

グラフ表現型画像処理プログラミングを用いたクラウド分散型 インターネットライブ配信システム

松本哲¹ 義久智樹¹ 川上朋也^{2,1} 寺西裕一^{3,1}

A Cloud-Distributed Internet Live Distribution System using Image Processing Programming Formed by Graphs

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

1. はじめに

インターネットの普及に伴い、YouTube や TwitCasting といったインターネットライブ配信が広く利用されている。近年のインターネットライブ配信では、背景にぼかしをかけたり、物体にアノテーションを表示するといった映像効果（音声効果を含む）を付加することがある。例えば、自宅から配信する場合に、配信したくないものが映っている背景をぼかししたり、映像の顔付近に注釈で氏名を表示させるといった映像効果を付加することがある。

インターネットライブ配信では、処理時間が長すぎると、フレームレートが低下したり、映像が途切れるといった問題が発生する。これらの問題が発生すると、視聴者は快適にインターネットライブ配信を視聴できないため、処理時間を短縮することが求められる。処理時間を短縮するために、計算能力の高い計算機（クラウド計算機）で映像効果を付加することで、配信者の計算機（配信端末）で処理を行う場合と比べて処理時間を短縮するクラウド分散型インターネットライブ配信システムが提案されている。クラウド計算機とは、Microsoft Azure や Amazon AWS といったクラウドサービスで提供される計算機であり、一般に利用できる計算能力の高い計算機である。

しかし、クラウド計算機に映像データを送信する時間が余分にかかるため、映像効果付加に伴う処理量が少ない場合には配信端末で処理を行う方が短時間で映像効果を付加することがあった。クラウド計算機および配信端末で映像効果付加を行った場合の処理時間と、クラウド計算機に映像データを送信するのにかかる時間をあらかじめ把握することで、短時間で処理できる方法を選択できる。クラウド計算機に映像データを送信するのにかかる時間は、通信帯域等から概算できるものの、処理にかかる時間を把握するためには、処理内容を把握する必要があり、非常に困難である。画像処理の幾つかは、プログラミングを簡素化し処理分散を行いやすくするために、グラフで映像処理手順を表現（グラフ表現型画像処理プログラミング）している。

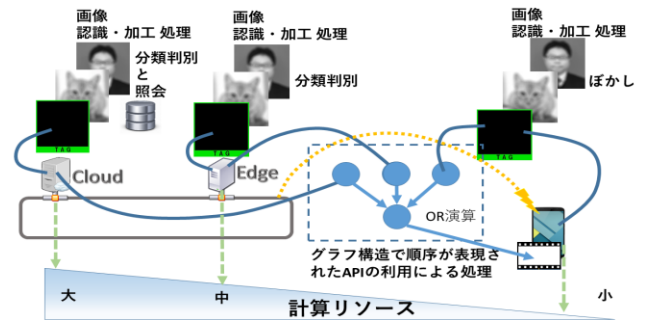


図1. 配信システムの構成図

グラフ表現型画像処理プログラミングでは、映像効果の手順をあらかじめ把握できるため、クラウド計算機で処理する方が早く行える映像処理を把握しやすい。

そこで本研究では、グラフ表現型画像処理プログラミングを用いたクラウド分散型インターネットライブ配信システムを提案する。提案システムの構成図を図1に示す。クラウド計算機の計算リソースが大きく、配信端末との間の計算機（エッジ）は中程、配信端末の計算リソースは小さい場合を想定している。提案システムでは、グラフのノードで表現された各映像処理を、各フレームの映像効果の付加にかかる時間が短くなるように処理する端末を決定する。本研究では、グラフ表現型画像処理プログラミングとしてOpenCVのGraph APIを利用し、クラウド計算機で映像処理を行った場合の計算時間と、配信端末で映像処理を行った場合の計算時間を比較し、グラフ表現型画像処理プログラミングの映像処理時間について評価を行う。

以下、2章で関連研究について説明し、3章でグラフ表現型画像処理プログラミングを用いた処理分散の設計を行う。4章で実験環境と評価について説明し、5章で本論文をまとめる。

2. 関連研究

クラウド、エッジ、またはフォグコンピューティングモデル等[1-11]、多くの負荷分散処理システムが提案されている。Ningらがモバイル向けストームとしてリアルタイム

¹ 大阪大学

² 福井大学

³ 情報通信研究機構

ムストリーム配信処理システムをクラウド利用により提案した[7]。著者らの研究グループでは、YouTuber 等による配信映像では、顔の位置に余り変化が無い事に着目して、1 フレーム中の顔画像を探索する面積の範囲を絞り、映像処理の安定化を目指した研究を行った(詳細は[11])。本稿では、映像中で認識されたオブジェクトについて、[11]の高速化手法とは異なり、詳細な分類として処理の詳細を把握する目的で切り出して、エッジサーバやクラウドサーバに送信する。一度目の顔認証の後、より詳細な情報を得るという順列を伴った処理を行い、かつ、各サーバで処理した適切な部分処理後の画像を受け取り、同期・合成するために、グラフ構造で順序が表現される API を利用して、サーバ連携の効果を確かめる。

また、著者らはコンテンツに応じたプライバシー保護を行うプライバシー指向インターネットライブ配信システムの構築を目指し、映像中のコンテンツに応じた画像処理についての高速化手法の研究を行った。コンテンツに応じたプライバシー指向の映像配信は、非常に困難と考えられていた ([12, 13, 14])。本研究では、著者らは順列を持った画像処理をシームレスにクラウド分散型のシステムで高速処理する事により、上記問題を解決している。

3. グラフ表現型画像処理プログラミングを用いた処理分散

3.1 従来システムの問題点

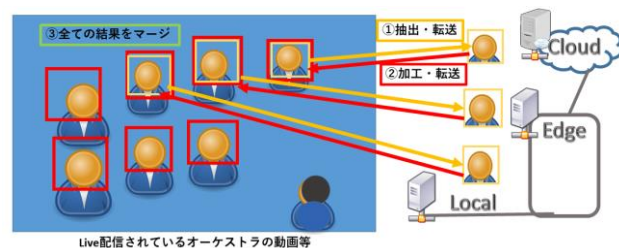
映像効果の付加に伴う処理量は、映像効果の内容に依存するため、あらかじめ予測することが難しい。このため、これまでのシステムでは、クラウド計算機で負荷する映像効果を ECA ルール等で宣言的に決定していた(詳細は[15])。付加する映像効果がルール等に記述されていない場合には、処理時間を短縮するように処理を分散できなかった。

提案システムでは、画像処理の手順をグラフで表現することで即応性、認識の精度と計算負荷の負荷分散、コントロールできるグラフ表現型画像処理プログラミングを用いたアプローチを採用する。

3.2 適切な計算資源管理の設計

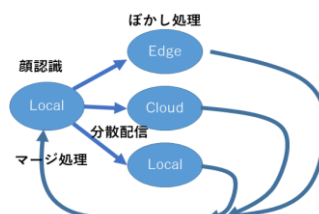
クライアント端末の負荷軽減の為に、事前設計された部分処理について、適切な装置に割り振る設計を行い、画像認識・加工を分散処理し、認識・加工のそれぞれの結果画像については、最終的に画像同士の論理和をとり、クライアント端末上で映像を集約し、表示する方式とした。視聴用のクライアント端末として計算性能が余り高くはない端末を用いても、処理速度の向上が得られ、より精度の高い画像認識が行われる事を目指す。

本研究では、グラフ表現型画像処理プログラミングとして OpenCV の Graph API を用い、図 1 に示す構成のクラウド分散型インターネットライブ配信システムを実装した。



TCP ソケットライブラリにより画像を送受信

図 2. 実験環境のイメージ



Graph 処理：
OpenCV4.3 コード内
GAPI 処理

図 3. 実験で用いたグラフで表現された画像処理プログラム

3.3 グラフ表現型画像処理プログラミング

3.3 グラフ表現型画像処理プログラミング

グラフ表現型画像処理プログラミングとは、グラフ形式で画像処理の手順を表現するプログラミングである。グラフのノードで処理を、エッジで処理の遷移を表す。映像データを取得すると、最初の処理を示すノードの画像処理が行われ、その処理が終了すると、エッジで繋がった次の処理が実行される。同じ映像データに対して複数の処理を並列に行う場合には、あるノードから複数のエッジが出ることになる。また、複数の処理結果を必要とする画像処理では、あるノードに複数のエッジが入ることになる。

グラフ表現型画像処理プログラミングは、手順のある処理を記述する場合に適しており、画像処理やストリーム処理を行うプログラムで近年よく用いられている。例えば、OpenCV の Graph API や Apache Flink 等で用いられている。グラフ表現型画像処理プログラミングを用いることで、ノード単位で処理の手順をあらかじめ把握できる。各ノードの処理内容を把握する場合には、各ノードに対応するプログラムを詳細に読解することで可能と考えられる。

3.4 インターネットライブ配信への応用

映像処理を伴うクラウド分散型インターネットライブ配信において、映像処理をグラフ表現型画像処理プログラムで表現することで、処理内容からクラウド計算機または配信端末で行う方が早く完了する処理を把握しやすくなる。例えば、多数の GPU を利用する画像処理であれば、GPU を多数搭載したクラウド計算機で実行する方が短時間で処理が完了する確率が高い。また、非常に簡素な画像処理であれば、データ送信にかかる時間がオーバーヘッドとなり、配信端末で実行する方が短時間で処理が完了する確率が高い。実際には、クラウド計算機や配信端末の余剰計算能力

表 1 サーバ機器の性能

構成箇所	構成要素	性能
クラウドサーバ	VM プラン	・ Standard B2s(2vcpu 4 Gib memory)
	OS	・ Windows 10 Pro 1809 17763.973
	CPU	Intel(R) Xeon(R) platinum 8171M CPU @2.60Ghz
	メモリ	RAM 4.00GB
エッジサーバ	OS	Windows10 Pro 1909 18363.836
	CPU	Intel(R) Core(TM) i7-7660U CPU @2.50GHz
	メモリ	RAM 8.00GB
Local PC (視聴用)	OS	Windows10 Pro 1909 18363.836
	CPU	Intel(R)Core(TM)i5-6300U CPU @2.40GHz
	メモリ	RAM 8.00GB

や、ネットワーク帯域にも依存するため、予想とは異なることがあるが、確率的にどちらで処理を実行する方が早くなるか参考にできる。しかし、これまでにグラフ表現型画像処理プログラムを用いた映像処理を伴うクラウド分散型インターネットライブ配信を想定した研究が行われておらず、クラウド計算機や配信端末においてグラフ表現型画像処理プログラムを用いた画像処理を行った場合の処理時間が明らかになっていなかった。本研究では、次章でクラウド分散型インターネットライブ配信を想定した評価を行い、これらを明らかにする。

4. 評価

4.1 実験環境

評価に使用した計算処理を分散するのに用いるサーバ及び PC の、OS 及び CPU、メインメモリの性能を表 1 に示す。今回利用したクラウドコンピューティング環境は、Microsoft Azure サービスの Virtual Machine であり、VM のプランも表 1 に示す。視聴端末と主な計算処理を行う Local PC には、ノート PC を利用する。エッジサーバには、Local PC と同ルータにて接続されている Local PC より性能の良い PC を使用する。この実験環境における顔認識には OpenCV 4.3 の Haar-like 特徴分類器を利用し、グラフ処理には、コード内での Graph API のシーケンスを用いている。クラウドサーバ、エッジサーバへの送信は TCP を使用した Socket ライブラリにて通信を行う設計とした。

4.2 評価方法

評価方法は、視聴端末と主な計算処理を行う Local PC により、Harr-Like 特徴分類器を利用して認識された顔画像部分を切り出し、ぼかし処理をかけて、原画像に合成して表示し、1 フレームの画像処理時間を計測した。この結果と、Local PC での特徴分類認識処理の後、切り出した部分のみをクラウドサーバ、エッジサーバに送信し、分散処理した結果を受信して、原画像に合成して表示する処理を計測し、比較した。今後、クラウドやエッジサーバによる個人ごとの識別や物体の詳細検証等の複雑な処理を担わせる狙いがある。図 2 と図 3 に構成と実験方法の概要図を示す。送受

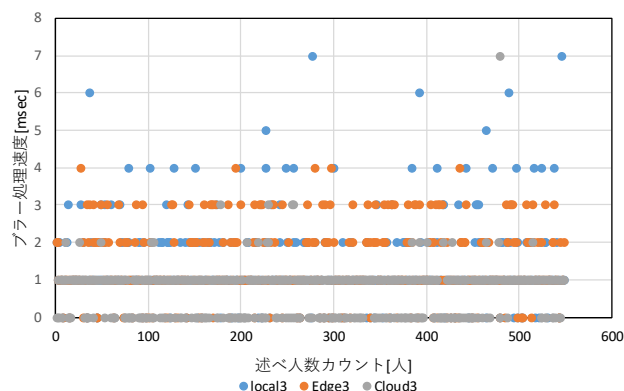


図 4. 画像の顔領域にぼかしをかけるのに
かかった時間

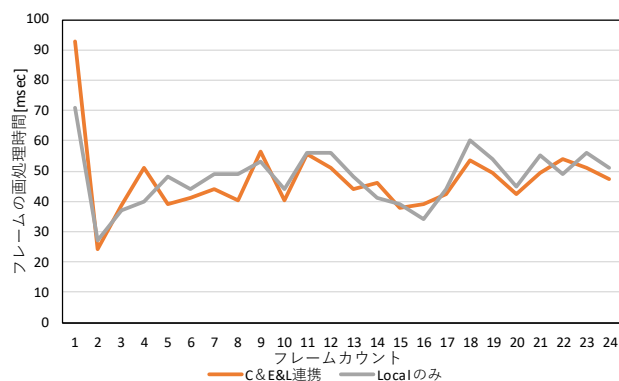


図 5. サーバ連携時と Local 処理のみの処理時間

信部分で OpenCV4.3 利用上でのコーディング部分の都合上、同じシミュレートプログラムを 3 つの機器で動作させ、通信による送受信に掛るターンアラウンドタイムをシミュレートし、計測を行った。ターンアラウンドタイムは、送受信した画像のサイズと通信プロトコルに掛る部分の一定数時間を送受信時間に必要な時間として、38 名が正面を向いて演奏しているオーケストラの映像を用いて 1 フレーム毎の処理時間の測定をした。通信プロトコルに掛る処理時間として、著者らが以前行った（詳細は[16]）、Cloud ネットワーク環境を用いた計算負荷分散処理の実験時に要した、通信プロトコルに掛る処理時間の平均値 (1.28msec=ターンアラウンドタイム平均値 16.28msec - 処理時間 15.00msec) をシミュレートに用いた。この実験では、顔を切り出した部分にぼかしをかける処理時間をそれぞれ三つの計算機で画像処理に掛った時間を計測し用いた。

4.3 評価結果

4.2 の評価方法で 1 フレーム毎の処理時間について、Local PC (視聴用) のみで処理を全て行った場合と、Cloud サーバ、エッジサーバに分散させて処理を行った場合に分けて測定し、その結果を用いて通信に掛る部分をシミュレートした。その結果、オーケストラが演奏している映像の様に、映像に映る人々の増減が無く、人々が一定位置から余り動かない状況の場合、処理の高速化が行えた。

また、順列を持ったグラフの処理により、Local PC、Cloud サーバ、エッジサーバに均等に処理を割り当てる等のコントロールが行え、安定した高速化が行える事が分かった。

計測し、その結果を用いて、通信経路部分の処理時間をシミュレートした。オーケストラの映像を用いて、24 フレーム中、延べ 550 人の顔認証を行い、サーバ連携時と Local PC のみの処理とを比較した結果を図 4 に示す。サーバ連携時の処理時間が僅かに下回り、僅かに高速化される事が分かった。また、全く同じオーケストラの映像を用いて、24 フレーム中、延べ 550 人の顔認証を行い、顔画像を切り出してブラー処理をそれぞれの機器単独で計算処理を行った処理時間を図 5 に示す。クラウドサーバ > エッジサーバ > Local PC の順の性能差である。

4.4 議論

まず、画像処理に関する問題点として、細いマイクスタンドや楽器の一部などにより顔前面が一部隠れている場合、顔認識が行われず、顔画像の加工が行えない。人がこの画像を見た場合、マイクスタンド越しにその本人と認識できてしまう場合が多かった。この度試用した映像では、39 名中、Harr-Like 抽出器の類似度パラメータ 1.08 にて、67 フレームの測定結果では平均 25.6 名が顔として認識されて処理された。1 フレーム内で平均 13 名程は未加工のままであった。プライバシー保護等を考慮した画像加工を行いたい場合、カメラから被写体までの間にはむしろ、遮蔽物が無い環境が適していると分かった。処理速度を向上できれば、類似度パラメータの精度を上げた詳細な認識も可能となる。

本論文では、単純なグラフ表現型画像処理プログラムで評価を行ったが、分岐や統合がある場合には、他のノードで表現された処理の時間も影響するため、一概にクラウドサーバによる処理が高速化できるとはいえないと考えられる。

5. まとめ

本研究では、グラフ表現型画像処理プログラミングを用いたクラウド分散型インターネットライブ配信システムを提案、実装し、分散処理についてシミュレートした。シミュレート結果では、視聴用のクライアント端末として計算性能が余り高くない端末を用いても、処理速度の向上が得られ、より精度の高い画像認識が行える事が分かった。

今後、他の処理をグラフ表現型画像処理プログラムで表現して評価を行うことや、クラウド計算機への割り当て等を詳細に検証する予定である。

謝辞

本研究の一部は科学研究費補助金(課題番号 JP20K11829)および G-7 奨学財団研究開発助成事業の助成による。

関連文献

- [1] X. Zhao, H. Ma, H. Zhang, Y. Tang, and Y. Kou, HVPI: Extending Hadoop to Support Video Analytic Applications, in Proceedings of the 8th IEEE International Conference on Cloud Computing (CLOUD 2015), pp. 789-796 (2015).
- [2] W. Kou, H. Li, and K. Zhou, Turning Video Resource Management into Cloud Computing, Future Internet 8(3), 35, 10 pages (2016).
- [3] T. Yoshihisa, T. Hara, A Low-Load Stream Processing Scheme for IoT Environments, in Proceedings of the 2016 IEEE International Conference on Big Data (Big Data 2016), pp. 263-272 (2016).
- [4] N. Chen, Y. Chen, Y. You, H. Ling, P. Liang, and R. Zimmermann, Dynamic Urban Surveillance Video Stream Processing Using Fog Computing, in Proceedings of the 2016 IEEE Second International Conference on Multimedia Big Data (BigMM 2016), pp. 105-112 (2016).
- [5] Q. Ning, C.-A. Chen, R. Stoleru, and C. Chen, Mobile Storm: Distributed Real-Time Stream Processing for Mobile Clouds, in Proceedings of the 4th IEEE International Conference on Cloud Networking (CloudNet 2015) (2015).
- [6] T. Li, J. Tang, and J. Xu, A Predictive Scheduling Framework for Fast and Distributed Stream Data Processing, in Proceedings of the 2015 IEEE International Conference on Big Data (Big Data 2015), pp. 333-338 (2015).
- [7] J.-H. Choi, J. Park, H. D. Park, and O.-G. Min, DART: Fast and Efficient Distributed Stream Processing Framework for Internet of Things, ETRI Journal, Vol. 39, No. 2, pp. 202-212 (2017).
- [8] Y.-K. Kim, Y. Kim, and C.-S. Jeong, RIDE: Real-Time Massive Image Processing Platform on Distributed Environment, EURASIP Journal on Image and Video Processing, 13 pages (2018).
- [9] J. Yang, B. Jiang, and H. Song, A Distributed Image-Retrieval Method in Multi-Camera System of Smart City Based on Cloud Computing, Future Generation Computer Systems, Vol. 81, pp. 244-251 (2018).
- [10] L. O'Gorman and X. Wang, Balancing Video Analytics Processing and Bandwidth for Edge-Cloud Networks, in Proceedings of the 24th International Conference on Pattern Recognition (ICPR 2018), pp. 2618-2623 (2018).
- [11] Tomoki Yoshihisa, Satoru Matsumoto, Tomoya Kawakami, and Yuuichi Teranishi, "A Video Processing System to Stabilize Frame Rates on Trust-Oriented Internet Live Video Distributions," in Proc. of the International Workshop on Informatics (IWIN 2019), pp. 31-38 (2019).
- [12] X. Zhang, et. al., Keyword-driven Image Captioning via Context-dependent Bilateral LSTM, IEEE ICME2017, pp. 781-786 (2017).
- [13] J. Li, et. al., Attention Transfer from Web Images for Video Recognition, ACM Multimedia, 9 pages (2017).
- [14] Z. Naor, et. al., Content Placement for Video Streaming over Cellular Networks, IEEE ICNC, pp. 133-137 (2015).
- [15] 松本哲, 義久智樹, 川上朋也, 寺西裕一: "自律映像処理を伴

うトラスト指向インターネットライブ配信システムの検討,"

(DICO2019) シンポジウム論文集, pp. 1298-1300 (2019).

[16] 松本 哲, 石 芳正, 義久智樹, 川上朋也, 寺西 裕一, "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," (DICO2018) シンポジウム論文集, pp. 523-529
(2018).