様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　６ 年　 ６ 月 １４ 日現在

機関番号：３４４０６
研究種目：基盤研究(C)（一般）
研究期間：2018～2023
課題番号：１８Ｋ０２９１６

研究課題名（和文）図形アニメーションを基盤とした並列プログラミングの学習支援システムの研究

研究課題名（英文）Learning Support System for Parallel Programming using Graphical Animation

研究代表者
水谷　泰治（Mizutani, Yasuharu）

大阪工業大学・情報科学部・准教授

研究者番号：１０４１１４１４

交付決定額（研究期間全体）：（直接経費）　　2,600,000円

研究成果の概要（和文）：本研究では図形アニメーションを基盤とした並列プログラミング環境を開発した．本環境は，Processing言語とその開発環境を用いて，図形アニメーションを描画するマルチスレッド並列プログラムを容易に作成できるフレームワークを提供する．本環境を用いて並列プログラミングの初学者に対して演習を実施し，アンケートを実施したところ，アニメーションの動作速度が変化することで並列化による高速化を実感できた，図形の描画結果によってスレッドセーフでない動作を実感できた，との結果を得ることができた．これより，並列プログラミングを初めて学ぶ際の環境として適切であることがわかった．

研究成果の学術的意義や社会的意義
マルチコアCPUやGPUの普及など，並列処理を行えるハードウェア環境が普及しつつある一方で，それらを効果的に扱うためのソフトウェア開発技術の普及が遅れている．従来の並列プログラミングの学習では，並列プログラムの作成や教材自体が難しかったが，本研究で開発した並列プログラミング環境により，図形アニメーションといった高速処理の効果が見た目にわかりやすい環境を提供することができた．また，実際の授業において実施してアンケートを行ったことで，初学者に対して並列プログラミングの興味を持たせることに貢献できていることがわかった．

研究成果の概要（英文）：We developed a parallel programming environment based on graphical animation. This environment provides a framework for easily writing multi-thread parallel programs which can draw graphical animations using the Processing language. We used this programming environment to a parallel programing exercises for beginners of parallel programming. The questionnaire results show that students were able to realize the speed-up by parallelization and the non-thread-safe operation, and this environment is appropriate to start learning parallel programming.

研究分野：並列計算

キーワード：並列プログラミング　プログラミング教育

様　式　Ｃ－１９、Ｆ－１９－１（共通）

１．研究開始当初の背景

近年では一般的な PC に搭載される CPU ですらマルチコア構成になっている．マルチコア CPU による高速計算の恩恵を得るためには，プログラム開発者自身がマルチコア CPU を効率良く用いるプログラム（並列プログラム）を書く必要がある．しかし現状では，最先端の科学技術分野を除き，一般的な開発者にまで並列プログラミングが普及するには至っていない．その理由は，様々なツールを用いたとしても並列化できるか否かの判断は容易ではなく，開発者自身が並列計算の素養を身に付けた上で判断し，適切にプログラムを記述しなければいけないなど，並列プログラミング自体が難しいためである．そのため，一般的な開発者にまで真に並列計算の活用を広めるためには，開発者にとって並列プログラミングを学びやすい環境を普及させる必要がある．

並列プログラミングの初学者に対して学習を難しくしている要因として，並列プログラミングに対する学習意欲の維持の難しさが挙げられる．並列プログラミングの入門的な題材としてよく用いられるものは，行列演算や連立方程式の求解などの数値計算問題である．これらは並列プログラミングを学ぶにあたり数値計算の知識が必要となり，数値計算自体に興味がある者以外には難解である．また，並列プログラムの実行環境は端末ベースであり，面白みに欠ける．これは「並列プログラミングは難しい，面白くない」といった印象を与えることにつながり，並列プログラミングの学習における入口の時点で敬遠される原因の１つとなっていると考えられる．

２．研究の目的

本研究では，前述の問題を解決できる並列プログラミング学習環境の開発とその効果の検証を目的とし，図形アニメーションを基盤とした並列プログラミング環境と学習教材を構築する．図形アニメーションを用いることで，アニメーションの動作の滑らかさや教材の単純さによって，並列計算の効果が見た目にわかりやすくなり，並列プログラミングの学習に対する興味を維持させることが可能になると期待できる．さらに，構築した並列プログラミング環境と教材を用いて並列プログラミング演習を実施し，その効果について検証する．

３．研究の方法

図形アニメーションに基づく並列プログラミング環境については，Processing 言語を用いて構築する．Processing は Java をベースとしたグラフィック機能に特化したプログラミング言語および統合開発環境であり，プログラミング初心者でも容易に図形やアニメーションを描画できるプログラムを作成できるという特徴がある．また，Processing は Java で作成したクラスを使用することも可能である．並列プログラミング環境を構築するために，Java のマルチスレッド機能を初学者が扱いやすいように抽象化する．この並列プログラミング環境を用いた教材としては，N 体問題などの物理シミュレーションプログラムを対象とする．N 体問題のような計算量の大きいシミュレーションを用いることでプログラムの動作を視覚的に実感させることと，並列化による高速化の効果を実感させることを狙う．また，並列プログラミングの初学者を対象としたプログラミング演習を実施し，開発した並列プログラミング環境と教材についてアンケートを実施することで評価する．

４．研究成果

(1) 図形アニメーションに基づく並列プログラミング環境の構築

図形アニメーションに基づく並列プログラミング環境として，Java のマルチスレッドの機能を用いた抽象クラスを作成し，ユーザはそのクラスを継承することで並列処理および図形アニメーションを行える環境（以降，本環境と呼ぶ）を構築した．

図 1 に本環境を用いた並列プログラミングを示す．本環境では並列に動作させる処理を記述するためのクラス（ParallelTask クラス）を提供する．ParallelTask クラスには run メソッドという名前の抽象メソッドがあり，ユーザは ParallelTask クラスを継承し，サブクラスにおいて run メソッドを実装する．この run メソッドが並列に動作する処理となる．run メソッド内では，図 1 の中央上部のプログラムのように getMyRank メソッドを用いてスレッド ID を取得できる．この ID を用いて各スレッドに異なる動作をさせることができる．次に，本環境が提供するもう 1 つのクラスである Parallel クラスを用い，図 1 の中央下部に示すように，ユーザが作成した処理と並列実行数の登録（Parallel.init メソッド），並列呼び出し（Parallel.fexec メソッド），画像統合（Parallel.merge メソッド）を行う．

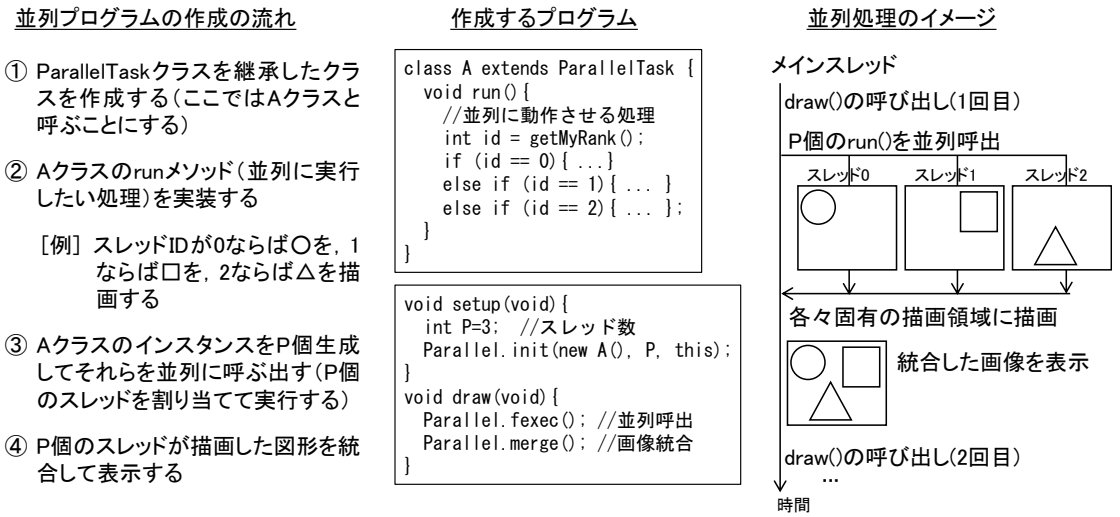


図 1. 構築した環境を用いた並列プログラミング

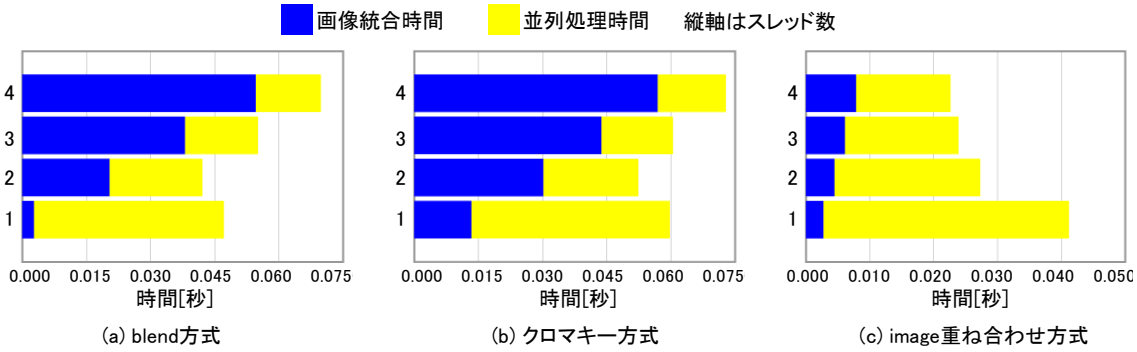


図 2. プランクトン生態シミュレーションプログラムの実行時間

各スレッドはそれぞれ固有の描画領域をもち，そこに図形を描画することができる．並列呼び出しを終えた後，画像統合を行うことで全体として 1 つの画像にまとめられ，画面に表示される．このような形式としている理由は，Processing の描画はスレッドセーフではなく，各スレッドが同時に直接表示用の領域に描画を行うと競合が発生して意図した図形が描画されないためである．そのため，各スレッドが固有の描画領域に描画し，その後，それらを統合する方式とした．

(2)画像統合処理の高速化

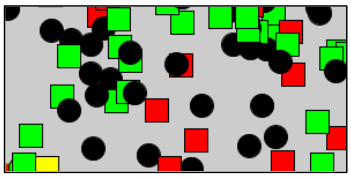
Processing の draw メソッドはデフォルトでは 1 秒間に 60 回呼び出される．1 回の draw メソッドの処理に 1/60 秒以上を要する場合，アニメーションのコマ送りが遅くなり，滑らかなアニメーションの表示ができなくなる．本環境では，(1)で述べたように 1 回の draw の実行の中で並列呼び出しと画像統合が行われており，画像統合を高速に行うことが図形アニメーションを滑らかに行うことにつながる．本環境では，各スレッドが描画した領域を統合するための手法として，Processing に含まれる blend メソッドを用いる方式（blend 方式），描画領域の各画素を比較して表示する画素を決める方式（クロマキー方式），描画領域に透明色を設定した上で Processing に含まれる image メソッドを用いる方式（image 重ね合わせ方式）を実装した．

本環境を用いてプランクトン生態シミュレーションプログラムを並列化し，前述の 3 種類の画像統合方式の性能を計測した．図 2 はこのプログラムを並列実行したときの 1 コマあたりの実行時間である．図 2 より，全てのスレッド数において画像統合時間が最も短いのは image 重ね合わせ方式であることがわかる．4 スレッドを使用したとき，画像統合に要した時間は blend 方式，クロマキー方式，image 重ね合わせ方式のそれぞれについて 55ms，57ms，8ms であり，3 種類の画像統合方式の中で image 重ね合わせ方式が画像統合に要する時間を最も小さくできていることがわかった．

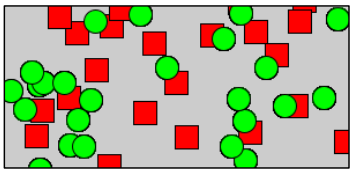
(3) 開発した並列プログラミング環境を用いた授業の実施

本環境を用いて並列プログラミングに関する授業を実施した．この授業の出席者は大阪工業大学の大学院情報科学研究科の大学院生および情報科学部 4 年生の計 42 名であり，授業の実施回数は 1 回（100 分）である．

本授業では並列プログラミングについて解説した．まず，並列計算の概要を説明し，その後，本環境を各自のノート PC に導入させ，並列プログラミングの方法について述べ，その後，1)競



(a) 2スレッドが共通の描画領域に描画



(b) 2スレッドが固有の描画領域に描画し，その後，描画結果を統合

図 3. 競合の実演

表 1. アンケートの質問項目の一覧

質問1	この授業を受講する前に並列処理について勉強したことがありますか？
質問2	質問1で2,3,4を回答した人に質問します．今回の授業では図形のアニメーション描画を題材して並列処理の実演やプログラミングを行いましたが，以前に勉強したときと比べて今回の授業の題材はどう感じましたか？
質問3	今回の授業における「競合」の説明において，Processing言語の図形描画を用いた説明はどうでしたか？
質問4	今回の授業における「競合」の説明において，cnt変数を用いた説明はどうでしたか？
質問5	本授業における惑星運動プログラムにおいて，並列処理によるプログラムの高速化に関する実感はどうでしたか？

質問1

1. 全くない	20
2. 少し勉強したことがある	15
3. 勉強したことがある	5
4. 研究等で実際に活用している	1

質問2(質問1で2,3,4と回答した者のみ)

1. わかりにくかった	0
2. どちらかというとわかりにくかった	0
3. どちらともいえない	2
4. どちらかというとわかりやすかった	9
5. わかりやすかった	10

	質問3	質問4
1. 全くわからなかった	0	0
2. あまりわからなかった	0	0
3. どちらともいえない	1	3
4. 少しわかった	14	15
5. よくわかった	26	23

質問5

1. 全く実感できなかった	0
2. あまり実感できなかった	0
3. どちらともいえない	1
4. 少し実感できた	19
5. しっかり実感できた	21

図 4. アンケートの結果（選択肢と回答者数）

合による不整合，2)並列計算によるプログラムの高速化，の 2 点について解説した．

1)については，複数のスレッドが同時に同一の大域変数を更新した場合に，適切に対応をしなければ意図通りに更新されないことを理解させることを目的としている．本授業では，2 節で示した本環境を用いて 2 つのスレッドがそれぞれ赤四角と緑丸を描画するプログラムを記述して実行させたとき，競合に対して適切に対応した場合は意図通りに赤四角と緑丸が描画されるが，そうでない場合は異なる色や形の図形が描画されることを示した（図 3）．これにより，競合による不具合を視覚的に確認させることができる．また，比較として，図形描画を用いずに複数スレッドが同時に単一の変数の値を更新するプログラムも提示し，競合への未対応による不具合についても示した．2)については，惑星運動シミュレーションの並列プログラムを作成させ，使用スレッド数を変化させることで惑星運動のアニメーションの実行速度の変化を体感させた．

本授業後にアンケートを実施した．得られた回答数は 41 であった．表 1 にアンケートの質問項目の一覧を，図 4 に回答結果を示す．質問 1 および 2 より，並列計算の学習経験者 21 名のうち 19 名は，図形アニメーションを用いた並列計算の学習が「わかりやすい」または「どちらかというとわかりやすい」と回答している．また，質問 3 より，Processing を用いた競合の説明（図 3）について肯定的であることがわかった．ただし，比較のために行った図形描画を用いない競合の説明についても同様の回答であった．質問 5 より，本環境を用いて並列処理にプログラムの高速化も実感できていることがわかった．また，自由記述欄には「図形描画という分かりやすい動きで見れたのは良かった」,「1 講義で並列処理の感覚を掴むにはちょうどいい講義だと思った」といった肯定的な感想も得られた．

５．主な発表論文等

〔雑誌論文〕　計0件

〔学会発表〕　計4件（うち招待講演　0件／うち国際学会　0件）

１．発表者名 水谷泰治，橋本渉，西口敏司
２．発表標題 Processing言語による図形アニメーションを用いた並列プログラミング演習
３．学会等名 情報処理学会第86回全国大会
４．発表年 2024年

１．発表者名 前川翔，水谷泰治，西口敏司，橋本渉
２．発表標題 Processingを用いた学習向け並列プログラミング環境の改善
３．学会等名 電子情報通信学会2022総合大会
４．発表年 2022年

１．発表者名 岸野竜司，笠波将太郎，鈴木優大，西口敏司，橋本渉，水谷泰治
２．発表標題 Processingで視覚的な並列処理を行うためのフレームワークの開発と評価
３．学会等名 教育システム情報学会2019年度JSiSE学生研究発表会
４．発表年 2020年

１．発表者名 水谷泰治， 井垣宏， 尾花将輝，西口敏司，橋本渉
２．発表標題 大阪工業大学情報科学部の初年次Cプログラミング演習におけるBYODのためのプログラミング環境
３．学会等名 情報処理学会第81回全国大会
４．発表年 2019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