

遊休資源を活用したオンプレミス型 CDN による 対外的なデータ通信量削減の提案

川戸聡也^{1,2} 東野正幸¹ 高橋健一¹ 川村尚生¹

概要: コンピュータの性能は向上, 用途によっては有する資源が十分に活用されず, 遊休の状態となっている. 一方, コンピュータを活用した教育が推進されており, 自宅に限らず学校内での授業や自学自習においても, ネットワーク上のコンテンツを利用する機会が多くなっている. この際, 大勢が同じコンテンツを利用することがあるが, 各自がそれぞれに取得しようとすると, その分のデータ通信が当該コンテンツを所持するサーバおよびそのサーバまでのネットワークに発生する. 特に, 当該コンテンツが学校外のネットワーク上にある場合にはその帯域を利用することとなり, 限りある共有のネットワークを経由するために, 学校内から学校外への対外的なデータ通信量は少なくすることが望ましい. そこで, 学校内のネットワーク上に存在する遊休の資源を持つコンピュータを活用してオンプレミスに独自の CDN を構築することで, 対外的なデータ通信量を削減することを提案する. 学校外のネットワークにあるコンテンツについて, その複製を学校内のネットワーク上に配置して配信することで対外的に発生するデータ通信量を削減する. また, 複製の配置や配信などに, 遊休の資源を持つ既存のコンピュータを活用することで, 独自の CDN を安価に構築できる.

Proposal for Reduction of the Amount of External Data Traffic by On-Premise CDN using Idle Resources

TOSHIYA KAWATO^{†1,2} MASAYUKI HIGASHINO^{†1}
KENICHI TAKAHASHI^{†1} TAKAO KAWAMURA^{†1}

1. はじめに

コンピュータの性能は技術の進歩に伴い向上している. 一方, 相対的に価格は下落しているために, 以前と比較して高性能なコンピュータを安価に利用することが可能になっている. 例えば, データを長期間に亘って保存するための補助記憶装置であるストレージについて, 以前は数百 GB の容量を持つ HDD が一般的に普及していた. しかし, 近年は速度や静音性に優れた SSD の普及が進んでおり, HDD であれば TB 台の容量を持つことが一般的になっている. これは, SSD と HDD とともに大容量化が進んだ一方, 単位容量あたりの価格が安価になったことが要因である.

このように, コンピュータは高性能化しているが, その用途によっては高い性能を十分に活用できていないとは限らない. 主な用途がブラウジングや事務作業といった比較的軽い処理の場合, CPU やメモリなどの計算処理に係る資源が性能の上限まで利用されることは稀であり, 余力を有しているのが基本的な状態である. また, オンラインストレージの普及により, コンピュータの内部のストレージでは OS やアプリケーションなどのコンピュータとしての動作に必要なデータのみを保持し, 動画像や文書といったコンテンツは外部のオンラインストレージに保存するという運用が可能となっている. この場合にはコンピュータの内部

のストレージは上限まで利用されず, 空きが存在することになる. 計算処理やデータ保存以外にも, ネットワークの I/O も余力を有する資源として挙げられる.

よって, コンピュータが稼働している間において, CPU, メモリ, ストレージ, ネットワーク I/O といった資源は, 余力を有する場合に利用できるのにもかかわらず利用していない遊休の状態となる. このコンピュータが有する遊休の状態の資源 (以下, 「遊休資源」という) は既存のものであり, 用途さえあれば新たな費用の負担なく利用できるため, 有効に活用したい. 我々はこれまでに, 分散ストレージとしての活用[1]や e-Learning システムにおける負荷分散での活用[2]を提案しており, 更に他の用途でも有効に活用できないか検討している.

2020 年に入り, COVID-19 の影響により人間社会の生活様式には変化が生じている. 密閉, 密集, 密接の状況を避けることが求められ, 様々な場で対策が取られている. 教育機関である各種学校においては, 原則として行ってきた対面授業を可能な限り避け, 遠隔授業や分散登校が実施されている. 遠隔授業では, 生徒や学生は通学せずに自宅で学習することが主な形態となる. この際, 以前と比較して, 資料の閲覧や動画の視聴, レポートの作成などにコンピュータを利用する機会が多くなっている. 遠隔授業の実施により, 授業や自学自習におけるコンピュータの利用が急速に推進且つ実践されており, これは今後, 授業や自学自習の場所が学校内に移った場合でも継続されると考えられる. 文部科学省が主導している GIGA スクール構想[3]においても, 生徒や学生に対する 1 人 1 台のコンピュータの提供や

¹ 鳥取大学
Tottori University
² 米子工業高等専門学校
National Institute of Technology, Yonago College

ネットワークの増強などを通し、ICT を活用した教育改善が図られており、学校内における授業や自学自習でのコンピュータの利用についても浸透していくと考えられる。

コンピュータで授業や自学自習を行う場合、ネットワークを経由して動画や文書などのコンテンツを取得して利用することがある。どのようなコンテンツを利用するかは授業や自学自習の内容によるが、それらの主題が決まっている場合、大勢が同じコンテンツを利用するという状況が考えられる。この状況で各自がそれぞれコンテンツを取得しようとする、コンテンツを保持するサーバおよびそのサーバへのネットワークにその分のデータ通信が発生することになる。特に、授業中などにリアルタイムで特定のコンテンツを利用する場合にはその分の通信が同時に発生することになり、サーバやネットワークの負荷が短期間に急激に高まることで遅延や障害が発生する可能性がある。また、対象となるコンテンツが学校外のネットワーク上に存在する場合、当該コンテンツの取得に係るデータ通信が共有のネットワーク上に生じることになり、限りある帯域をその分占有することとなる。COVID-19 の影響で共有のネットワークにおけるデータ通信量は増加しており[4]、教育現場においては遠隔授業などにおけるデータ通信量を極力小さくするように呼びかけられている[5]。このため、学校内のネットワークから学校外の共有のネットワークに対する対外的なデータ通信量は少なくすることが望ましい。対外的なデータ通信量を少なくすれば、対内的なデータ通信量も少なくなり、総じて共有のネットワークにおけるデータ通信量を削減することができる。

ネットワーク上のコンテンツを配信する基盤としては CDN (Content Delivery Network) [6]があり、OS のアップデートや動画などの主に大容量のコンテンツを配信する手法として広く利用されている。基本的な CDN は、コンテンツのオリジナルを保持するサーバ(以下、「オリジンサーバ」という)と、コンテンツの複製をキャッシュとして保持するサーバ(以下、「キャッシュサーバ」という)から構成される。オリジンサーバが保持するコンテンツはその複製がキャッシュサーバに配置される。そして、エンドユーザへのコンテンツの配信はキャッシュサーバからの複製の配信により行われ、オリジンサーバからは直接配信されない。キャッシュサーバは地理的にもネットワーク的にも分散して複数配置することで、CDN 全体やエンドユーザの状況に最も適したキャッシュサーバが選択して配信できる。このようにコンテンツを分散して配置および配信することにより、アクセスの分散や耐障害性の向上を実現している。CDN は Akamai などのプロバイダが提供しているサービスでの利用が一般的であるが、クラウド上に構築されているために学校内のネットワークに閉じて利用することはできない。よって、共有のネットワーク回線におけるデータ通信量を削減することはできない。削減するためには、学校

内のネットワーク上に CDN を構築し、コンテンツの複製を学校内のネットワーク上でやり取りする方法が考えられるが、この場合には各サーバを用意するための費用が問題となる。

そこで、学校内のネットワーク上にある遊休資源を有するコンピュータ(以下、「遊休コンピュータ」という)を活用し、独自の CDN をオンプレミスに構築することで、対外的なデータ通信量を削減することを提案する。CDN の利用者は、CDN を構築する学校内のネットワークに接続されたコンピュータを利用する生徒や学生とし、学校外のネットワークからの利用は想定しない。また、オリジナルのコンテンツは管理しないためにオリジンサーバは無く、CDN を管理する管理サーバと、コンテンツの複製を配置および配信するキャッシュサーバにより構成する。学校外のネットワーク上にあるコンテンツを利用する際は、初回のみ対外的なデータ通信により当該コンテンツを取得し、複製としてキャッシュサーバに配置する。以降はその複製を配信することで、対外的なデータ通信を不要とし、対外的なデータ通信量を削減する。また、当該コンテンツを取得するための経路を短くすることができる。遊休コンピュータとしては、学校内において事務処理や情報システム運用などの本来の用途があり、既存で利用されているコンピュータを活用する。既存の遊休コンピュータを活用することで、サーバなどとして新規に導入が必要な機器を減らし、独自の CDN を安価に構築することができる。

既存で運用や提案されている CDN では、CDN を構成するコンピュータが CDN の管理者の制御下にあり、専用で利用することを前提としている場合がほとんどである。提案する CDN では、活用する遊休コンピュータに本来の用途があるため、本来の用途への影響や本来の用途による制限がある。例えば、常時起動しているとは限らない。本来の用途に応じてコンピュータの起動と停止が発生およびその頻度やタイミングはまちまちとなり、CDN の管理者はこれらを制御することができない。また、オリジナルのコンテンツそのものについても、構成するコンピュータと同様に CDN の管理者の制御下にはない。提案する CDN においては、これらを考慮することが必要となる。

本稿では、対外的なデータ通信量を削減することで共有のネットワークにおけるデータ通信量を削減する、遊休コンピュータを活用したオンプレミス型の CDN の設計を行う。また、想定する実装を示す。

2. 設計

提案する CDN について、想定する環境や対象とする遊休コンピュータを整理した上で、構成や処理の流れを示す。

2.1 想定環境

提案する CDN は、学校内にあるネットワーク上に構築し、そのネットワーク上には後述する遊休コンピュータが

存在することを前提とする。対象とするコンテンツは、学校教育における授業や自学自習で利用される、インターネット上に存在する動画や文書とする。授業や自学自習を行う生徒や学生が利用者であり、利用者がコンテンツを取得や利用するために用いるコンピュータ（以下、「クライアント」という）は学校内のネットワークに接続されたものとし、学校外のネットワークからの利用は想定しない。なお、具体的な環境として学校を想定しているが、同じような条件を満たす環境であれば、企業などの他の組織にも置き換えることができる。

2.2 対象の遊休コンピュータ

遊休資源とは、コンピュータが有する演算処理、データ保存、通信などに利用可能な資源のうち、そのコンピュータの本来の用途において実用されていない資源を指す。また、遊休コンピュータは、遊休資源を有するコンピュータを指す。遊休コンピュータには本来の用途があるため、別の用途で利用する場合には本来の用途への影響を最小限とすることが望ましい。

提案する CDN において利用する遊休コンピュータは学校内のネットワークに接続されたもののみとし、クライアント、業務サーバ、業務パソコンの3種類に分類する。

クライアントは、生徒や学生がコンテンツを利用するために用いるコンピュータである。パソコンをはじめ、タブレットやスマートフォンなども対象とする。クライアントとしての純粋な用途はコンテンツを取得して授業や自学自習に利用することであるが、この用途だけでコンピュータが有する資源を上限まで使うことは考えづらい。よって、クライアントにも遊休資源が存在すると考えられ、他の遊休コンピュータと同様に別の用途を用意することで有効に活用したい。ここで、クライアントには、学校所有と個人所有の2通りが考えられる。学校所有であれば管理者は学校であるため、利用にあたり学校の裁量で制御することができる。しかし、個人所有の場合には、プライバシーやセキュリティなどの事情により、クライアントとしての純粋な用途以外での利用を強制できないことが考えられる。また、ソフトウェアのインストールなどの事前準備が必要となると、クライアントとしての利用が煩雑になってしまう。このため、クライアントが個人所有の場合には、純粋にコンテンツを利用するのみのコンピュータとして扱い、学校所有の場合のみそれ以外の用途にも利用するといった使い分けができることが必要となる。

また、学校内のネットワーク上にある遊休コンピュータについて、常時起動しているか否かにより業務サーバと業務パソコンの2つに分類する。業務サーバは、常時起動しているコンピュータであり、情報システムを常時稼働させるために利用されるサーバなどを想定する。常時起動しているために、そのコンピュータからクライアントに対してサービスを提供する場合には、障害発生時を除いて原則と

して常時利用できることになる。ただし、クラウドサービスなどの利用により、学校内のネットワークには業務サーバが全く存在しないこともある。このため、業務サーバは必須とはせず、存在すれば利用することとする。一方、業務パソコンは、事務処理などで必要な時にのみ利用されるコンピュータである。このコンピュータは必要時のみ起動し、用途を終えて必要でなくなった場合には停止する。この起動と停止の頻度やタイミングは業務パソコンの利用者や用途によってまちまちであり、制御することはできない。このため、業務パソコンを利用する場合には、頻度やタイミングが一律ではない起動と停止が発生することを前提とする必要がある。また、業務パソコンはクライアントとなる可能性もあり、これらの状況を考慮する必要がある。なお、業務サーバも同様にクライアントとなる可能性があるが、今回は想定しない状況とする。

2.3 構成

提案する CDN の構成を図 1 に示す。管理サーバ、業務キャッシュサーバ、クライアントにより構成される。キャッシュサーバとクライアントの双方を兼ねる遊休コンピュータも存在するため、図中では双方の群に所属する形で表現している。

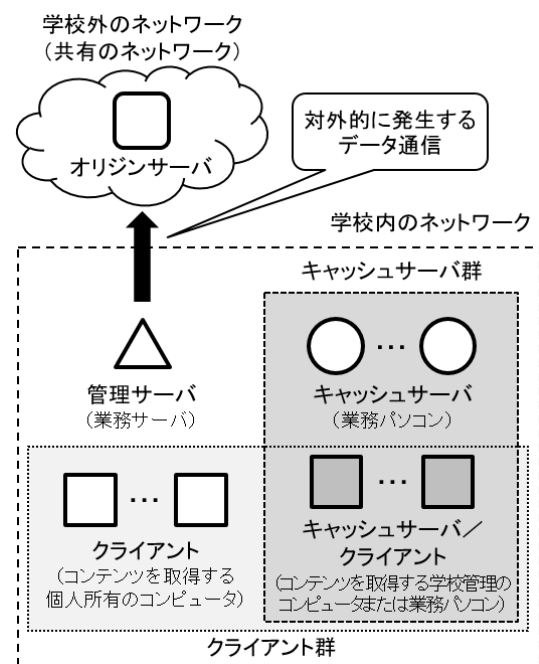


図 1 提案する CDN の構成

Figure 1 The configuration of the proposed CDN.

管理サーバは、クライアントからのコンテンツを取得する通信を中継するプロキシサーバとしての機能を持ち、クライアントがコンテンツを取得する際には始めに必ず管理サーバを中継するように構成する。また、キャッシュサーバと、キャッシュサーバが有するコンテンツのそれぞれの一覧を保持する。これらにより、クライアントが取得したいコンテンツの複製をキャッシュサーバに配置する、キャッ

シュサーバに配置した複製をクライアントに配信するといった処理を担う。管理サーバにより処理を分岐する構成とすることでクライアントでの処理を不要とし、導入や利用を容易とする。このような処理を行うため、管理サーバは常時利用可能であることが求められる。このため、業務サーバを管理サーバとして利用することを想定するが、業務サーバが存在しない場合には別途専用のコンピュータを用意する。なお、構成上、管理サーバが1台のみの場合は単一障害点となるため、複数用意しておくなどの対応により負荷分散や耐障害性の向上を図る必要がある。

キャッシュサーバは、管理サーバの指示によりコンテンツの複製を保持し、保持する複製をクライアントに配信する。このため、キャッシュサーバは、コンテンツの複製を保持するストレージとしての利用が主となる。キャッシュサーバには、業務パソコンおよびキャッシュサーバとして利用してもよいクライアントの2種類のコンピュータを利用する。これらのコンピュータは、起動と停止が制御できず、利用できるか否かがその時々状況により異なる。このため、特定の業務パソコンやクライアントをキャッシュサーバとして常時利用するのではなく、キャッシュサーバとして活用可能な業務パソコンやクライアントは予め全て管理サーバに登録および管理サーバにて起動状態を監視しておき、コンテンツの複製の配置や取得の際に利用できるものを実際に利用する。なお、キャッシュサーバとして利用可能なコンピュータが全く存在しない場合も十分考えられるが、この場合でもオリジナルのコンテンツはオリジンサーバが保持しているため、対外的な通信が発生することになるが、当該コンテンツをクライアントが利用できないという状態にはならない。

提案する CDN では、対象とするコンテンツが学校外のネットワーク上にある場合、オリジンサーバも学校外のネットワーク上に存在することになる。コンテンツを利用するクライアントが多数存在する場合に、オリジンサーバからの取得を初回のみとして以降は複製を配信することで、対外的な通信を少なくする。なお、コンテンツが学校内のネットワークに存在することも考えられる。この場合には学校外のネットワークにおけるデータ通信量を削減することには繋がらないが、当該コンテンツを持つオリジンサーバへの負荷の集中を避け、負荷の分散を図ることができる。オリジンサーバが学校内と学校外のネットワークのどちらにあるのかは、コンテンツ取得時の URI の違いでしかないので、CDN の構成を変更する必要はない。

2.4 コンテンツの配置と配信

管理サーバは、クライアントからのコンテンツを取得する際の通信に含まれる URI を取得することで、クライアントが取得したいコンテンツを識別する。URI はネットワーク上で一意という特性があるため、URI によりコンテンツを特定することができる。この情報と、利用可能なキャ

ッシュサーバおよびそのキャッシュサーバが保持するコンテンツの一覧とを突き合わせることで、処理を分岐する。

利用可能なキャッシュサーバ上に取得したいコンテンツの複製が存在しない場合における、基本的な処理の流れを図2に示す。クライアントがコンテンツを取得しようとする、管理サーバによりその通信が中継される(①)。管理サーバは処理を分岐の上、クライアントに代わりコンテンツをオリジンサーバから取得し(②と③)、利用可能なキャッシュサーバに複製を配置する(④)。併せて、キャッシュサーバとコンテンツの一覧を更新する。その後、クライアントに対して、複製を配信するキャッシュサーバへのリダイレクトを要求し(⑤)、クライアントは当該キャッシュサーバに接続してコンテンツを取得する(⑥と⑦)。利用可能なキャッシュサーバ上に利用したいコンテンツが存在する場合には、②から④の処理をとばし、①の次の処理が⑤のリダイレクトを要求する処理となる。

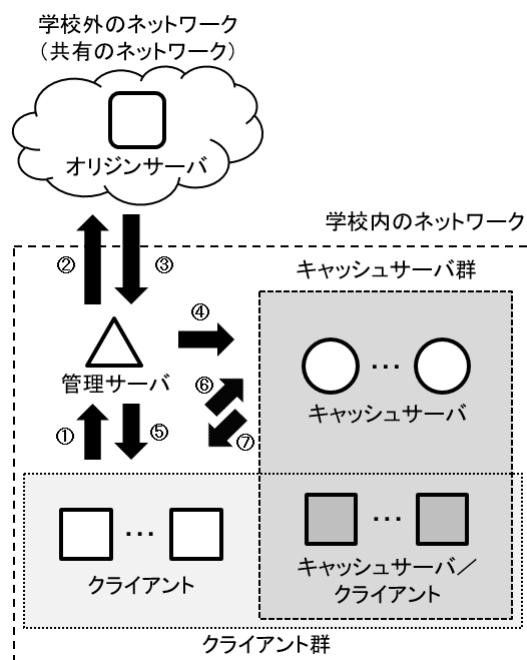


図2 処理の流れ

Figure 2 The Process flow.

複製の配置先は、利用可能なキャッシュサーバの中からストレージの空き容量が多い順番に複数を選択して配置する。また、配信元は当該コンテンツを所持している利用可能なキャッシュサーバからランダムに選択する。なお、将来的な課題として、キャッシュサーバの本来の用途への影響や本来の用途による制限、クライアントとの位置関係などを考慮した適切な複製の配置先、配置数、配信元の選択が挙げられる。

また、利用可能なキャッシュサーバ上に存在しないコンテンツに対して同時に大量のアクセスが発生する場合には、そのコンテンツのキャッシュサーバへの配置およびキャッシュサーバからの配信が、クライアントからの要求に間に

合わないことが考えられる。間に合わない場合には、対外的なデータ通信量を削減することを優先し、キャッシュサーバへの配置が完了するまでクライアントはコンテンツの取得を待つこととする。ただし、事前にそのコンテンツを利用することが分かっている場合における大量の同時アクセスであれば、事前に当該コンテンツをクライアントで取得する操作をしておけば、キャッシュサーバに当該コンテンツが配置されるために、待ちが発生する状況を回避することができる。現状ではこのような方針とするが、待ち時間を削減する手法の検討や、対外的なデータ通信量と待ち時間とのバランスを取ることが課題となる。

2.5 キャッシュサーバとコンテンツの管理

キャッシュサーバとコンテンツの情報は管理サーバにて管理する。この情報は、コンテンツの配置先や配信元を選定するために利用する。

キャッシュサーバの情報は、予めキャッシュサーバとして利用する業務パソコンおよびクライアント側から管理サーバに接続することで登録しておく。通信するために必要な IP アドレスや起動状況、コンテンツの複製を配置するために必要なストレージの空き状況などの情報を管理し、これらの情報は管理サーバとキャッシュサーバ間の定期的な通信や、キャッシュサーバからの起動や停止に伴う通信により更新する。定期的な通信を行うことで、キャッシュサーバの予期せぬ停止などの障害発生に対応する。

コンテンツの情報は、コンテンツの複製を配置する際に登録する。管理サーバがコンテンツの識別やクライアントへのリダイレクト先を示すために必要な、オリジナルのコンテンツを取得する際の URI やコンテンツの複製を保持するキャッシュサーバの情報を管理する。また、コンテンツの直近の取得回数とコンテンツの取得日時を保持しておき、直近の取得回数が閾値を下回り且つ一定期間が過ぎたコンテンツはキャッシュサーバから削除する。これにより、コンテンツの複製がキャッシュサーバに残り続けることを防ぐ。ただし、この手法ではオリジナルのコンテンツが更新される場合に複製を自動で更新できない。管理者が複製を削除するなどにより手動での対応は可能であるが、利便性が悪いため、今後の検討課題である。

3. 実装

設計に基づき、想定する実装を示す。

3.1 プロキシサーバ

クライアントからのコンテンツへの通信を中継するプロキシサーバは、OpenResty[7]にて構築できる。OpenResty は Nginx に lua-nginx-module など標準で組み込んだものであり、通信内容に応じた処理を軽量なスクリプト言語である Lua により記述できる。これにより、コンテンツの URI を取得し、後述するテーブルの情報と突き合わせ、複製の配置やリダイレクトなどに処理を分岐する。なお、負荷分

散や冗長化が必要な場合には、DNS ラウンドロビンやロードバランスを導入するなどにより対応できる。

3.2 キャッシュサーバとコンテンツの管理

キャッシュサーバとコンテンツの情報の管理には、管理サーバ上に MariaDB などで RDBMS を構築することで対応できる。RDBMS では、表 1 に示すキャッシュサーバの一覧を管理するテーブルと、表 2 に示すコンテンツの一覧を管理するテーブルを保持する。コンテンツを配置する際には、表 1 より利用可能なキャッシュサーバを得ることで処理を行う。また、コンテンツを配信する際には、表 2 より配置先のキャッシュサーバの情報を得た後、表 1 よりそのキャッシュサーバの情報を得ることで処理を行う。なお、表 2 の、コンテンツを保持するキャッシュサーバの識別番号とそれに対応するコンテンツへの配置日時の項目は、コンテンツの複製を配置する数に応じて増やす。なお、プロキシサーバと同様にデータベースについても負荷分散や冗長化を行う場合には、複数のサーバ上の RDBMS で同期を取る、Cassandra[8]などの NoSQL のデータベースを利用するなどにより対応できる。

表 1 キャッシュサーバの一覧を保持するテーブル

Table 1 The table that holds a list of cache servers.

項目
キャッシュサーバの識別番号
キャッシュサーバの IP アドレス
キャッシュサーバの起動状況
キャッシュサーバのストレージの空き容量

表 2 コンテンツの一覧を保持するテーブル

Table 2 The table that holds a list of content.

項目
コンテンツの識別番号
コンテンツを取得する際の URI
コンテンツのファイル名
コンテンツの直近の取得回数
コンテンツを保持するキャッシュサーバの識別番号
上記のキャッシュサーバへのコンテンツの配置日時

3.3 キャッシュサーバにおけるコンテンツの配置と配信

キャッシュサーバにおけるコンテンツの配置と配信は、minIO[9]によりオブジェクトストレージを構築することで実現できる。minIO は Windows, macOS, Linux などの複数の OS 上でオブジェクトストレージを構築できるオープンソースソフトウェアであり、異なる OS が混在する環境でも同じように扱うことができる。キャッシュサーバとして利用する業務パソコンやクライアントには、事前に minIO と、管理サーバと通信するためのソフトウェアをインストール

ールしておく。なお、キャッシュサーバとして利用したくないクライアントには、これらのソフトウェアをインストールしないことで純粋なクライアントとして利用できる。

4. 関連研究

コンピュータが有する遊休資源を活用する代表的な事例として、ボランティアコンピューティング[10]がある。これは、ボランティアにて提供されたコンピュータが有する遊休資源を集め、大規模で高速な計算能力を有するスーパーコンピュータとして活用する取り組みである。ウイルスの解析などに活用されているが、主に大規模な計算処理を行うことを想定しており、ボランティアコンピューティングにより CDN などのコンテンツの配信を実現している事例は見当たらない。

CDN においては、コンテンツの配置や配信のアルゴリズムに関する研究が多く存在し、データ通信量の削減やクライアントへの高速な配信などを実現する様々な手法が提案および評価されている[11][12]。また、CDN のプロバイダや ISP が提供する CDN を利用するのではなく、自組織内への CDN の構築を想定したものとして[13]がある。これは、組織を構成する拠点が物理的に分散して存在することを前提とし、CDN のプロバイダや ISP が運用する CDN に対して自組織内に構築する場合の相違点を整理した上で有効な設計について提案している。しかし、CDN を構成するコンピュータは CDN の管理者の制御下にあることが前提であり、起動停止などの制御が他者に委ねられているようなコンピュータで CDN を構築することは想定されていない。また、コンテンツの利用者は、自組織内の者ではなく外部の者が想定されている。

また、コンテンツを共有する手法としては、BitTorrent[14]に代表される P2P (Peer to Peer) 技術を基盤とするものがある。これは、コンテンツを持つサーバがコンテンツを要求するクライアントに対して配信するという基本的な CDN などの方式に対し、システムに参加するコンピュータが対等な役割や機能を有することで、コンテンツを相互に要求および配信するという方式である。これにより負荷の分散や冗長性の向上を実現できる。CDN の一部に P2P 技術によるコンテンツ共有を適用した事例[15]もあり、P2P 技術は提案する CDN の一部として適用可能だが、今回は構成や運用の簡素化のために採用していない。

5. おわりに

対外的なデータ通信量を削減することで共有のネットワークにおけるデータ通信量を削減する、遊休資源を活用したオンプレミス型の CDN を提案した。学校内のネットワーク上に独自の CDN を構築することで、学校外のネットワーク上にあるコンテンツを各自がそれぞれ取得する場合に対して、対外的なデータ通信量を削減することができ

る。また、CDN を構成するコンピュータとして学校内のネットワーク上に存在する遊休コンピュータを活用することで、独自の CDN を安価に構築できる。今後は実装を終え、有効性の評価を行う。また、遊休コンピュータの本来の用途への影響の評価、遊休コンピュータやクライアントの性能や状態を考慮したコンテンツの配置や取得の手法などを検討していく。

謝辞 本研究は JSPS 科研費 19K03081 の助成を受けている。

参考文献

- [1] Toshiya Kawato, Masayuki Higashino, Kenichi Takahashi, Takao Kawamura. Attempt to Utilize Surplus Storage Capacity as Distributed Storage. The 3rd International Conference on Information and Computer Technologies, 2020, pp. 351-355.
- [2] Toshiya Kawato, Masayuki Higashino, Kenichi Takahashi, Takao Kawamura. Proposal of Distributed e-Learning System using Idle Resources. The 5th International Conference on Computer and Communication Systems, 2020, pp. 557-561.
- [3] “GIGA スクール構想の実現について：文部科学省”. https://www.mext.go.jp/a_menu/other/index_00001.htm, (参照 2020-08-19).
- [4] “インターネットトラフィック関連動向について”. https://www.nii.ac.jp/news/upload/20200417-4_Mic.pdf, (参照 2020-08-19).
- [5] “データダイエットへの協力のお願い：遠隔授業を主催される先生方へ”. <https://www.nii.ac.jp/event/other/decs/tips.html>, (参照 2020-08-19).
- [6] Vijay K. Adhikari, Yang Guo, Fang Hao, Volker Hilt, Zhi-Li Zhang, Matteo Varvello, Moritz Steiner. Measurement Study of Netflix, Hulu, and a Tale of Three CDNs. IEEE/ACM Transactions on Networking, 2015, vol.23, no.6, pp.1984-1997.
- [7] “OpenResty® - Official Site”. <https://openresty.org/>, (参照 2020-08-19).
- [8] “Apache Cassandra”. <https://cassandra.apache.org/>, (参照 2020-08-19).
- [9] “MinIO | High Performance, Kubernetes Native Object Storage”. <https://min.io/>, (参照 2020-08-19).
- [10] Muhammad Nouman Durrani, Jawwad A. Shamsi. Volunteer computing: requirements, challenges, and solutions. Journal of Network and Computer Applications, 2014, Vol. 39, pp. 369-380.
- [11] Takuma Nakajima, Masato Yoshimi, Celimuge Wu, and Tsutomu Yoshinaga. Color-based Cooperative Cache and its Routing Scheme for Telco-CDNs. IEICE Transactions on Information and Systems, 2017, vol.E100-D, no.12, pp.2847-2856.
- [12] M. Zubair Shafiq, Amir R. Khakpour, Alex X. Liu. Characterizing caching workload of a large commercial Content Delivery Network. The 35th Annual IEEE International Conference on Computer Communications, 2016.
- [13] Riccardo Lancellotti, Claudia Canali, Andrea Corbelli. On Private CDNs with Off-Sourced Network Infrastructures: a Model and a Case Study. Journal of Communications Software and Systems, 2018, vol.14, no.4, pp.376-185.
- [14] “BitTorrent.org”. <https://www.bittorrent.org/>, (参照 2020-08-19).
- [15] 西村敏, 田中壮, 遠藤洋介. P2P-CDN 連携型ライブ配信システムの試作. 2013 年映像情報メディア学会冬季大会, 2013, p.12-9.