実装も行っているが、レイテンシ等の面で難があるものとなっている。そこで、最新の MPI 実装の機能等を活用した実装改善により、処理系のノード間通信の基本性能を向上させる。

- ワークスティール戦略の改善: 既存の Tascell は大規模並列環境での実行に対する想定が十分ではなく、ノードを跨ぐスティールが極端に増えたり、小さいタスクを少数のワーカ間で何度も盗み合うことによる性能劣化が無視できなくなっている。そこで、アプリケーション

ョンの知識を利用してプログラムレベルで適切なスレッド先を選択できるようにする機能追加、およびそれを用いた戦略改善などによりこの性能劣化の問題を解消する。

- ワーカー間の知識共有支援機能：既存の Tascell は、タスク授受時以外でのワーカー間の情報共有は特にサポートしておらず、プログラマがたとえば MPI や pthread のロック機能を直接呼び出す必要がある等、ワーカー間の知識共有のための言語機能が不十分である。そこで、キャッシング戦略やコンシステンシモデルの指定等により、簡便に知識共有を実現できるようなプログラミングインターフェースを設計し、MPI 等を活用した効率的な実装を行う。

- アプリケーションの実装に関しては、上記で開発される新機能も活用して、蛋白質解析等の Tascell プログラムの開発・チューニングを行う。たとえば、解析対象となるグラフの性質を分析し、その結果に基づいてグラフを分割して各計算ノードにある程度静的に割り当てたり、知識を全ワーカーで共有せず内容に応じて共有範囲を限定するなどの手法を検討・開発する。

3. 研究の方法

平成 29 年度は、Tascell の通信層の実装の改良およびワークスティーリング戦略の改善に関する機能の設計・実装を行う。平成 30 年度から 31 年度の前半にかけて、ワーカー間の知識の実現を支援するためのプログラミングインターフェースの設計・実装を行う。平成 31 年度後半以降は、それまでに開発した Tascell の機能を利用した、グラフ探索アプリケーションの並列実装およびチューニングを行う。また、Tascell の実装や新しいプログラミングインターフェースの妥当性の検証およびフィードバックも行う。

4. 研究成果

(1) Tascell の実装・機能改善

Tascell の通信機構の実装改善

他の多くのタスク並列言語にはない Tascell の特徴として、分散環境での並列計算にも対応しており、かつノードを跨いだワーカー間の自動負荷分散にも対応しているという点がある。初期の Tascell での分散環境対応の実装では、TCP/IP を用いてノード間通信を実現していた。これは性能上問題であるほか、計算ノード間の通信に TCP/IP をサポートしないスーパーコンピュータも多いという問題点があった。そこで、MPI ライブラリを用いた Tascell のプロトタイプ実装を行っていたが、この実装では、開発当時に普及していた MPI ライブラリの制約下での動作を実現するため、性能上は不利となる実装が行われていた。具体的には、複数のスレッドが同時に MPI 関数を呼び出さないように保証する必要があるという制約を守るため、単一の通信用スレッドが外部ノードとの送受信の両方を担い、ワーカーはこの通信用スレッドにメッセージの送信を依頼する、という実装方式をとっていた。このため、この通信スレッドの負荷が重くなり、通信の遅延が大きくなるという問題があった。

その後、複数のスレッドが同時に MPI 関数を呼び出すことが許される実装が一般的になってきた。そこで本課題の研究では、マルチスレッド対応 MPI のサポートを生かしてワーカースレッドが直接送信ノードに対してメッセージを送信し、またメッセージの受信のみを担う受信スレッドを置くという方式により、メッセージ送受信にかかる遅延の低減を目指した。京都大学のスーパーコンピュータ（36 コアを持つ計算ノード最大 10 台）を用いた性能評価の結果、多くのベンチマークプログラムにおいて、従来の MPI 実装を上回る性能を実現できた。

この時点の実装では、ワーカーがデータを外部ノードに送信する際には、データを一旦バッファにコピーして MPI 関数で送信し、受信側も MPI 関数でバッファ領域に取り込んだデータをワーカーのプライベート領域にコピーする（図 1）という実装となっており、性能劣化の原因となっていた。そこで、通信機構のさらなる実装改善として、異なる計算ノードのワーカー間が、バッファへのコピーを行うことなく MPI 通信により直接データのやりとりを行う実装（図 2）を行った。上記と同じ計算環境での性能評価の結果、特に大きな配列データを扱うベンチマークプログラムに対して性能改善を達成することができた。

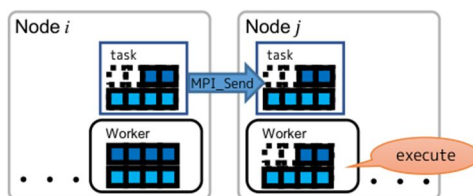


図 1 バッファへのコピーをとともなうノード間のデータ通信

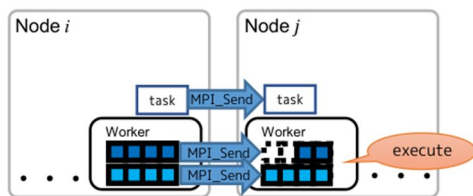


図 2 ワーカースレッド間の直接データ通信

Node-aware ワークスティーリング機能

配列の最大値を求める計算のように、計算の種類によっては明らかにマルチノード並列化による性能向上が見込めないものがある。マルチノード並列化による性能向上が見込める計算と見込めない計算が混在するようなアプリケーションでも適切な並列計算が行えるように、TasceII の並列 for などの並列構文にアノテーションを追加することで、当該の並列ループに関してはノード外からのワークスティーリングを拒否するという指示ができるようにした。なお、このアノテーションは、同一ノードのワーカからのワークスティーリングを拒否するためにも用いることができ、たとえばサイズの大きなタスクについてはなるべく外部ノードのワーカにスティーリングされるようにしたいケースなどに活用できる。この機構は、以下で述べる階層型行列生成の分散環境での実装に利用し、その有効性を確認した。

(2) アプリケーションの並列実装

グラフマイニング

対象としたグラフマイニングは、各頂点が複数のアイテムと紐付けられたようなグラフと閾値（整数）を入力とし、全頂点が閾値以上の数の共通アイテムを持つような部分グラフを全列挙するというものである。このようなグラフマイニングはソーシャルネットワーク解析や、タンパク質ネットワークから共通する複数の薬品に反応するような部分を抽出するという創薬研究などに応用可能である。このグラフマイニング問題を解く効率的な逐次アルゴリズムは COPINE という名前ですでに提案されており、このアルゴリズムの並列化および TasceII による実装も本研究課題以前に行ってきた。

本課題の研究では、TasceII の分散環境における中断処理機能を用いたさらなる高速化に関する検討を行った。COPINE は、探索中に得た知識をテーブルに書き込み、その知識を利用して不要な部分木の探索を行わないようにする（枝刈り）ことで、探索空間を大幅に削減するが、この仕組みを並列探索にも適用しようとする、次の問題が生じる。逐次探索においては、探索部分木に対する枝刈りは、当然その探索開始前に行われる。ところが、並列探索では、あるワーカがある部分木に対する枝刈りを実施しようとしたとき、その部分木が他のワーカによってすでに走査が開始されているということが起こりうる。これに対して何の対処も行わなければ、枝刈り対象であった部分木が完全に走査されてしまい、無駄な探索が無視できないほど増えてしまう。

この問題を解決するため、ワーカが探索部分木に対する枝刈りを行う際、別のワーカがその部分木を探索中でないかをチェックし、もし探索中であればそのワーカにその部分木は枝刈り対象であることを通知することで不要な探索の中断を促す仕組みを以前提案している。通知を受けたワーカは、前節で述べた例外処理機構における例外を投げることで不要な探索を中断する。この機構が共有メモリ環境で有効なことは確認していたが、本課題ではこの中断機構を用いた探索を分散環境で実装・評価した。

評価は京都大学のスーパーコンピュータ（16 コアを持つ計算ノード最大 16 台）を用いて行った。結果、分散環境においても中断機構により探索空間の削減ならびに実行時間の短縮を実現できることが確認できた。

階層型行列の生成

境界要素解析や多体問題シミュレーションでは、物理要素間の相互関係を表す密行列を係数行列とする連立一次方程式を解くことが必要になることが多い。しかし、 N 個の物理要素の全相互関係を表す密行列の要素数は N^2 となり、これを直接扱う計算手法では、必要なメモリ容量や計算時間が N が増えるにつれて急激に増大してしまう。そこで、そのような密行列の近似圧縮表現である階層型行列（図 3）を用いる手法が提案されている。階層型行列とは区分けされた行列の各ブロック（葉行列）を、積がその葉行列の近似となるような 2 つの低ランク行列で表現したものである。これにより必要メモリサイズを $O(N \log N)$ に抑えることができる。

階層型行列の生成は、行列の区分けの決定、その各区分けに対する葉行列要素の計算という二段階で行われることが多い。行列の区分け処理はさらに、要素集合を要素間距離に基づいて二分割する作業を再帰的に繰り返すことによる「クラスタ木」(CT) の構築と、クラスタ木の同レベルにある 2 ノードの組に対応するクラスタ間距離をそれぞれ評価することによる「ブロッククラスタ木」(BCT) の構築により行うことができる。ここで構築・走査される CT, BCT は一般に不規則な構造をしているため、その効率的な並列化は難しく、従来の階層型行列ライブラリの実装では逐次実装しか行われていない部分であった。そこで、本課題の研究では TasceII を用いてこの処理の並列化を実施した。

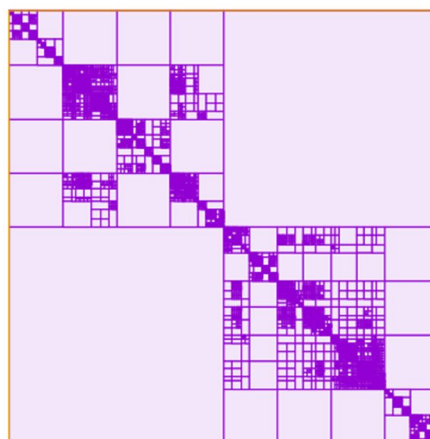


図 3 階層型行列の例

BCT の構築は比較的単純な処理であるため、Tascel の機能を用いれば分散環境においてもコア数に応じた性能向上を簡単に得ることができた。一方、CT の構築においては各 CT ノードの構築のために、分割対象となる要素を並び替える(クイックソートのピボッティングに相当する処理を行う)という処理が必要となるため、CT ノード間のみではなく CT ノード内の処理についても並列化するという二段階の並列化を行う必要があった。加えて、ノード間の並列化についてはマルチノード並列による性能向上が見込めるが、ノード内の処理はマルチノード並列による性能向上がまったく見込めない種類の計算であるため、上で述べた node-aware ワークスティーリング機構を開発・採用した。

京都大学のスーパーコンピュータ(36 コアを持つ計算ノード最大 8 台)および 3 次元電磁場解析に用いられる最大 1 億要素のデータを用いて評価を行った結果、CT ノード構築では C 言語の逐次実装に対して最大約 18 倍、BCT ノード構築では最大約 140 倍、区分け決定処理全体では最大約 17 倍の性能向上が得られた。特に CT 構築部分の分散環境における並列化については、ワークスティーリング戦略の改善など課題は残されているが、これまで並列化が実施されていなかった計算の並列化を行い有用な性能向上が得られたという点で、大きな成果が得られたといえる。

5. 主な発表論文等

〔雑誌論文〕 計6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 5件）

1. 著者名 Ryusuke Nakashima, Masahiro Yasugi, Hiroshi Yoritaka, Tasuku Hiraishi, Seiji Umatani	4. 巻 29
2. 論文標題 Work-stealing Strategies That Consider Work Amount and Hierarchy	5. 発行年 2021年
3. 雑誌名 Journal of Information Processing	6. 最初と最後の頁 478-489
掲載論文のDOI（デジタルオブジェクト識別子） 10.2197/ipsjjip.29.478	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 Tatsuya Abe, Tasuku Hiraishi	4. 巻 29
2. 論文標題 An Extensionally Equivalence-ensured Language for Task Parallel Processing with Backtracking-based Load Balancing	5. 発行年 2021年
3. 雑誌名 Journal of Information Processing	6. 最初と最後の頁 434-448
掲載論文のDOI（デジタルオブジェクト識別子） 10.2197/ipsjjip.29.434	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 Masahiro Yasugi, Daisuke Muraoka, Tasuku Hiraishi, Seiji Umatani, Kento Emoto	4. 巻 Article 77
2. 論文標題 HOPE: A Parallel Execution Model Based on Hierarchical Omission	5. 発行年 2019年
3. 雑誌名 ICPP 2019: Proceedings of the 48th International Conference on Parallel Processing	6. 最初と最後の頁 1-11
掲載論文のDOI（デジタルオブジェクト識別子） 10.1145/3337821.3337899	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 Bai Zhengyang, Hiraishi Tasuku, Nakashima Hiroshi, Ida Akihiro, Yasugi Masahiro	4. 巻 27
2. 論文標題 Parallelization of Matrix Partitioning in Construction of Hierarchical Matrices using Task Parallel Languages	5. 発行年 2019年
3. 雑誌名 Journal of Information Processing	6. 最初と最後の頁 840 ~ 851
掲載論文のDOI（デジタルオブジェクト識別子） 10.2197/ipsjjip.27.840	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 Masahiro Yasugi, Reichi Ikeuchi, Tasuku Hiraishi, Tsuyeyasu Komiya	4. 巻 27
2. 論文標題 Evaluating Portable Mechanisms for Legitimate Execution Stack Access with a Scheme Interpreter in an Extended SC Language	5. 発行年 2019年
3. 雑誌名 Journal of Information Processing	6. 最初と最後の頁 177-189
掲載論文のDOI (デジタルオブジェクト識別子) 10.2197/ipsjjip.27.177	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 Hiroshi Yoritaka, Ken Matsui, Masahiro Yasugi, Tasuku Hiraishi, Seiji Umatani	4. 巻 82
2. 論文標題 Probabilistic guards: A mechanism for increasing the granularity of work-stealing programs	5. 発行年 2019年
3. 雑誌名 Parallel Computing	6. 最初と最後の頁 19-36
掲載論文のDOI (デジタルオブジェクト識別子) 10.1016/j.parco.2018.06.003	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 -

〔学会発表〕 計33件 (うち招待講演 0件 / うち国際学会 5件)

1. 発表者名 白 正陽, 平石 拓, 伊田 明弘, 八杉 昌宏
2. 発表標題 Parallelization of Matrix Partitioning in Hierarchical Matrix Construction Using Node-aware Work Stealing
3. 学会等名 情報処理学会第138回プログラミング研究会
4. 発表年 2022年

1. 発表者名 蔡 皇誠, 平石 拓, 中島 浩, 八杉 昌宏
2. 発表標題 Copy Reduction and Latency Hiding for MPI-Based Implementations of the Tascell Task-Parallel Language
3. 学会等名 日本ソフトウェア科学会プログラミング論研究会第22回プログラミングおよびプログラミング言語ワークショップ (PPL2021) カテゴリ 3
4. 発表年 2021年

1. 発表者名 中嶋 隆介, 八杉 昌宏, 寄高 啓司, 平石 拓, 馬谷 誠二
2. 発表標題 Work-Stealing Strategies That Consider Work Amount and Hierarchy
3. 学会等名 第132回プログラミング研究会 (PRO-2020-4)
4. 発表年 2021年

1. 発表者名 安部 達也, 平石 拓
2. 発表標題 An Extensionally Equivalence-ensured Language for Task Parallel Processing with Backtracking-based Load Balancing
3. 学会等名 第131回プログラミング研究会 (PRO-2020-3)
4. 発表年 2020年

1. 発表者名 中嶋 隆介, 八杉 昌宏, 平石 拓, 馬谷 誠二
2. 発表標題 分散メモリ環境における階層性と仕事量を考慮したワークスティーリング戦略
3. 学会等名 日本ソフトウェア科学会第37回大会 (JSSST2020)
4. 発表年 2020年

1. 発表者名 八杉 昌宏, 村岡 大輔, 平石 拓, 馬谷 誠二, 江本 健斗
2. 発表標題 HOPE: 階層的計算省略に基づく耐障害性を備えた並列実行モデル
3. 学会等名 日本ソフトウェア科学会第37回大会 (JSSST2020)
4. 発表年 2020年

1. 発表者名 八杉 昌宏, 村岡 大輔, 平石 拓, 馬谷 誠二, 江本 健斗
2. 発表標題 HOPE: A Parallel Execution Model Based on Hierarchical Omission
3. 学会等名 日本ソフトウェア科学会プログラミング論研究会第22回プログラミングおよびプログラミング言語ワークショップ (PPL2020) カテゴリ 2
4. 発表年 2020年

1. 発表者名 山口 左近, 平石 拓, 中島 浩, 八杉 昌宏
2. 発表標題 並列言語Tascellのタスク定義自動生成による記述性向上
3. 学会等名 日本ソフトウェア科学会プログラミング論研究会第22回プログラミングおよびプログラミング言語ワークショップ (PPL2020) カテゴリ 3
4. 発表年 2020年

1. 発表者名 Daiki Kojima, Tasuku Hiraishi, Hiroshi Nakashima, Masahiro Yasugi
2. 発表標題 Implementing the Tascell Task-Parallel Language Tascell Using Multithreaded MPI
3. 学会等名 International Conference on High Performance Computing in Asia-Pacific Region (HPC Asia 2020), Poster presentation (国際学会)
4. 発表年 2020年

1. 発表者名 Ryusuke Nakashima, Hiroshi Yoritaka, Masahiro Yasugi, Tasuku Hiraishi, Seiji Umatani
2. 発表標題 Extending a Work-Stealing Framework with Priorities and Weights
3. 学会等名 9th Workshop on Irregular Applications: Architectures and Algorithms IA3 2019 (SC19 workshop) (国際学会)
4. 発表年 2019年

1. 発表者名 Zhengyang Bai, Tasuku Hiraishi, Hiroshi Nakashima, Akihiro Ida, Masahiro Yasugi
2. 発表標題 Implementation of Partitioning of Hierarchical Matrices using Task Parallel Languages
3. 学会等名 The 48th International Conference on Parallel Processing (ICPP 2019), Poster presentation (国際学会)
4. 発表年 2019年

1. 発表者名 Zhengyang Bai, Tasuku Hiraishi, Hiroshi Nakashima, Akihiro Ida, Masahiro Yasugi
2. 発表標題 Parallelization of Matrix Partitioning in Construction of Hierarchical Matrices using Task Parallel Languages
3. 学会等名 The 3rd cross-disciplinary Workshop on Computing Systems, Infrastructures, and Programming (xSIG 2019)
4. 発表年 2019年

1. 発表者名 Zhengyang Bai, Tasuku Hiraishi, Hiroshi Nakashima, Masahiro Yasugi, Akihiro Ida
2. 発表標題 Parallelization of Matrix Partitioning in Construction of Hierarchical Matrices using Task Parallel Languages
3. 学会等名 情報処理学会第122回プログラミング研究会 (PRO-2018-4)
4. 発表年 2019年

1. 発表者名 Tasuku Hiraishi
2. 発表標題 Dynamic Load Balancing for Construction and Arithmetic of Hierarchical Matrices
3. 学会等名 SIAM Conference on Parallel Processing for Scientific Computing (SIAM PP18) (国際学会)
4. 発表年 2018年

1. 発表者名 白 正陽, 平石 拓, 伊田 明弘, 中島 浩
2. 発表標題 階層型行列の区分け決定処理のCilk Plusによる並列化
3. 学会等名 日本ソフトウェア科学会プログラミング論研究会第20回プログラミングおよびプログラミング言語ワークショップ (PPL2018) カテゴリ 3
4. 発表年 2018年

1. 発表者名 平石 拓, 村岡 大輔, 八杉 昌宏
2. 発表標題 タスク並列言語におけるノード間通信の実装方式の検討
3. 学会等名 第59回プログラミング・シンポジウム
4. 発表年 2018年

1. 発表者名 奥野 伸吾, 平石 拓, 中島 浩, 八杉 昌宏, 瀬々 潤
2. 発表標題 分散環境での並列グラフマイニングにおけるタスク中断処理による冗長探索削減
3. 学会等名 並列/分散/協調処理に関するサマー・ワークショップ (SWoPP2017) : 第115回プログラミング研究会 (PRO-2017-2)
4. 発表年 2017年

1. 発表者名 寄高 啓司, 八杉 昌宏, 平石 拓, 馬谷 誠二
2. 発表標題 優先度ならびに重みを用いたワークスティールフレームワークの性能改善
3. 学会等名 The 1st cross-disciplinary Workshop on Computing Systems, Infrastructures, and Programming (xSIG 2017)
4. 発表年 2017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
--	---------------------------	-----------------------	----

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
---------	---------