

学 位 論 文 要 旨 (Summary of the Doctoral Dissertation)	
学位論文題目 (Dissertation Title)	サーバ支援型通信に基づく並列ボランティアコンピューティングに関する研究
氏 名 (Name)	猪原 圭一
コンピュータの性能向上に伴い，様々な研究分野で科学技術計算が不可欠になっており，取り扱われるデータ量や計算問題の規模が増加している．大規模な計算問題を実行するためのシステムの 1 つに，ボランティアコンピューティング (VC) がある．VC は，インターネットに接続されたパーソナルコンピュータ (PC) に内在する余剰計算資源を提供してもらい，大規模な分散計算を行う手法である．VC には，高性能な分散計算環境を安価に構築できる利点があるため，大規模な計算問題の実行基盤として注目を集めている． VC では，一般ユーザが所有する PC を計算ノード（ワーカ）として使用するため，ワーカ同士が直接通信を行うようには構成されておらず，シャットダウンやネットワーク障害などにより，ワーカが任意のタイミングで計算から離脱するという特性がある．このため，既存の VC では，個々の計算（ジョブ）が独立した分散計算のみが対象とされ，計算中にワーカ間の通信を伴う並列計算は対象外とされてきた．このことが，VC の適用範囲を狭める要因となっており，言い換えれば，VC の環境で並列計算を実行できれば，VC の多様な分野への貢献が期待できる． 並列計算が実行可能な VC（並列 VC）を実現するためには，1) セキュリティを確保しながら，インターネット上でワーカ間通信が可能であること，2) 参加に対する負担を軽減し，参加しやすいこと，3) 離脱が発生する環境で効率よく計算可能であることの 3 つを満たすことが要件となる．1) については，並列 VC には，ワーカ間の通信が必要になるが，VC のノードは，インターネットに接続された一般ユーザの PC であるため，通常，サーバのように外部から通信を受け付けるようにはなっていない．また，PC 同士が直接通信可能な Peer-to-peer (P2P) のように，ワーカがサーバのように外部からの通信を受ける設定が必要にならない．2) については，VC に参加するために追加でソフトウェアをダウンロードとインストールする必要があるれば，ユーザの PC 環境が煩雑になり，さらには，VC への参加の敷居は高くなる．3) については，VC の特徴として，ユーザが PC をシャットダウンするなどにより，計算中に離脱することがある．離脱が発生すれば，通信が行えなくなり，並列計算は終わらない．これらの要件を満たし，並列 VC の実現させるための研究は存在しない． そこで，本研究では，並列 VC の実現を目的として，前述した 3 つの要件を満たす，並列 VC システム，並列 VC クライアント，チェックポイント機能を提案する．1) については，ワーカ間の直接通信を管理サーバを介した通信に置き換えることにより，ワーカ間の通信を可能にする．2) については，ブラウザ上でプログラムを実行可能な WebAssembly を使用することで，ブラウザのみで参加可能な VC を実現する．3) については，ワーカが離脱しても，他のワーカが途中から計算を始めるためのチェックポイント機能を実装することにより可能にする． 本文の構成は，以下のとおりである．	

第 1 章では，緒言であり，本研究の背景と目的について述べる．

第 2 章では，VC の種類と既存手法について述べる．既存の VC の計算モデルと実例を述べ，並列 VC の計算モデルと実現に満たすべき要件を説明する．また，既存手法を挙げ，それらの欠点を述べる．

第 3 章では，サーバ支援型通信に基づく並列 VC システムを提案する．本システムは，2 つのサブシステムから構成されており，まず，ワーカ間の通信を実現するためのサーバ支援型通信システムについて説明する．次に，ジョブの配布・管理を行うジョブ配布システムについて説明する．また，既存の並列プログラムを本システムで実行可能な形式に変換するツールについて説明する．本手法を評価するために，各サブシステムの基本性能の評価と並列計算のベンチマークを使用した実行時間の評価を行う．

第 4 章では，WebAssembly を使用した並列 VC クライアントを提案する．本章では，本手法の概要を述べ，ブラウザから第 3 章で説明したサーバ支援型通信システムと通信するための通信関数の実装について説明する．本手法の性能評価では，並列計算のベンチマークを使用し，本手法と C 言語で作成したクライアントを比較し，実行時間のオーバーヘッドを評価する．

第 5 章では，ソースコードレベルで実現するチェックポイント機能を提案する．本章では，提案するチェックポイント機能の概要を説明し，チェックポイントの保存・復元を実行する関数の実装を説明する．また，チェックポイントで指定された位置から再開する機能について説明する．本手法の性能評価では，チェックポイントの保存/復元にかかる時間と実際の計算において本手法が実行時間に与える影響を評価する．

最後に第 6 章で，サーバ支援型通信を基にした並列 VC について総括し，今後の課題について述べる．

学 位 論 文 要 旨 (Summary of the Doctoral Dissertation)	
学位論文題目 (Dissertation Title)	A Study on Parallel Volunteer Computing Based on Server Assisted Communication
氏 名(Name)	INOHARA Keiichi
As computer performance has been increased, scientific and technical computing has become indispensable across various research fields, leading to an increase in both data volume and the scale of computational problems. One system used to execute large-scale computational tasks is Volunteer Computing (VC). VC is a large-scale distributed computing paradigm, which utilizes the idle computational resources of personal computers (PCs) connected to the internet. Because VC allows for the construction of high-performance distributed environments at a low cost, it has gained significant attention as a platform for large-scale computation. In VC, since computing nodes (workers) is owned by general users serve, the system is not configured for direct communication between workers. Furthermore, workers may leave the computation at any time due to shutdowns or network failures. Consequently, existing VC systems have primarily focused on distributed computing where individual tasks (jobs) are independent, excluding parallel computing that requires communication between workers. This limitation has narrowed the scope of VC applications. In other words, if parallel computing could be executed in a VC environment, it would significantly broaden VC's contribution to diverse fields. To realize Parallel Volunteer Computing (Parallel VC), three key requirements must be met: 1). The communication between workers must be possible over the internet while ensuring security. 2). The system must be easy to join, minimizing the burden of participation. 3). Computation must be efficient even in environments where workers frequently leave. Regarding 1), while Parallel VC requires communication between workers, VC nodes are typically general users PCs that do not accept unsolicited external communications like servers. Furthermore, unlike Peer-to-Peer (P2P) models, workers should not be required to configure their systems to accept such communications. Regarding 2), if users must download and install additional software to participate, the environment becomes cluttered, raising the barrier to entry. Regarding 3), a characteristic of VC is that users may shut down their PCs, causing them to drop out mid-computation. If a dropout occurs, communication is severed, and the parallel calculation cannot be completed. Currently, no research satisfies all these requirements to realize parallel VC.	

Therefore, this study proposes a parallel VC system, a parallel VC client, and a checkpointing function to satisfy these three requirements. Regarding 1), we enable inter-worker communication by replacing direct communication with communication mediated through a management server. Regarding 2), we utilize WebAssembly (Wasm), which allows programs to run in a web browser, enabling participation using only a browser. Regarding 3), we implement a checkpointing function that allows other workers to resume a calculation from an intermediate point if a worker leaves.

This paper is organized as follows.

Chapter 1 serves as the introduction, describing the background and objectives of this research.

Chapter 2 discusses the types of VC and existing methods. It outlines the computational models and practical examples of existing VC, explains the requirements for parallel VC, and identifies the drawbacks of current methods.

Chapter 3 proposes a parallel VC system based on server assisted communication. This system consists of two subsystems: a server-assisted communication system to facilitate communication between workers, and a job distribution system for task management. This chapter also describes a tool for converting existing parallel programs into a format compatible with this system. To evaluate this approach, we assess the basic performance of each subsystem and the execution time using parallel computing benchmarks.

Chapter 4 proposes a parallel VC client using WebAssembly. This chapter provides an overview of the method and details the implementation of communication functions that allow the browser to interact with the server-assisted system described in Chapter 3. The performance evaluation compares this method with a C-based client to assess execution time overhead.

Chapter 5 proposes a source-code-level checkpointing function. We explain the overview of the proposed function and the implementation of functions for saving and restoring checkpoints. Additionally, we describe the mechanism for resuming from specific checkpointed locations. The performance evaluation measures the time required for saving/restoring and the impact on overall execution time.

Chapter 6 concludes the study by summarizing parallel VC based on server assisted communication and discusses future research directions.

(様式 9 号)

学位論文審査の結果及び最終試験の結果報告書

山口大学大学院創成科学研究科

氏 名	猪原 圭一
審 査 委 員	主 査： 福士 将
	副 査： 山口 真悟
	副 査： 爲末 隆弘
	副 査： 井田 悠太
	副 査： 黒川 陽太
論 文 題 目	サーバ支援型通信に基づく並列ボランティアコンピューティングに関する研究 (A Study on Parallel Volunteer Computing Based on Server Assisted Communication)
【論文審査の結果及び最終試験の結果】 本学位論文は、インターネットに接続される多数のコンピュータを用いて大規模な計算を行うボランティアコンピューティング (Volunteer Computing: VC) において、並列計算を実行可能な並列 VC の実現方法を議論したものである。従来の VC では、個々の計算ジョブが独立している分散計算は広く取り扱われているものの、ジョブ間で通信を必要とする並列計算は実行そのものが困難であった。これに対し、本論文では、VC の管理サーバがノード間の通信を中継するサーバ支援型通信のアプローチに基づき、並列 VC を実現するための VC サーバ及び VC クライアント、さらには、並列計算の効率化を図るチェックポイント機能を提案し、その有効性を検証している。 まず、並列 VC 向けのサーバとして、サーバ支援型通信の機能を提供するシステムを提案、実装している。本サーバは、一般的な Web サーバとデータベースサーバで構成され、多数のノードからの同時アクセスの並行処理と通信データ及び制御情報の効率的な管理を容易に可能にするものである。本サーバを用いた評価により、並列計算における基本的な一対一通信及び集団通信の性能特性を明らかにし、その有効性を示している。また、実用的な流体計算を含むいくつかの並列計算のベンチマークを用いて並列 VC の実現可能性を実証し、従来の VC に対する並列 VC の優位性を示している。 次に、並列 VC 向けのクライアントとして、多くのコンピュータに標準的に搭載されている Web ブラウザを用いて VC に参加可能なクライアントを提案、実装している。本クライアントは、WebAssembly 技術とブラウザ標準の通信インターフェースを用いることで、既存の並列プログラムを Web ブラウザ上で実行可能にするものである。本クライアントを用いた評価により、従来のクライアントとの計算・通信性能の差異を明らかにし、異なるオペレーティングシステム (Operating System: OS) のノードが混在する環境において、並列計算が実行可能なことを実証している。	

さらに、ノードの離脱が発生する VC において、並列計算の効率化を図るためのチェックポイント機能を提案している。本手法は、プログラムのソースコードにデータの保存・復元、再開処理機能を自動的に埋め込むことで、プログラム開発者に負担をかけることなく、異なる OS のノード間で計算の引継ぎを可能にするものである。性能評価により、プログラムで一般的に用いられる配列・ポインタ・構造体変数に対するデータ量と保存・復元時間の関係を明らかにし、並列計算に本手法を適用した場合の効果を明らかにしている。

本論文のアプローチであるサーバ支援型通信は、これまでの VC の研究ではあまり見られない革新的なものである。本アプローチに基づいて提案したシステムおよび手法は、これまで困難であった並列計算を容易に効率よく実行可能にするものであり、本論文で得られた成果はその有効性を裏付けるものである。

公聴会における主な質問内容は、サーバ支援型通信の制御方法に関するもの、評価で用いたベンチマークに関するもの、チェックポイントにおけるポインタ変数の保存方法とオーバーヘッドに関するもの、本研究における評価環境とインターネットにおける実環境の違いに関するものなどであった。いずれの質問に対しても、発表者からの的確な回答がなされた。

以上より、本研究は、独創性、信頼性、有効性、実用性ともに優れ、博士（工学）の論文に十分値するものと判断した。

論文内容及び審査会、公聴会での質問に対する応答などから、最終試験は合格とした。

なお、主要な関連論文の発表状況は下記のとおりである。（関連論文 計 5 編、参考論文 計 2 編）

- 1) Keiichi Inohara, Yota Kurokawa, Masaru Fukushi, "A Parallel Volunteer Computing System Based on Server Assisted Communication," *IEEJ Transactions on Electrical and Electronic Engineering*, Vol. 20, Issue 9, pp. 1415–1423, 2025.
- 2) Keiichi Inohara, Sun Zhiding, Yota Kurokawa, Masaru Fukushi, "A Browser-based Parallel VC Client using WebAssembly," *Proceedings of The Thirty-first International Symposium on Artificial Life and Robotics 2026*, pp. 1260–1264, 2026.
- 3) Keiichi Inohara, Masaru Fukushi, "An implementation of checkpoint function for the realization of parallel volunteer computing," *Proceedings of The Twenty-Seventh International Symposium on Artificial Life and Robotics 2022*, pp. 1068–1072, 2022.