

## 修士論文

ボランティアコンピューティング資源活用によるクラウ  
ドゲーミングのQoE向上に関する研究

前田 健登

2021年1月25日

奈良先端科学技術大学院大学  
先端科学技術研究科 情報科学領域

本論文は奈良先端科学技術大学院大学先端科学技術研究科情報科学領域に  
修士(工学)授与の要件として提出した修士論文である。

前田 健登

審査委員 :

飯田 元 教授 (主指導教員)  
藤川 和利 教授 (副指導教員)  
市川 春平 准教授 (副指導教員)  
高橋 慧智 助教 (副指導教員)

# ボランティアコンピューティング資源活用によるクラウドゲーミングのQoE向上に関する研究\*

前田 健登

內容梗概

人類がこの地上に現われて以来、 $\pi$  の計算には多くの関心が払はれてきた。

本論文では、太陽と月を利用して  $\pi$  を低速に計算するための画期的なアルゴリズムを与える。

ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。  
ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。  
ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。  
ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。  
ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。

ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。  
ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。  
ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。  
ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。  
ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。

## キーワード

ネットワーク、クラウド、クラウドゲーミング、ボランティアコンピューティング

\*奈良先端科学技術大学院大学 先端科学技術研究科 情報科学領域 修士論文, 2021年1月25日.

# **Research About QoE Approaving of Cloud Gaming Using Resouces of Volunteer Computing\***

Kento Maeda

## **Abstract**

The calculation of  $\pi$  has been paid much attention since human beings appeared on the earth.

This thesis presents novel low-speed algorithms to calculate  $\pi$  utilizing the sun and the moon.

This is a sample abstract. This is a sample abstract. This is a sample abstract.  
This is a sample abstract. This is a sample abstract. This is a sample abstract.  
This is a sample abstract. This is a sample abstract. This is a sample abstract.  
This is a sample abstract.

This is a sample abstract. This is a sample abstract. This is a sample abstract.  
This is a sample abstract. This is a sample abstract. This is a sample abstract.  
This is a sample abstract. This is a sample abstract. This is a sample abstract.  
This is a sample abstract.

## **Keywords:**

network, cloud, cloud gaming, volunteer computing

---

\*Master's Thesis, Division of Information Science, Graduate School of Science and Technology, Nara Institute of Science and Technology, January 25, 2021.

# 目 次

1. はじめに	1
2. 背景	6
2.1 クラウドゲーミング	6
2.2 ボランティアコンピューティング	8
3. 設計	9
3.1 提案システムの概要	9
3.2 実装上の課題	10
3.3 構成コンポーネント	11
3.3.1 VC コントローラ	11
3.3.2 VC クライアントエージェント	11
3.3.3 VC ホストエージェント	11
3.3.4 クラウドゲームサーバ/クライアント	12
4. 実装	13
4.1 VC コントローラとエージェントの連携	13
4.2 クラウドゲームサーバ/クライアント間のP2P通信	13
4.3 システム動作	15
5. 評価	17
5.1 評価環境	17
5.2 クラウドゲームサーバ/クライアント間の通信性能	18
5.2.1 リンクに対する生の遅延の大小の影響	18
5.2.2 リンクに対する遅延の大小のスループットへの影響	18
5.3 ゲームプレイ時のフレームレート	20
5.3.1 ネットワーク帯域の大小の影響	20
5.4 考察	21
6. まとめと今後の課題	27

謝辞 28

参考文献 29

## 図 目 次

1	クラウドゲーミングシステム	2
2	提案システムの概要	10
3	gRPC の概要	14
4	システム動作	16
5	評価環境	18
6	EdgeVPN リンクに対する遅延挿入の影響	19
7	EdgeVPN リンクへの遅延挿入の帯域への影響	20
8	EdgeVPN を使用していないリンクへの遅延挿入の帯域への影響	21
9	提案システムを使用した Albion Online (MMORPG) のプレイの様子	22
10	提案システムを使用した Red Ecliplse 2 (FPS, Action) のプレイの 様子	23
11	提案システムを使用した Simply Chess (ボードゲーム) のプレイの 様子	24
12	帯域制限下でのゲームプレイ時のフレームレートの変化 (Albion Online (MMORPG) プレイ時)	24
13	帯域制限下でのゲームプレイ時のフレームレートの変化 (Red Ecli- plse 2 (FPS, Action) プレイ時)	25
14	帯域制限下でのゲームプレイ時のフレームレートの変化 (Simply Chess (ボードゲーム) プレイ時)	26

## 表 目 次

## 1. はじめに

ゲーム産業は娯楽産業の中でも大きな収益を上げている産業であり、2019年の世界ゲームコンテンツ市場の規模は15兆6898億円と推定されている[23]。国内でも10年連続で成長しており、市場規模は1兆7330億円となっている。その中でも家庭用ハードや家庭用ソフトと比較して、スマートフォンのアプリやPC向けのオンラインゲームなどといったオンラインプラットフォームのゲームの市場規模が年々大きく拡大している。このようにゲームのプレイスタイルがゲーム専用のハードを購入するというものから汎用デバイスでのゲームプレイへと変化を見せており、こういった状況下で注目されている新たな手法で展開されるゲームサービスの一つにクラウドゲーミングがある。

従来のゲームプレイは、プレイヤーがゲーム専用ハードやゲーミングPC等を所有し、その上でゲームを動作させることによって実現されている。一方、クラウドゲーミングというサービスにおいては、図1のようにクラウドサーバ上でゲームを動作させてその画面をクライアントであるプレイヤーの端末にストリーミングすることで、ゲームをネットワーク越しにプレイすることを可能にしている。このとき、プレイヤーがゲームプレイに使用する端末は、クラウドサーバより送信されるゲーム画面の再生とプレイヤーの操作のサーバへの送信だけを行う。この仕組みによって、スマートフォンやタブレット端末等の性能が貧弱なデバイスでも、従来は高価なゲームハードやゲーミングPCを使用しなければ体験できなかつたような高品質なゲーム体験を得られることが期待されている。

クラウドゲーミングは商用のサービスの展開もある。過去には英国のOnLive[1]やアメリカのGaikaiがクラウドゲーミングサービスを展開していた。これらは既にサービスを終了しているが、ソニーが2012年にGaikaiを買収し、2015年にサービスを閉鎖したOnLiveの資産を取得した[2]。同年に、ソニーは新たにクラウドゲーミングサービスのPlayStation Now[11]を開始した。また、Googleも2019年にクラウドゲーミングサービスであるGoolge Stadia[13]を開始した。同社のウェブブラウザであるGoogle Chromeをインターフェースとしているのが特徴で、ユーザーへのメディアのストリーミングに動画配信サービスのYouTubeを使用している。現在は日本を含まない14カ国で展開されている。また、グラフィッ

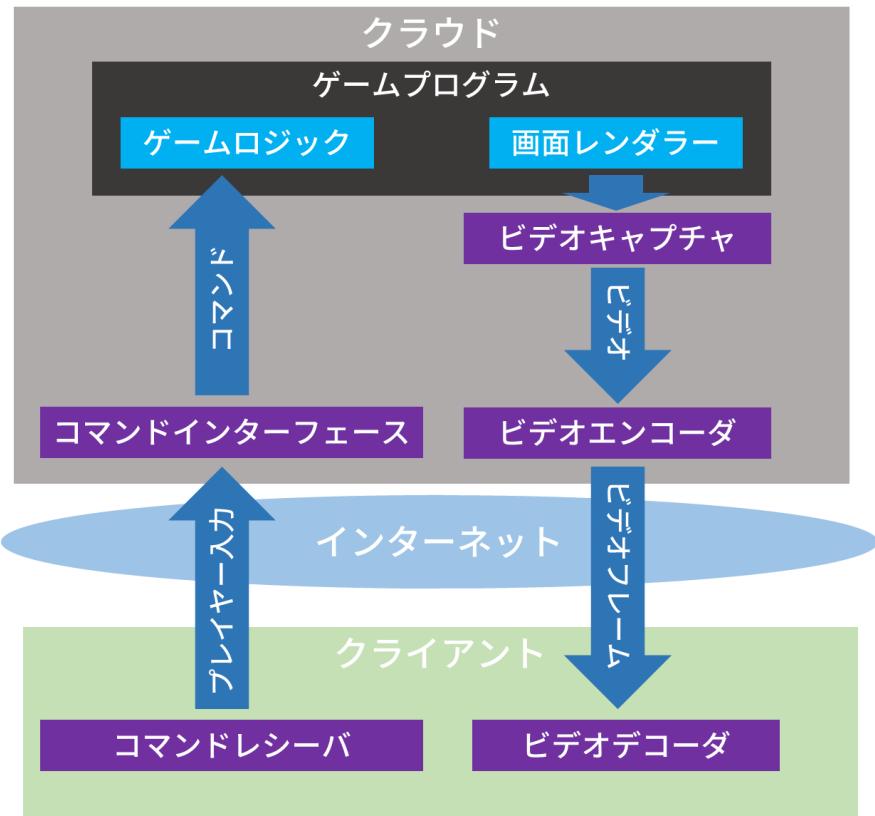


図 1 クラウドゲーミングシステム

クプロセッシングユニットのメーカーとして知られる NVIDIA が提供するクラウドゲーミングサービス GeForce NOW[5] は、従来 PC ゲームをプレイしていたプレイヤーをメインターゲットに据えており注目されている。他に、Microsoft の Project xCloud や Amazon の Luna が海外でサービス開始されている。

商用サービスだけでなく、研究開発用のクラウドゲーミングプラットフォームも存在する。Huang ら [20] は、既存のクローズドソースのシステムではクラウドゲーミングを体験するためのテストベッドの設置が困難であることから、オープンソースのクラウドゲーミングプラットフォームである GamingAnywhere を開発した。GamingAnywhere は Windows、Linux、OS X 上で実装されており、クライアントは iOS や Android 等の他の OS にも移植が可能である。また、GamingAnywhere は詳細な設定を可能にしていて、更にオープンソースであるため拡張的な実装が

可能であるなど、クラウドゲーミングのシステム研究のテストベッドの構築に適している。

従来のゲームシステムとは一線を画するクラウドゲーミングには、依然として重要な課題が存在している[17]。クラウドゲーミングプロバイダの立場から見た課題としては、ゲーム環境の仮想化やサーバにおける負荷分散などといった課題がある。一方、システムのユーザーであるプレイヤーが知覚するゲーム体験の品質の評価指標である Quality of Experience(QoE) を確保することも重要な課題である。クラウドゲーミングのプレイにおける QoE の確保に必要な課題の構成要素としては、以下のようなものがある。

- ストリーミングされるゲーム画面の高品質な画質の担保
- ゲーム画面のリアルタイムストリーミングに耐えうる、充分なネットワーク帯域の確保
- 伝送データ圧縮や効率的なストリーミング技術
- プレイヤー端末での画面表示やプレイヤーによる操作が画面に反映されるまでの遅延の最小化

中でもプレイヤーの操作に対する応答性に直結する遅延の問題は重要である。Raaen[25] らの調査によると、一部のゲーマーは 40ms 未満のゲームの応答遅延を認識可能であり、また半数のゲーマーが 100ms 以上の応答遅延を許容できないということが観測されている。動画配信プラットフォームにおけるライブストリーミングの場合、途切れることなく安定した動画の配信を行うためにバッファリングを行うことで対処する場合がある。しかしクラウドゲーミングは多くの場合、リアルタイムかつインタラクティブな性質を持つコンテンツであるためこの方法を使用することができない。そのため、クラウドサーバ上でのゲーム画面生成の高速化/効率化や、伝送データの圧縮、ネットワーク遅延の最小化などが課題となっている。

クラウドのデータセンターは、通常国内には数箇所しか設置されていない。このため、データセンターに地理的に遠いプレイヤーの端末から接続するとネット

ワーク遅延が大きくなってしまうという問題がある。データセンターに計算機資源を集約する現在のクラウドアーキテクチャを利用する限り、この遅延を解消することはできない。プレイヤーとクラウド上の計算機資源との間の物理的距離に起因する遅延の問題解決するためには、プレイヤーの近傍に計算機資源を設置することが必要である。

ところで、多量のタスクを小規模なタスクに分割して、ボランティアの提供する多数のコンピュータリソースに広域に分散して処理するボランティアコンピューティングという枠組みがある。SETI@home[15] や Folding@home[21] に代表されるボランティアコンピューティングプロジェクトが、高エネルギー物理学、分子生物学、医学、天体物理学、気候研究などの分野の研究で利用された。クラウドコンピューティングの枠組みにおいては、クラウドサーバに一極集中したリソースにユーザがアクセスして利用する。それに対し、ボランティアコンピューティングにおいては、広域に分散した一般ユーザーのリソースを活用するという特徴がある。国内に数箇所しか設置されていないクラウドのデータセンターに比べて、広域に分散した一般ユーザーのリソースは地理的に近傍である可能性が高い。この性質を利用し、よりネットワーク遅延の小さい計算機同士での通信でクラウドゲーミングを行えば、遅延の問題を解消する可能性がある。

本研究では、ボランティアが提供する地理的に近傍の遊休コンピュータのリソースを活用することによるクラウドゲーミングシステムを提案する。既存のクラウドゲーミングアーキテクチャにおいては、クラウドのデータセンターでゲームが動作しているために、プレイヤーの端末からデータセンターまでのネットワーク遅延を回避することは不可能である。国内のデータセンターまでのネットワーク遅延は最大 50ms 程度と言われている（出典がない）。これはゲームプレイにおけるレスポンスの遅延としては無視できない値であり、著しくプレイヤーの QoE 低下の原因となり得る。本研究では、プレイヤーの端末からデータセンターまでのネットワーク遅延を削減し、プレイヤーがクラウドゲーミングのプレイを通して体験する画面表示や操作の反映の遅延を最小化することでのプレイヤーの QoE の向上を目的とする。広域に分散したボランティアの提供する遊休コンピュータの中から、プレイヤーの端末から見て地理的に近傍のものを選択し、その上でクラ

ウドゲームサーバを動作させる。これによって遅延の削減を目指す。

本論文の以後の部分は次のように構成されている。2章では、研究分野の背景、関連研究について述べる。3章では、提案システムの設計について述べる。4章では、提案システムの実装について述べる。5章では、提案システムの性能評価について述べる。6章では、まとめと今後の展望について述べる。

## 2. 背景

本章では、研究の背景および既存の研究について述べる。

### 2.1 クラウドゲーミング

クラウドゲーミングシステムは OnLive や Gaikai といった商用サービスから始まったが、それらのクローズドシステムではゲームプレイ体験や可用性、スケーラビリティといった要素のテストベッドとして活用することは困難である。Huang ら [20] は、研究開発に利用できる高い拡張性、移植性を備えたオープンソースのクラウドゲーミングプラットフォームである GamingAnywhere を開発した。GamingAnywhere は以下の 3 つの設計目標に沿って設計されている。

- オープンソースのシステムであり、ビデオストリーミングなどのコンポーネントを、異なるアルゴリズムや規格、プロトコルで実装した別のコンポーネントに容易に置き換えることができる
- クロスプラットフォームであり、Windows、Linux、OS X で利用可能である
- メモリコピー回数を減らす等の時間的/空間的なオーバーヘッドを最小化し、効率的に動作する

GamingAnywhere はこれらの実験を通して実装された。大規模な実験を通して商用クラウドゲーミングシステムである OnLive と StreamMyGame を大幅に凌駕することが示された。GamingAnywhere と比較すると、OnLive と StreamMyGame は内部における処理遅延が最大で 3 倍および 10 倍になり、またビデオ品質についてはそれぞれ 3db および 19db 低くなるという結果が示されている。また、GamingAnywhere はクラウドゲーム開発者やシステム研究者に公開され、クラウドゲーミングシステムのテストベッドとして使用することができる。

Shea ら [27] は、OnLive を対象としてクラウドゲーミングプラットフォームのパフォーマンスを測定した。その結果、ストリーミング品質およびインタラクションの遅延が、クラウドゲーミングを構成するゲーム、計算機、ネットワークのす

べての部分で課題を抱えていることを指摘した。また、携帯電話のネットワークにおいて 200ms を超えるネットワーク遅延の回線は珍しくなく、これにより多くのゲームにおいてインタラクション遅延が高くなりすぎる可能性について指摘した。さらに、ゲーム画面のレンダリングとゲームロジックの一部をプレイヤー端末のローカルで実行することにより、インタラクションの遅延などのクラウドゲーミングの課題の一部を隠蔽するシステムの可能性についても言及している。

クラウドゲーミングが抱える遅延の課題について、Lee ら [22] は、将来起こりうる画面フレームを投機的にレンダリングする Outatime を提案した。Outatime は以下の処理を組み合わせて構成されている。

- 将来のゲームの状態予測
- 画像ベースレンダリングとゲームイベントのタイムシフトによる状態近似
- チェックポイントとロールバックの機構による予測の誤りが前方に伝播することの防止
- 帯域幅の節約のための状態圧縮

投機的にレンダリングされた画面フレームは一つの RTT に事前にプレイヤーの端末に配信され、予測と異なる状態に遷移した際に迅速に回復する。これにより、クラウドゲーミングで発生するネットワーク遅延を最大 120ms 隠蔽できると述べている。一方で、ゲームのステージ内をワームホールでテレポートするようなイベントについては、この手法では対応できないということも指摘している。

クラウドゲーミングで発生する遅延のうちでも大きな要因であるネットワーク遅延の課題に対し、Hong ら [19] はクラウドゲームサーバとして動作する VM 配置の最適化によって解決を試みた。プレイヤーの QoE とプロバイダの純利益との間のトレードオフであるクラウドゲーミングの最適化を行うためのヒューリスティックアルゴリズムを開発した。この中で、クラウドゲームサーバの役割を果たす VM を、プレイヤーの端末から最もネットワーク遅延が小さい位置にあるサーバに割り当てるという最適化を行っている。また、開発したアルゴリズムを評価するために大規模なシミュレーションを実施しており、結果としてアルゴリズムが以下のことを示している。

- 最適に近いソリューションが得られる
- 20000 台のサーバと 40000 人のプレイヤーを抱える大規模なクラウドゲーミングサービスへの拡張が可能である
- 最先端の既存ヒューリスティックアルゴリズムを上回る結果が得られること

一方、この手法においてもプレイヤー端末からデータセンターまでのネットワーク遅延を回避できないという課題の解決には至っていない。

## 2.2 ボランティアコンピューティング

ボランティアコンピューティングは多量のタスクを小規模なタスクに分割して、ボランティアの提供する多数のコンピューティングリソースに分散して処理する枠組みである。ボランティアのリソースで動作するクライアントは、定期的にサーバと通信を行い、完了したタスクの報告と新しいタスクの取得を行う。サーバがタスクをディスパッチする速度によって、ボランティアコンピューティングプロジェクトで使用できる計算能力が制限される可能性がある。Anderson ら [14] は、ボランティアコンピューティング用のミドルウェアシステムである Berkeley Open Infrastructure for Network Computing (BOINC) を開発した。BOINC のタスクサーバの様々なコンポーネントが使用する CPU 時間とディスク帯域幅の測定が行われ、BOINC は 1 日に 8880 万個のタスクをディスパッチできると結論付けられた。各クライアントが 1 日に 1 つのタスクを発行し、各タスクが 1GFLOPS のコンピュータで 12 時間 CPU を使用する場合、ボランティアコンピューティングプロジェクトが 880 万のクライアントをサポートし、4.4PetaFLOPS の計算能力を得ることができると述べられている。

### 3. 設計

本章では、従来のクラウドゲーミングにおいてクラウドのデータセンターとプレイヤー端末間の遅延を避けることができないという課題を解決するための、ボランティアコンピューティング資源を活用するクラウドゲーミングシステムを提案する。まず提案システムの概要を述べた後、システムを実装するに当たっての課題について述べる。最後にシステムを構成するコンポーネントとその役割について述べる。

#### 3.1 提案システムの概要

提案システムの概要を図2に示す。提案システムは、クラウドのデータセンターに比べてより近いボランティアが提供する遊休コンピュータでクラウドゲームサーバをホストする。それにより、プレイヤーがクラウドゲーミングのプレイを通して体験する遅延を削減ということを目的とするものである。

システムの構成要素として、プレイヤーPC、クラウド上のボランティアクラウドゲーミングコントローラ、およびボランティアが提供する遊休コンピュータの3つのハードウェアがある。プレイヤーPCは、クラウドゲーミングをプレイするプレイヤーの所有するPCである。遊休コンピュータはボランティアが所有しているコンピュータの、一時的に使用していないコンピュータリソースを貸与している状態のものを指す。ボランティアクラウドゲーミングコントローラはプレイヤーPCと遊休コンピュータのマッチングを行う。

それぞれのハードウェアで動作するソフトウェアの構成要素について述べる。プレイヤーPCで動作するボランティアクラウド(VC)クライアントエージェントは、プレイヤーの希望に応じてボランティアクラウドゲーミングコントローラにゲームプレイを要求する。ボランティアクラウドゲーミングコントローラ上で動作するVCコントローラは、プレイヤーPCから要求を受け取ると遊休コンピュータとのマッチングを行う。遊休コンピュータ上で動作するVCホストエージェントは、VCコントローラからのクラウドゲームの実行要求に応じてクラウドゲームサーバの起動を行う。クラウドゲームサーバとクラウドゲームクライアントは、

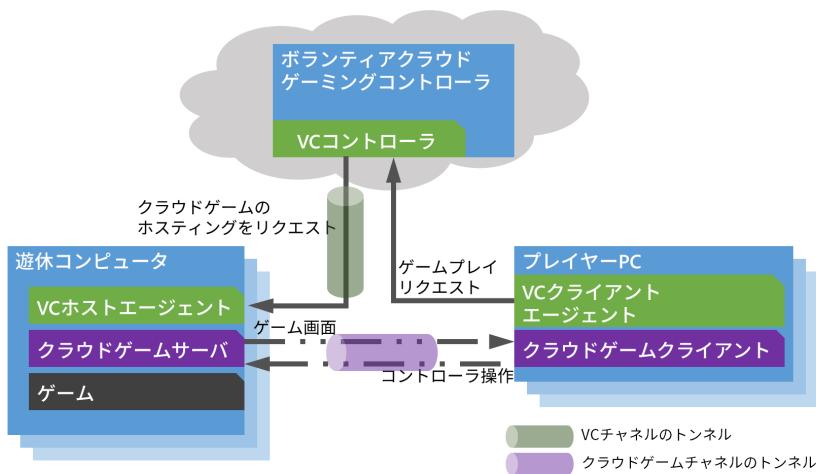


図 2 提案システムの概要

遊休コンピュータとプレイヤー PC を直接接続してクラウドゲーミングのプレイを実現する。

### 3.2 実装上の課題

パブリッククラウドやビジネス向けのサーバでクラウドゲーミングサービスを提供する場合と異なり、提案システムはクラウドゲームサーバをボランティアの提供するユーザコンピュータ上で実行する。しかしユーザコンピュータは NAT やファイアウォールの背後にあり、また固定 IP アドレスを有しないことが一般的であるため遊休コンピュータとプレイヤー PC が直接通信することが困難である。そのため、提案システムの実装にあたり次の 2 点の課題が生じる。

- クラウド上のコントローラから遊休コンピュータへクラウドゲームの実行等の直接命令を送ることができない
- クラウドゲームサーバ/クライアントを展開する遊休コンピュータとプレイヤー PC 間での通信において、双方向的な直接通信を展開できない

### 3.3 構成コンポーネント

本項では各構成要素の詳細を述べる。

#### 3.3.1 VC コントローラ

クラウド上に配備する VC コントローラはゲームをプレイしたいプレイヤーの PC と利用可能な遊休コンピュータをマッチングする役割を果たす。主な機能は以下の通りである。

- プレイヤーからのゲームプレイ要求を受け付ける
- 最適な遊休コンピュータを発見して割り当てる
- プレイヤー PC と遊休コンピュータが通信を確立するための接続情報を配布する

マッチングの要件として、プレイヤー PC と遊休コンピュータが接続する際のネットワーク遅延が小さいことがある。そのため、VC コントローラはプレイヤーからのゲームプレイ要求に基づき、利用可能な遊休コンピュータの中から最も遅延の小さくなるものを探す。

#### 3.3.2 VC クライアントエージェント

VC クライアントエージェントは、プレイヤーの希望に応じてボランティアクラウドゲーミングコントローラにゲームプレイの要求をする役割を持つ。プレイしたいゲームやサーバとリンクを張るために必要なネットワーク情報などを付帯して、VC コントローラへゲームプレイの要求を送信する。

#### 3.3.3 VC ホストエージェント

VC ホストエージェントは VC コントローラの実行要求に応じてクラウドゲームサーバの起動を行う役割を果たす。また、プレイヤーが実際にプレイするゲー

ムの起動も行う。起動が完了すると、接続に必要な情報と共に VC コントローラを経由して VC クライアントエージェントに起動完了の通知を送信する。

### 3.3.4 クラウドゲームサーバ/クライアント

クラウドゲームサーバ/クライアントは遊休コンピュータとプレイヤーPCを接続して、クラウドゲーミングのゲームプレイを提供する役割を果たす。クラウドゲームサーバはゲーム画面をキャプチャして、ビデオストリームとしてクラウドゲームクライアントに送信する。一方で、クラウドゲームクライアントは受け取ったゲーム画面の描画を行う。また、クラウドゲームクライアントはプレイヤーの入力操作をキャプチャしてクラウドゲームサーバ上のゲームの入力となるように送信を行う。

## 4. 実装

本章では提案システムの実装について述べる。まず3章で述べた設計を実装する際に生じる課題の解決方法について述べた後、システムの動作について述べる。

### 4.1 VC コントローラとエージェントの連携

VC コントローラと遊休コンピュータ上で動作する VC ホストエージェントとの通信の課題については gRPC[18] を用いる。gRPC は Google が開発しているオープンソースの RPC フレームワークで、異なるコンピュータ間情報をやり取りするため使用される。gRPC ではクライアントアプリケーションがローカルで実装されたメソッドを使用するかのようにサーバアプリケーションのメソッドを直接呼び出すことができるため、分散アプリケーション等の実装に適している(図3)。サーバ側ではサービスを定義してそのインターフェースを実装する。クライアント側ではサーバと同じメソッドを提供するスタブを介してサーバアプリケーションの機能を使えるようにしているのが特徴である。

gRPC の Response Streaming gRPC という機能は、単一の要求に対して複数のレスポンスを任意のタイミングで返すことが可能である。これを使用することで、VC クライアントが送信した単一のゲームプレイ要求に対して、ACK や起動報告、実行終了時の完了報告、エラーの通知など様々なレスポンスを返すことができる。

### 4.2 クラウドゲームサーバ/クライアント間の P2P 通信

実際にリモートでのゲームプレイを実現するクラウドゲームサーバとクラウドゲームクライアントには GamingAnywhere を使用する。遊休コンピュータ上に GamingAnywhere サーバ、プレイヤー PC 上に GamingAnywhere クライアントを起動し、GamingAnywhere サーバが展開する RTSP サーバにクライアントが接続することでクラウドゲームのプレイが開始される。

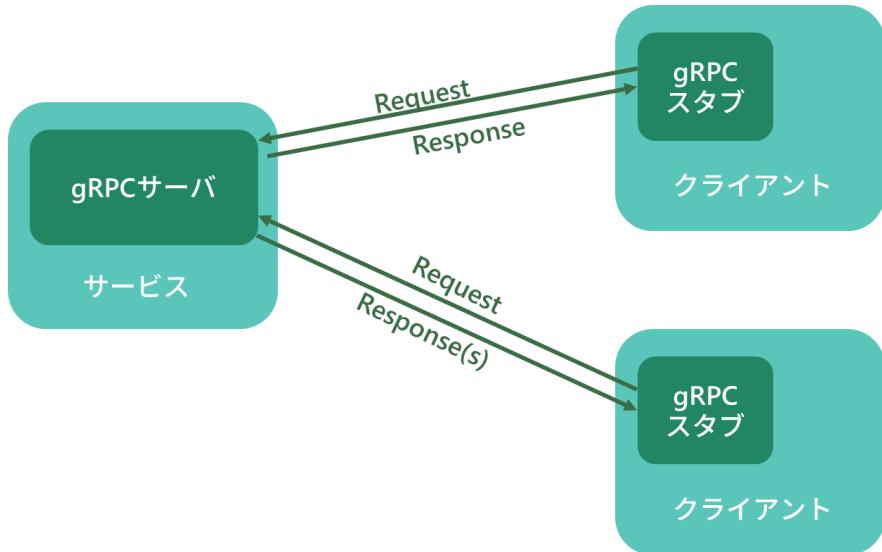


図 3 gRPC の概要

ここで、ユーザコンピュータで動作する GamingAnywhere サーバ/クライアント間で双方向的な直接通信を行えないという問題がある。この問題の解決策として、遊休コンピュータおよびプレイヤーPCのファイアウォールを解除する、あるいは特定の通信を許可する設定をするという方法がある。しかし、これはセキュリティ上の危険や、煩雑な設定をユーザに強いるという課題がある。そこで、本研究では GamingAnywhere の通信を行うリンクに対し EdgeVPN[10] を使用する。EdgeVPN は、分散コンピューティング環境のエッジに存在するリソース間の通信にスケーラブルな VPN オーバレイを展開するためのオープンソースソフトウェアである。EdgeVPN は IP-over-P2P(IPOP) プロジェクト [29] の進化版である。IPOP は、個人の端末を対象とした IP ベースの P2P オーバレイであり、一元化されたユーザー/グループ管理をサポートしている。EdgeVPN を使用することで、エッジデバイス同士が、NAT/ファイアウォール、およびクラウドコンピューティングリソースの背後にあるネットワークアドレスに透過的に接続し、インターネットを介したトラフィックを P2P で暗号化およびトンネリングすることができる。また、EdgeVPN は XMPP プロトコル [26] を使用してピアとの接続情報を検出および交換する。パケット交換とルーティングは分散されているため、スケーラブル

ルな P2P オーバレイを展開しつつ、メンバーシップの一元管理も可能である。

EdgeVPN はピア間でカプセル化/暗号化されたイーサネットフレームを伝送する仮想リンクの構築に、Juste ら [28] が開発した TinCan を使用している。TinCan は、XMPP サーバを使用してエンドツーエンドの VPN トンネルをブートストラップすることにより、コントローラとデータパスを分離するモデルを実現している。また、ノードが接続先に自身のパブリック IP やポートを知らせるためのリフレクションサービスとして STUN プロトコル [30] を使用する。一方、制限の強いファイアウォールや Symmetric NAT の背後にあり直接 P2P 接続を構築できないノードがある場合はリレーサービスとして TURN プロトコル [24] を実行するサーバによりトラフィックをプロキシしている。

### 4.3 システム動作

本節ではシステムの動作のシーケンスについて述べる。まず始めにクラウド上に存在する XMPP サーバを使用して、プレイヤー PC と遊休コンピュータが gRPC および EdgeVPN のリンクを張るための接続情報をそれぞれに通知する(図 4 矢印 1)。次に、プレイヤーの希望に応じて、プレイヤー PC 上の VC クライアントエージェントが gRPC クライアントとして、遊休コンピュータで動作する VC ホストエージェント上の gRPC サーバにゲーム起動要求を送信する(矢印 2)。これを受信した VC ホストエージェントは、クラウドゲームサーバの役割を果たす GamingAnywhere サーバとプレイヤーが指定した所望のゲームを起動し(矢印 3)、GamingAnywhere サーバがゲーム画面のキャプチャを開始した後(矢印 4)、了解を返す(矢印 5)。その後、遊休コンピュータとプレイヤー PC 間に EdgeVPN のリンクを張り、プレイヤー PC が GamingAnywhere クライアントを起動して、トンネルを介して GamingAnywhere サーバに接続することでゲームプレイを開始する(矢印 6)。また、ゲームプレイ終了時は GamingAnywhere クライアントの終了によって接続の切断が GamingAnywhere サーバに通知されるため、それを観測することで VC クライアントエージェントは終了を検知し、GamingAnywhere とゲームを終了する(矢印 7)。

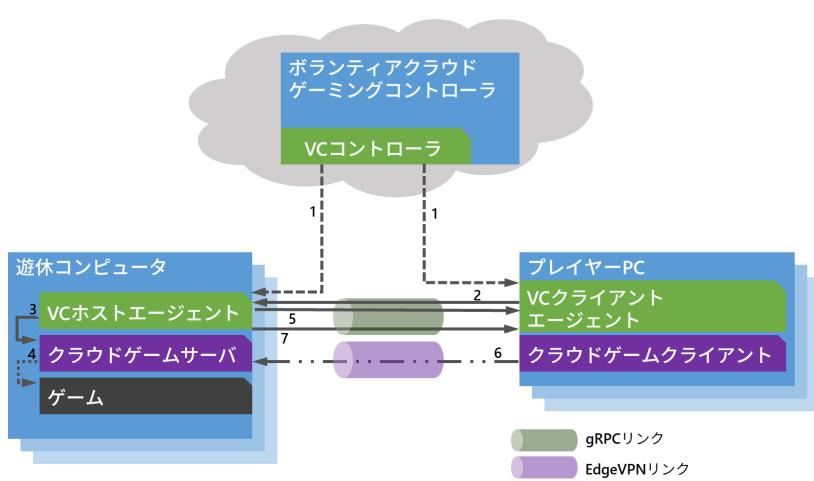


図 4 システム動作

## 5. 評価

本章では、提案システムの性能評価について述べる。本評価ではボランティアコンピューティングを活用したクラウドゲーミングシステムの実現可能性を評価することを目的とする。また、提案システムが実現可能であるときそれが有用な条件は、プレイヤーPCからデータセンターよりもボランティアの提供する遊休コンピュータに接続する方がネットワーク遅延が小さい状況である。提案システムの実現可能性を評価するため、通信性能の評価および、ゲームプレイにおけるQoEを測定するためにフレームレートの評価も行う。

まず、評価を行う環境について述べる。次に、クラウドゲームサーバ/クライアント間の通信性能の評価について述べる。その後、ゲームプレイ時のフレームレートの評価について述べる。最後に評価についての考察を述べる。

### 5.1 評価環境

既存のクラウドゲーミングシステムはクラウドのデータセンター上でクラウドゲームサーバが動作している。これに対し、提案システムはボランティアが提供する遊休コンピュータ上でクラウドゲームサーバを動作させることで、プレイヤーからデータセンターまでの遅延を削減することを目指した。一般にユーザコンピュータからデータセンターへの遅延は大きくても50ms程度であるが、提案システムでのゲームプレイ中に発生する遅延がこの基準を下回るかどうかを評価する。また、提案システムの通信を実現するために組み込んだトンネリングのオーバヘッドについても評価を行う。

評価を行う環境を図5のように構築した。ボランティアクラウドゲーミングコントローラは用意せず、クラウドサーバ上にはEdgeVPNのリンクを確立するために必要なXMPPサーバのみを用意した。プレイヤーPCの役割を果たすsiciliaではgRPCクライアントとGamingAnywhereクライアントが動作する。また、遊休コンピュータの役割を果たすfirenzeではgRPCサーバ、GamingAnywhereサーバおよびゲームが動作する。siciliaとfirenzeはそれぞれUbuntu20.04で動作するマシンを用い、1Gbpsのリンクを持つネットワークで接続した。このリンクに遅

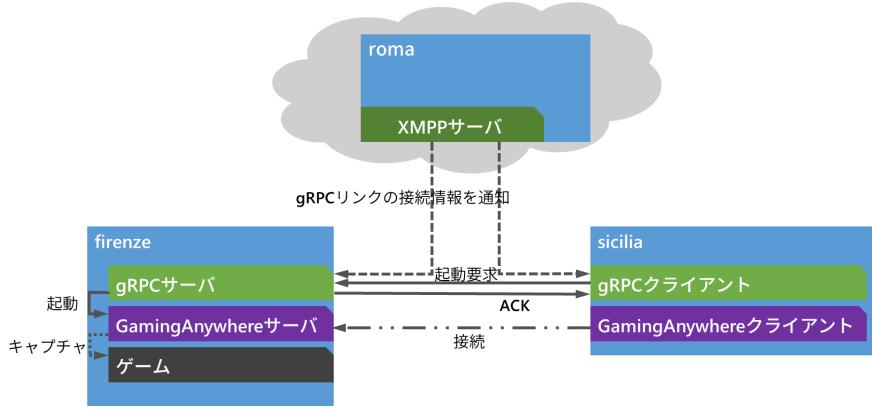


図 5 評価環境

延挿入や帯域制限をかけることで様々なネットワーク環境での評価を実現する。

## 5.2 クラウドゲームサーバ/クライアント間の通信性能

### 5.2.1 リンクに対する生の遅延の大小の影響

sicilia と firenze の間のネットワーク遅延の大小によって、GamingAnywhere サーバ/クライアント間のリンクを張るために使用している EdgeVPN のオーバーヘッドがリンクの遅延に与える影響について調査した。tc[7] を用いてリンクに 0-60ms の遅延を挿入し、EdgeVPN を利用する場合と直接接続する場合の遅延増加の様子 ping[16] を用いて計測した。計測遅延と挿入遅延の値の差をプロットしたものが図 6 である。計測値には ping を 10 回実行した際の値の平均値を使用している。

直接通信に比べて、EdgeVPN を利用する場合は平均して 1ms 程度遅延のオーバーヘッドが存在することがわかる。また、遅延のオーバーヘッドは挿入遅延の大きさに関わらずほぼ一定であるといえる。

### 5.2.2 リンクに対する遅延の大小のスループットへの影響

sicilia と firenze の間のネットワーク遅延の大小の影響で、GamingAnywhere サーバ/クライアント間のリンクで展開した EdgeVPN がリンクのネットワークス

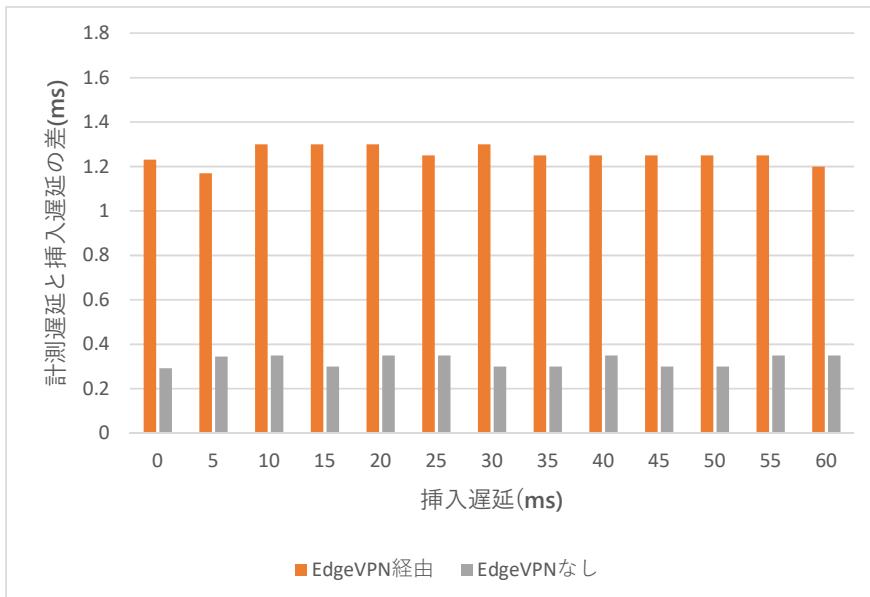


図 6 EdgeVPN リンクに対する遅延挿入の影響

ループットに与える影響について調査した。5.2.1 と同様に  $tc$  を用いてリンクに 0-60ms の遅延を挿入し、EdgeVPN を利用する場合と直接接続する場合の帯域減少の様子を iperf[6] を使用して計測した。計測値は iperf を 60 秒間の設定で 10 回実行した結果を用いている。EdgeVPN を利用する通信についての結果をプロットしたものが図 7、直接接続する通信についての結果をプロットしたものが図 8 である。

EdgeVPN を利用する通信において、スループットが指数関数的に減少していることがわかる。一方で、最も大きい 60ms まで遅延が増大した状況下においても 40Mbps 程度のスループットを GamingAnywhere サーバ/クライアント間のリンクにおいて維持できている。(これは許容量であるみたいなことを関連研究のどこ書いてから書く)

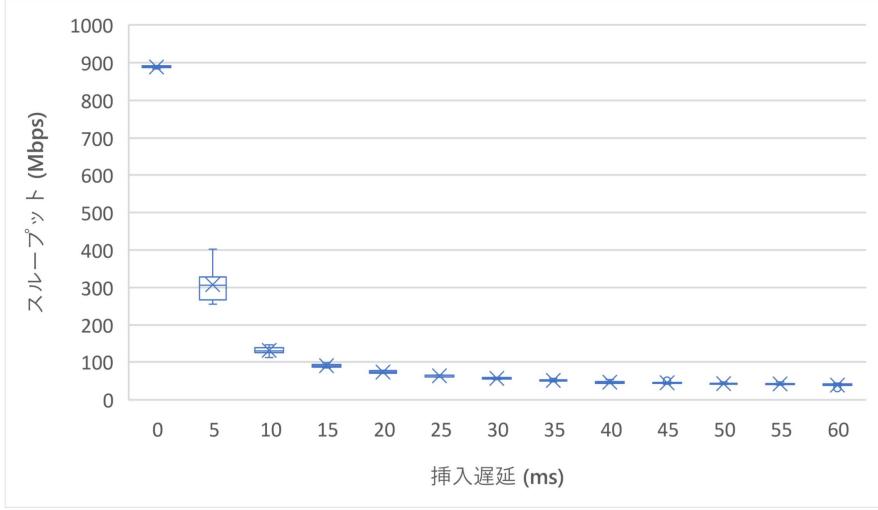


図 7 EdgeVPN リンクへの遅延挿入の帯域への影響

### 5.3 ゲームプレイ時のフレームレート

#### 5.3.1 ネットワーク帯域の大小の影響

提案システムを使用して実際にゲームをプレイしている間、sicilia と firenze の間のネットワーク帯域幅のサイズがゲーム画面のフレームレートにどのように影響するかを調査した。

実験は GamingAnywhere サーバの設定でフレームレートを 60fps に設定した状態で行った。ゲームのジャンルにより画面の動きが激しく、フレームレートの変動の影響が大きいものとそうでないものがあるため、複数のジャンルから選んだゲームを実験に使用した。実験に使用したゲームは MMORPG に分類される Albion Online[4]、FPS/アクションに分類される Red Eclipse 2[8]、およびボードゲームである Simply Chess[9] の 3 種類である。いずれも PC ゲームのダウンロード販売等を行うプラットフォームである Steam[12] で公開されている、Linux に対応したゲームである。

GamingAnywhere サーバ/クライアント間のリンクに tc を使用して帯域制限を施し、EdgeVPN を利用する場合と直接接続する場合のそれについてフレームレートの変化を観測した。帯域制限は、制限をかけていない 1Gbps および 100Mbps

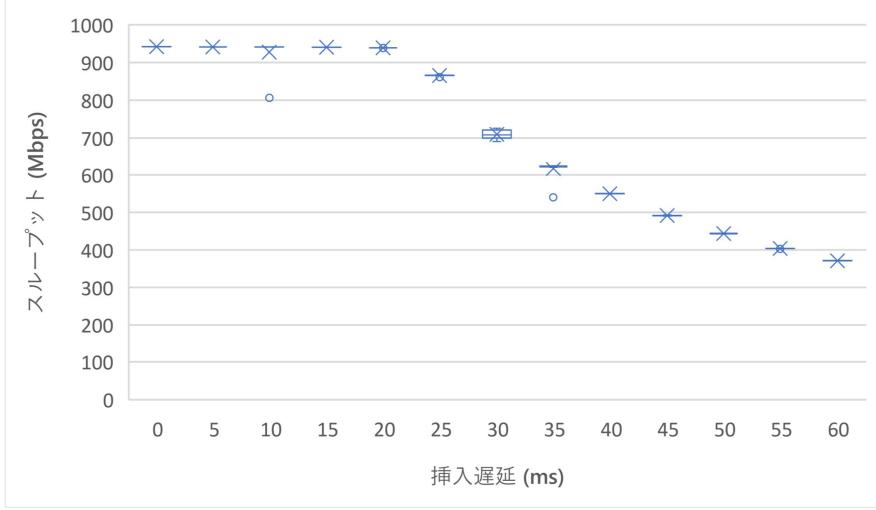


図 8 EdgeVPN を使用していないリンクへの遅延挿入の帯域への影響

と 10Mbps の 3 つの条件で行った。計測値は GamingAnywhere クライアントが定期的に出力するフレームレートの値を 10 回分計測し、その平均値を使用している。Albion Online をプレイした際の結果を図 12、Red Eclipse 2 をプレイした際の結果を図 13、Simply Chess をプレイした際の結果を図 14 にそれぞれ示した。

動きのあまり少ないボードゲームの Simply Chess のプレイ中においては EdgeVPN を利用するかどうかに関わらず、帯域制限下でもほぼ 60fps のフレームレートを保っている。また、常に画面の少なくとも一部に動きが生じるジャンルである MMORPG の Albion Online のプレイ中においても、帯域制限下で 60fps を大きく下回ることなくパフォーマンスが安定している。さらに、常に画面全体に激しい動きが発生する FPS/アクションに分類される Red Eclipse 2 のプレイ中においても、帯域制限下でフレームレートの低下は確認されなかった。

#### 5.4 考察

国内のデータセンターにユーザコンピュータが接続する場合のネットワーク遅延は最大 50ms 程度である。提案システムにおいて、プレイヤー PC が接続するボランティアが提供する遊休コンピュータまでのネットワーク遅延がこれよりも



図 9 提案システムを使用した Albion Online (MMORPG) のプレイの様子

小さい場合に、クラウドゲーミングの QoE を損なわないパフォーマンスを提供できるかどうかを考察する。

まず、クラウドゲーム/サーバ間のリンクのネットワーク遅延について、EdgeVPN のオーバーヘッドにより直接接続のネットワークよりも 1ms 程度遅延が増えていく。この遅延の増大はゲームプレイに影響を及ぼす程度のものではなく許容量であるといえる。

また、リンクに EdgeVPN を使用したことにより、リンクのネットワーク遅延が大きくなるに伴ってスループットが低下する。計測値のうち最も低くなるリンクのネットワーク遅延が 60ms の場合のスループットは 40Mbps 程度である。Google Stadia をプレイするのに必要なネットワーク要件として最低 25Mbps のスループットのネットワーク接続が必要であると言われている [3]。また Playstation Now は公式サイトにて、快適にプレイするため必要なネットワーク要件として 12Mbps 以上のスループットを上げている [11]。提案システムはこれらの要件を満たしており、クラウドゲーミングのプレイに充分なスループットを維持できると言える。

さらに、EdgeVPN のオーバーヘッドがゲームプレイ時のフレームレートに与える影響について調べた。その結果、10Mbps の低スループットのネットワークを使用した場合においてもフレームレートの低下は見られなかった。これは、ボー



図 10 提案システムを使用した Red Eclipse 2 (FPS, Action) のプレイの様子

ドゲームのような画面の動きが小さいゲームのみならず、FPSのような常に激しい画面の動きを伴うゲームにおいても同様である。

以上より、提案システムはデータセンターよりもネットワーク遅延の小さい遊休コンピュータでクラウドサーバをホストした場合に、QoE を損なわないクラウドゲーミングのプレイに十分な性能を示した。



図 11 提案システムを使用した Simply Chess (ボードゲーム) のプレイの様子

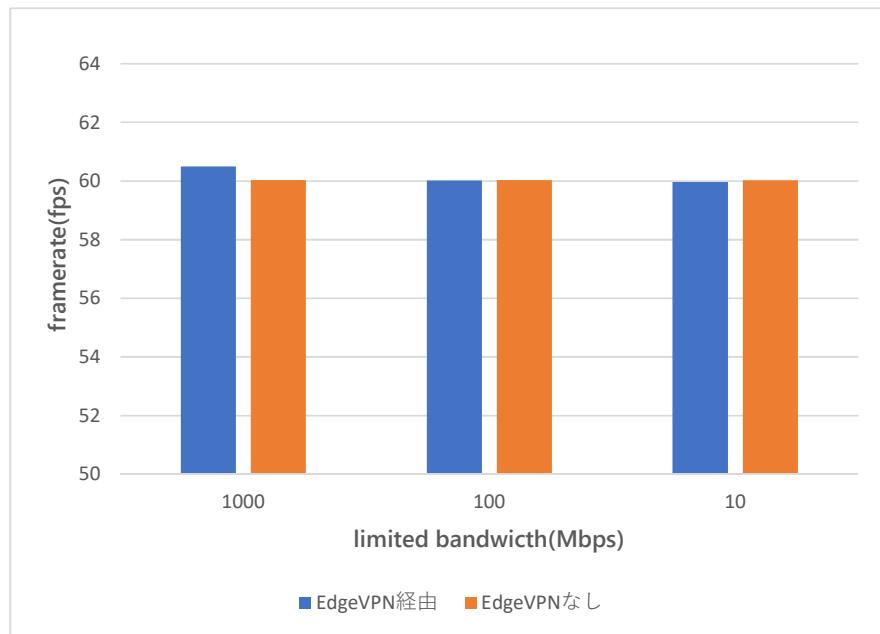


図 12 帯域制限下でのゲームプレイ時のフレームレートの変化 (Albion Online (MMORPG) プレイ時)

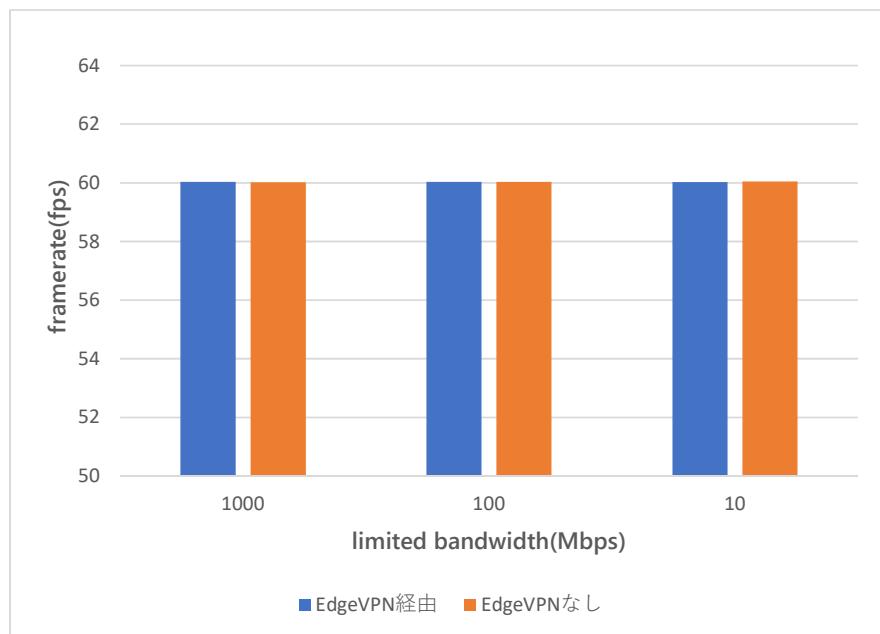


図 13 帯域制限下でのゲームプレイ時のフレームレートの変化 (Red Eclipse 2 (FPS, Action) プレイ時)

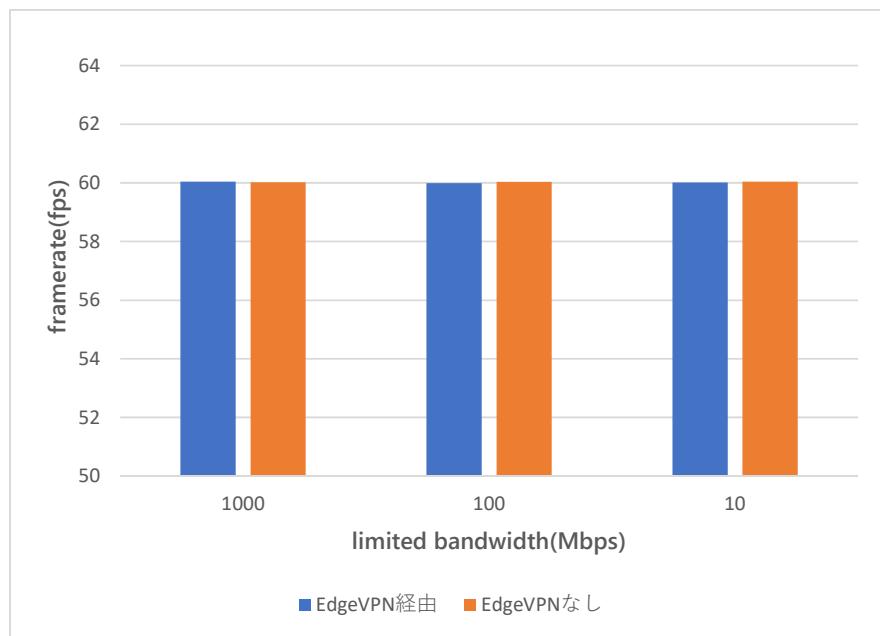


図 14 帯域制限下でのゲームプレイ時のフレームレートの変化 (Simply Chess (ボードゲーム) プレイ時)

## 6. まとめと今後の課題

本研究では、ボランティアが研究する地理的に近傍の遊休コンピュータのリソースを活用することによるクラウドゲーミングシステムを提案した。既存のクラウドゲーミングアーキテクチャにおいて、プレイヤーの端末からデータセンターへ接続する際のネットワーク遅延を回避することはできない。この課題を解決するため、広域に分散したボランティアの計算機資源でクラウドゲームサーバをホストするシステムを考案した。実験を通して、提案システムはクラウドゲーミングのプレイにおいてネットワーク遅延やスループットが許容量に収まることを示した。また、ゲームプレイ中のフレームレートについて、低スループットのネットワークを使用した際でも低下が見られないことを示した。これにより、提案システムがクラウドゲーミングのプレイにおける QoE を低下させることなくサービスを提供できることを示した。

今後の展望としては、利用可能な遊休コンピュータの中から、プレイヤー PC からのネットワーク遅延が小さいものを探索する機能を持つボランティアクラウドゲーミングコントローラの実装を検討している。また、今回の実験は遊休コンピュータとプレイヤー PC がそれぞれ 1 台ずつの構成で行ったが、これらの数を増やした上でシステムの負荷試験についても検討する。

## 謝辞

ありあとやす

## 参考文献

- [1] OnLive GAMES ON DEMAND, 2015. <http://onlive.com/>.
- [2] ゲームストリーミングサービスの「OnLive」がサービスを閉鎖、資産をソニーが取得, 2015. <https://automaton-media.com/articles/newsjp/onlive-sony-gaikai/>.
- [3] Google's Project Stream is a working preview of the future of game streaming, 2018. <https://www.theverge.com/2018/10/8/17950998/google-project-stream-gaming-assasins-creed-odyssey-first-impression>.
- [4] Albion Online: The Fantasy Sandbox MMORPG, 2020. <https://albiononline.com/en/home>.
- [5] GeForce NOW - あなたのPCゲームをどこでもプレイするための「力」, 2020. <https://www.nvidia.com/ja-jp/geforce-now/>.
- [6] iperf3, 2020. <https://software.es.net/iperf/>.
- [7] iproute2, 2020. <https://wiki.linuxfoundation.org/networking/iproute2>.
- [8] Red Eclipse, 2020. <https://www.redeclipse.net/>.
- [9] Simply Chess, 2020. <http://bluelinegamestudios.com/simply-chess/>.
- [10] Open-source VPN for Edge Computing, 2021. <https://edgevpn.io/>.
- [11] PlayStaion Now, 2021. <https://www.playstation.com/ja-jp/ps-now/>.
- [12] Simply Chess, 2021. <https://store.steampowered.com/?l=japanese>.
- [13] Stadia - One place for all the ways we play, 2021. <https://stadia.google.com/>.

- [14] D. P. Anderson, E. Korpela, and R. Walton. High-performance task distribution for volunteer computing. In *First International Conference on e-Science and Grid Computing (e-Science'05)*, pages 8 pp.–203, 2005.
- [15] David P Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, and Dan Werthimer. SETI@ home: an experiment in public-resource computing. *Communications of the ACM*, 45(11):56–61, 2002.
- [16] Robert Braden. RFC1122: Requirements for Internet hosts-communication layers, 1989.
- [17] Wei Cai, Ryan Shea, Chun-Ying Huang, Kuan-Ta Chen, Jiangchuan Liu, Victor CM Leung, and Cheng-Hsin Hsu. A survey on cloud gaming: Future of computer games. *IEEE Access*, 4:7605–7620, 2016.
- [18] The Linux Foundation. gRPC A high performance, open source universal RPC framework, 2020. <https://grpc.io/>.
- [19] Hua-Jun Hong, De-Yu Chen, Chun-Ying Huang, Kuan-Ta Chen, and Cheng-Hsin Hsu. Placing Virtual Machines to Optimize Cloud Gaming Experience. *IEEE Transactions on Cloud Computing*, 3(1):42–53, 2014.
- [20] Chun-Ying Huang, Cheng-Hsin Hsu, Yu-Chun Chang, and Kuan-Ta Chen. GamingAnywhere: an open cloud gaming system. In *Proceedings of the 4th ACM multimedia systems conference*, pages 36–47, 2013.
- [21] Stefan M. Larson, Christopher D. Snow, Michael Shirts, and Vijay S. Pande. Folding@ Home and Genome@ Home: Using distributed computing to tackle previously intractable problems in computational biology. *arXiv preprint arXiv:0901.0866*, 2009.
- [22] Kyungmin Lee, David Chu, Eduardo Cuervo, Johannes Kopf, Yury Degtyarev, Sergey Grizan, Alec Wolman, and Jason Flinn. Outatime: Using speculation to enable low-latency continuous interaction for mobile cloud

gaming. MobiSys '15, page 151–165, New York, NY, USA, 2015. Association for Computing Machinery.

- [23] 株式会社KADOKAWA Game Linkage (KADOKAWA グループ) . 2019年世界ゲームコンテンツ市場は前年比約2割増の15兆6898億円。国内は10年連続で成長！過去最高の1兆7330億円に。～『ファミ通ゲーム白書2020』発刊～, 2020. <https://prtimes.jp/main/html/rd/p/000007188.000007006.html>.
- [24] Rohan Mahy, Philip Matthews, and Jonathan Rosenberg. Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN). Technical report, RFC 5766, 2010.
- [25] K. Raaen, R. Eg, and C. Griwodz. Can gamers detect cloud delay? In *2014 13th Annual Workshop on Network and Systems Support for Games*, pages 1–3, 2014.
- [26] Peter Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Core. 2004.
- [27] Ryan Shea, Jiangchuan Liu, Edith C-H. Ngai, and Yong Cui. Cloud Gaming: Architecture and Performance. *IEEE network*, 27(4):16–21, 2013.
- [28] Pierre St Juste, Kyuho Jeong, Heungsik Eom, Corey Baker, and Renato Figueiredo. TinCan: User-Defined P2P Virtual Network Overlays for Ad-hoc Collaboration. *EAI Endorsed Transactions on Collaborative Computing*, 1(2):e4, 2014.
- [29] Kensworth Subratie, Saumitra Aditya, Vahid Daneshmand, Kohei Ichikawa, and Renato Figueiredo. On the Design and Implementation of IP-over-P2P Overlay Virtual Private Networks. *IEICE Transactions on Communications*, 103(1):2–10, 2020.

- [30] Dan Wing, Philip Matthews, Rohan Mahy, and Jonathan Rosenberg. Session Traversal Utilities for NAT (STUN). *RFC5389*, October, 2008.