

全天球カメラを用いたクラウド分散型インターネットライブ放送システムの評価

松本哲¹ 石芳正¹ 義久智樹¹ 川上朋也^{2,1} 寺西裕一^{3,1}

概要: 近年のインターネットライブ放送では、放送局や映像配信者が、映像効果を加えて視覚や音声の臨場感を向上させることがある。また、配信者や視聴者の端末の映像配信に伴う負荷を軽減するために、高い計算処理能力を有するサーバを用いて効果や処理を付加するため、商用のクラウドコンピューティングサービスを容易に利用できるようになった。我々の研究グループでは、カメラから得られた映像ストリームに対し、リアルタイムに映像処理を加えることで異世界から映像を配信しているかのように見える異世界放送システムを提案してきた。一方で、市販されているカメラデバイスの多様化にともない、一度に360度の視野の映像を撮影し、リアルタイムに映像ストリームとして符号化する全天球カメラが容易に入手できるようになった。これまでの異世界放送システムは従来のカメラで得られる映像のみを配信対象としてきたが、本研究では、全天球映像を取り込み、クラウドサービスで提供されている計算機を用いて多様な映像処理を加える異世界放送システムの評価について述べる。クラウド分散型異世界放送システムでは、視聴者は所望の計算処理サーバを選択し、撮影した映像データと効果に関する情報(タイプ、パラメータなど)を指定してビデオ効果や処理を行う。

An Evaluation of a Distributed Internet Live Broadcasting System Enhanced by Cloud Services with Omnidirectional Camera

Satoru Matsumoto¹ Yoshimasa Ishi¹ Tomoki Yoshihisa¹
Tomoya Kawakami^{2,1} Yuuichi Teranishi^{3,1}

1. 概要

近年のインターネットライブ放送では、放送局や映像配信者が、映像効果を加えて視覚や音声の臨場感を向上させることがある。また、配信者や視聴者の端末の映像配信に伴う負荷を軽減するために、高い計算処理能力を有するサーバを用いて効果や処理を付加するため、商用のクラウドコンピューティングサービスを容易に利用できるようになった。我々の研究グループでは、カメラから得られた映像ストリームに対し、リアルタイムに映像処理を加えることで異世界から映像を配信しているかのように見える異世界放送システムを提案してきた[1]。一方で、市販されているカメラデバイスの多様化にともない、一度に360度の視野の映像を撮影し、リアルタイムに映像ストリームとして符号化する全天球カメラが容易に入手できるようになった。これまでの異世界放送システムは従来のカメラで得られる映像のみを配信対象としてきたが、本研究では、全天球映像を取り込み、クラウドサービスで提供されている計算機を用いて多様な映像処理を加える異世界放送システムの評価について述べる。クラウド分散型異世界放送システムでは、視聴者は所望の計算処理サーバを選択し、撮影した映

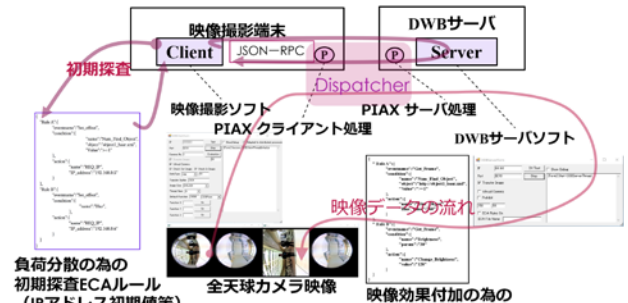


図1 ECAルールに基づくリアルタイムな映像効果付加処理フロー

像データと効果に関する情報(タイプ、パラメータなど)を指定してビデオ効果や処理を行う。

2. 関連研究

処理負荷を分散するための様々な方法が提案されている。しかし、それらの多くは、事前に負荷分散システムを構築する必要がある。インターネット放送における生放送を開始することは近年容易であり、映像配信端末の数は一般的には不特定となる。そのため、多数の映像配信端末の負荷分散システムを予め構築することが困難な場合が多い。映像録画端末以外のコンピュータを使用する分散型ビデオ処理システムについては幾つかの研究が行われている。

1 大阪大学
2 奈良先端科学技術大学院大学
3 情報通信研究機構

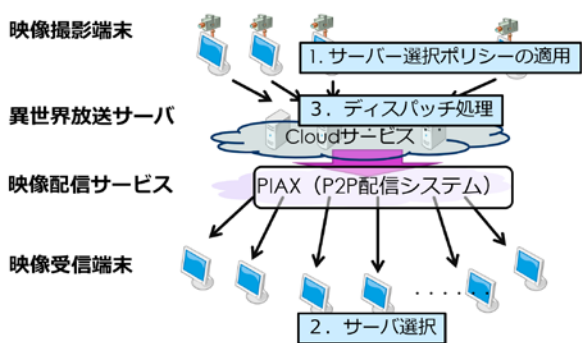


図 2 ECA ルールの選択ポリシーに基づく分散システム環境

生放送のための PaaS (Platform as a Service) タイプのクラウドコンピューティングサービスとして設計された MediaPaaS では、サービスプロバイダ側のコンピュータを使用してビデオをエンコード、再エンコード、および配信することが可能です[2]。MediaPaaS とは異なり、提案システムにおいては、P2P エージェントプラットフォームである PIAX [9]を用いた負荷分散システムを実装している。

提案方式では、映像記録端末が異機種放送サーバに接続する際に、異なる世界放送サーバから接続先をランダムに選択して接続を中断し、負荷を分散する。数多くの異世界放送サーバや映像録画端末に処理負荷を分散することで、ビデオ処理時間が短縮されることがこれまで著者らの実験により確認されている[1]。本研究では、より多くの異なる放送サーバを利用できるように、提案システムを実装し、商用クラウドサービスが提供する仮想計算機を用いて映像処理時間を評価した。

P2P ネットワークによって実装された、生放送のために最近録画されたビデオ (追跡再生) を見ることを可能にするビデオ配信システムでは[3]、放送事業者がビデオエフェクトの追加を選択肢から選択する事により簡易にできる。本研究においても、提案したシステムの特徴の 1 つとし、ビデオエフェクトの追加を選択肢から選択する事により簡易に可能としている。ライブインターネット放送の映像配信の遅延時間を短縮する方法として、いくつかの方法が提案されている。SmoothCache2.0[4]を用いたライブ放送では、他のピアからのビデオデータをキャッシュし、キャッシュされたピアから P2P ネットワークを用いて配信することにより、映像録画端末の通信負荷や遅延時間を削減することが可能である。Dai 他においては、遅延時間を最小にする P2P ネットワーク分散経路決定法を提案している[5]。提案された HD 方式では、[6]で一对一の通信を組み合わせた 1 対多数の放送配信を用いて、複数の視聴端末に同時に

映像データを送信することで、通信が減少する。提案システムであっても、映像を配信する際にこれらの遅延削減手法を適用することができるが、我々の研究は映像処理システムの設計が異なる。さらに、蓄積されたビデオデータに対してビデオ処理を行うシステムに関する研究が議論されている。Gibbon 他においては、カメラで撮影した映像データを処理能力の高いコンピュータに転送して映像処理を行うシステムを提案した[7]。Ting らは、低処理能力のコンピュータ (例えば、スマートフォン) によって捕捉された画像をクラウドストレージなどの外部記憶装置に直接記憶するシステムを提案した[8]。しかし、これらのシステムは蓄積されたビデオデータをターゲットとし、ライブインターネット放送には適用できない。

3. 天球カメラを用いたクラウド分散型インターネットライブ放送システム

3.1 システム概要

クラウド分散型異世界放送システムでは、視聴者と配信者が煩雑な手順を踏むことなく、あらかじめ記述されたルールに基づいて自動的に計算資源を選択する [1]。クラウドサービスとして、例えば、Windows Azure サービスで提供される仮想計算機を異世界放送サーバとして用いることが考えられる。異世界放送サーバへのデータの送受信には P2P ネットワークを用いている。高負荷の異世界放送サーバに処理要求が集中しないように、負荷を分散している。ECA ルールはクラウド分散型異世界放送システム上の計算処理サーバに持たせる。これら、分散処理と映像効果設定の為の ECA ルールを用いたリアルタイムな映像付加処理の処理フローを図 1 に示す。

3.2 想定環境

本研究で想定する環境を図 2 に示す。想定環境には、映像録画端末、異世界放送サーバ、およびビデオ受信端末の 3 種類の端末またはサーバがある。録画端末は、所望のビデオエフェクトを実行可能な異世界放送サーバを選択し、ビデオエフェクトライブラリと録画済みビデオを送信する。クラウドサービスの仮想マシン上で動作する異世界放送サーバは、端末の要求に応じて、映像記録端末から送信された映像に対して映像処理を実行する。異世界放送サーバで処理された映像は、映像配信サービスを介して映像受信端末に配信される。ビデオ受信端末は、ビデオ配信サービスのサーバまたはチャンネルを選択した後、処理されたビデオを受信する。

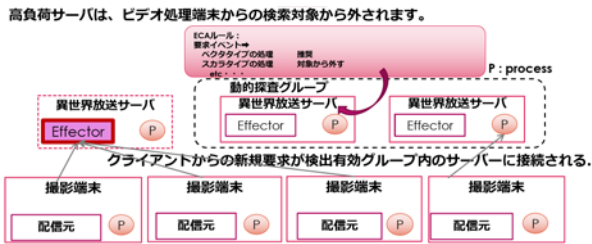


図3 録画端末と異世界放送サーバの内部構成

本研究では、異世界放送サーバの負荷分散を考慮したうえで録画端末が異世界放送サーバを選択する。従来提案されているシステムでは、ロードバランサ等の負荷分散機構を介して映像記録端末を接続することにより負荷分散を実現している。この方法では、負荷分散機構は、ビデオが送信されている間に別の異世界放送サーバに切り替える必要がある場合に、映像録画端末が使用していない異世界放送サーバを選択する。例えば、表示する場面または効果に適した異世界放送サーバに切り替える場合、映像録画端末は、既存の接続を終了させるが、ここにおいて確実な計算性能保証を持った再接続なしでは再開できず、映像が途切れる事がある。このため、映像配信を継続しながら、異種の計算性能（特性）を持った放送サーバ間をスムーズに切り替えることは困難とされる。つまり、負荷分散機構が映像記録端末からの要求に応じて異世界放送サーバを変更した場合でも、通信データを単純なバイトストリームとして中継することで、データ構造（例えば、ビデオフレームの境界）が無視され、機構がサーバを切り替えるときにビデオデータが破損する可能性があることを意味する。そこで、本研究では、負荷分散機構は、映像記録端末の要求に基づいて、異世界放送サーバを選択するだけであり、映像記録端末は、異世界放送サーバのいずれかに処理を要求する際に、ECAルールを用いてシームレスに接続できるようにした。このようにして、映像記録端末は、シーンまたはビデオフレームの境界が来ても、いつでも異世界放送サーバを切り替えることができる。

3.3 分散システム構成

図3に、録画端末と異世界放送サーバの内部構成を示す。録画端末には録画端末ソフトウェアとクライアント側PIAXシステムがインストールされる。異世界放送サーバソフトウェアとサーバ側のPIAXシステムが異世界放送サーバにインストールされる。PIAX [9]は、オーバーレイネットワークの検索機能を利用してサーバレスで効率的なリソース検索を可能にするJavaベースのプラットフォームミドルウェアです。PIAXはオープンソースソフトウェアとして提供されている。映像録画端末のPIAXシステムと異世界放送

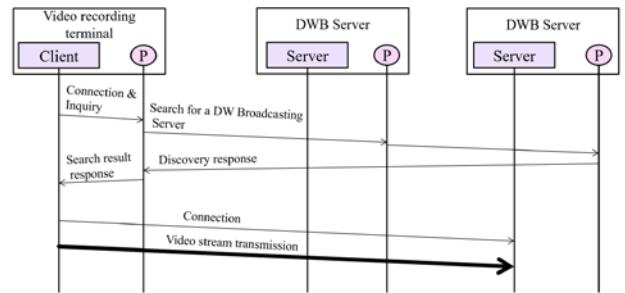


図4 通信手順のデータフローとタイミングチャート

サーバは、オーバーレイネットワークを介して接続することができる。録画端末のクライアント側PIAXシステムは、録画端末ソフトウェアの要求に応じてオーバーレイネットワークを検索する。システムは、異世界放送サーバのホストから異世界放送サーバを選択し、選択されたサーバのIPアドレスおよびサーバソフトウェアのリソース番号を返す。この応答に基づいて、録画端末ソフトウェアは異世界放送サーバソフトウェアとの接続を確立し、ビデオの送信を開始する。JSON-RPCは、録画端末ソフトウェアとクライアント側のPIAXシステム間の通信にTCP経由で直接使用される。

異なる世界放送サーバ上のサーバ側PIAXシステムは、異なる世界放送サーバソフトウェアからの指示に基づいて、その検索状態を決定する。異世界放送サーバが高負荷状態であると見なされると、ローカルホスト上のサーバ側PIAXシステムの制御ポートに接続し、検索のための「離脱」要求コマンドを送信することにより、異なる映像放送端末からの更なる接続を受信しない。異世界放送サーバの負荷が軽減され、サーバが新たな接続を受け入れるマージンを得た場合、放送サーバはローカルホスト上のサーバ側PIAXシステムの制御ポートに再接続し、「離脱」要求を送信するコマンド。このコマンドは、サーバ側のPIAXシステムは再び映像録画端末からの検索受諾状態に入る。

以上の動作により、異世界放送サーバの負荷状態に基づいて録画端末からの新たな接続が制御されるため、異世界放送サーバソフトウェアの過負荷を回避することができる。図4に通信手順のデータフローとタイミングチャートを示す。

3.4 ECA ルールの設計

イベント、条件、およびアクションのリストは、前述の研究[1]で説明されている。このリストに加えて、クラウドコンピューティングサービスを使用するためのシステムを強化するために、1つのイベントと1つのアクションを追加する。新しいビデオエフェクトがシステムで使用されると、「Set_effect」イベントが発生する。「Req_IP」アクションは、最初に、条件部で指定されたビデオエフェクトを

追加する異世界放送サーバの IP アドレスを設定する. 指定されたサーバの負荷が重い場合, アドレスは PIAX によって自動的に変更される. 図 5 は, 具体的な ECA ルールセットを示している. この例では, 「Num_Find_Object」, 「Spherical_coordinates_Convert」という名前の映像録画端末からのビデオ処理要求に対して, それぞれ, 初期値として 192.168.0.5 と 6 の IP アドレスのサーバが割り当てられている.

3.5 分散システムの実装

Microsoft Azure をクラウドサービスとして使用して, 異機種放送システムを展開した. 異世界放送サーバは, Azure サービスが提供する仮想マシン上で動作する. 各仮想マシンは, Microsoft Azure によって提供されるサービスの 1 つである仮想ネットワーク (VNet) によって論理的に接続されている. 図 6 は, サーバソフト・クライアントソフトのインターフェースを示す.

ビデオエフェクト追加プロセスを開始するとき, 異世界放送サーバソフトウェアのインターフェースを提供する. 図 9 は, 映像録画端末にビデオを配信する映像録画端末ソフトウェアのユーザインターフェースを示す. 選択したエフェクトを適用した結果を視覚的に認識できる.

異なる世界放送サーバダイアログボックスの「ECA ルールオン」チェックボックスにチェックが入っていれば, ユーザの操作なしに異世界放送サーバ ECA ルールに基づいて画像効果が自動的に追加される.

映像録画端末ソフトウェアには, 映像処理を要求する異世界放送サーバの IP アドレスが保持されている. 端末ソフトウェアは, 録画端末ソフトウェアダイアログボックスの「分散処理の適用」チェックボックスにチェックが入っていれば, IP アドレスを用いてプルダウンメニューで指定された映像処理を異世界放送サーバに要求し, 要求は, クライアントソフトウェアによって保持される ECA 規則に基づいて, 負荷分散機構を介して異世界放送サーバに送信される. 処理要求は, 処理負荷の低い異機種放送サーバに送信される. この場合, ECA ルールは, 負荷分散メカニズムにその実行を要求するために, 初期 IP アドレスを記述する. 本研究で実装した上記システムの処理フローは 1 章の図 1 に従う.

3.6 全天球カメラ画像の分散処理

全天球カメラを用いると, 360 度パノラマ画像を作成する場合等に, ある視点からの全方位の写真を数十枚撮影してコンピュータ上で合成処理する必要がなくなり, 一度の撮影でパノラマ画像が作成でき, リアルタイムに全方位の画像情報が得られる. この度用いる全天球カメラによる画像は, 魚眼レンズが前後に合わさった, 前後 180 度ずつの図 6 左下部のような 360 度の視野の画像が得られる. これらの映像を切り出して平面画像に変換する必要がある, [11] で説明されている様々な手法がある. 本研究での実装

```

{
  "Rule A": {
    "eventname": "Set_effect",
    "condition": {
      "name": "Num_Find_Object",
      "object": "object1_haar.xml",
      "Value": ">=1"
    },
    "action": {
      "name": "REQ_IP",
      "IP_address": "192.168.0.5"
    }
  },
  "Rule B": {
    "eventname": "Set_effect",
    "condition": {
      "name": "Spherical_coordinates_Convert",
    },
    "action": {
      "name": "REQ_IP",
      "IP_address": "192.168.0.6"
    }
  }
}

```

図 5 ECA ルールの例

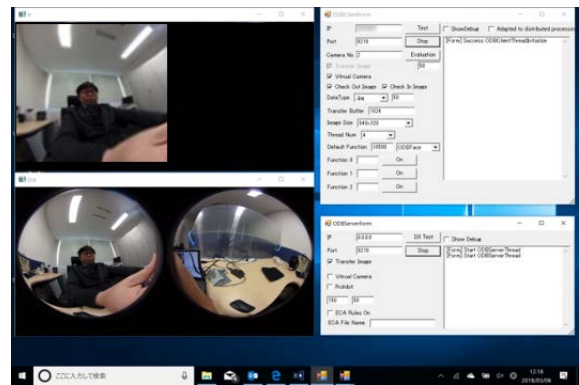


図 6 サーバソフト・クライアントソフトのインターフェース

は魚眼レンズで得られた仮想半球状の映像に接する平面に対して射影して得られた映像を切り出し, 図 6 の左上部のような画像として閲覧するシステムとした. 仮想半球状の極座標から直交座標に変換する処理や多くの全天球カメラが高い解像度の映像をリアルタイムに配信する事から, 多くの計算資源を一度に扱う為, 変換時に映像が途切れる場合がある. この度は異世界放送サーバを用いて極座標・直交座標変換処理の分散処理を行った.

4. 評価

Microsoft Azure サービスが提供する仮想マシン上に構築された実装システムを使用して, 提案された方法の有効性を評価した. 評価方法, 環境, 結果は以下に述べる.

4.1 評価方法

実験では, RICOH 社製 Theta S を全天球カメラとして用いた. 動画映像の各 1 フレームを jpeg 形式で符号化して送信し, RICOH THETA UVC Blender 経由で USB の仮想カメラ映像として送信する. 極座標・直交座標変換処理等の各フレームの映像効果を分散処理し, 異世界放送を実現させる. 評価では, 人物が撮影されている映像の顔認識によ

り得られた人物周辺を矩形で囲った際の矩形内の画像データをデータベースへ登録するまでのターンアラウンドタイムを計測し、分散処理により短縮できるかを確認する。

提案システムの効率性を評価するために、ECA ルールの処理時間を含めたビデオ処理時間とターンアラウンドタイムを評価指標として取り上げた。

著者らは次の2点について評価した、

(1) ビデオエフェクト処理要求が ECA ルールを使用せずにビデオエフェクト処理サーバに集中している場合のターンアラウンドタイムの比較評価を行った。

(2) 映像効果処理要求が集中せず、ECA ルールが採用された。評価のために、クラウドサービスと同じパフォーマンスパラメータを持つ複数のコンピュータを割り当てた。

利用可能な異世界放送サーバを選択するには、前章で説明した PIAX オーバレイネットワークを使用した。特定の異世界放送サーバが過負荷になった場合、サーバはサーバ側の PIAX プロセスに通知を送信し、負荷が低くなるのを待つ。評価のターンアラウンドタイムは2つのケースで測定した。1つは、3つの映像録画端末が3つの異世界放送サーバのうちの1つに要求を出した集中的なケースである。もう1つは完全に分散されたケースで、3つの映像録画端末のそれぞれが3つの独立した異世界放送サーバへとサービスを要求したケースである。

この評価ではライブ放送を使用した。評価には、ECA ルールに記載されたビデオエフェクトとして、前述の極座標・直交座標変換後に顔検出ビデオ処理を施した処理を採用した。フレームデータの送受信に要した時間をターンアラウンドタイムと定義した。これには、以下の a) ~ d) の4つの項目が含まれる。

a) クライアントがビデオデータを取得する前処理時間(前のフレームデータ受信の終了から次のフレームデータ送信の開始までの時間に等しい)。

b) 異世界放送サーバがフレームデータを受信している間の通信時間。

c) 異世界放送サーバでの処理時間。

d) 異世界放送サーバからのフレームデータをクライアントが受信する通信時間。映像処理時間は、映像データ受信開始時刻を除いた映像処理開始から処理終了までの時間として定義される。

4.2 評価環境

この評価では、異なる世界のブロードキャストサーバが Microsoft Azure サービスによって提供される仮想マシンで動作した。表1に仮想マシンの仕様と OS を示す。異世界放送サーバには5つの異なる仮想マシンを使用した。インテルの並列コンピューティング・ライブラリ TBB [10]によって並列化された Open CV は、異世界放送サーバでビデオ処理を実行するためのライブラリとして使用した。映像録画端末は、大阪大学に設置された PC である。表2

に、録画端末 PC の仕様を示す。2台の PC を使用し、映像録画端末ソフトウェアを一度に2つのプロセスで起動することにより、映像録画端末ソフトウェアを合計4つのプロセスで動作させた。うち、1台だけに全天球カメラを付けた。これらの PC は、異なる家庭用光回線網を介して異なる世界放送サーバにアクセスし、ネットワーク (NTT, フレッツ光による KT Opticom によるフレッツ光) の輻輳を回避した。

4.3 評価結果

評価結果として、分散処理によりターンアラウンドタイムの短縮につながった。その結果、異世界放送サーバ間で ECA ルールに基づいて処理が割り当てられ、負荷を分散できることを確認した。

図7および図8は、4.2節で説明した評価環境下でのターンアラウンド時間の評価結果を示す。横軸は記録されたフレーム番号を示し、縦軸はターンアラウンド時間(ミリ秒)を示します。負荷が単一の異世界放送サーバに集中している図7では、ターンアラウンドタイムは2500msecの物も目立ち、なだらかに直線的に周期的に増加する。また、約7500msecの遅れた処理が発生し、周期的に4500msec近辺の遅れた処理が数回発生している。図8では、3つの異世界放送サーバに映像処理要求が分散し、1500msecまでターンアラウンドタイムが抑えられており、約500フレームの処理中7500msec近辺の処理遅れが1度のみであった。

また、PC2が映像効果処理を要求した異世界放送サーバは Microsoft Azure の実環境では東日本リージョンに存在する VM であった。処理要求元の端末は西日本からのものであり、経路上の他の要因か、処理のターンアラウンドタイムにばらつきが見受けられた。

表1 仮想マシン仕様と OS

OS	Microsoft windows Server 2016	
Microsoft Azure Plan	Standalone Server Microsoft Corporation Virtual Machine x64-based PC	
CPU	Intel E5-2697 v3 Equivalent 2.4GHz	
Main memory	3.584MB	

表2 録画端末 PC (クライアント) の仕様

OS	Client PC 1~3	Microsoft Windows 10 Pro Version 1709,1511
	Client PC1	Intel i7-7660U Equivalent 2.5GHz
	Client PC2	Intel i5-6300U Equivalent 2.4GHz
CPU	Client PC3	Intel i3-4020Y Equivalent 1.5GHz
	Client PC1	8.00 MB
	Client PC2	8.00 MB
Main memory	Client PC3	4.00 MB

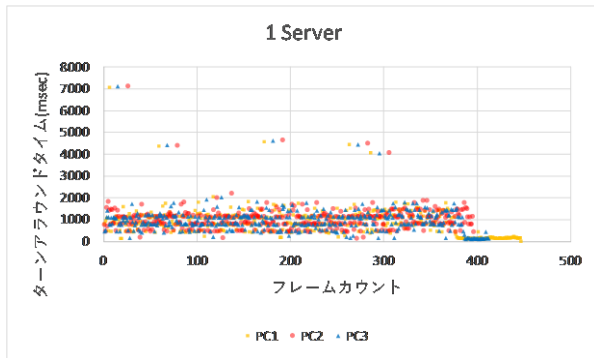


図7 外部クラウドサーバー1台の時の
ターンアラウンドタイム

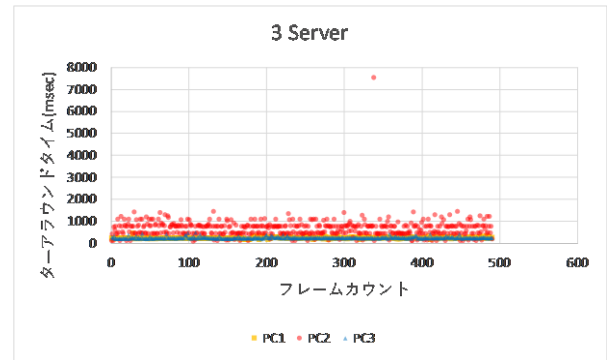


図8 外部クラウドサーバー3台の時の
ターンアラウンドタイム

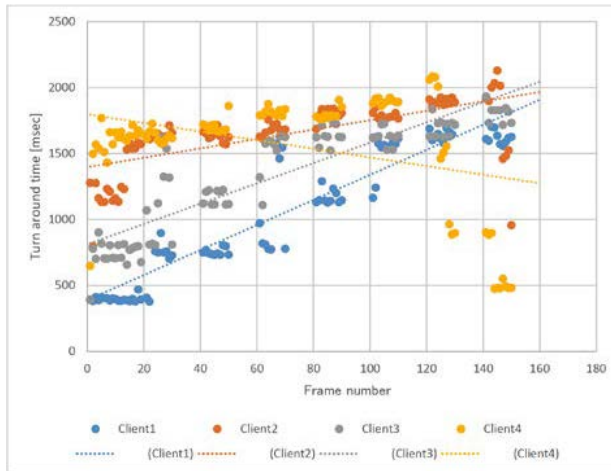


図9 外部クラウドサーバー1台の時の
テストパターンターンアラウンドタイム

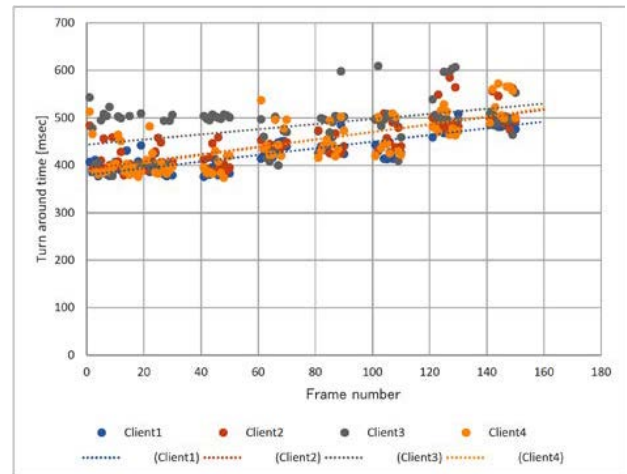


図10 外部クラウドサーバー4台の時の
テストパターンターンアラウンドタイム

た、推奨されている異世界放送サーバに問い合わせるのに必要なターンアラウンドタイムも測定した。1回の問い合わせの平均値は、50回の試行で16.28ミリ秒であった。

その結果、処理要求はECAルールに基づいて異世界放送サーバに割り当てられ、負荷が分散されることが明らかになった。また、通信遅延などの影響で仮想マシンのハードウェア性能が同等であっても、ターンアラウンドタイムが変動する可能性があることを確認した。

5. 評価についての考察

この評価では、計算負荷が3台のサーバに分散されている場合でも、1台のサーバにビデオ処理が集中することがあった。

評価には2つのネットワークを使用した。(NTTのフレッツ光, K-Opticomのeo光)。映像録画端末からの要求が1つの異世界放送サーバに集中するようになると、異世界放送サーバに処理負荷が分散された。アクセス時間を比較すると、ターンアラウンドタイムは比較的長くなった。

5.1 評価結果について

評価では、ビデオ処理は、ECAルールに記載された指定

された効果で、ビデオ内の人物の顔を検出していた。以前の実験と同様に、人間の顔検出による負荷を徐々に増加させ、ターンアラウンドタイムを測定した。フレームデータを送受信するのに必要な時間をターンアラウンドタイムと比較した。そのテストパターンを用いた結果を図9及び図10に示す。クラウドコンピューティングサービスの仮想マシン間でターンアラウンドタイムの変動が確認された。これは、ネットワーク距離のクラウド環境の違いで実際のサーバのパフォーマンスの変動などの要因によって発生する。ユーザがシステムを構成するときには、これらの問題を検討する事を推奨する。また、ECAルール定義にはさらなる改善が必要である。

5.2 ECAルールについて

ビデオエフェクトには手順があり、例えば、顔検出処理は、一般に、検出された顔のモザイク効果の前に実行される。提案されたシステムにおけるECAルールの「Timer」または「Message」機能は、そのような手順を定義することができる。その手順がECAルールで定義され、ECAルールに順序依存性がある場合、システムは順序に沿ってそのルールと処理要求を同時に実行する必要がある。それ以外の、ECAルールが処理要求に依存しない場合、システムは

ルールを並行して実行することができ、順序依存の関係を
持つ ECA ルールと比較して処理時間を短縮できる。現在の
システムでは、ECA ルールを並行して処理することが
できず、クラウドコンピューティングサービスに対する並
列処理が今後の課題となる。

6. まとめ

本研究では、全天球映像を取り込み、クラウドサービス
で提供されている計算機を用いて多様な映像処理を加える
分散型ライブインターネット（異世界）放送システムの実
装と評価を行った。ECA ルールで処理割り当てを記述す
ることにより、ビデオ処理に仮想マシンのパフォーマンス
を柔軟に割り当てることができた。実装されたシステムで
は、PIAX プラットフォームを使用して仮想マシンの検索
と通信を行い、仮想マシンの数が変化してもライブイン
ターネットブロードキャスト中にビデオ処理を継続的に割
り当てることができた。システム評価により、提案されたシ
ステムを使用してビデオ処理のターンアラウンドタイムを
短縮できることが確認された。

今後は、ECA ルールを階層化し、端末からサーバ間、サ
ーバ内部処理、サーバから端末間のそれぞれのフェーズの
データフローに対してルールを明確にし、エッジコンピ
ューティング処理とクラウドコンピューティング処理を適材
適所で連動させる予定である。

謝辞

本研究の一部は、文部科学省科学研究費補助金・基盤研
究(B) (15H02702)および、本研究の一部は NICT・大阪大学
共同研究「大規模エッジコンピューティングのための高信
頼ネットワークプラットフォーム」による成果である。こ
こに記して謝意を表す。

参考文献

- [1] 松本哲, 義久智樹, 石芳正, 川上朋也, 寺西裕一: “全天球
映像処理を伴うクラウド分散型インターネットライブ放送シ
ステムの設計と実装,” 情報処理学会第 80 回全国大会論文集, 4B-1, pp.
1-357-358 (Mar. 2018).
- [2] B. Cheng, “MediaPaaS: A cloud-based media processing platform
for elastic live broadcasting,” Proceedings of the 7th IEEE International
Conference on Cloud Computing (CLOUD 2014), pp. 713-720 (2014).
- [3] Y. Gotoh, T. Yoshihisa, H. Taniguchi, and M. Kanazawa,
“Brossom: a P2P streaming system for webcast,” Journal of Networking
Technology, Vol. 2, No. 4, pp. 169-181 (2011).
- [4] R. Roverso, R. Reale, S. El-Ansary, and S. Haridi, “Smooth-Cache
2.0: CDN-quality adaptive HTTP live streaming on peer-to-peer
overlays,” Proceedings of the 6th ACM Multi-media Systems
Conference (MMSys 2015), pp. 61-72 (2015).
- [5] J. Dai, Z. Chang, and G.S.H. Chan, “Delay optimization for
multi-source multi-channel overlay live streaming,” Pro-ceedings of the
IEEE International Conference on Commu-nications (ICC 2015), pp.
6959-6964 (2015).
- [6] T. Yoshihisa and S. Nishio, “A division-based broadcasting method

considering channel bandwidths for NVoD services,” IEEE Transactions
on Broadcasting, vol. 59, no. 1, pp. 62-71 (2013).

- [7] D. Gibbon and L. Begaja, “Distributed processing for big data
video analytics,” IEEE ComSoc MMTC E-Letter, vol. 9, no. 3, pp.
29-31 (2014).
- [8] W.-C. Ting, K.-H. Lu, C.-W. Lo, S.-H. Chang, and P.C. Liu,
“Smart video hosting and processing platform for Internet-of-Things,”
Proceedings of the 2014 IEEE International Conference on Internet of
Things (iThings 2014), pp. 169-176 (2014).
- [9] M. Yoshida, T. Okuda, Y. Teranishi, K. Harumoto, S. Shimojo,
“PIAX: A P2P Platform for Integration of Multi-overlay and
Distributed Agent Mechanisms,” Transactions of Information
Processing Society of Japan / Information Processing Society of Japan,
Vol. 49, No. 1, pp. 402-413, (2008).
- [10] Thread Building Blocks,
<https://www.threadingbuildingblocks.org/>,
(referred October 1, 2017).
- [11] 庄原 誠, 佐藤 裕之, 山本 勝也: “2. 全天周撮影,” 映像情
報メディア学会誌 69(9), pp.652-657(2015)