D-03

P2P ファイル共有ネットワークを利用した フラッシュクラウド耐性のある協調型負荷分散手法

Cooperative Load Distribution for Addressing Flash Crowds Using P2P File Sharing Network

岡本 大樹† 岡部 寿男‡
Hiroki Okamoto Yasuo Okabe

1. はじめに

世界のインターネット人口は年々増加を続けており，現在約 22 億人に達している[1]．また新興国の発展に伴い，今後その数はさらに増加することが見込まれる．そのような状況で，近年，フラッシュクラウド(flash crowd)と呼ばれる Web サーバへの急激な負荷の増加が問題となっている．フラッシュクラウドとは，ネットワークや Web サーバが突然大量のトラフィックを受ける現象で，その負荷は平常時の数十倍から数百倍に及ぶ[2]．フラッシュクラウドは，典型的には有名な Web サイトやブログで当該の Web コンテンツが紹介されたことが契機で生じ，その発生も規模も予測が困難である．

これまで，Web サーバの負荷の高まりに対するサーバ側の一般的な対策としては，リバースプロキシによるキャッシュやロードバランサーによる複数マシンへの振り分け，Content Delivery Networks[3]へのキャッシュの分散などが用いられてきた．しかし，フラッシュクラウドは恒常的な高負荷とは違い，短時間の間に多くのトラッフィクが発生し比較的短時間に収束することや発生の予測が難しいといった特徴があり，予めの見積もりが必要となるこれらの方法ではうまく対処する事ができない．

また，近年いわゆるクラウドサービスの中でも，IaaS(Infrastructure as a Service)と呼ばれるインフラの設備を仮想化しサービスとして提供する手法が注目されている．IaaS では，必要に応じて柔軟に設備を増強する事ができるため，予測できない負荷に対しても対応が可能である．しかし，設備の増強に応じて料金が発生するため，金銭的な理由などによって，平常時の設備以上に設備の増強が出来ない場合は，フラッシュクラウドなどの短時間の急激な高負荷に対処することが出来ない．また，そのような設備の増強は通常は管理者による設備の追加契約と設定が必要であり，迅速な対応にも限界がある．

そこで，本研究では，個人などが運営する Web サーバ同士が P2P ファイル共有ネットワークを通じてデータをやり取りできるようにし，協調させることでフラッシュクラウドに柔軟に対処する協調型の負荷分散手法を提案する．本提案手法では，それぞれの持つ Web サーバが遊休資源を柔軟に融通し合い，また P2P ファイル共有ネットワークへコンテンツを拡散することでフラッシュクラウドに対処する．そのため，金銭的な理由などによって，物理的，仮想的問わず，新たに設備を増強することが出来ない場合でも，フラッシュクラウドのような短時間の急激な高負荷に対処することが可能である．

以下，2 章では Web サーバの協調型の負荷分散手法と P2P 型の Web システムに関する関連研究について述べる．3 章では，提案手法の要件と基本設計について述べ，4 章でシステムとしての実装に関する検討を行い，5 章でまとめと今後の方針について述べる．なお，提案システムでは，P2P ファイル共有ネッワークとして BitTorrent を用いる．これは，BitTorrent のシステムは他の P2P ファイル共有ネットワークと比較してネットワークへ参加することによる負荷が小さく，また BitTorrent ネットワークからファイルをダウンロードする機能を備えたブラウザが存在しており Web のシステムとの親和性が高いと判断したためである．そのため，性能を別にすれば，BitTorrent に限らず，ネットワークに参加するピア同士で自由にデータの受け渡しができる P2P ファイル共有ネットワークであれば，同様の役割として利用することが可能であると考えている．

2. 関連研究

本研究が考える協調型の負荷分散と同様の研究として，T.Stading らは，HTTP リダイレクトによって協調する Web サーバに負荷を分散する手法を提案している[4]．Backslash と呼ばれるこのシステムでは，コンテンツの提供は Web サーバのみが行い，高負荷時は HTTP リダイレクトによって協調する他の Web サーバへリクエストを分散させる．これに対して，本提案手法では，クライアントの一部も負荷分散に参加し，過去に自身が取得したコンテンツを提供するため，フラッシュクラウド発生時により広範囲に負荷を分散することができる．

V.Padmanabhan らは，同様に CoopNet と呼ばれるクライアントも負荷分散に参加するシステム[5]を提案している．このシステムでは， P2P ネットワークに参加するクライアントへのルーティングを，Web サーバが全て担っている．そのため，急激な負荷によって Web サーバがダウンしてしまうとコンテンツの提供が止まってしまう．これに対して，本提案手法では，P2P ファイル共有ネットワークに参加するクライアントへのルーティングは，Web サーバではなく，P2P ファイル共有ネットワーク自身の仕組みによって行う．そのため，急激な負荷によって Web サーバがダウンしてしまった状況でも，コンテンツにアクセスすることが可能である．

Yokota らは，リバースプロキシを Web サーバの前に設置し，そこでフラッシュクラウドを検知するとともに，これまでデータを取得したクライアントを記録しておき，フラッシュクラウド時は，この記録しておいたクライアントへリダイレクトする方式を提案している[6]．この方式では，コンテンツへのルーティングをプロキシサーバに依存するため，プロキシサーバのダウンによってコンテンツへのアクセスが不可能になってしまう．また，プロキシサーバを管理/運用する労力も発生する．これに対し，提案手法では

† 京都大学 大学院情報学研究科
Graduate School of Informatics, Kyoto University
‡ 京都大学 学術情報メディアセンター
Academic Center for Computing and Media Studies, Kyoto University

Web サーバがリクエストのリダイレクトを行うため，新たに管理/運用の必要な機器を導入する必要が無い．また高負荷によって自身のサーバがダウンしてしまった場合でも，協調サーバやファイル共有ネットワークから直接コンテンツを取得することが可能である．

C.Pan らは，フラッシュクラウド下にあるデータへのリクエストを，キャッシュを行うプロキシサーバで構成されるオーバレイネットワークにリダイレクトする方式を提案している[2][7]．この方式は，フラッシュクラウドが発生した際に，動的にオーバレイネットワークを構築し負荷分散できるため，柔軟にフラッシュクラウドに対処することができる．しかし，当該の Web サーバがオーバレイネットワークの構築の指示や管理などを全て担っているため，フラッシュクラウド時に Web サーバが何も出来ないままダウンしてしまった場合や上流のネットワークの輻輳によってメッセージが伝達できない場合に問題が残る．これに対して本提案手法では，協調関係にあるサーバ同士が互いに生存確認を行い，ダウンしている場合には代理で Web コンテンツの提供を行うため，より可用性が高くなっている．

また，Client Server モデルに基づいた従来の Web とは違った P2P モデルに基づいた Web に関する研究も行われている[8][9][10]．これらの P2P モデルに基づいた Web は，従来の Web とは違い，負荷の集中が起き難いという利点があるが，これまでの Web の資源が活用できないため，システムを移行する際のコストが大きいという問題がある．これに対し本提案手法では，既存の Web の資源を活用することができるため，システムの移行に掛かるコストが小さく，比較的容易に既存のシステムに適用することが出来る．

3. 提案手法

3.1 提案手法の要件

以下，問題を簡単にするため，Web サイトは静的なコンテンツのみで構成されているものとするが，動的コンテンツであってもデータベースなどへの書き込みがなく単純に分散できるようなものについては同様に扱える．また，本提案手法はフラッシュクラウドのような短時間の急激な高負荷に対処することを目的としているため，恒常的な高負荷は対象外とする．

フラッシュクラウドは，短時間に大量のトラフィックが発生することと発生の予測が難しいことが特徴である．これに対して，従来の負荷分散手法では，管理/運用の労力，費用，柔軟な設備の増強が出来ないなどの問題からうまく対処する事ができない．特に柔軟な設備の増強ができない問題は深刻であり，従来の負荷分散手法では，フラッシュクラウドによる短時間の急激な高負荷に合わせて設備を準備した場合，平常時には無駄が多くなり，また平常時に合わせて設備を準備した場合，フラッシュクラウド時にコンテンツを提供できなくなってしまう．そのため，フラッシュクラウドに対処するためには，柔軟に設備を増強できることが必要となる．また，リバースプロキシやロードバランサー等の機器などは増強するほどハードウェアの台数が増え，管理や運用の労力が増加する．そのため，管理や運用のための人員が必要となり，その数が負荷分散能力の限界となってしまう．そこで，この問題を解決するためには，設備の増強に合わせて新たに管理や運用の労力が増加しない仕組みが必要となる．また，IaaS などは柔軟に設備の増強ができ，また仮想化された設備をサービスとして利用するため，ハードウェアの管理や運用の労力も必要ない．しかし，使用した分量に合わせて従量制で費用が発生するため，フラッシュクラウドの場合は想定外の高負荷に応じた料金を請求されるなど，あまり費用を掛けられない Web サイトの運営者などと相性が悪い．そのため，このような費用を掛けられない Web サイトもフラッシュクラウドに対処するためには，無償で設備を融通し合う，相互扶助的な仕組みが必要となる．

これらにより，提案手法は以下の要件を満たす必要がある．

（1）負荷に応じた柔軟な設備の増減が出来る
（2）負荷に応じた料金が発生しない
（3）負荷分散のために新たに HW の管理/運用の労力が生じない

3.2 提案手法の設計

提案手法では，平常時は通常通り自身の Web サーバで処理を行い，負荷の高まりに応じて予め協調関係を結んでおいた他のサーバへ負荷を分散する．また，協調関係にある他のサーバの負荷が高まった場合は，そのサーバに代わって自身の Web サーバでリクエストを処理する．この協調関係にあるサーバは，自身のサイトを代理で提供することを許可するほか，Dynamic DNS を用いた DNS レコードの変更なども許可するため，相応に信頼でき，またお互いに責任の取れる関係のものでなければならない．そのため，その数は 50 台から 100 台程度を想定している．

また，提案手法では，負荷が高まっている協調関係にあるサーバの Web サイトを自身のサーバで提供する際，その Web サイトのデータの取得を，P2P ファイル共有ネットワークから行う．P2P ファイル共有ネットワークは，多くの人がアクセスする人気のあるファイルほど，多くのキャッシュが存在し，ダウンロードに掛かる時間が短くなるという特徴があるため，P2P ファイル共有ネットワークをデータ交換に利用することによって，高負荷の状況において素早く Web サイトのデータを取得し代理で提供を開始することが可能となる．また，P2P ファイル共有ネットワーク上にデータが拡散することで，P2P ファイル共有ネットワークからクライアントが直接 Web サイトを取得することも可能となるため，オリジナルの Web サーバや協調関係にある Web サーバが全てダウンしてしまった状況でも部分的にコンテンツの提供を続ける事が出来るという利点も生まれる．これらの理由から，本提案手法は P2P ファイル共有ネットワークをデータ交換に用いる．またこれにより，本提案手法の P2P ファイル共有ネットワークは協調関係にある Web サーバのグループと一部の P2P ファイル共有ネットワークに参加可能なクライアントによって構成されることになる．

提案手法では，この比較的少数で構成される協調サーバ群への負荷分散と Web サーバとクライアントで構成される P2P ファイル共有ネットワークへの負荷分散の 2 つにより柔軟な設備の増強を実現する．また，提案手法において，Web サイトの運営者にとって管理が必要となる機器は，自身の Web サーバのみであり，負荷分散のために新たにハードウェアの管理や運用の労力は生じないようになっている．また，提案手法では，前述の相互扶助に基づく仕組みと Web サーバとクライアントで構成される P2P ファイル

共有ネットワークを使って負荷分散を行うため，負荷に応じて利用料が発生することがなく，金銭的な制約により設備を増強できない Web サイトの運営者でもフラッシュクラウドに備えることが可能となっている．

4. システムの設計

4.1 システムの構成

本提案手法のシステムは，図１に示す様に自身の Web サーバ，協調する他の Web サーバ，DNS，Web サイトのクライアント，BitTorrent トラッカーの５つの要素で構成される．なお，DNS はそれぞれの Web サイトが使用しているものを用いるため新たに用意する必要は無い．ただし，提案システムにおいて用いるためには，Dynamic DNS の機能が使用可能である必要がある．また，BitTorrent トラッカーについても既存の BitTorrent ネットワークで一般に公開し使用されているものを用いる．

（1）自身の Web サーバ
（2）協調する他の Web サーバ
（3）DNS
（4）クライアント
（5）BitTorrent トラッカー

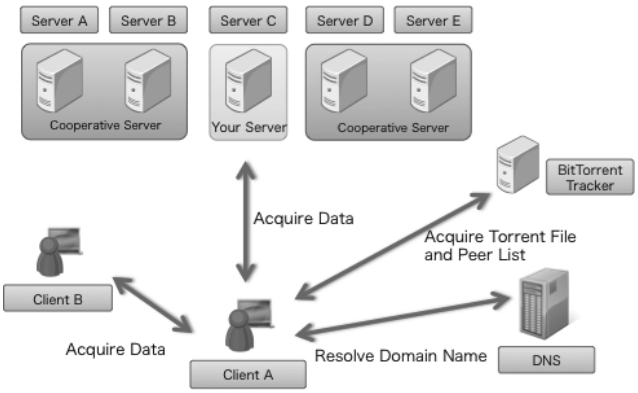

図 1. 提案手法の構成

4.2 システムの備える機能

負荷の測定とフラッシュクラウドの検知

提案システムにおける Web サーバは自身の負荷を常に測定し，予め定めた閾値を超えるとフラッシュクラウドの状況にあると判断し，協調関係にある他のサーバへ通知を行い，負荷の分散を開始する．また，ここで用いる負荷の指標は，Web サーバ内で設定されている，同時接続可能なクライアント数の上限，プロセス数・スレッド数の上限，保留状態のコネクションの最大数などの設定値に対する現在の値の割合などを用いることを検討している．
また，急激な負荷によって通知をする間もなくダウンしてしまう場合も考えられるため，協調関係を結んだサーバ同士はお互いに定期的に生存確認を行う．なお，この生存確認には，Ping，TCP コネクション,HTTP GET などを用いる事を検討している．

リクエストの分散

提案システムでは，HTTP リダイレクトと Dynamic DNS によって DNS レコードを追加した後の DNS ラウンドロビンの２つによって Web サーバへのリクエストを分散する．提案手法における Web サーバは，フラッシュクラウドを検知すると HTTP リダイレクトによって，一定量のリクエストを協調関係を結んだ他のサーバへ振り分ける．また，この HTTP リダイレクトは協調関係にあるサーバへ助けを求めるメッセージも兼ねている．HTTP リダイレクトによって他のサーバのコンテンツを要求された Web サーバは，BitTorrent ネットワークから対応するコンテンツをダウンロードし提供を開始する．また，この時 Dynamic DNS を用いて，DNS へ自身のレコードを追加する．そのため，協調関係を結んだ Web サーバ同士は予め DNS に相手のサーバが登録情報の変更を行うことを許可する設定をしておく必要がある．

データの拡散

提案システムでは，Web サイトのデータを拡散させる方式について３つの方式を想定している．

（1）予め協調サーバへ配布
（2）高負荷時に協調サーバへ配布
（3）クライアントへの拡散

まず予め協調サーバへ配布する方式は，フラッシュクラウドによってクライアントにも協調関係にある Web サーバにもデータを渡せずにダウンしてしまう場合を想定しており，Web サイトのデータを新たにアップロードしたり更新したりする際に，事前に承諾を得られた特定の協調関係にある Web サーバへ，BitTorrent 経由で Web サイト全体のデータを配布しておく方式である．ただし，Web サイト全体のデータのサイズによっては，相手側のサーバが受け入れられない可能性もあるため，協調関係にある他のサーバは Web サイト全体のデータの受け入れをどの程度のサイズまで許可するか設定できる必要があると考えている．
また，高負荷時に協調サーバへ配布する方式は，Web サイト全体ではなく，協調サーバへ HTTP リダイレクトしたデータのみを BitTorrent 経由で配布する方式である．この方式は，必要最小限のデータのみを配布するため，協調関係にあるサーバのストレージを圧迫しない利点がある．
また，提案システムでは Web サイトのデータを BitTorrent 経由で取得することが可能であるため，BitTorrent が使用可能なクライアントは，BitTorrent ネットワークに参加することで，BitTorrent から Web サイトのデータを取得し閲覧できる．この際，このクライアントも Web サイトのデータを保持することになり，次の BitTorrent 経由でのリクエストの際には，データの提供者の１つとして機能する．

クライアントの種類

提案システムでは，BitTorrent ネットワークに参加するクライアントと BitTorrent ネットワークに参加しないクライアントの 2 種類のクライアントが存在する．BitTorrent ネットワークに参加するクライアントは，閲覧したい Web サイトについて，通常通り直接 Web サイトからデータを

取得する方法と，torrent ファイルを取得して BitTorrent ネットワークから Web サイトのデータを取得する方法のどちらかを選ぶことが出来る．このクライアントは，負荷が高まって取得しづらくなっているデータについて，P2P ファイル共有ネットワークの利点を生かして素早く取得できる他，サーバが全てダウンして，通常であれば完全に閲覧できなくなっている Web サイトについても BitTorrent 経由で閲覧できる可能性がある等のメリットがある．対して，BitTorrent ネットワークに参加しないクライアントは，通常通り直接 Web サイトからのみデータの取得を行う．そのため，高負荷時には，閲覧しづらくなったり，全ての Web サーバがダウンしてしまった場合は，Web サイトを閲覧できなくなる可能性がある．

状態の遷移

提案システムにおける Web サーバは，低負荷の状態とフラッシュクラウドの状態の 2 つの状態を遷移する．予め設定した閾値を超えると低負荷の状態からフラッシュクラウドの状態へ遷移し，協調関係にある他のサーバへ助けを求める．また，フラッシュクラウドの状態から低負荷の状態への遷移は，設定しておいた一定時間の経過によって自動的に遷移することを検討している．これは，フラッシュクラウドの特徴から高負荷の状態は長い持間続かないためである． またこの際，Web サーバは，フラッシュクラウド時に自身が利用する DNS に追加された協調サーバのレコードを削除する．

4.3 システムの動作

ここでは，想定される状況における提案システムの動作について述べる．なお，ここでは，便宜上，自分の運営する Web サーバへあるクライアントがアクセスしようとしている状況を想定する．

低負荷の状態＋通常アクセスの場合

自身の Web サーバが低負荷で，クライアントが通常通りの方法でアクセスしてくる場合は，従来の Web における Web ページの閲覧と同様である．そのため，クライアントは DNS で名前解決を行った後，Web サーバからデータを取得する．以下図 2 にその具体的な動作の流れを示す．

1. クライアントは DNS で自サーバへ名前解決.
2. クライアントは自サーバからデータを取得.

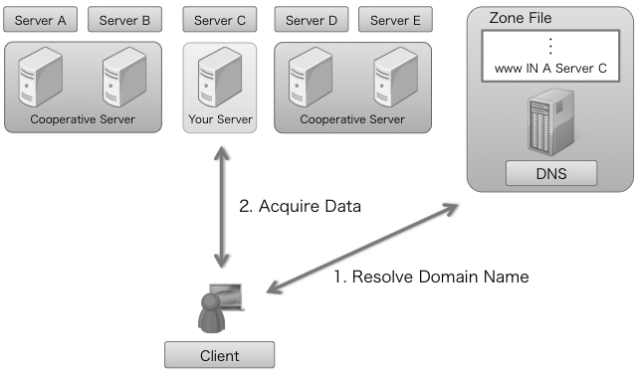

図 2. 低負荷の状態＋通常アクセス時の動作

低負荷の状態＋BitTorrent アクセスの場合

自身の Web サーバが低負荷の状態で，クライアントが BitTorrent ネットワーク経由でアクセスしてくる場合は，クライアントは Web サーバから.torrent ファイルを取得し，トラッカーからファイルを保持するピアのリストを取得した後に，BitTorrent を用いてファイルの取得を行う．以下の図 3 にその具体的な動作の流れを示す.

1. クライアントは DNS で自サーバへ名前解決.
2. クライアントは自サーバから.torrent ファイル取得.
3. クライアントは BitTorrent トラッカーからピアのリストを取得.
4. クライアントは BitTorrent からデータを取得.

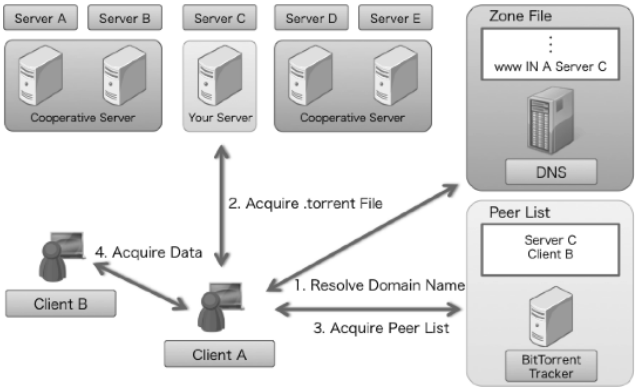

図 3. 低負荷の状態＋BitTorrent アクセス時の動作

フラッシュクラウドの状態＋通常アクセス

自身の Web サーバがフラッシュクラウドの状況でまだダウンしていない時にクライアントが通常の方法でアクセスしてきた場合，自身の Web サーバは直接データを返さず，協調サーバへ HTTP リダイレクトによってリダイレクトする．リダイレクトを受けた協調サーバは，BitTorrent から該当するデータを取得し，クライアントへ送信するとともに，自サーバがフラッシュクラウドを受けていると判断して，負荷分散を開始する．以下の図 4 にその具体的な動作の流れを示す.

1. クライアントは DNS で自サーバへ名前解決.
2. クライアントは自サーバへ HTTP でアクセス.
3. 自サーバは協調サーバへ HTTP リダイレクト.
4. クライアントは協調サーバへ HTTP でアクセス.
5. 協調サーバはトラッカーから.torrent ファイルとピアのリストを取得.
6. 協調サーバは BitTorrent からデータを取得.
7. 協調サーバはクライアントへデータを送信.
8. 協調サーバは DNS へ自身のレコードを追加.

これに対して，自身の Web サーバがフラッシュクラウドの状況で，その負荷によってダウンしてしまった時にクライアントが通常の方法でアクセスしてきた場合は，協調サーバが定期的な生存確認によって ダウンしていることを発見し，ダウンしている Web サーバが現在フラッシュク

ラウドを受けていると判断して，代理で Web サイト全体の提供を開始する．その際の具体的な動作は，以下の図 5 の様になる．

1. 協調サーバが定期的な生存確認によって自サーバのダウンを検知.
2. 協調サーバはトラッカーから.torrent ファイルとピアのリストを取得する.
3. 協調サーバは BitTorrent から自サーバの Web サイト全体のデータを取得.
4. 協調サーバは DNS に自身のレコードを追加.
5. 協調サーバは自サーバに代わって Web サイトの提供を開始する.
6. クライアントは DNS で協調サーバへ名前解決.
7. クライアントは協調サーバからデータを取得

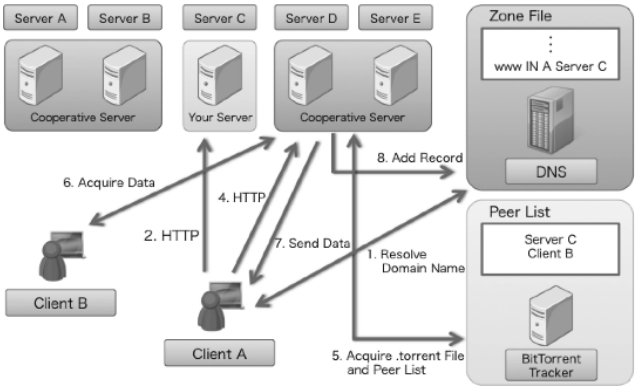

図 4. フラッシュクラウドの状態＋通常アクセス＋自サーバがダウンしていない時の動作

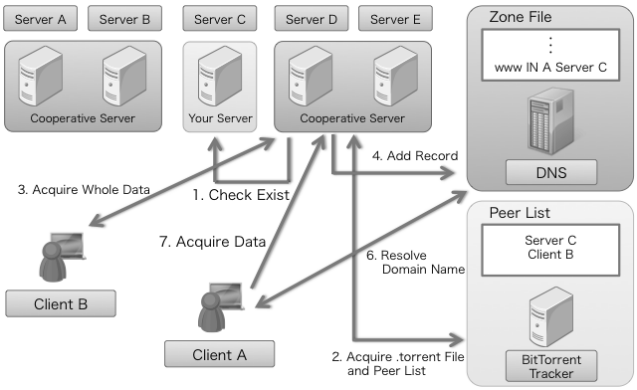

図 5. フラッシュクラウドの状態＋通常アクセス＋自サーバがダウンしている時の動作

フラッシュクラウドの状態＋BitTorrent アクセスの場合

自身の Web サーバがフラッシュクラウドの状況で，クライアントが BitTorrent 経由でアクセスしてくる場合は，クライアントは，トラッカーから対応する.torrent ファイルとファイルを保持するピアのリストを取得し，BitTorrent から直接データを取得する．以下の図 6 にその具体的な動作を示す.

1. クライアントはトラッカーから.torrent ファイルとピアのリストを取得.
2. クライアントは BitTorrent からデータを取得.

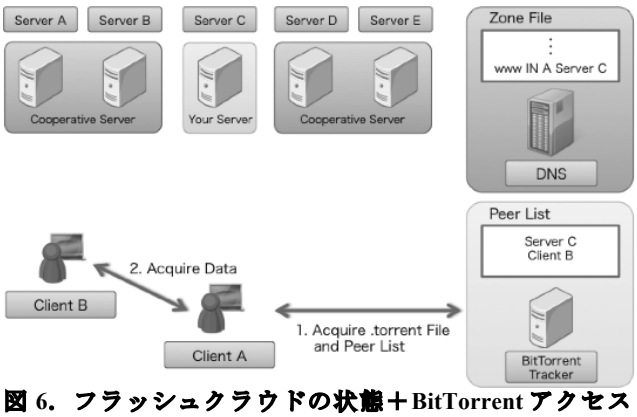

図 6. フラッシュクラウドの状態＋BitTorrent アクセス時の動作

5. おわりに

本稿では，個人などが運営する Web サーバ同士が P2P ファイル共有ネットワークを通じてデータをやり取りし，協調することでフラッシュクラウドに柔軟に対処する協調型の負荷分散手法を提案した．今後は，実際に提案手法のシステムを実装し，テストベッドなどでその性能を評価することで，その有効性を検証したいと考えている．なお，現在，実装については，Apache のモジュールとして Web サーバに必要なフラッシュクラウドの検知や協調サーバへのリダイレクト，DNS レコードの変更などの機能を実装し，Opera などのブラウザのプラグインとしてクライアントが P2P ファイル共有ネットワークに参加する際の機能を実装する方針で検討を進めている．また，テストベッドとしては，Planetlab などを用いることを検討している.

参考文献

[1] Internet World Stats. http://www.internetworldstats.com/stats.htm

[2] C. Pan, M. Atajanov, M. Belayet Hossain, T. Shimokawa, and N. Yoshida. FCAN: Flash crowds alleviation network. IEICE transactions on communications. Vol.E89-B. No.4. pages 1119-1126. April 2006.

[3] Akamai. http://www.akamai.com

[4] T. Stading, P. Maniatis, and M. Baker. Peer-to-peer caching schemes to address flash crowds. Peer-to-Peer Systems, pages 203–213, 2002.

[5] V. Padmanabhan, and K. Sripanidkulchai. The case for cooperative networking*. Peer-to-Peer Systems, pages 178-190, 2002.

[6] K. Yokota, T. Asaka, and T. Takahashi. A load reduction system to mitigate flash crowds on web server. ISADS 2011, pp. 503–508. IEEE, 2011.

[7] C. Pan, M. Atajanov, M. B. Hossain, T. Shimokawa, and N. Yoshida. FCAN: Flash crowds alleviation network. SAC'06, pages 759-765, 2006.

[8] Roberto J. Bayardo Jr., Rakesh Agrawal, Daniel Gruhl, and Amit Somani. Youserv: a web-hosting and content sharing tool for the masses. WWW '02, pp. 345–354, 2002. ACM.

[9] 池嶋俊, 阿部洋丈, 加藤和彦. Asagumoweb: P2P 技術を用いた web システム. 日本ソフトウェア科学会第 22 回大会, 2005.

[10] M. Bari, M. Haque, R. Ahmed, R. Boutaba, B. Mathieu, et al. Persistent naming for p2p web hosting. P2P'11, 2011 IEEE International Conference on, pp. 270–279. IEEE, 2011.