

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 晃平 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

内容梗概

人類がこの地上に現われて以来、 π の計算には多くの関心が払われてきた。

本論文では、太陽と月を利用して π 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。

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 π has been paid much attention since human beings appeared on the earth.

This thesis presents novel low-speed algorithms to calculate π 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.

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	2
2.1 クラウドゲーミング	2
2.2 ボランティアコンピューティング	2
3. 設計	3
3.1 提案システムの概要	3
3.2 実装上の課題	4
3.3 構成コンポーネント	5
3.3.1 VC コントローラ	5
3.3.2 VC クライアントエージェント	5
3.3.3 VC ホストエージェント	5
3.3.4 クラウドゲームサーバ/クライアント	6
4. 実装	7
4.1 VC コントローラとエージェントの連携	7
4.2 クラウドゲームサーバ/クライアント間の P2P 通信	7
4.3 システム動作	8
5. 評価	10
5.1 評価環境	10
5.2 クラウドゲームサーバ・クライアント間の通信性能	10
5.2.1 リンクに対する生の遅延の大小の影響	10
5.2.2 リンクに対する遅延の大小の帯域への影響	10
5.3 ゲームプレイ時のフレームレート	10
5.3.1 ネットワーク帯域の大小の影響	10
5.4 考察	10
6. まとめと今後の課題	16

謝辞	17
参考文献	18

図 目 次

1	提案システムの概要	4
2	評価環境	11
3	EdgeVPN リンクに対する遅延挿入の影響	11
4	EdgeVPN リンクへの遅延挿入の帯域への影響	12
5	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	12
6	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)	13
7	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- pse 2 (FPS, Action) プレイ時)	14
8	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)	15

表 目 次

1. はじめに

従来のゲームプレイは、プレイヤーがゲームハードやゲーミングPC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。クラウドゲーミングは、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するサービスである。プレイヤーの使用する端末は、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などの性能が貧弱なデバイスでも高価なゲームハードやゲーミングPCでプレイするのと同様の高品質なゲーム体験を得られることが期待できる。(この辺の出典どうしよう)

商用のクラウドゲーミングサービスも展開されており、GoogleのGoogle Stadia、SONYのPlayStation NOW、NVIDIAのGeForce NOWなどがある。(もうちょっと膨らませたい気がする)

(この辺でGamingAnywhereの話とかする?)

(クラウドゲーミングは遅延が課題ですという話を論文引用しながら書く)(サーベイ論文使ったらもっといろんな課題の話できるな) クラウドゲーミングの課題はユーザ目線で高品質な画質の担保、十分なネットワーク帯域幅の確保、伝送データ圧縮・ストリーミング技術、画面表示や操作の遅延の最小化など。プロバイダ目線でゲーム環境の仮想化、サーバにおける負荷分散

(ボランティアコンピューティングの話はBOINCの引用でいいかな)

(研究目的を書く) クラウドのデータセンターでゲームが動作しているとデータセンターまでの遅延を避けることは不可能。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利用するクラウドゲーミングフレームワークを提案。プレイヤーから見て近傍の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。ネットワーク遅延削減によりプレイヤーが体験する遅延を減少させる

??節では、過去における研究について述べ、6章では、現状と今後の課題について述べる。また、付録??におまけその1を添付する。

過去における研究としては[6]などがある。

2. 背景

2.1 クラウドゲーミング

GamingAnywhere: An Open Cloud Gaming System Huang ら [5] は、
Placing Virtual Machines to Optimize Cloud Gaming Experience Hong ら [4]
は、

2.2 ボランティアコンピューティング

High-Performance Task Distribution for Volunteer Computing Anderson ら [2]
は、
(EdgeVPN(TinCan) の話は実装の章で)

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によ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1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、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

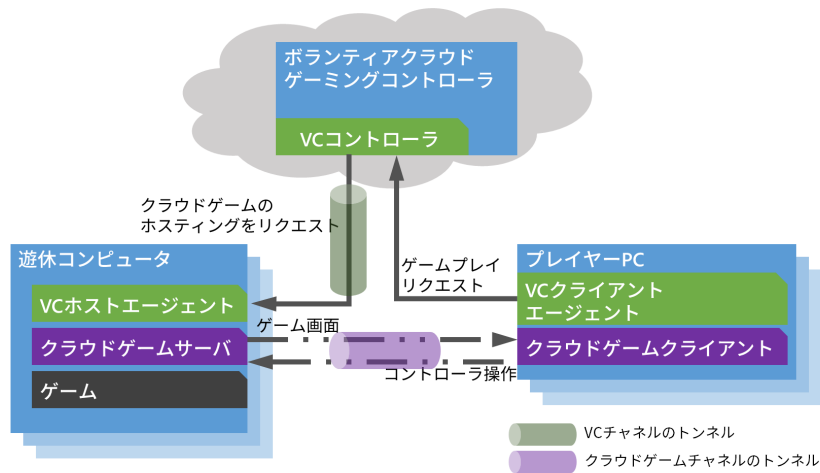


図 1 提案システムの概要

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[3] 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に使用される。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。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。GamingAnywhere は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り、ユーザのコンピュータにインストールして設定を最適に変更しつつ実験を行えるクラウドゲームテストベッドである。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。

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。これを解決するため、GamingAnywhere の通信を行うリンクに対し EdgeVPN[1] を使用する。EdgeVPN は、分散エッジソース全体にスケーラブル VPN を展開するためのオープンソースソフトウェアである。パケットキャプチャ、暗号化、トンネリング、転送および NAT トラバーサルをサポートが組み込まれている。エッジデバイスと 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することで、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。EdgeVPN リンクはオープンソースの WebRTC フレームワークによって実装された SSL ベースのトランスポートレイヤプロトコルセキュリティで暗号化および認証され、ノード間の通信はトンネルを介して UDP に基づく Datagram TLS(DTLS)[8] を使用する。また、EdgeVPN は XMPP プロトコル [9] 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 St Juste ら [10] が開発した TinCan に基づいている。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し、分離されたコントローラ/データパスモデルをサポートしている。また、ノードが接続先にパブリック 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 [11] を使用し、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 [7] を使用して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。

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。これを受信した VC ホストエージェントは了解を返し、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、ゲーム画面を GamingAnywhere サーバがキャプチャする。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。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。

5. 評価

5.1 評価環境

評価の目的・手法・実験環境

評価環境も図にする

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

tc を使って任意に遅延を挿入し、ping の値で遅延の増え方に影響がないか見る。遅延が増えたときの遅延の増え方が線形みたいなことを言う。遅延が増えたときの帯域の減り方の話をする。

5.2.2 リンクに対する遅延の大小の帯域への影響

tc を使って任意に遅延を挿入し、iperf で帯域の減り方への影響を見る遅延が増えたときの帯域の減り方の話をする。

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

tc を使って帯域に制限をかけて、実際に複数のゲームをプレイしたときのフレームレートへの影響を見る。使用したゲームは Steam で公開されている Albion Online(MMORPG)、Red Eclipse 2(FPS, Action)、Simply Chess(Board Game)

5.4 考察

どれぐらい数値が良ければ既存に勝てるのかみたいなこと

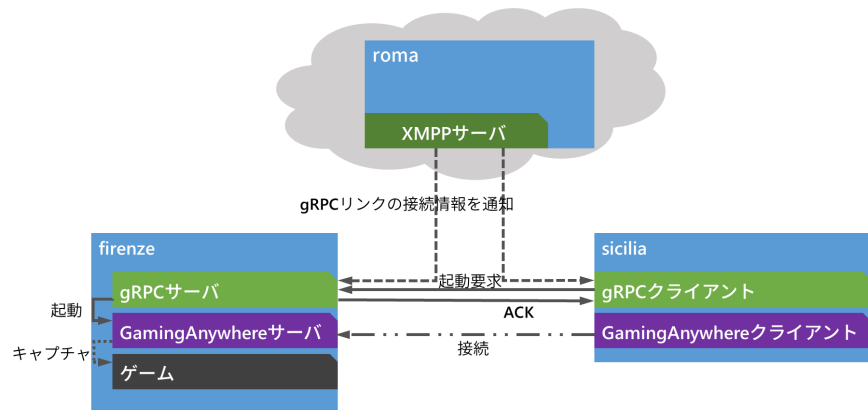


図 2 評価環境

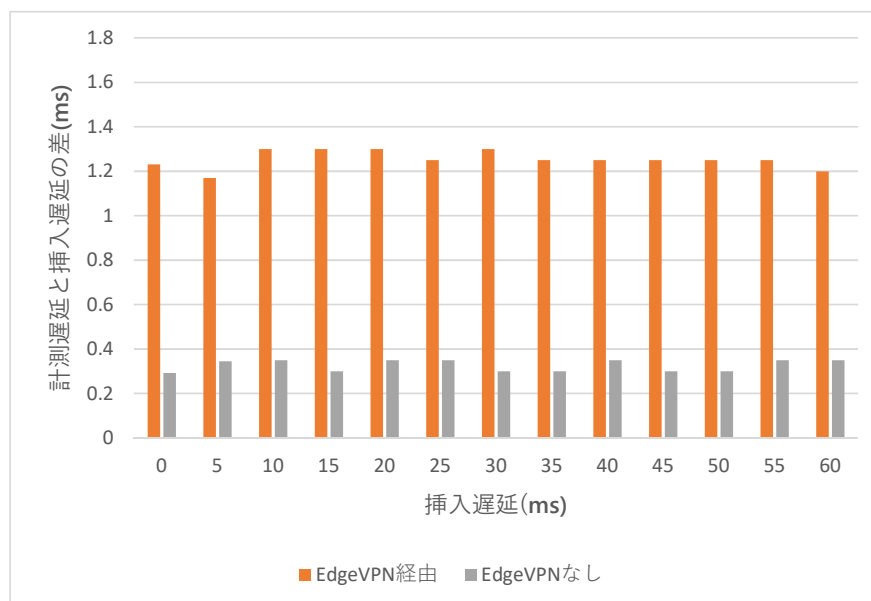


図 3 EdgeVPN リンクに対する遅延挿入の影響

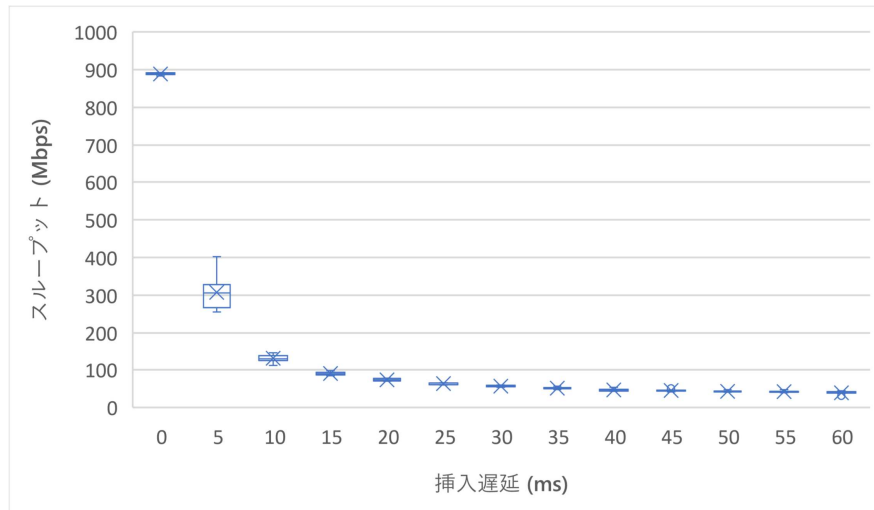


図 4 EdgeVPN リンクへの遅延挿入の帯域への影響

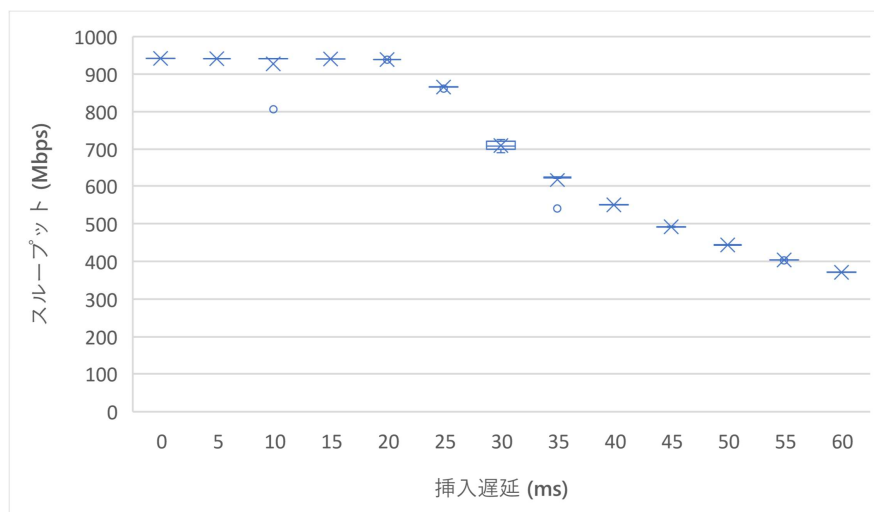


図 5 EdgeVPN を使用していないリンクへの遅延挿入の帯域への影響

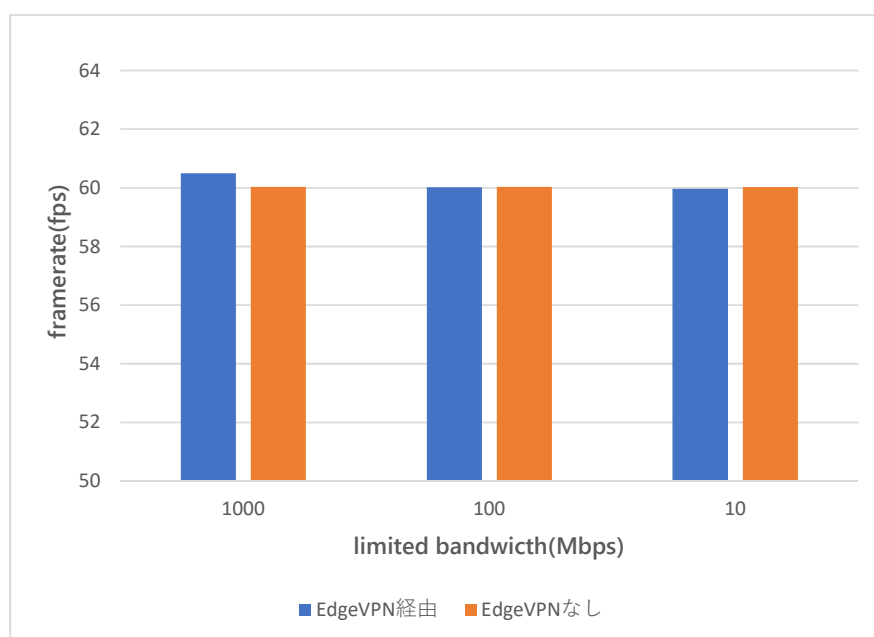


図 6 帯域制限下でのゲームプレイ時のフレームレートの変化 (Albion Online (MMORPG) プレイ時)

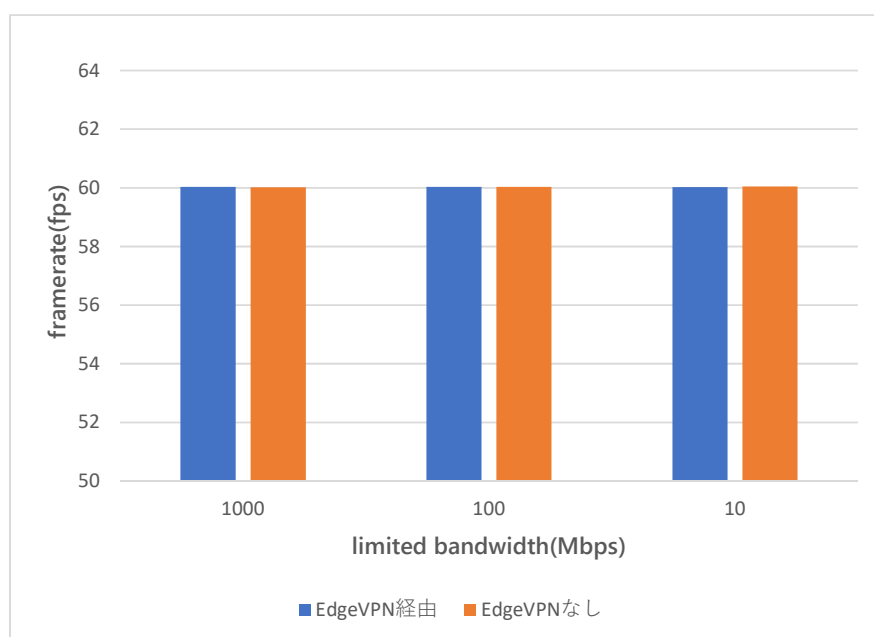


図 7 帯域制限下でのゲームプレイ時のフレームレートの変化 (Red Eclipse 2 (FPS, Action) プレイ時)

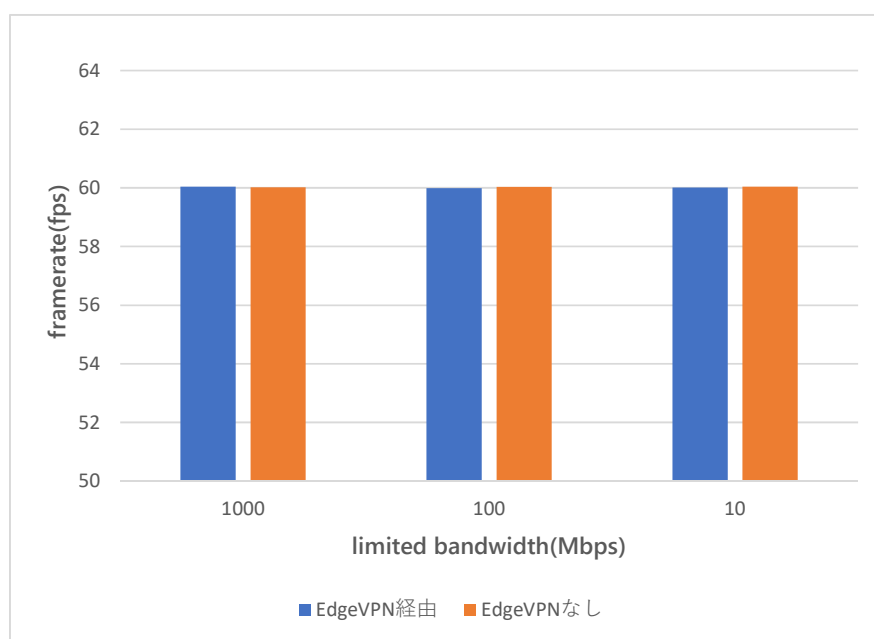


図 8 帯域制限下でのゲームプレイ時のフレームレートの変化 (Simply Chess (ボードゲーム) プレイ時)

6. まとめと今後の課題

今後、ボランティアクラウドゲームコントローラの実装。遊休コンピュータ、プレイヤーPCの数を増やしての動作での負荷試験。

謝辞

ありあとやす

参考文献

- [1] Open-source VPN for Edge Computing, 2021. <https://edgevpn.io/>.
- [2] David P Anderson, Eric Korpela, and Rom Walton. High-performance task distribution for volunteer computing. In *First International Conference on e-Science and Grid Computing (e-Science'05)*, pages 8–pp. IEEE, 2005.
- [3] The Linux Foundation. gRPC A high performance, open source universal RPC framework, 2020. <https://grpc.io/>.
- [4] 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.
- [5] 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.
- [6] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25(NIPS'12)*, pages 1097–1105, 2012.
- [7] 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.
- [8] Eric Rescorla, Nagendra Modadugu, et al. Datagram Transport Layer Security, 2006.
- [9] Peter Saint-Andre et al. Extensible Messaging and Presence Protocol (XMPP): Core. 2004.
- [10] 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), 2014.

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