

クラウド支援型 P2P ライブストリーミングにおける バッチ制御手法の提案と評価

冨永菜都美[†] Bahaa Aldeen Alghazawy[‡] 藤田 聡[‡]

広島大学 工学部[†] 広島大学 大学院工学研究院[‡]

1 はじめに

フラッシュクラウドとは、多数のユーザの同時アクセスによってサービス品質(QoS)が一時的に低下する現象のことであり、P2P 型のライブストリーミングシステムにおいて解決すべき重要な課題の一つである。この問題に対して Jinbo らは、クラウドの支援を想定したバッチ制御手法を提案した[1]。この手法では、新規参加ピアによって構成されるバッチと呼ばれる部分オーバーレイのサイズに条件をつけることで新規参加ピアの最大視聴待ち時間を大幅に短縮している。本稿では、バッチのメインオーバーレイへの接続部分に焦点を当て、クラウドコストを抑えつつ QoS を保つためのバッチ制御手法の提案と評価を行う。

2 モデル

本稿で対象とする P2P システムは、メディアサーバ、トラッカー、および複数台のピアから構成される。以下では、フラッシュクラウド発生前のシステムをメインシステムと呼ぶ。ビデオストリームは、メディアサーバによって等しく分割され、固定長のチャンクの列として与えられる。メインシステムにおけるトラッカーの役割は、新規ピアからの参加リクエストに対してオンラインピア集合の中から任意に選択された K 台のピアのリストを返すことである。リストを受け取ったピアは、適切なアルゴリズムにしたがって P2P オーバーレイに参加し、一定量のチャンクをバッファした後に動画の視聴を開始する。

クラウドはフラッシュクラウド発生時にトラッカーにより呼び出され、適切な支援を行う。AWS (Amazon web services) モデルに基づくクラウドの使用には、使用するメモリー量に関して GB/秒あたり \$0.00001504 のコストが、送信データ量に関して 1GB あたり約 \$0.14 のコストがそれぞれ必要である。

3. 既存手法

[1]の手法では、トラッカー機能がクラウド上で実装される。クラウド上で呼び出されるトラッカー機能のことを**クラスター**と呼ぶ。フラッシュクラウドを検知すると、トラッカーはクラウド内で複数のクラスターを呼び出し、各クラスターにはトラッカーによって同程度のサイズの新規ピアの集合が割り当てられる(ピア集合のサイズは、メインオーバーレイ上の各ピアの平均アップロード帯域に基づいて、視聴待ち時間が決められた時間内に収まるように設定される)。新規ピアの集合を受け取ったクラスターは、それらのピアに対して、各ピアが高々一つのツリー上で内部ノードとなるようにスパニングマルチツリー(バッチ)を構築する。クラスターは各ピアに対して親を提案する責任を持ち、親を見つけることができなかったピアに対する配信はクラウドストレージにより支援される。

バッチが完成すると、クラスターはクラウドコスト節約のためにキャンセルされ、バッチの根ノードたちによる(クラウド CDN を介した)クラウドストレージからのチャンクの取得が開始される。その後各根ピアは、トラッカーに対してメインオーバーレイに参加するためのリクエストを再び行い、メインオーバーレイ内の親を探す。根ノードに対するクラウド支援は、メインオーバーレイからのストリームの取得が始まるとキャンセルされる。

Proposal and evaluation of batch control method in cloud-assisted P2P live streaming

[†]*** TOMINAGA, Hiroshima University

[‡] Satoshi FUJITA, Hiroshima University

本研究の一部は科学研究費補助金基盤研究(B)16H02807の補助を受けています。

4.提案手法

[1]の手法では、根ノードに対するクラウドストレージからのストリームの配信は、バッチの完成後に行われていた。そのためバッチサイズをあまり大きくすることができず、結果的に多数の根ノードたちがメインオーバーレイに接続されていた。フラッシュクラウド発生時に到着する全新規ピア数に比べればその数は少なくなるものの、トラッカーに向けて再度リクエストを行うピア数は全新規ピア数に比例して増加し、メインオーバーレイへの接続時間とクラウドコストを増加させる要因となっていた。

本稿ではバッチへのクラウドからのストリーム配信を「バッチを成長させながら」行う手法を提案する。これにより、バッチの個数を大きく減少させることが可能となり、バッチのメインシステムへの接続が容易に行えるようになる。

提案手法では、各ピアは K 台のピアを子ノードとして持つのに十分なアップロード帯域を有するものとする。ここで K はサブストリームの数であり、生成されるスパニングツリーの個数とも一致する。各スパニングマルチツリーは、クラスターの処理に伴って少しずつ増えていく内部ノードを使用して、ツリーの高さが可能な限り浅くなるように構築していく。各ツリーの葉ピアは、アップロード帯域に余裕のある内部ノードに接続されるが、ツリーのより浅い位置に内部ノードを配置できるように、要請があったときにはいつでも内部ノードに場所を譲るものとする。葉ピアが内部ノードとなる新規ピアに場所を譲る際、動画再生の遅延や中断が起こる可能性があるが、それを防ぐため、移動する葉ピアは元の親に接続したまま新たな親に接続し、新しい親への接続が完了した後に元の親との接続を切断し場所を譲るという手法をとる。内部ノード数の不足によって、ツリーに参加できず、上記の手続きの中でパケットが受け取れないピアが発生した場合は、クラウドが支援を行う。クラウドからの支援を受けているピアがツリーに参加するとクラウドの支援は終了する。

5.評価

性能を評価するためにシミュレーションによる評価を行った。シミュレーションにはイベント駆動型

シミュレータを用いた。シミュレータは C++ 言語で記述されており、[2]で開発されたものに本研究室で独自の拡張がなされている。評価尺度はピアの視聴待ち時間とクラウドから取得される合計データ量である。具体的には、提案手法によって、視聴待ち時間を従来手法と同程度以上に保ったままでクラウドコストを大幅に削減できることを示す。

実験の結果を図1と図2に示す。視聴待ち時間は従来と同程度に抑えられている。各グラフにおいて、(a)提案手法、(b)従来手法、(c)従来手法(1つのバッチ)を示している。図1はバッチの各ルートがクラウドから取得するデータ量の合計であり、図2はピアがクラウドから取得するデータ量の合計である。

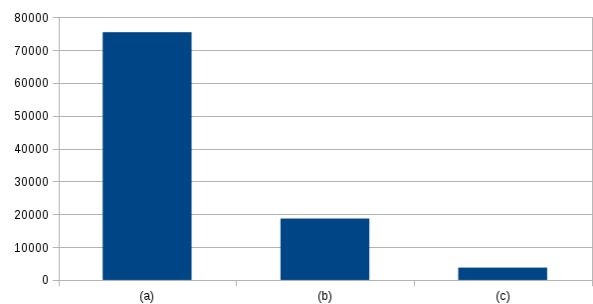


図1 クラウドから取得したデータ量:ルート

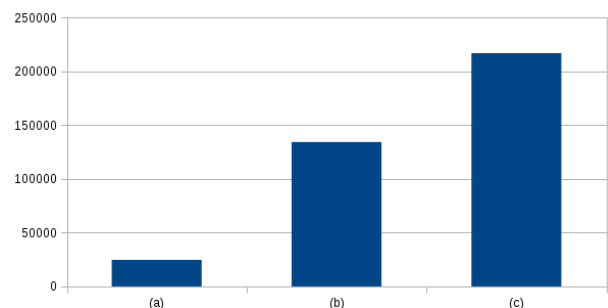


図2 クラウドから取得したデータ量:ピア

提案手法ではルートが取得するデータ量は、バッチ構築時からデータを取得するため従来手法より大きくなっている。一方で、ピアが取得するデータ量は減少している。

参考文献

- [1] Shuji Jinbo, Bahaa Aldeen Alghazawy, Satoshi Fujita:Batch-based flash crowd relaxation in cloud-assisted P2P live streaming. Proc. SNPD 2017: 95-100.
- [2] Meng Zhang, Peer to Peer streaming simulator, [online], <http://media.cs.tsinghua.edu.cn/~zhangm/>.