

2.2.2.3 動画像広域分散配信システムおよびクラウドを用いた Web コンテンツ 大規模配信実験

寺田 直美

1. はじめに

本学電子図書館では現在、書籍、資料の電子化に加え、講義講演等を高画質映像で自動アーカイブする機能を用意している。文字、画像等の視覚情報をはじめとして聴覚、触覚、嗅覚情報までもデジタル化可能になりつつある現在、従来から資料的価値があるとされてきた文献にとどまらず、生活環境情報を丸ごと保存する動きも高まっている。図書館が情報集積の中心として将来的にあらゆる情報をデジタルアーカイブ化していく際、有限であるリソースを有効活用しつつ広く配信する技術が必要となるため、附属図書館研究開発室では、デジタルコンテンツの大規模配信について研究開発を進めている。

2. 動画像広域分散配信システムの開発

動画像を含むコンテンツ配信では、サーバ(あるいはサーバ群)よりクライアントにサービスを提供するサーバクライアント方式が一般的である。この方式ではサーバに情報が集約されるため管理が比較的容易であるが、アクセスが集中した場合にサーバ付近のネットワークリソースおよびサーバリソースが不足するといった問題点がある。この問題を解決するために、サーバ台数を増やして1台あたりの負荷を軽減する、ネットワーク上の複数箇所にサーバ群を分散配置するなどの方法が採用されてきた。

これらの方式とは別に、ネットワーク上に特定のサーバを置かず、端末間を相互接続しデータを交換するオーバレイネットワークが提案されており、オーバレイネットワーク基盤の上で各種配信を行う方式が注目されている。この方式はサーバクライアントベースのシステムと比較して低コストで配信できる、配信規模が拡大してもシステムの拡張性が確保できるとされているが、ネットワーク上の端末(ピア)同士を接続しているため、切断/再接続の問題が常に発生する。また、サーバクライアント方式のように一点に集約することがなく、端末の切断あるいは新規参加により時々刻々と変化する配信網を形成するため、接続先ピアが切断した際の再接続や、新たなコンテンツを取得するため別のピアへ接続し直す際に親となるピアの探索、接続に一定の時間を必要とする。

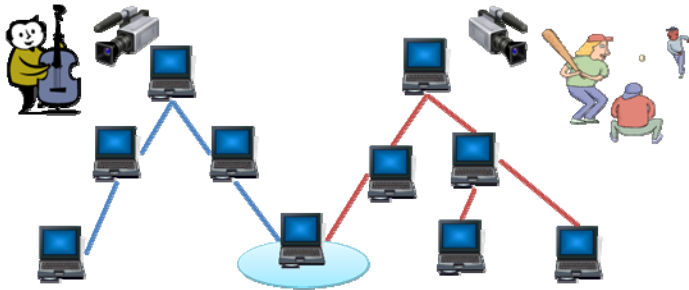


図1 オーバレイネットワークを用いたマルチソースストリーミングの簡単な例

本研究では、ユーザがストリーミング視聴時にテレビでチャンネルを頻繁に切り替える(ザッピング)ように、複数のコンテンツを短時間に切り替えることに着目した。オーバレイネットワーク上で動作する既存のストリーミング配信システムでは、ユーザが視聴するストリームを切り替えるたびに、切り替えようとするコンテンツを配信しているピアを探索、接続するため、ストリーミングにおけるチャンネルの切り替えに時間が掛かるという問題がある。既存システムで実験した際には20秒から長い場合は1分近くの待ち時間が発生することが分かった。この待ち時間はユーザへのサービス品質を大きく低下させるだけでなく、この間にユーザを逃す可能性もある。そこで本研究では1) 次に接続される可能性が高いストリーミングコンテンツを予想し、ザッピングにおける切り替え時間を短縮するためにあらかじめストリーミングデータを読み込んでおく 2) 予想外の切断に備えて、バックアップリンクを用意しておく という2つの方法を用いて設計、実装を行った。

複数のストリーミングコンテンツを配信するための基盤がいくつか既存研究として挙げられ、実験により高いスケーラビリティを確保しているが、接続時間の短縮については考慮されてこなかった。

接続時間を短縮するには、究極的にはそれぞれのピアが配信システム上を流れるすべてのストリーミングコンテンツにリンクを張ることで実現される。しかしながら、変化し続ける配信網で大量のリンクを確保し続けるためには、各ピアがお互いのピアの生存確認のための通信(Keep-alive traffic)が必要となり、ネットワークトラフィックがあふれる結果となり非現実的である。

本研究の基本的な方式は、次に切り替えられる可能性の高い予想リンクをあらかじめ設定し、それぞれ最適なピアに対して事前に接続を張っておくことによりスムーズな再接続とチャンネル切り替えを実現する。予想リンクを決定する際、ストリームコンテンツ毎に設定されたキーワードをもとに視聴履歴から重複するキーワードを抽出し、次に選択される可能性が高い類似コンテンツを選出する。

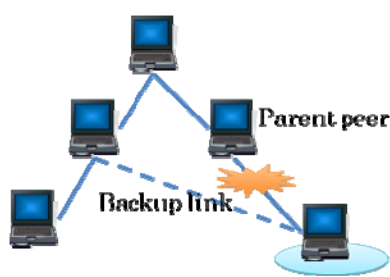


図 2 バックアップリンク

バックアップリンクは図 2 に示すように、データを受け取る親となるピアの他に、親ピアの切断に備えて用意された別のリンクを指す。

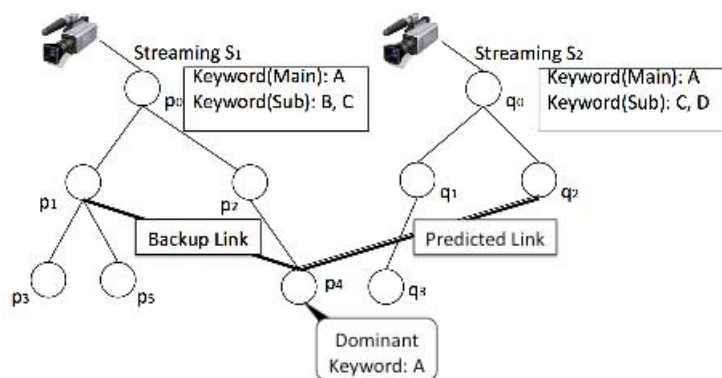


図 3 バックアップリンクと予想リンク

図 3 に示すように、ストリーミングコンテンツには内容に応じて 1 つのメインキーワードとサブキーワードを少なくとも 1 つ設定する。予備調査の結果から、コンテンツ切り替え時にはこういった場所で行われているかといった情報が重要になることから、「球場」や「コンサートホール」といった情報をメインキーワードとし、付加的な情報をサブキーワードとしている。ユーザの視聴履

歴には直近に視聴した複数コンテンツのメインキーワード、サブキーワードを保存しておき、その中で頻出するメインキーワードを優先キーワード、頻出するサブキーワードを従属キーワードとした。図 4 の例では、” stadium ” が優先キーワード、” baseball ” が従属キーワードとなる。

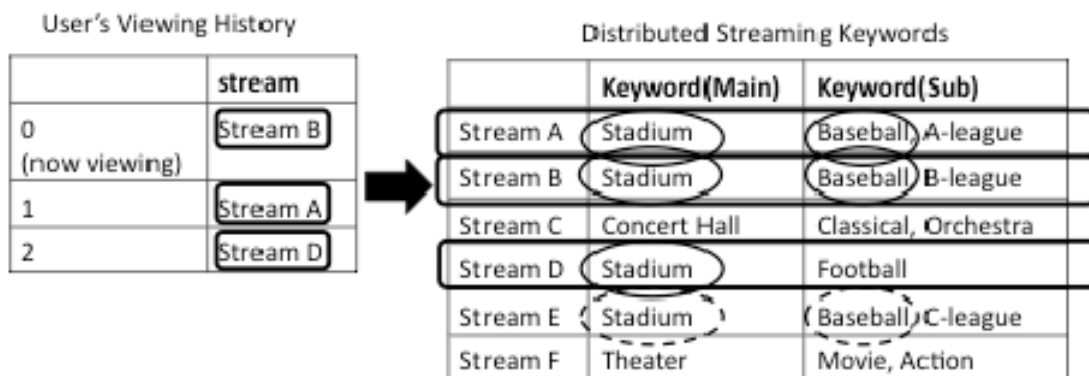


図 4 キーワードの設定例

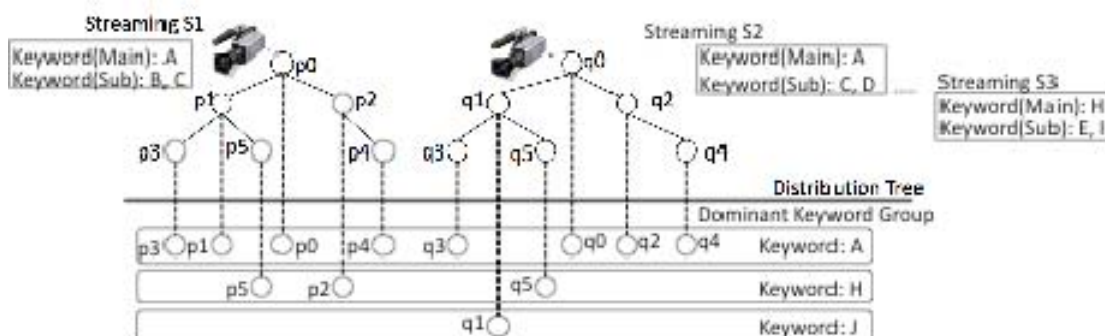


図 5 配信木とキーワードグループの例

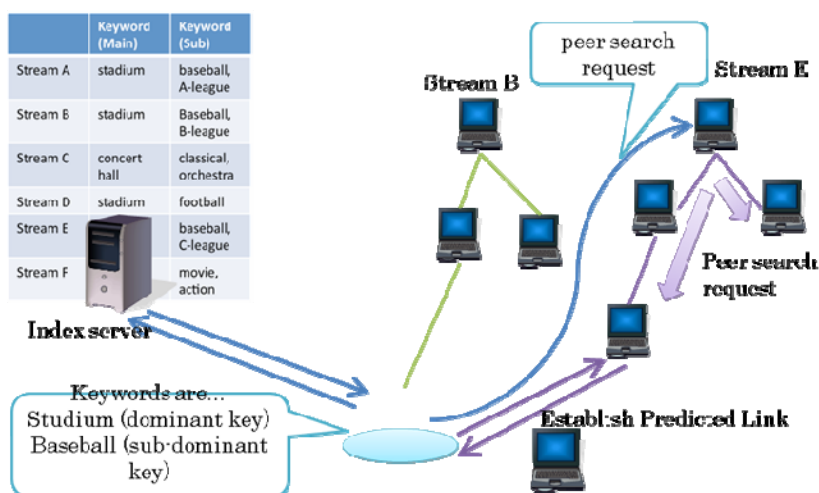


図 5 ピア探索方法

視聴履歴の状態によって3つのケースに分類し、1) 視聴履歴が存在しない場合 2) 視聴履歴の中に重複するキーワードが存在しない場合 3) 視聴履歴中のいくつかのキーワードが重複している場合、としている。本方式が最も高い効果を発揮するのは3)のケースであり、このようなケースでは本方式を適用しない場合と比較して接続待ち時間の最大90%減、1)あるいは2)のケースでも接続待ち時間の5%減を実現することができた。配信網の構成とキーワード予測についてさらに改善を進めているところである。

3. クラウドを用いた Web コンテンツ大規模配信実験

近年、ネットワークやサーバリソースの有効活用という視点から、企業、大学システムにおけるクラウド化の重要性が議論されている。本研究では Web コンテンツ大規模配信におけるクラウド利用の可能性について、実サービス環境で広域にまたがるクラウドを用いて実験を行った。

奈良先端科学技術大学院大学では1996年より朝日放送株式会社と連携して夏の高校野球選手権大会インターネット中継を行ってきている。本中継ではここ数年リソースの動的割り当ておよび負荷分散に関する実験を継続的に行っており、今回は WIDE プロジェクトで進められている実験クラウドシステムを用いて配信実験を実施した。

クライアントがサイトへアクセスした場合、大阪にある朝日放送サーバ群に送られ、そこで複数台のサーバに分散させるローカル負荷分散を行っている。実験では図7で示すように、WIDE クラウド上に仮想的に構築された計算機 VM にローカル負荷分散で用いているものと同じサーバを用意し、クライアントから一部トラフィックを WIDE クラウドに送ることで広域での分散を行った。なお、クラウドを構成する実計算機の性能は一定ではないため、あらかじめ行った負荷実験の結果をもとにクライアントからの接続を割り振る割合を決め、受付となるリバースプロキシサーバが接続を分散させる。今回の実験では奈良先端科学技術大学院大学内にある WIDE クラウド計算機群と東京大学内にある WIDE クラウド計算機群を稼働させた。

配信システムは Web ページそのものを配信する HTML 系サーバ群と、中継映像

を画像データとして切り出し表示するための JPEG 系サーバ群に分割されており、いずれも Apache により http 配信されている。今回の実験では 1) 実験トラブル時にサービスへの影響が出ないこと 2) 実験の評価がとりやすいよう、ネットワーク帯域における分散の効果が大きいこと、を考慮し、HTML 系サーバ群は従来のままにしておき、JPEG 配信系についてクラウドへの分散設定をした。



図 6 夏の高校野球選手権大会インターネット中継 Web サイト

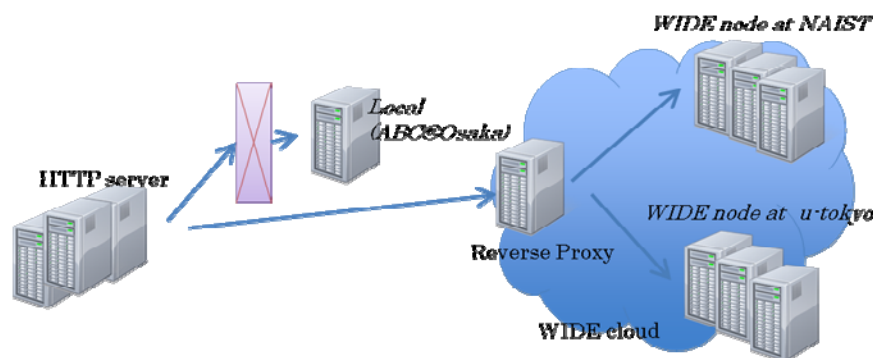
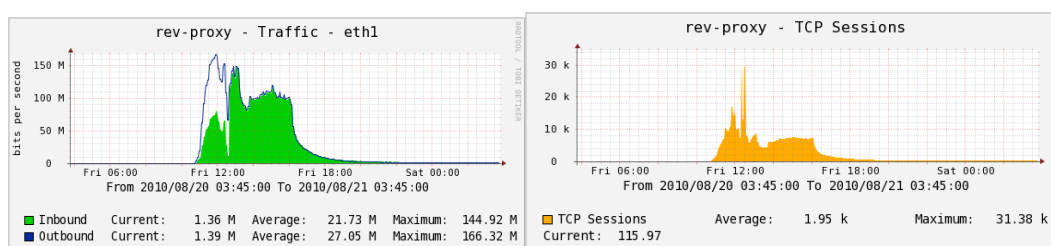


図 7 WIDE クラウドへのトラフィック分散

実験ではクライアントからの接続状況を観測しながら、試合と試合の間に順次クラウドの投入を実施した。これまでは大阪に設置したサーバ群で全てのアクセスを処理していたため、決勝戦など多数のアクセスが集中するタイミングでは帯域が逼迫し、一時的にコンテンツの品質を下げることで対応する必要があったが、実験では奈良、東京の遠隔地へトラフィックを分散させることができたため、大阪への一点集中を避けられ、結果的にコンテンツ品質変更によるトラフィック抑制も少なく抑えることができた。

クラウドの特徴ともいえるサーバリソースの動的制御については、配信開始後にクラウド側で持つ帯域に対してサーバリソースが少なかったため、サーバVMの増設を行った。配信中に実機サーバを増設することは、機材の確保といった金銭的、時間的問題のほかにも収容スペースや電源、排熱の問題等から障壁が多いが、実験クラウド上では必要に応じて容易にVMを増設できるので予想を上回るアクセスにも対処しやすい。また、リソースを必要な時に「借りる」ことができるため、システム全体で見たとき、それぞれのサービスに固定的にリソースを割り当てる方式と比較してリソースの有効活用が可能である。しかしながら、今回実験で判明したことは、実計算機上でVMの性能がどの程度確保できるかはVMの実装にまだ大きく依存しているという現状であり、1台の実計算機上で複数のVMを起動させた時には予想以上に大幅な性能劣化が見られる場面が多々あった。また、負荷実験でベンチマークテストを行った結果に比べて、実配信では性能が得られないケースもあった。リバースプロキシを通した場合、一部のサーバについてのみ性能劣化が起きる現象も観測されている。これらの点については事前の負荷実験で確認できなかった。



今回実験においてクラウドを導入したことにより、特に帯域の面からサービスに制約が出ていた部分が大きく改善された。しかしながら、実際に運用した場合、同スペックでありながらハードウェアの違いによって得られる性能が大きい

く変わる、複数 VM を起動すると大幅な性能劣化が起きる、実運用とベンチマークテストの差が大きい、リバースプロキシを用いた場合の性能劣化問題など、数多くの問題が発生した。特に定常運用システムに導入する場合は十分な検討と予備実験を行った上で、発生するトラブルに対して常時対処可能な体制を組む必要があると考える。今回観測された複数のトラブルについて、細かい検証と追実験を進めているところである。