

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

内容梗概

従来のプレイヤーがゲーム専用機やゲーミングPCを所有して行うゲームプレイに対し、クラウドサーバでゲームを動作させてゲーム画面をストリーミングするクラウドゲーミングがゲーム業界で注目されている。クラウドゲーミングはプレイヤーの端末ではストリーミングされた画面の描画とプレイヤーの入力の送信のみを行うため、スマートフォンやタブレットなどの性能が貧弱な端末でも高品質なゲーム体験を得られる。一方で、クラウドゲーミングはプレイヤーが入力を行ってからストリーミングされる画面に反映されるまでの応答遅延が課題である。この応答遅延は、クラウドゲームサーバにおけるレンダリング処理時間やストリームのエンコード・デコード処理時間を含むが、特にクラウドゲームサーバとプレイヤー端末間のネットワーク遅延の割合が大きい。クラウドゲーミングのプレイヤーの Quality of Experience (QoE) を担保するために、このネットワーク遅延を解消する必要がある。本研究では、広域分散した計算機資源を活用する枠組みであるボランティアコンピューティングの仕組みを応用し、クラウドサーバよりも地理的に近傍の計算機でクラウドゲームサーバを動作させることによってネットワーク遅延を解消するシステムを提案する。また、提案手法を評価するために、システムの実装を行い、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を行った。その結果、提案システムが有效地に働く条件と、ゲームプレイにおけるQoEを低下させることなくサービスを提供できることを明らかにした。

*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。

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

Research on QoE Improvement of Cloud Gaming by Utilizing Volunteer Computing Resources*

Kento Maeda

Abstract

Cloud gaming, which runs games on cloud servers instead of player-owned gaming consoles or PCs, has been attracting much attention from the gaming industry. Since a player's device only receives the game screen streamed from the cloud server, even poorly performing mobile devices such as smartphones or tablets can provide a high-quality gaming experience. However, cloud gaming suffers from high response delay (duration between the player's input and when the input is reflected on the screen). The breakdown of the response delay includes the rendering time, video encoding/decoding time, and network delay between the player's device and the cloud server, but recent studies have shown that network delay is most significant. Therefore, reducing the network delay is crucial to improve the Quality of Experience (QoE) of cloud gaming. In this thesis, I propose a system that reduces the network latency by applying volunteer computing, a concept where computation is distributed across a large number of computing resources contributed by volunteers, and by running the cloud game server on a computer geographically close to the player. I implemented a prototype of the proposed system and evaluated the communication performance and

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

the frame rate during gameplay to quantify the QoE. The results reveal the conditions where the proposed system provides better QoE than conventional cloud gaming systems and demonstrates that the proposed system does not degrade the QoE of gameplay.

Keywords:

network, cloud, cloud gaming, volunteer computing

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	10
3.1 提案システムの概要	10
3.2 実装上の課題とアプローチ	11
3.3 構成コンポーネント	12
3.3.1 VC コントローラ	12
3.3.2 VC クライアントエージェント	12
3.3.3 VC ホストエージェント	13
3.3.4 クラウドゲームサーバ/クライアント	13
3.4 提案システムの範囲及び制限	13
4.実装	14
4.1 VC コントローラとエージェントの連携	14
4.2 クラウドゲームサーバ/クライアント間のP2P通信	16
4.3 システム動作	19
5.評価	21
5.1 評価環境	21
5.2 クラウドゲームサーバ/クライアント間の通信性能	23
5.2.1 リンクに対する遅延の大小の影響	23
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	24
5.2.3 10Gbps ネットワークにおける EdgeVPN のオーバーヘッド	26
5.3 ゲームプレイ時のフレームレート	28
5.3.1 ネットワーク帯域の大小の影響	28

5.4 考察	30
6. まとめと今後の課題	34
謝辞	35
参考文献	37

図 目 次

1	クラウドゲーミングプラットフォーム	2
2	提案システムの概要	11
3	gRPC の概要	14
4	TinCan のシステム概要	16
5	EdgeVPN のオーバレイネットワーク	17
6	システム動作	20
7	評価環境	22
8	評価環境における通信	22
9	EdgeVPN リンクに対する遅延挿入の影響	24
10	EdgeVPN リンクへの遅延挿入の帯域への影響	25
11	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	26
12	10Gbps ネットワークにおける EdgeVPN リンクへの遅延挿入の帯域への影響	27
13	10Gbps ネットワークにおける 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	27
14	提案システムを使用した Albion Online (MMORPG) のプレイの様子	28
15	提案システムを使用した Red Ecliplse 2 (FPS, Action) のプレイの様子	29
16	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	29
17	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)	31
18	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)	32
19	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)	33

表 目 次

1 評価に用いたマシンのスペック	23
----------------------------	----

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[24]。国内でもゲーム産業は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のゲームの市場規模が年々拡大している。このようにゲームのプレイスタイルがゲーム専用機を購入するというものから汎用端末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機やゲーミング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のみを行う。この仕組みによって、スマートフォンやタブレット等の性能が貧弱な端末でも、従来は高価なゲーム専用機やゲーミングPCを使用しなければ体験できなかった高品質なゲーム体験を得られる。

クラウドゲーミングは既に商用サービスとして提供されている。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。Google Stadiaは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。Google Stadiaは現在日本を含まない14カ国で展開されている。また、GPU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

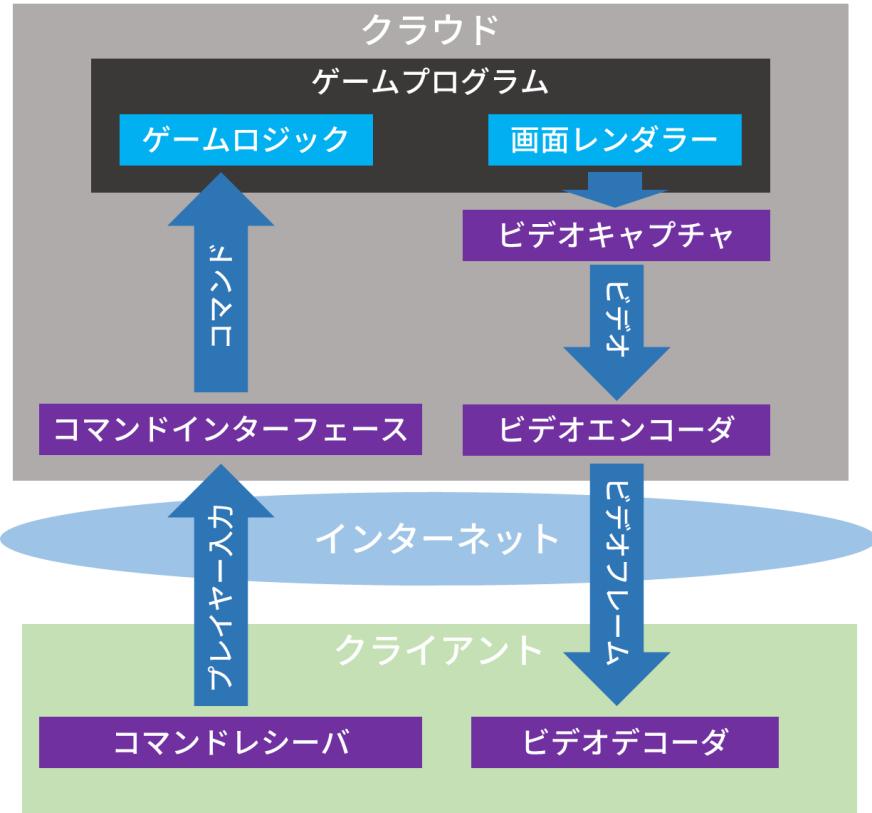


図 1 クラウドゲーミングプラットフォーム

イしていたプレイヤー層をメインターゲットに据えており、PC ゲーマーからの注目を集めている。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 [21] は、既存のクローズドソースのシステムでは研究開発のための修正や拡張を行うこと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、macOS 上で実装されており、クライアントは iOS や Android 等の他の OS にも容易に移植が可能である。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に関する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サービスの提供者の視点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といった課題がある。一方、クラウドゲーミングシステムのユーザであるプレイヤーの視点から見た課題としては、ゲーム体験の品質、すなわち Quality of Experience (QoE) がある。クラウドゲーミングにおける QoE の担保に必要な課題としては、以下がある。

- ストリーミングされるゲーム画面のレンダリング品質の担保
- 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
- 高圧縮率と高画質を両立する動画ストリーミングコーデックの実現
- 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

これらの課題の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最小化は特に重要な課題である。Raaen[28] 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。クラウドゲーミングサービスにおける応答時間は主に以下で構成される。

- クライアント/サーバ間のネットワーク遅延
- サーバにおけるゲーム画面のレンダリング処理時間
- 動画ストリームのエンコード・デコード処理時間

これらの中でもネットワーク遅延が支配的であり、Lee[23] らはネットワークの RTT が 128ms である場合、応答時間の約 8 割をネットワーク遅延が占めると指摘している。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。

クラウドサービスを提供するサーバが配備されるデータセンターは、通常国内には数箇所しか存在しない。既存のクラウドゲーミングアーキテクチャにおいて

は、クラウドのサーバ上でゲームが動作している。このため、データセンターから地理的に遠いプレイヤーの端末からクラウドゲーミングサービスに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。これがゲームプレイにおける応答遅延を増大させることとなり、プレイヤーのQoE低下の原因となり得る。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を解消するためには、プレイヤーの地理的近傍に計算機資源を設置することが必要である。そのため、中央集権的なクラウドデータセンターのようなアーキテクチャの利用ではなく、広域に分散した計算機資源を活用する必要があると考えられる。

広域分散した計算機資源の活用形態としては、エッジコンピューティングや一般ユーザがボランティアとして計算機資源を提供するボランティアコンピューティングなどの活用が考えられる。しかしながら、エッジコンピューティングは一般的にIoTアプリケーションなどで用いられる比較的小規模な計算機資源であることから、クラウドゲーミングの用途には適さないと考えられる。一方で、一般ユーザが計算機を提供するボランティアコンピューティングは、大規模な科学技術計算を多数の小規模なタスクに分割して分散処理するべくして設計されたもので、ゲームをホストするために十分な計算機性能を確保できると考えられる。現在までに、SETI@home[15] や Folding@home[22] 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。既存の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計算機資源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。国内に数箇所しか存在し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計算機資源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。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、研究の背景と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 [21] 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。StreamMyGame は、Windows 向けのゲームおよびアプリケーションに対し、Windows および Linux の端末でのリモートプレイを可能にするソフトウェアである。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 [30] 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 [23] 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 [20] 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。

ボランティアコンピューティングは、前述のように本来は科学技術計算を膨大な数の一般ユーザの計算機資源を用いて分散処理する仕組みを提供するものである。しかし、多数の一般ユーザの計算機資源を利用している点に注目すると、中

央集権的なクラウドコンピューティングのアーキテクチャと異なり、エンドユーザに地理的に近傍に利用可能な計算機環境を広く配備することが可能なプラットフォームと考えられる。このような広域分散システムの活用により、既存のクラウドゲーミングが抱える遅延の課題を解決できる可能性がある。ただし、既存のボランティアコンピューティングにおいて、ボランティアが提供する計算機ノードは基本的にタスク配信を行う中央サーバのみと通信を行い、計算機ノード間で直接通信を行う仕組みはない。ボランティアコンピューティングを物理的に近傍な計算機資源を利用する仕組みとしてクラウドゲーミングに応用する際には、この課題を解決する必要があると考えられる。

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、

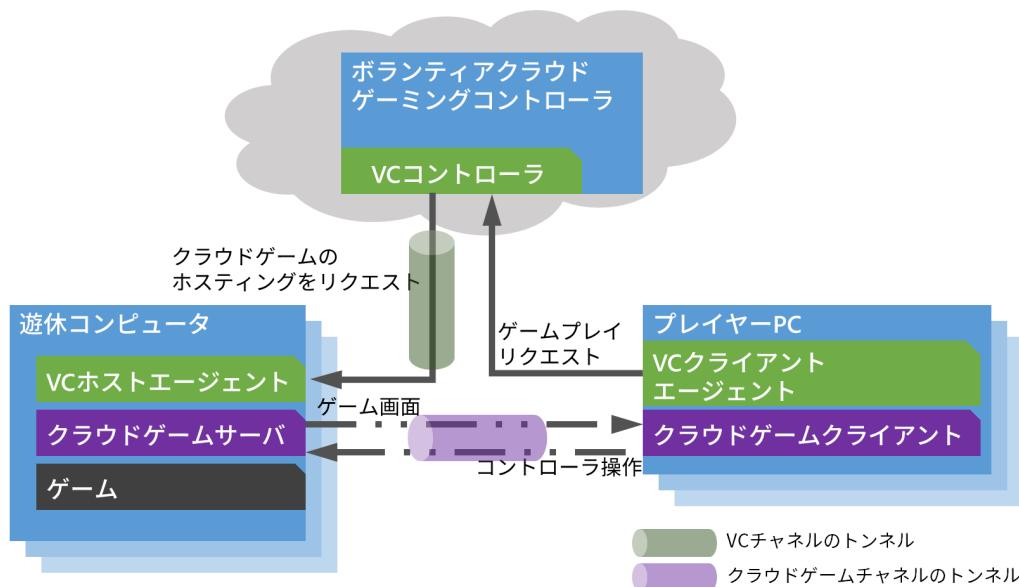


図 2 提案システムの概要

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 実装上の課題とアプローチ

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

コントローラと遊休コンピュータ間の通信については、RPC の技術を用いてトンネリングを行い、命令の伝送を行うことを試みる。クラウドゲームサーバ/クライアント間の通信については、P2P 型の VPN を活用することでの接続を試みる。その性能についてネットワークと QoE の 2 つの観点から評価し、ボランティアコンピューティングをクラウドゲーミングに利用することが有効な条件を本研究で明らかにする。

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。

3.4 提案システムの範囲及び制限

提案システムには、ボランティアクラウドゲーミングコントローラにてプレイヤー PC と遊休コンピュータのマッチングを行う機能を含んでいる。しかし、プレイヤー端末とクラウドゲームサーバ間のネットワーク遅延に着目したマッチングは、既存の VM 配置最適化 [20] と同様の問題を扱うことになる。したがって本研究ではこの機能については対象外とし、ユーザコンピュータ間の通信によるクラウドゲーミングシステムの実現を主眼に置くこととする。

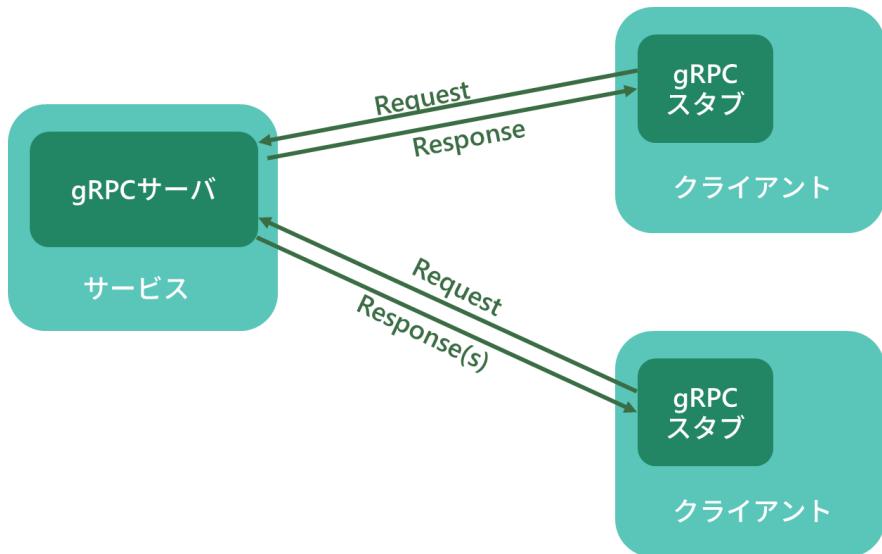


図 3 gRPC の概要

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の解決手法としては、REST API と Long poling[25] を組み合わせる方法や Websocket[18] の活用が考えられる。しかし、Long poling にはデータ転送のたびに再接続を行うためメモリ消費が多いなどの問題がある。また、提案システムではプレイヤーからのゲームプレイリクエストに対し、了解や起動報告、終了報告など複数のレスポンスをサーバ側の任意のタイミングで返す必要があり、Websocket ではこれを実現できない。そこで本研究では、gRPC[19] 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。RPC は Remote Procedure Call の略で、プログラムから別のアドレス空間にあるサブルーチンや

手続きを実行することを可能にする技術である。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。

gRPCはgRPCサーバとgRPCスタブのやりとりを行うためにprotoファイルにインターフェースを定義する必要がある。やりとりされるデータ構造はmessageとして指定し、コンパイラによってデータアクセスクラスが生成される。また、gRPCサーバが提供するサービスもproto上に定義し、入力と戻り値をmessageとして指定する。定義したprotoファイルをListing1に示す。実装したgRPCのAPIの動作は以下である。

1. gRPCスタブがAwakeningサービスを使用してAwakeRequestを送信する
2. gRPCサーバがAwakeRequestを受け取るとGamingAnywhereサーバを起動し、gRPCスタブにAwakeReplyを返す
3. gRPCスタブがAwakeReplyを受け取るとGamingAnywhereクライアントを起動し、GamingAnywhereサーバに接続する

Listing 1 protoファイル

```
1 syntax = "proto3";
2
3 package signal;
4
5 service Signal{
```

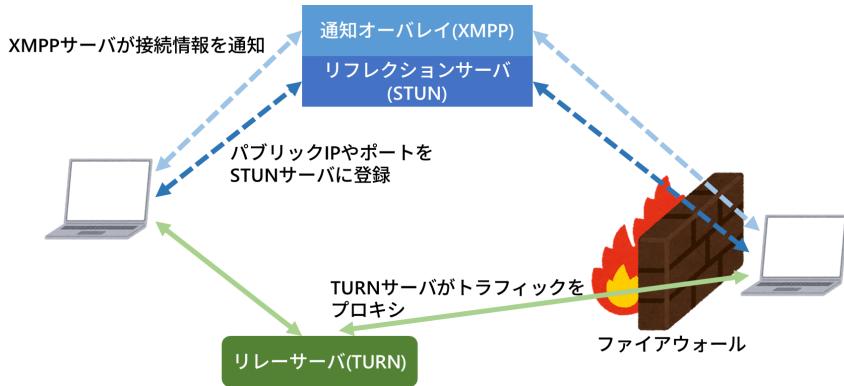


図 4 TinCan のシステム概要

```

6      rpc Awakening (AwakeRequest) returns (AwakeReply) {}
7  }
8
9  message AwakeRequest {
10    string name = 1;
11  }
12
13 message AwakeReply {
14   string message = 1;
15 }
```

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。

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

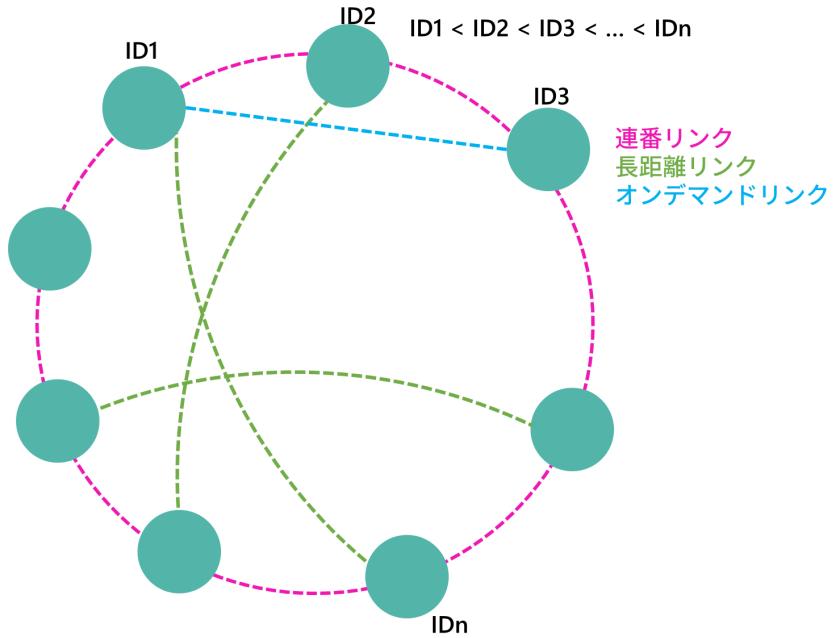


図 5 EdgeVPN のオーバレイネットワーク

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。

ボランティアコンピューティングをクラウドゲーミングにおいて活用するにあたって、NAT やファイアウォールの背後にあるユーザコンピュータのペアを安全かつ透過的に接続することが必要である。また、遊休コンピュータを提供するボランティアやプレイヤーが行う設定は少ないことが好ましい。

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。VPN は Virtual Private Network の略で、本来公的なネットワークであるインターネットに跨ってプライベートネットワーク間が専用線で接続されているように拡張する技術及びそのネットワークである。EdgeVPN は通常の VPN とは異なり、通信を集約する VPN サーバを必要としない。接続するピア同士が P2P ベースの分散プロトコルに基づいてパケットを自律的にルーティングすることによって、ピア間の通信を確保する。そのため、各ピア上やピアが

所属するネットワークにおいて特別な設定は必要なく、透過的に遊休コンピュータとプレイヤーPC間の通信を確立できる。

EdgeVPNはIP-over-P2P(IPOP)プロジェクト[32]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[29]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[31]が開発したTinCanを使用している。TinCanは図4のように、XMPPサーバを使用してエンドツーエンドのVPNトンネルを配備することにより、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[33]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[26]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。

EdgeVPNのオーバレイネットワークはネットワーカトポジをSymphonyプロトコル[27]を基に構築している。Symphonyプロトコルは、図5のようにネットワークに参加する全ノードに一意の識別子を割り当て、識別子によって順序付けられたリング構造でネットワークを構成する。このリングに沿って連番リンクをたどりながらパケットを転送することで、あるノードと他の任意のノードとの間で通信が可能になる。しかし、単純なリングではスケールしないため、リングを横切る長距離リンクを追加してパス長を短くする。これによりSymphonyプロトコルは $O(\log(N))$ ステップでパケットを転送可能である。EdgeVPNは更に、トラフィックの多いノード間には図5のノードID1とノードID3間のリンクのよう

なオンデマンドリンクを自動で追加して転送を効率化する機能を備えている。

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 6 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)。

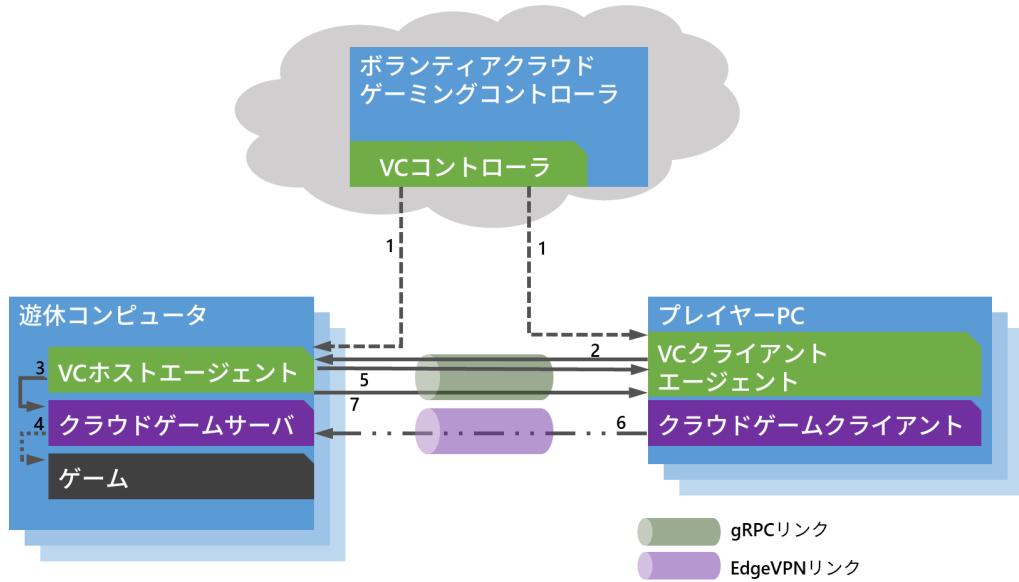


図 6 システム動作

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。提案システムは遊休コンピュータとプレイヤーPCとの間でEdgeVPNによるVPNを確立する。このVPNが挿入するネットワーク遅延やスループットのオーバーヘッドが大きすぎる場合、クラウドデータセンターではなくボランティアが提供する遊休コンピュータを利用する価値はなくなる。したがって本評価では、提案システムにおけるネットワーク遅延が十分に小さいことと、提案システムが達成可能なスループットがクラウドゲーミングを行うに当たって十分なパフォーマンスを有することを評価する。

評価を行う環境を図7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の役割を果たすプレイヤーPCではgRPCクライアントとGamingAnywhereクライアントが動作す

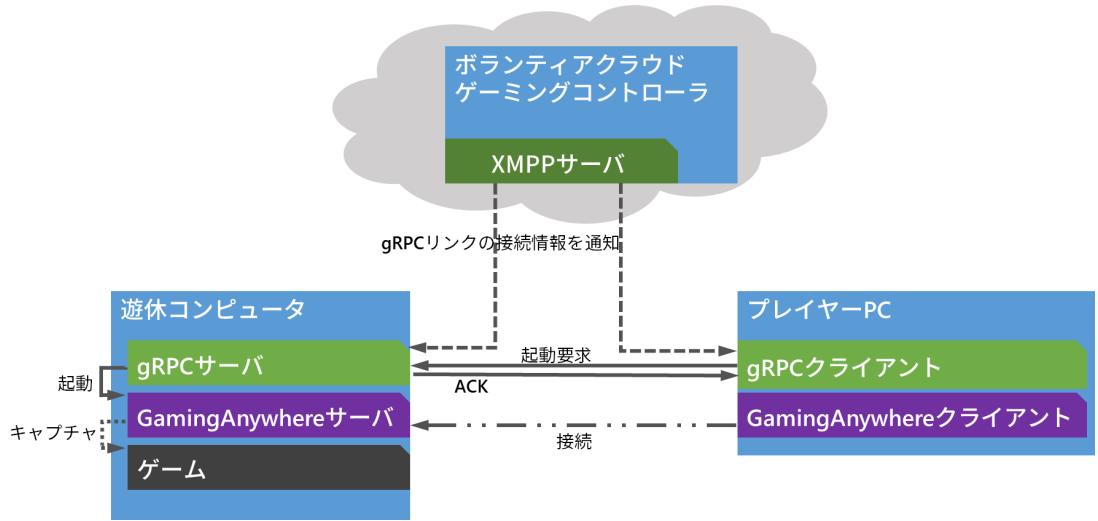


図 7 評価環境

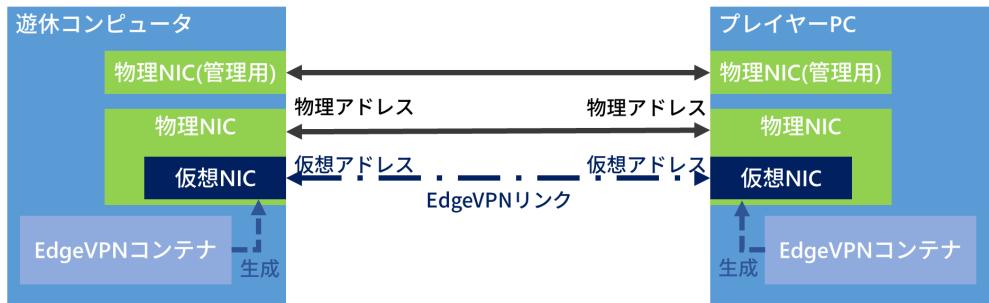


図 8 評価環境における通信

る。また、遊休コンピュータの役割を果たす遊休コンピュータではgRPCサーバ、GamingAnywhereサーバおよびゲームが動作する。

プレイヤーPCと遊休コンピュータには表1に示したスペックのマシンを使用し、1Gbpsのリンクを持つネットワークで接続した。それぞれのマシンに2つのネットワークインターフェースを用意した。一つは管理用に使用し、もう一方はクラウドゲーミングのゲームデータの通信に使用している。ゲームデータ通信用のインターフェースに遅延の挿入や帯域制限をかけることにより、様々な環境のネットワークを再現して評価を行う。図8のように、EdgeVPNを起動するとコンテナ型の仮想環境プラットフォームであるDockerのコンテナが立ち上がり、それ

表 1 評価に用いたマシンのスペック

	遊休コンピュータ	プレイヤー PC
CPU	Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz	Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz
メモリ	32GB	16GB
NIC	Intel Corporation I350 Gigabit Network Connection (rev 01)	Intel Corporation I350 Gigabit Network Connection (rev 01)
OS	Ubuntu 20.04.1 LTS	Ubuntu 20.04 LTS
GPU	GeForce GTX 1060 6GB	GeForce GTX 1060 6GB

によってゲームデータ通信用のインターフェース上に仮想ネットワークインターフェースが立ち上がる。そこに仮想的な IP アドレスが割り当てられ、この仮想アドレスを持つインターフェースのペアで EdgeVPN のリンクが確立する。

EdgeVPN を利用しない条件で行う測定については、物理的なネットワークインターフェースに割り当てられている IP アドレスを用いてクラウドゲームサーバ/クライアントの接続を行い、直接通信をさせる。EdgeVPN を利用する条件での測定は、EdgeVPN によって生成された仮想ネットワークインターフェースに割り当てられている仮想 IP アドレスを用いて通信を行うことで、VPN 上における性能を測定する。

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 リンクに対する遅延の大小の影響

プレイヤー PC と遊休コンピュータ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。tc は、qdisc(Queueing Discipline) に対する設定等を行うコマンドである。qdisc はカーネルとネットワー

クドライバの間にあり、送信パケットの送信順序の変更、遅延、廃棄等の機能を有する。遅延を挿入する際に Listing2 のようにした。計測遅延と挿入遅延の値の差をプロットしたものが図 9 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。

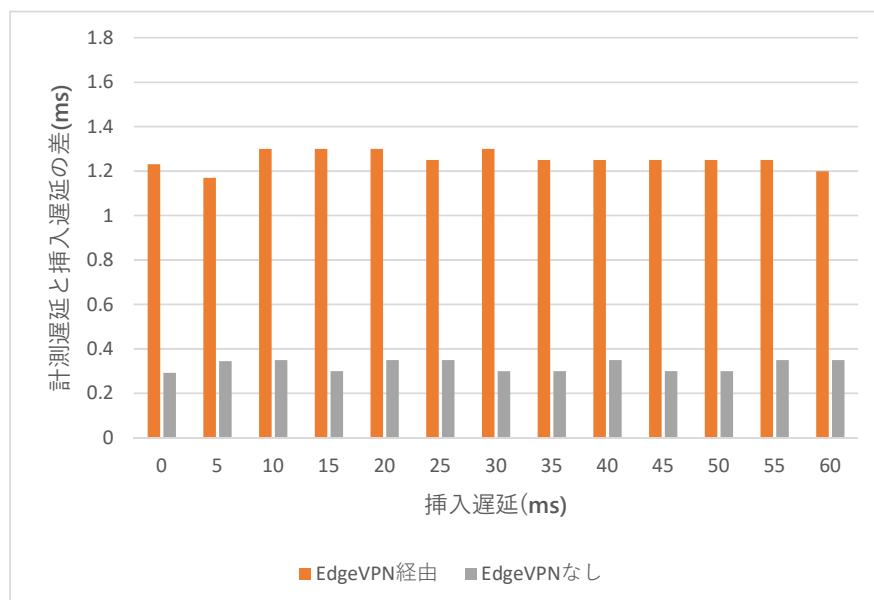


図 9 EdgeVPN リンクに対する遅延挿入の影響

Listing 2 遅延挿入

```
1 tc qdisc add dev brl101000F root netem delay 10ms
```

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

プレイヤー PC と遊休コンピュータ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スループット減少の様子を iperf[6] を使用して計測した。iperf はクライアント/サーバ間のネットワークの帯域幅を測定するツールである。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。その際クライアント側で Listing3 のようにした。EdgeVPN を利用する通信についての結果をプロットしたもののが図 10、直接接続する通信についての結果をプロットしたもののが図 11 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。

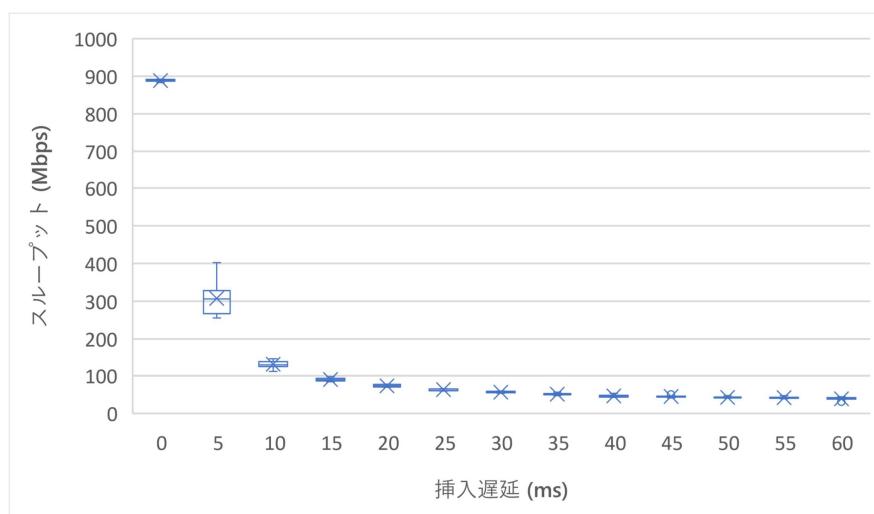


図 10 EdgeVPN リンクへの遅延挿入の帯域への影響

Listing 3 帯域計測

```
1 i=1; while [ $i -le 10 ]; do iperf -c 10.10.100.2 -t 60; i
   =$(expr $i + 1);done
```

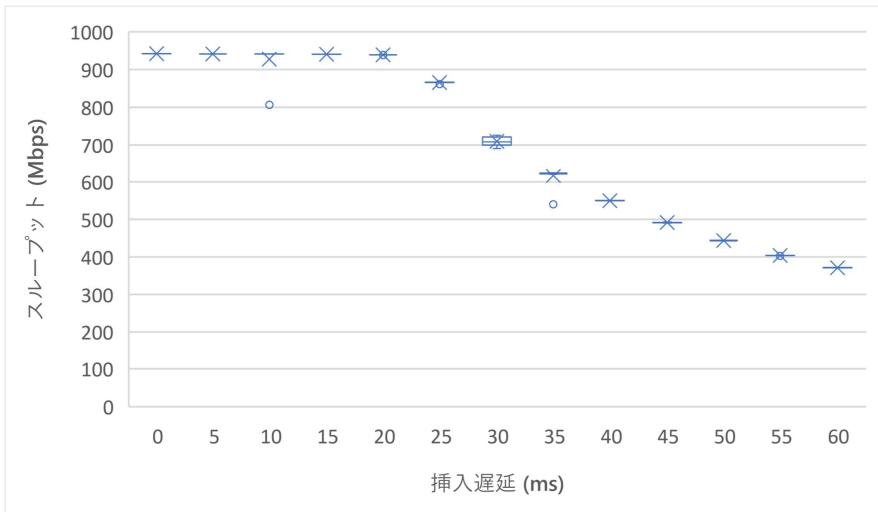


図 11 EdgeVPN を使用していないリンクへの遅延挿入の帯域への影響

5.2.3 10Gbps ネットワークにおける EdgeVPN のオーバーヘッド

ネットワークインターフェースが 1Gbps の環境では、スループットの上限が 1Gbps に制限されているので、挿入遅延が 0ms の場合に EdgeVPN のオーバーヘッドの影響が分かりづらい (図 10、11)。この結果は EdgeVPN による性能低下を正確に測定できていない可能性がある。

そこで、仮想環境にはなるが 10Gbps ネットワークが利用可能な環境における実験を行った。異なる物理マシンで動作する 10Gbps で接続された 2 つの VM を用意し、EdgeVPN でリンクを張るインターフェースと直接接続をするインターフェースをそれぞれ用意した。5.2.2 節と同様に遅延挿入を行ってスループット減少の様子を観測した。EdgeVPN を利用する通信についての結果は図 12、直接接続する通信についての結果は図 13 である。10Gbps ネットワークにおいては EdgeVPN を利用すると、遅延挿入を行っていなくても 550Mbps 程度までスループットが下がっていることがわかる。この値が 1Gbps ネットワークでの結果よりも低いのは、VM のオーバーヘッドによるものと考えられる。また、遅延挿入に伴つてばらつきはあるもののスループットの減少が見られ、60ms のネットワーク遅延がある状態では最低で 35Mbps 程度まで低下している。オーバーヘッドは大きいものの、10Gbps 環境においてもクラウドゲーミングのプレイに耐える程度の性能を

発揮していると言える。

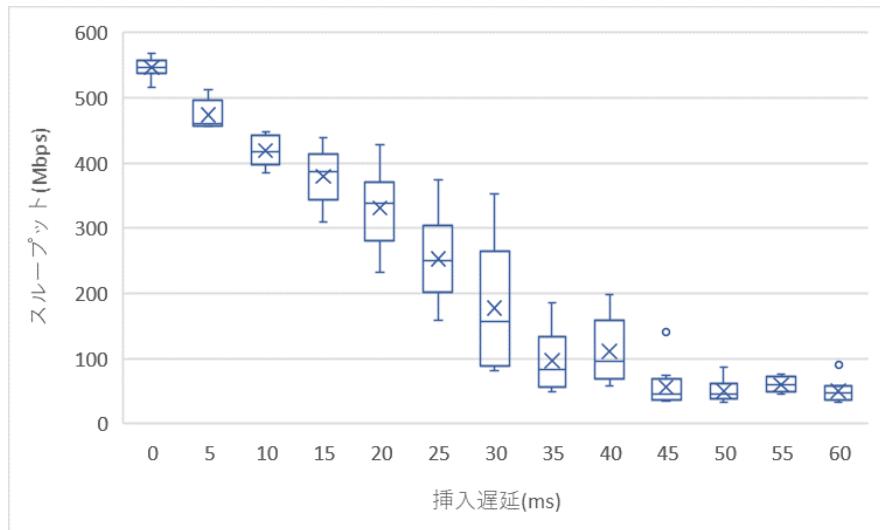


図 12 10Gbps ネットワークにおける EdgeVPN リンクへの遅延挿入の帯域への影響

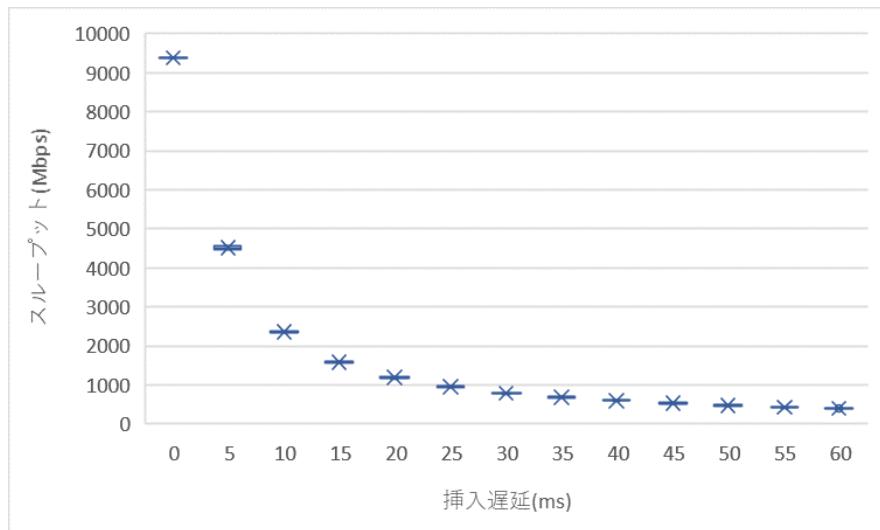


図 13 10Gbps ネットワークにおける EdgeVPN を使用していないリンクへの遅延挿入の帯域への影響

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、プレイヤーPCと遊休コンピュータ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。



図 14 提案システムを使用した Albion Online (MMORPG) のプレイの様子

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

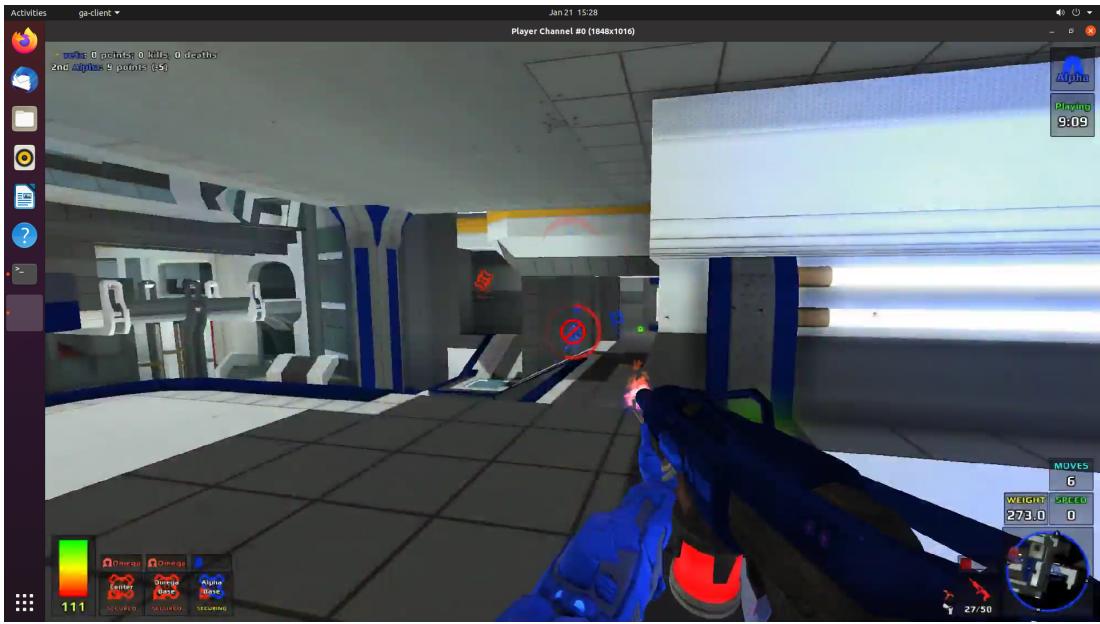


図 15 提案システムを使用した Red Eclipse 2 (FPS, Action) のプレイの様子

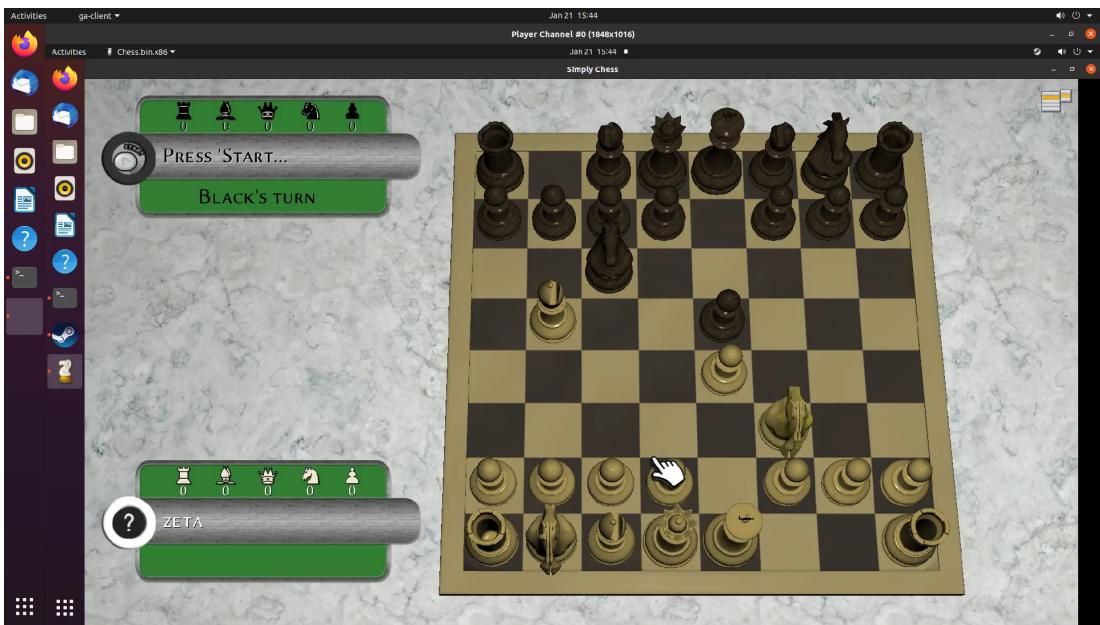


図 16 提案システムを使用した Simply Chess (ボードゲーム) のプレイの様子

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と10Mbpsの3つの条件で行った。Listing4のようにした。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17、Red Eclipse 2をプレイした際の結果を図18、Simply Chessをプレイした際の結果を図19にそれぞれ示した。

Listing 4 帯域制限

```
1      tc qdisc add dev brl101000F root tbf limit 10mb buffer 10kb
      rate 0.375mbps
```

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

本節では、提案システムによるオーバーヘッドを分析し、ボランティアコンピューティングを活用することによるクラウドゲーミングの実現可能性について考察を行う。

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。これにより、遊休コンピュータへの接続がクラウドデータセンターに接続する場合と比較して、1ms以上のネットワーク遅延の短縮が可能であるならば、提案システムが提供するQoEが既存のクラウドゲーミングシステムを上回ると言える。また、ゲームのフレームレートは高品質なゲームの場合でも60fps程度で

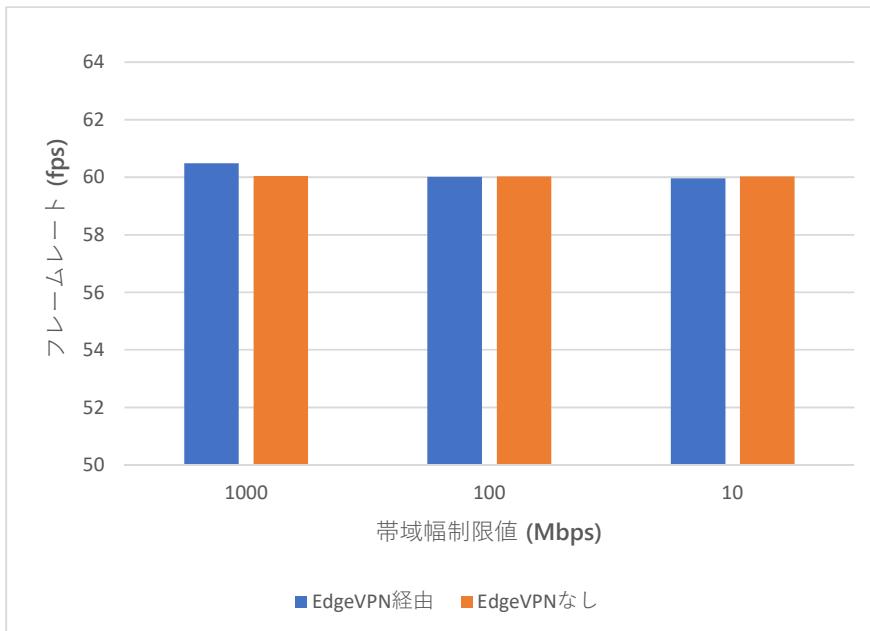


図 17 帯域制限下でのゲームプレイ時のフレームレートの変化 (Albion Online (MMORPG) プレイ時)

あり、このとき 1 フレームあたりの画面更新は約 17ms であるから、1ms 程度の遅延は十分に許容範囲である。提案システムでは、クラウドゲーミングプラットフォームにおけるレンダリングやエンコードの部分には修正を加えていない。1 章で述べたように、クラウドゲーミングの応答遅延はネットワーク遅延・レンダリング時間・エンコード/デコード時間で構成される。提案システムはこのうちネットワーク遅延の削減を行い、応答遅延全体もこの分だけ改善する。

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。

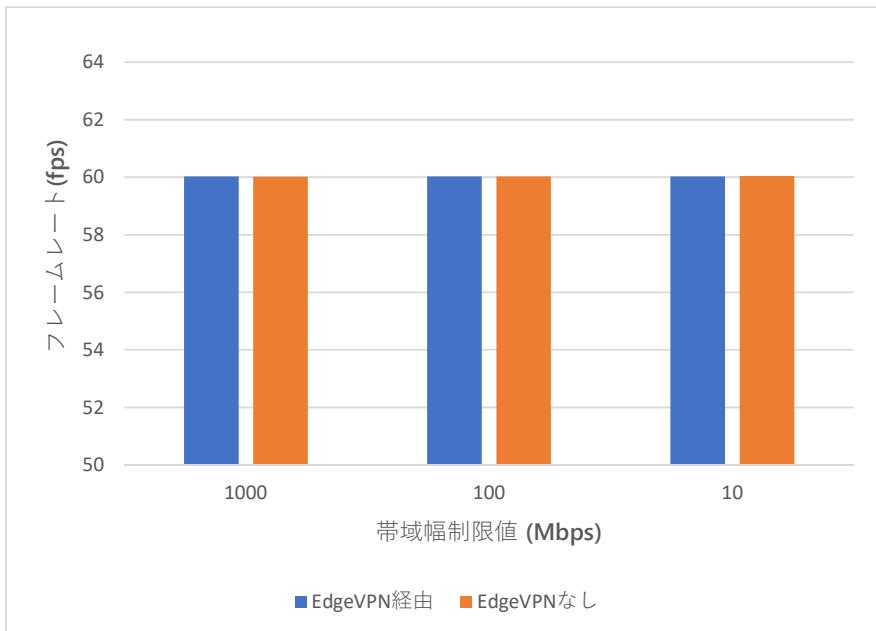


図 18 帯域制限下でのゲームプレイ時のフレームレートの変化 (Red Eclipse 2 (FPS, Action) プレイ時)

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、ボーダ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。

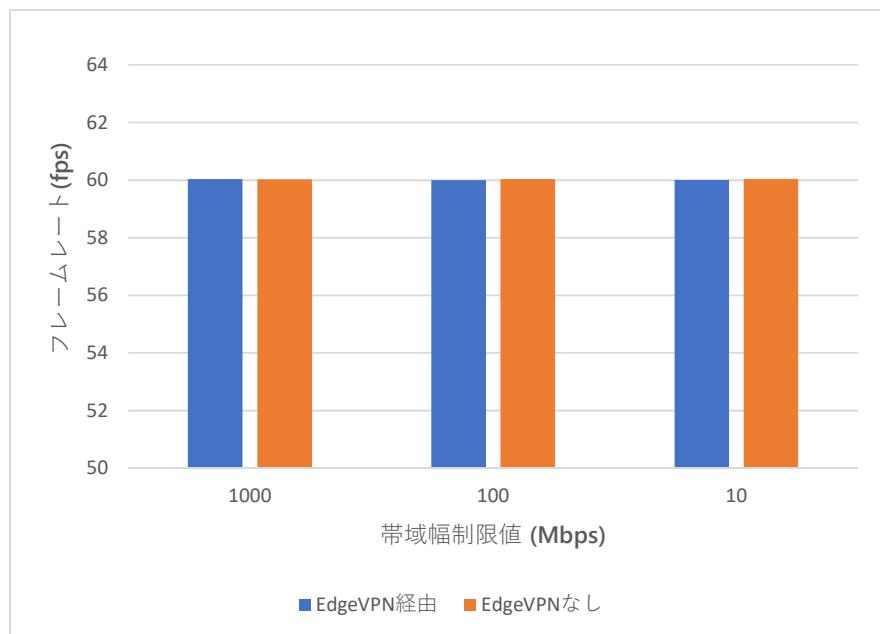


図 19 帯域制限下でのゲームプレイ時のフレームレートの変化 (Simply Chess (ボードゲーム) プレイ時)

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。この際、ゲーム毎にレンダリングに必要な時間に違いがある可能性についても考慮する必要が考えられる。高画質かつ高フレームレートのゲームなどのレンダリングやエンコードの負荷が大きいゲームの場合、総応答遅延が長くなる。これは相対的にネットワーク遅延が小さくなることを意味する。このとき、提案システムで QoE を維持するために、より近傍の遊休コンピュータを使用して総応答遅延を削減する必要が想定される。また、ゲームの種類によって許容される応答遅延に差がある可能性についても考慮が必要である。FPS のような低遅延が求められる対戦形式のゲームにおいては、遊休コンピュータのマッチングにおいてプレイヤー端末から最も近傍であることを優先しなければならない可能性がある。一方で、操作や対戦相手の行動が画面に反映されるまでにある程度の遅延が許容されるボードゲームのようなジャンルにおいては、マッチングにおいて近傍であることの優先度を下げるということも考えられる。

謝辞

本論文の執筆を進めるに当たり、多くの方々に御指導、御協力を賜りました。ここに御名前を記すと共に、深く感謝の意を示します。

主指導教員であり、本論文の審査委員を務めていただいた、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 飯田 元 教授には、研究の進め方を始めとして、研究の捉え方等、あらゆる面で非常に多くの御指導・御助言を賜りました。心より感謝を申し上げます。

副指導教員であり、本論文の審査委員を務めていただいた、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 藤川 和利 教授には、本研究の扱うべき課題や方針について非常に貴重な御意見を賜り、本研究の方向性を定めていく上で非常に大きな助けとなりました。心より感謝を申し上げます。

副指導教員であり、本論文の審査委員を務めていただいた、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 市川 昊平 准教授には、本研究の立ち上げから提案手法の立案、論文執筆に至るまで、様々な面で研究に関する御指導・御協力を賜りました。心より感謝を申し上げます。

副指導教員であり、本論文の審査委員を務めていただいた、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 高橋 慧智 助教には、研究の方針や提案システムの実装・評価など様々な面で御指導・御協力を賜りました。心より感謝を申し上げます。

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 ソフトウェア設計学研究室秘書の小川 晓子氏には、学生生活や TA/RA 業務における書類作成や諸連絡等、事務処理において多大なる御協力を頂きました。心より感謝を申し上げます。

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 ソフトウェア設計学研究室元秘書の仲田 綾奈氏には、学生生活や TA/RA 業務における書類作成や諸連絡等、事務処理において多大なる御協力を頂きました。心より感謝を申し上げます。

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 ソフトウェア設計学研究室の諸先輩方、同期生、後輩の皆様には、学生生活において公私共に

様々に御助力を頂き有意義な2年間を過ごすことができました。心より感謝を申し上げます。

LIPPS梅田ロフトのスタッフの皆様には、学生生活を送る上で自信を持てる姿を保つために多大なる尽力を頂きました。心より感謝を申し上げます。

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] Ian Fette and Alexey Melnikov. The websocket protocol, 2011.
- [19] The Linux Foundation. gRPC A high performance, open source universal RPC framework, 2020. <https://grpc.io/>.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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.

- [24] 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>.
- [25] Salvatore Loreto, P. Saint-Andre, Sd Salsano, and G. Wilkins. Known issues and best practices for the use of long polling and streaming in bidirectional http. *Internet Engineering Task Force, Request for Comments*, 6202(2070-1721):32, 2011.
- [26] 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.
- [27] Gurmeet Singh Manku, Mayank Bawa, and Prabhakar Raghavan. Symphony: Distributed hashing in a small world. USITS'03, page 10, USA, 2003. USENIX Association.
- [28] 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.
- [29] Peter Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Core. 2004.
- [30] Ryan Shea, Jiangchuan Liu, Edith C-H. Ngai, and Yong Cui. Cloud Gaming: Architecture and Performance. *IEEE network*, 27(4):16–21, 2013.

- [31] 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.
- [32] 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.
- [33] Dan Wing, Philip Matthews, Rohan Mahy, and Jonathan Rosenberg. Session Traversal Utilities for NAT (STUN). *RFC5389*, October, 2008.