

コンテンツオリエンテッドネットワークにおける

Web サーバの負荷分散手法の提案

Proposal of load balancing method of the Web server in the content-oriented network

丹羽 徹†
Toru Niwa

川橋 裕‡
Yutaka Kawahashi

1. はじめに

インターネットが普及し、コンテンツ流通サービスである Web はインターネットで最も多く利用されているサービスになっている。Web サーバの需要が高まるにつれて、Web は利用者の日常生活に不可欠なものとなりつつある。このような背景により、Web サーバには次のような能力が求められている。

- Web サーバの能力が不足した際に、容易にシステムを拡張できる能力（拡張性）
- Web サーバに障害が発生してもサービスが停止しない能力（可用性）

上記の能力を考慮した Web サーバの運用管理が要求され、年々運用管理の重要性は増している。これらの要求を満たす手法の 1 つが負荷分散技術である。負荷分散とは、並列に運用されている Web サーバ間での負荷がなるべく均等になるように処理を分散して割り当てることである。負荷分散の既存技術として DNS ラウンドロビン、ロードバランサが挙げられる。

既存の負荷分散技術はロケーションオリエンテッド通信モデルにもとづくアーキテクチャ上で設計されている。この通信モデルではクライアントは Web サーバからコンテンツを取得する際に URL を指定する。URL は Web サーバの場所を表すドメイン名とコンテンツ名から構成されている。

しかし、既存技術には問題がある。既存技術のシステムでは、URL によってクライアントのアクセス先が一部の場所に限定されるため、同じコンテンツを持つ Web サーバが負荷分散システムの機能しないネットワークに存在する場合に、ネットワークを超えた負荷分散ができない。本稿では以降、負荷分散システムが機能するネットワークを内部ネットワークと表記し、負荷分散システムが機能しないネットワークを外部ネットワークと表記する。

上記の問題を解決するためには、同じコンテンツを持つ Web サーバ間で Web サーバやネットワークの状況を考慮することができる負荷分散技術が必要である。ネットワークを超えて、Web サーバやネットワークの状況を考慮した負荷分散をおこなうためには、コンテンツ名だけでコンテンツを取得できる通信モデルが必要である。この通信モデルはコンテンツオリエンテッド通信モデルと呼ばれる。コンテンツオリエンテッド通信モデルを実現するアプローチに Internet Indirection Infrastructure(i3) [1][2]がある。i3 はオーバーレイネットワークを利用してランデブー型通信モデルを具体的に実現したものである。この通信では、クライアントは Web サーバからコンテンツを取得する際にコンテンツ名を指定する。よって、クライアントは Web サーバの場所を意識する必要はない。したがって、同じコンテ

ンツを持つ複数の Web サーバ間で、ネットワークを超えた負荷分散技術の手法を考えることができる。

本研究では i3 のコンテンツオリエンテッドネットワークにおける Web サーバの負荷分散を目的とする。i3 プロキシと OpenFlow [3]を用いて Web サーバの負荷分散手法の提案と実装をおこなった。提案手法では、同じコンテンツを持つ Web サーバとネットワークの負荷を考慮して、クライアントがアクセスする Web サーバを決定できる。提案手法の動作実験を、小規模なネットワークでおこなった。

2. 負荷分散技術

インターネット上で公開されている Web サーバには、クライアントからのアクセスが集中した際に大きな負荷がかかる。このため、同じ役割を担う Web サーバを複数台導入して負荷を分散させる様々な手法が提案されている。Web サーバを複数導入した場合、クライアントがどのようにして複数導入した Web サーバの 1 つを選択するかが重要である。本章では、クライアントのアクセスする Web サーバを選択することによる負荷分散技術を分類し、紹介する。

2.1 Web サーバ選択手法の分類

負荷分散技術はそれぞれの Web サーバを選択する手法によって分類することができる。既存の負荷分散技術を、Web サーバの選択場所を基準に以下の 3 つに分類した。

- クライアント側で Web サーバを選択する技術
 - サーバ側で Web サーバを選択する技術
 - 中間システムで Web サーバを選択する技術
- 以降の節でそれぞれの技術について説明する。

2.2 クライアント側で Web サーバを選択する技術

クライアント側で Web サーバを選択する技術は、クライアントまたはクライアントを操作するユーザが複数ある Web サーバから任意の 1 台を選択することで負荷分散をおこなう技術である。この手法はクライアント側が負荷分散に必要な知識を持っていることを前提とするが、一般的にこの前提は期待できない。

2.3 サーバ側で Web サーバを選択する技術

サーバ側で Web サーバを選択する技術は、クライアントからの要求を受け付けたサーバが、クライアントのアクセスする Web サーバを選択することで負荷分散をおこなう技術である。

このサーバにロードバランサがある。これは、クライアントからの要求を一括して管理し、複数ある Web サーバの一部にこの要求を転送することで負荷分散をおこなう技術である。ロードバランサの判断で様々な状況に対応することが可能であり、柔軟性は高い。ただし、ロードバラン

†和歌山大学, Wakayama University

‡和歌山大学システム情報学センター, Center for Information Science, Wakayama University

さは Web サーバへのアクセスを一括して管理するため、ボトルネックになる可能性がある。

2.4 中間システムで Web サーバを選択する技術

クライアントとサーバの間にあるシステムが Web サーバを選択することで負荷分散をおこなう技術である。ネットワーク中には多様な中間システムが存在し、以下の種類に分類できる。

- アプリケーションゲートウェイ
 - メタサーバ
- 以降でそれぞれの種類について説明する。

2.4.1 アプリケーションゲートウェイ

アプリケーションゲートウェイとは、通信料の削減を目的としたキャッシュサーバ等の負荷分散システムである。クライアントと Web サーバは直接通信せずに、Web サーバを選択するアプリケーションゲートウェイを介して通信する。

2.4.2 メタサーバ

クライアントが Web サーバへアクセスする前に利用するメタサービスがあり、ドメイン名と IP アドレスの間を名前解決する Domain Name System(DNS)等が挙げられる。このメタサービスを提供するメタサーバで、クライアントがアクセスする Web サーバを選択するシステムである。

DNS サーバで Web サーバを選択し、Web サーバの負荷分散をおこなう手法として DNS ラウンドロビンがある。この手法では DNS サーバ内で 1 つのドメイン名に複数の IP アドレスを割り当てて、クライアントからの問い合わせがあるたびに順番に IP アドレスを答えることにより、1 台の Web サーバへアクセスが集中するのを防ぐ。

3. 研究目的

本章では、2 章で紹介した負荷分散技術の問題を示し、これを踏まえて本研究での研究目的を述べる。

3.1 既存の負荷分散技術の問題点

2 章で紹介した負荷分散技術の種類ごとに問題を示す。

3.1.1 クライアント側で Web サーバを選択する技術

クライアント側で Web サーバを選択する技術は、2.2 節で述べたようにユーザに本来不要な判断を強いている上に、手動で Web サーバを選択する必要があるという問題がある。

3.1.2 サーバ側で Web サーバを選択する技術

サーバ側で Web サーバを選択する技術は内部ネットワークでしか Web サーバの選択、すなわち負荷分散ができない。

例えば、ロードバランサはその下位にある Web サーバ群を Web サーバ選択の対象とし、URL によって指定された場所やネットワークで負荷分散をおこなう。逆に言えば、同じコンテンツを持つ Web サーバが外部ネットワークに存在する場合でも、ロードバランサは外部ネットワークの Web サーバも考慮した負荷分散ができない。すなわち、ロードバランサは外部ネットワークの Web サーバや外部ネットワークの状況を考慮した負荷分散ができない。したが

って、この負荷分散技術は URL によって負荷分散できる範囲が限定されていると言える。

3.1.3 中間システムで Web サーバを選択する技術

中間システムで Web サーバを選択する技術の問題を 2.4 節で分類した種類ごとに示す。

- アプリケーションゲートウェイ
アプリケーションゲートウェイが Web サーバを選択するシステムは、中間システムがクライアントや Web サーバの代わりに通信しているに過ぎない。したがって、サーバ側で Web サーバを選択する技術と同様に URL によって負荷分散できる範囲が限定されている。

- メタサーバ
メタサーバが Web サーバを選択するシステムは、利用するメタサービスによって、Web サーバを評価する際の評価基準の変更と導入の難易度が決定する。

例えば、DNS ラウンドロビンは DNS に登録されている Web サーバ間でしか負荷分散をおこなうことができない。同じコンテンツを持つ Web サーバが外部ネットワークに存在している場合、これらの Web サーバ全てを 1 つの DNS サーバに同じ URL で登録すれば、同じコンテンツを持つ Web サーバ間で負荷分散が可能である。しかし、この手法は現実的ではない。したがって、このシステムは URL によって負荷分散できる範囲が限定されていると言える。さらに、Web サーバやネットワークの状況を考慮したサーバの選択ができない。

3.1.4 問題のまとめ

上記で説明した負荷分散技術の問題点は大きく分けて 2 つある。これらの問題点を図 1 を用いて説明する。前提として、Web サーバ 1-1、1-2、2-1 は同じコンテンツを持つ。

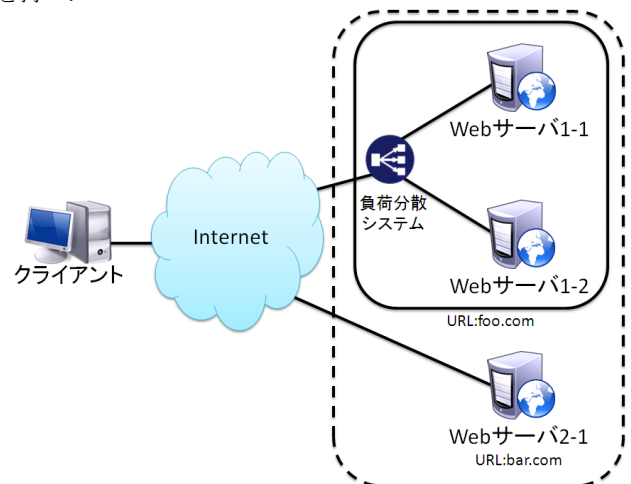


図 1 負荷分散システムの機能範囲

- URL によって負荷分散できる範囲が限定される
既存の負荷分散システムが URL:foo.com のネットワークで機能する場合、クライアントが URL:foo.com にアクセスする際の Web サーバの選択肢は Web サーバ 1-1、1-2 である。つまり、URL:bar.com に存在する Web サーバ 2-1 はこの選択肢に含まれない。

したがって、負荷分散できる範囲は実線で囲まれた領域内に限定される。

- Web サーバやネットワークの状況を考慮した Web サーバの選択ができない

既存の負荷分散システムが URL:foo.com のネットワークで機能する場合、クライアントが URL:foo.com にアクセスする際の Web サーバの選択肢は Web サーバ 1-1, 1-2 である。負荷分散システムはこれらの Web サーバにクライアントからのアクセスを割り振る。

しかし、負荷分散システムの種類によっては内部ネットワークにおける各 Web サーバや、負荷分散システムと各 Web サーバ間のネットワークの状況を考慮した Web サーバの選択ができない。

3.2 研究目的

本研究では 3.1 節で述べた問題点を解決した新しい負荷分散手法を提案することを目的とする。本節で、それぞれの問題点に対する解決策を述べる。詳しい技術概要については次章で述べる。

URL によって負荷分散できる範囲が限定されることなく、同じコンテンツを持つ Web サーバ間で負荷分散をするためには、コンテンツ名だけでコンテンツを取得できるコンテンツオリエンテッドネットワークが必要である。しかし、固定された場所間のユニキャストのロケーションオリエンテッドネットワークでは URL は必須の要素である。そこで、本研究では i3 というランデブーベース通信のオーバーレイネットワークを用いて、URL が必須でないコンテンツオリエンテッドネットワークを実現する。

i3 だけでは、Web サーバやネットワークの状況を考慮した Web サーバの選択ができない。そこで本研究では OpenFlow というプログラマブルスイッチを利用して Web サーバやネットワークの負荷を計測し、Web サーバを選択する際の判断基準に利用する。

提案手法では URL によって負荷分散できる範囲が限定されず、この範囲は図 1 に示す点線で囲まれた領域に広がる。さらに、Web サーバやネットワークの状況を考慮した Web サーバ選択と Web サーバに発生した障害の検知が可能である。

4. 提案手法

本研究の目的はコンテンツオリエンテッドネットワークにおける負荷を考慮した負荷分散手法の提案である。本章では、この研究目的を達成するための提案手法を述べる。提案手法を利用した負荷分散システムにおいて、負荷とは Web サーバ、OpenFlow スイッチ、OpenFlow スイッチに接続する回線の負荷を表す。

4.1 技術概要

本節では、システムの設計に利用した既存技術を提示する。コンテンツオリエンテッドネットワークを実現するためにオーバーレイネットワークのモデルである i3 を利用した。ネットワークの状況を監視して負荷を計測するために OpenFlow を利用した。以下でそれぞれの技術を説明する。

4.1.1 Internet Indirection Infrastructure(i3)

本項では、コンテンツオリエンテッドネットワークを実現するために利用した i3 について説明する。

提案手法は i3 で構成されたネットワークで機能する。i3 はランデブーベース通信のインフラストラクチャとして機能するオーバーレイネットワークであり、i3 サーバで構成

される。送信者は明示的に宛先へパケットを送信する代わりに、各送信データ(data)に論理的な ID(id)を関連付け、(id, data)としてオーバーレイネットワーク上に送信する。受信者 R はオーバーレイネットワークにトリガ(id, R)を送信することで ID(id)宛のデータを受信したい意思を表示する。本研究では ID を Web サーバが持つコンテンツ名から生成している。つまり、ID はコンテンツの ID と捉えることができる。本論文では以降、トリガをオーバーレイネットワークに送信することを、トリガを挿入すると表現する。

i3 サーバ

i3 サーバはトリガを格納し、i3 サーバ間およびエンドホストへ Internet Protocol(IP)を利用してパケットを転送する。

i3 のランデブーベース通信

i3 を使い、送信者 S と受信者 R が i3 サーバ I を介してデータを通信する例を示す。図 2 は S と R 間の通信を示しており、受信者 R は id に送信されたパケットを受信する意思があるとする。

1. 受信者 R は i3 サーバ I にトリガ(id, R)を挿入する
2. パケット(id, data)が送信されると、i3 サーバ I 上でパケットの ID とトリガの ID が一致する
3. パケットはトリガのアドレスである R に転送される

このランデブーベース通信は送信者と受信者を切り離す。送信者と受信者は互いの場所を意識する必要はない。

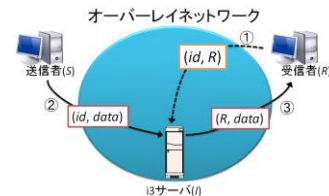


図 2 i3 のランデブーベース通信

i3 プロキシ

i3 プロキシは既存のアプリケーションのパケットと i3 パケットを相互変換し、アプリケーションのためにトリガを挿入・更新する。i3 プロキシによって Web クライアント等のアプリケーションは i3 ネットワークで動作できる。

i3 プロキシを導入した送信者 S と受信者 R が i3 サーバ I を介してデータを通信する例を示す。図 3 は S と R 間の通信を示しており、受信者 R は id に送信されたパケットを受信する意思があるとする。

1. 受信者 R は i3 サーバ I にトリガ(id, R)を挿入する
2. 送信者 S のアプリケーションがパケット(id, data)を送信しようとする、i3 プロキシは i3 サーバ I に ID が id であるトリガのアドレスを問い合わせる
3. i3 サーバ I は送信者 S に ID が id であるトリガのアドレス R を返す
4. 送信者 S の i3 プロキシはパケットを受信者 R に送信する

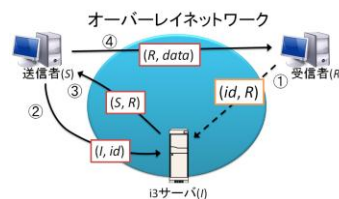


図 3 i3 プロキシを使用した i3 のランデブーベース通信

4.1.2 OpenFlow

本項では、主に負荷を計測するために利用した OpenFlow について説明する。

OpenFlow とは、OpenFlow スイッチングコンソーシアム [4] が提唱している、ソフトウェアによってネットワークの構成をおこない、制御する技術のことである。既存のネットワーク機器はネットワークの経路制御機能(C プレーン)とデータ転送機能(D プレーン)によって構成されている。OpenFlow の最大の特徴は、ネットワーク機器を構成する C プレーンと D プレーンが分離されていることである。OpenFlow は経路制御を担う OpenFlow コントローラと、データ転送を担う OpenFlow スイッチ、そしてコントローラとスイッチが通信するための OpenFlow プロトコルから構成される。

従来のネットワーク制御は、主に IP アドレスを利用した経路制御によっておこなわれてきたが、OpenFlow ではフロー単位で通信トラフィックを制御する。フローはヘッダフィールド、アクション、統計情報の 3 つの要素から構成される。

4.2 前提環境

本節では提案手法を実装し、動作検証をおこなったネットワークを説明する。このネットワークはここまでに説明した技術を使用して構築した。ネットワークの全体図を図 4 に示す。

すべての端末にはネットワークアドレスが異なる IP アドレスが設定されており、OpenFlow スイッチはルータとして動作している。Web サーバ 1、Web サーバ 2 とクライアントには i3 プロキシが導入されており、これを起動すると i3 サーバにトリガを挿入する。トリガは常に i3 サーバに挿入され続け、i3 サーバはその度に格納したトリガを更新している。Web サーバ 1、Web サーバ 2 は同じコンテンツを持つため、これらが挿入するトリガの ID は同じだと i3 コントローラに認識される。

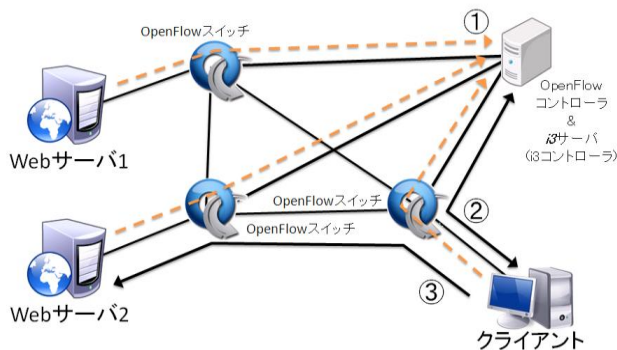


図 4 ネットワーク全体図

このネットワークで、クライアントが Web サーバにアクセスする手順を図 4 を使って説明する。

1. Web サーバ 1、Web サーバ 2 とクライアントは i3 コントローラにトリガを挿入する
2. クライアントがコンテンツを取得するためにアクセスする Web サーバの IP アドレスを i3 サーバから取得する。この際、取得した IP アドレスは Web サーバ 1 または Web サーバ 2 のものであり、これはランダムに選択される
3. クライアントは取得した IP アドレスの Web サーバにアクセスする

4.3 負荷の検出手法

本研究は負荷を考慮した Web サーバの負荷分散を目的の 1 つとする。負荷を考慮したサーバ選択を実現するためには、負荷を検出する必要がある。本節では前提環境における負荷の検出手法を述べる。

提案手法を利用した負荷分散システムにおいて、負荷とは Web サーバ、OpenFlow スイッチ、OpenFlow スイッチに接続する回線の負荷を表す。負荷を表す値は、一定時間に各 Web サーバに接続した OpenFlow スイッチを通過するパケットの数と定義する。このパケットは、Web サーバが処理したパケット、OpenFlow スイッチが処理したパケット、OpenFlow スイッチに接続する回線を通るパケットである。このため、このパケットの数を Web サーバ、OpenFlow スイッチ、OpenFlow スイッチに接続する回線の負荷の値として考えることは適切だと考えられる。ただし、提案手法は各 Web サーバが挿入・更新したトリガ数の割合を利用するため、このパケットの数にトリガのパケットの数は含めない。OpenFlow コントローラはトリガパケットとデータパケットの使用ポート番号が異なることを利用してこれらを区別する。OpenFlow コントローラは統計情報として負荷を 15 秒ごとに計測しており、Web サーバを選択する際の判断基準にする。負荷を計測する時間間隔は、クライアントがアクセスする Web サーバを切りかえる際に要する時間を考慮した値である。本論文では以降、この時間を遅延時間と表記する。

4.4 負荷分散手法

本節では、コンテンツオリエンテッドネットワークを利用して負荷を考慮した負荷分散手法を 2 つ提案する。

4.4.1 仮想障害法

Web サーバが i3 サーバに挿入・更新するパケットを OpenFlow スイッチ上で破棄することで仮想的に Web サーバに障害を起こし、クライアントがアクセスする Web サーバを動的に切りかえる手法である。

各 OpenFlow スイッチが計測した負荷に 10 倍以上の差が生じていた場合、負荷が最も高い値を示す OpenFlow スイッチはトリガを破棄するように動作する。トリガを破棄する判断基準である負荷の差の条件は、クライアントが Web サーバにアクセスしていない状態で誤って条件を満たしてしまわないように考慮した値である。

OpenFlow スイッチ上でトリガを破棄することにより、負荷が最も高いサーバが挿入・更新したトリガは i3 サーバに転送されず、トリガの挿入・更新処理は停止される。この手法で、i3 サーバ内に格納されている負荷の高い方の Web サーバのトリガを消滅させる。i3 サーバからは負荷の高い方の Web サーバに障害が発生したように見え、i3 サーバには負荷の低い方の Web サーバのトリガだけが格納される。その後、クライアントの i3 プロキシ内も負荷の低い方の Web サーバの ID だけが格納される。この手法により、クライアントは負荷の低い方の Web サーバにアクセスする。結果として、Web サーバ及びネットワークの負荷を考慮してクライアントがアクセスする Web サーバを選択できる。

本手法を施した環境で、クライアントがアクセスする Web サーバを変更する手順を図 5 を使って説明する。前提として、4.2 節で説明した手順は完了しており、

クライアントは Web サーバ 2 にアクセスし続けているとする。

1. i3 コントローラは Web サーバ 1 より Web サーバ 2 に負荷がかかっていることを検知する
2. i3 コントローラは Web サーバ 2 に接続している OpenFlow スイッチをトリガを破棄するように動作させる
3. i3 コントローラ内の Web サーバ 2 が挿入したトリガが消滅する
4. クライアントがコンテンツを取得するためにアクセスする Web サーバの IP アドレスを i3 サーバから取得する。この際、必ず Web サーバ 1 の IP アドレスを取得する
5. クライアントは Web サーバ 1 にアクセスする

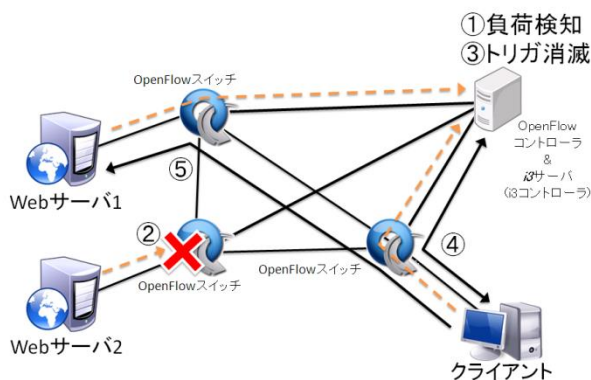


図5 仮想障害法

4.4.2 フロースイッチ法

OpenFlow によってクライアントが Web サーバにアクセスする際のパケットの IP アドレスを、同じコンテンツを持つ別の Web サーバの IP アドレスに変更する手法である。この IP アドレスの変更は確率によって制御される。

OpenFlow コントローラは各 Web サーバに接続した OpenFlow スイッチで計測した負荷を元に、トリガを破棄する確率を決定する。OpenFlow スイッチ s がトリガを破棄する確率 $P_{TriggerDrop}(s)$ は以下の式で求められる。

$$P_{TriggerDrop}(s) = \frac{Load_s}{Load_1 + Load_2 + \dots + Load_s + \dots + Load_m} \quad (1)$$

ここで、 $Load_s$ は OpenFlow スイッチ s が計測した負荷、 m は Web サーバに接続された OpenFlow スイッチの数をあらわす。負荷を考慮してトリガを破棄することで、OpenFlow スイッチと i3 サーバ間のネットワークのトリガによる負荷を軽減することができる。

これに加えて、Web サーバが挿入・更新したトリガの数も一定時間ごとに計測する。ただし、OpenFlow スイッチで破棄したトリガは i3 サーバに転送されないため、数に含めない。計測したトリガの数を元に、クライアントが Web サーバにアクセスする際の宛先 IP アドレスを変更する確率を決定する。この確率はトリガをより多く送信している Web サーバによりアクセスが集中するように決定する。クライアントが Web サーバにアクセスする際の宛先 IP アドレスを Web サーバ w の IP アドレスに変更する確率 $P_{Forward}(w)$ は以下の式で求められる。

$$P_{Forward}(w) = \frac{Trigger_w}{Trigger_1 + Trigger_2 + \dots + Trigger_w + \dots + Trigger_n} \quad (2)$$

ここで、 $Trigger_w$ は Web サーバ w が i3 サーバに挿入・更新したトリガ数、 n は Web サーバの数をあらわす。トリガの数をこの宛先 IP アドレスの変更確率に反映させること

で、障害が発生した Web サーバにクライアントからのアクセスが転送されないようにしている。この手法は、障害が発生した Web サーバはトリガを送信できないことを利用している。

i3 サーバに挿入・更新されるトリガの数は負荷の高さに反比例するので、結果として、クライアントの Web サーバへのアクセスは負荷が低い方の Web サーバに集中する。

本手法を施した環境で、クライアントがアクセスする Web サーバを変更する場合の手順を図 6 を使って説明する。図 6 では式(1)において $m = 2$ 、式(2)において $n = 2$ である。前提として、4.2 節で説明した手順は完了しており、クライアントは Web サーバ 2 にアクセスし続けているとする。

1. i3 コントローラは式(1)を利用して、Web サーバ 1 と Web サーバ 2 のトリガを破棄する確率を求める。この場合、Web サーバ 1 より Web サーバ 2 の方がこの確率は高く、トリガは破棄されやすい。結果として Web サーバ 2 より Web サーバ 1 の方が i3 サーバに挿入・更新するトリガの数が多くなる
2. i3 コントローラは式(2)を利用して、クライアントが Web サーバにアクセスする際の宛先 IP アドレスを変更する確率を求める。この場合、宛先 IP アドレスが Web サーバ 2 に変更される確率より Web サーバ 1 に変更される確率の方が高い
3. クライアントが Web サーバにアクセスする際の宛先 IP アドレスが Web サーバ 1 の IP アドレスに変更される
4. クライアントは Web サーバ 1 にアクセスする

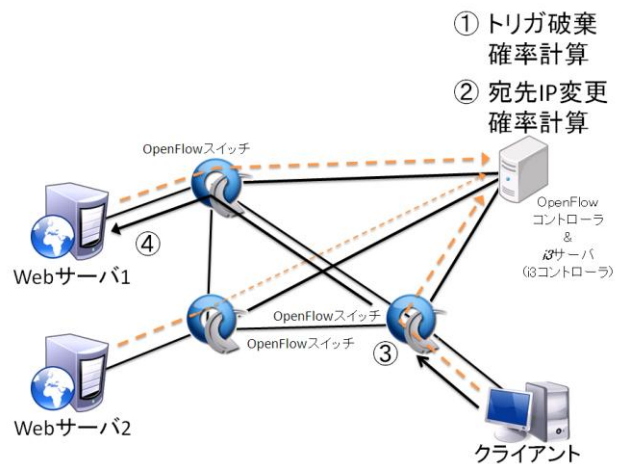


図6 フロースイッチ法

5. 実験

本章では、4 章で述べた提案手法の負荷分散の効率を評価するために、提案手法なし、仮想障害法、フロースイッチ法のそれぞれの場合について実験をおこなった。小規模な i3 ネットワークにおいて提案手法を施さなかった場合と、それぞれの提案手法を施した場合の Web サーバの選択による負荷分散比率を比較する。

5.1 実験概要

本節では、実験をおこなったネットワークの構成と実験手順を説明する。実験環境は図 7 に示すように、クライアント 1 台、Web サーバ 2 台、i3 サーバと OpenFlow コントローラを搭載した端末 1 台、OpenFlow スイッチ 3 台を使用した。

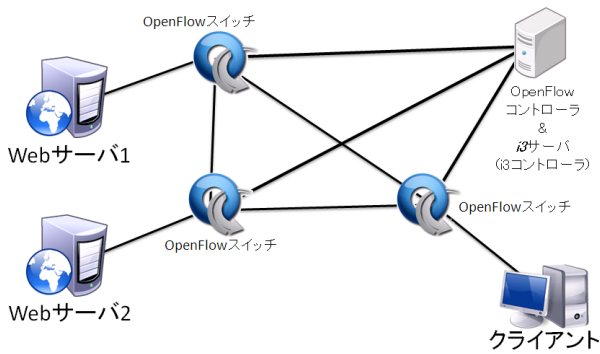


図7 実験環境

5.1.1 実験手順

本項では実験をおこなう手順を説明する。

実験では、Webサーバの性能、コンテンツサイズは同じ(8バイト)である。したがってクライアントから各Webサーバへのアクセスが均等に割り振られ、分散されることが理想である。

i3サーバにそれぞれのWebサーバとクライアントのトリガが格納され、クライアントからWebサーバへの通信ができることを確認してから実験をおこなった。

それぞれの提案手法で、クライアントはWebサーバ1とWebサーバ2から1時間の間wgetコマンド[5]を使ってコンテンツを取得し続け、各Webサーバへのアクセス回数を計測した。この実験を10回繰り返し、クライアントの各Webサーバへの平均アクセス回数を求めた。実験結果と考察を5.2節で述べる。

5.2 実験結果

本節では、5.1.1項で述べた実験手順に従って提案手法なし、仮想障害法、フロースイッチ法のそれぞれの場合について実験をおこなった結果をグラフで示す。

5.2.1 提案手法なし

提案手法の評価の基準として提案手法を施していない状態で実験をおこなった。結果のグラフを図8に示す。

図8より、各Webサーバへの平均アクセス回数に大きく差が生じていることが分かる。したがって、クライアントからのアクセスを各Webサーバに均等に割り振れておらず、効率的に負荷分散できていないと言えない。

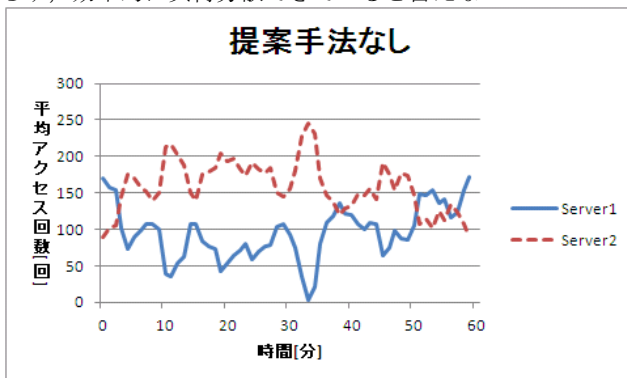


図8 提案手法なし

5.2.2 仮想障害法

仮想障害法を施した状態で実験をおこなった。結果のグラフを図9に示す。

各サーバへの平均アクセス回数に大きく差が生じている時間があるが、総合的に見ればアクセス回数が均等に割り振れていることが分かる。さらに、クライアントからサーバへのアクセスが切りかわるタイミングに規則性が見られる。したがって、トリガの破棄によって、提案手法を施していない場合より理想的な負荷分散が可能であることが示された。

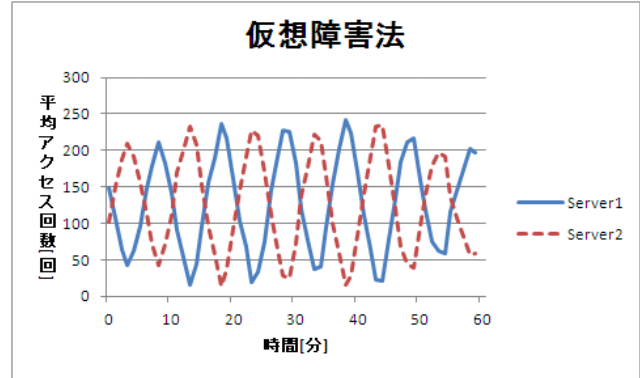


図9 仮想障害法

5.2.3 フロースイッチ法

フロースイッチ法を施した状態で実験をおこなった。式(1)において $m=2$ 、式(2)において $n=2$ である。結果のグラフを図10に示す。

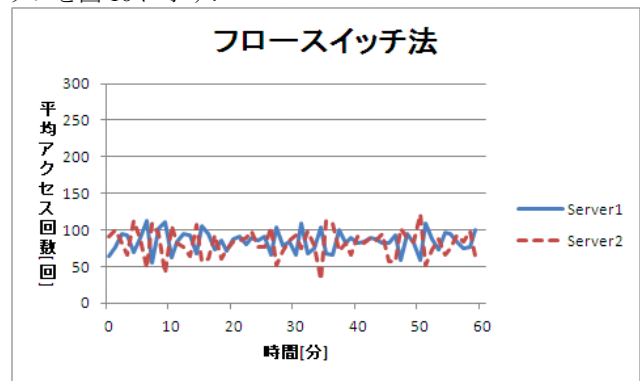


図10 フロースイッチ法

宛先IPアドレスの書き換えによって、頻繁にクライアントのWebサーバへのアクセスが切りかわる。このため、ある時間での各Webサーバへの平均アクセス回数がほぼ等しいことがグラフから分かる。よって、前述の2つの手法よりも理想的に負荷分散ができています。

6. 評価

本章では、本研究で提案する負荷分散手法の機能の評価をする。

評価はシステムの基本機能と分散方式の機能の2種類でおこなう。評価の比較対象は2章で挙げた負荷分散技術の中から選択した。しかし、ユーザにアクセスするWebサーバを選択させる手法は手動であるため選択しなかった。

6.1 システム基本機能の評価

本節では、負荷分散システムが持つ基本機能の評価をする。評価の比較対象は負荷分散技術として広く利用されているDNSラウンドロビンとロードバランサ、キャッシュサーバを選択した。評価項目は以下の3つである。

- 障害対応機能

Web サーバが複数ある状態で、一部の Web サーバに障害が発生した際にもサービスを継続できる機能

- 負荷考慮機能
Web サーバの負荷を考慮して Web サーバを選択できる機能
- 多地点協調機能
URL によってアクセスする Web サーバが限定されない機能

これらの項目がそれぞれの負荷分散システムで可能かどうかを評価した。評価結果を表 1 に示す。

表 1 システム基本機能の評価

| | 障害対応 | 負荷考慮 | 多地点協調 |
|-------------|------|------|-------|
| DNS ラウンドロビン | × | × | × |
| ロードバランサ | ○ | ○ | × |
| キャッシュサーバ | × | × | × |
| 仮想障害法 | ○ | ○ | ○ |
| フロースイッチ法 | ○ | ○ | ○ |

○:可能 ×:不可能

それぞれの評価について説明する。

障害対応機能

DNS サーバとキャッシュサーバには Web サーバの障害を検知する仕組みがないので障害対応が不可能だと評価した。

ロードバランサには Web サーバの障害を検知するモニタ機能が備わっている。仮想障害法とフロースイッチ法は i3 サーバにある Web サーバのトリガが格納されていなければ、この Web サーバに障害が発生していると認識する。そして、この Web サーバは Web サーバを選択する際の選択肢から外れる。よって、これらのシステムは障害対応が可能だと評価した。

負荷考慮機能

DNS ラウンドロビンは 3.1.3 項で述べたように、クライアントからの問い合わせがあるたびに順番に登録された IP アドレスを答えるだけである。ロードバランサは、負荷分散の方式によっては Web サーバの負荷を考慮した選択ができる。キャッシュサーバは Web サーバの代わりであるため、Web サーバの負荷を考慮してクライアントからのアクセスを転送するわけではなく、クライアントの代わりに Web サーバにアクセスするだけである。よって、これらのシステムは負荷考慮が不可能だと評価した。

仮想障害法とフロースイッチ法は OpenFlow を利用して Web サーバの負荷を計測し、Web サーバを選択する際の評価基準に利用している。よって、これらのシステムは負荷考慮が可能だと評価した。

多地点協調機能

DNS ラウンドロビンは 3.1.3 項で述べたように、実現が困難であると考えられる。ロードバランサは、2.3 節で述べたように、下位にある Web サーバ群を Web サーバ選択の対象とする。このため、多地点協調が不可能だと評価した。キャッシュサーバは URL ごとに Web サーバのコンテンツをキャッシュしている。ユー

ザの取得したいコンテンツが同じコンテンツであっても URL が異なれば、キャッシュサーバは別のコンテンツだと認識する。このコンテンツがキャッシュされていてもキャッシュサーバは機能しない。よって、これらのシステムは多地点協調が不可能だと評価した。

仮想障害法とフロースイッチ法はコンテンツオリエンテッドネットワークで動作しているため、Web サーバを選択する際の選択肢が URL によって限定されない。よって、これらのシステムは多地点協調が可能だと評価した。

表 1 より、提案手法の優位性が示された。

6.2 分散方式の評価

本節では、負荷分散システムが利用する分散方式の機能の評価をする。評価の比較対象は DNS ラウンドロビンとロードバランサで利用されている分散方式を選択した。キャッシュサーバには分散方式がないので選択しなかった。一方、ロードバランサには様々な分散方式が存在するので、代表的な 4 種類の方式を選択した。これらの方式を以下に示す。

Round Robin

クライアントのアクセスをサーバに均等に転送する。

Least Connections

一定時間の接続数が最も小さいサーバに転送する。

Fastest

1 番リクエストの応答が速いサーバに転送する。

Dynamic Ratio

CPU やメモリ使用率が低いサーバに転送する。

DNS ラウンドロビンの分散方式は上記の Round Robin に該当すると考える。よって、上記 4 種類の分散方式の評価の比較対象とした。

評価項目は以下の 3 つである。

- Web サーバの状況考慮
対象の分散方式が Web サーバの状況を考慮できるか
- ネットワークの状況考慮
対象の分散方式がネットワークの状況を考慮できるか。
- 分散比率の調節
負荷分散比率を静的に決定するか、動的に決定するか

これらの項目についてそれぞれの分散方式の機能の評価した。評価結果を表 2 に示す。

それぞれの評価について説明する。

表 2 分散方式の評価

| | Web サーバ の状況考慮 | ネットワーク の状況考慮 | 分散比率 の調節 |
|-------------------|------------------|-----------------|-------------|
| Round Robin | × | × | 静的 |
| Least Connections | ○ | × | 動的 |
| Fastest | ○ | ○ | 動的 |
| Dynamic Ratio | ○ | × | 動的 |
| 仮想障害法 | ○ | ○ | 動的 |
| フロースイッチ法 | ○ | ○ | 動的 |

○:可能 ×:不可能

Web サーバの状況考慮

Round Robin はクライアントからのアクセスを順番に各 Web サーバに転送するだけである。よって、Web サーバの状況考慮が不可能だと評価した。

Least Connections はクライアントの Web サーバへの接続数を Web サーバの負荷と捉えている。Fastest は Web サーバへのリクエストの応答速度を負荷と捉えている。この応答速度は Web サーバにかかっている負荷の影響を受ける。Dynamic Ratio は Web サーバの CPU やメモリ使用率を負荷と捉えている。仮想障害法とフロースイッチ法は Web サーバに接続する OpenFlow スイッチを通過するデータパケット数を負荷と捉えている。この負荷は Web サーバが処理したパケット数である。これらの方式は Web サーバの負荷を考慮してクライアントからのアクセスを Web サーバに転送するため、Web サーバの状況考慮が可能だと評価した。

ネットワークの状況考慮

Round Robin はクライアントからのアクセスを順番に各 Web サーバに転送するだけである。Least Connections はクライアントの Web サーバへの接続数をサーバ選択の評価基準にしている。Dynamic Ratio は Web サーバの CPU やメモリ使用率をサーバ選択の評価基準にしている。よって、これらの方式はネットワークの状況考慮は不可能だと評価した。Fastest は Web サーバへのリクエストの応答速度を負荷と捉えている。この応答速度はロードバランサと Web サーバ間のネットワークの負荷の影響を受ける。仮想障害法とフロースイッチ法は Web サーバに接続する OpenFlow スイッチを通過するデータパケット数を負荷と捉えている。この負荷は OpenFlow スイッチが設置されているネットワークを通過したパケット数である。これらの方式はネットワークの負荷を考慮してクライアントからのアクセスを Web サーバに転送するため、ネットワークの状況考慮が可能だと評価した。

分散比率の調節

Round Robin はクライアントからのアクセスを各 Web サーバに均等に転送する。よって、分散比率は静的だと評価した。Least Connections は一定時間のクライアントの Web サーバへの接続数を分散比率に反映させている。Fastest は Web サーバへのリクエストの応答速度を分散比率に反映させている。Dynamic Ratio は一定時間の Web サーバの CPU やメモリ使用率を分散比率に反映させている。よって、分散比率は動的だと評価した。仮想障害法とフロースイッチ法は Web サーバに接続する OpenFlow スイッチを通過するデータパケット数を分散比率に反映させている。よって、これらの方式の分散比率は動的だと評価した。

表 2 より、提案手法は既存の分散方式以上の機能を持つことが示された。

7. 考察と今後の課題

本章では、5 章の実験結果の考察と今後の課題について述べる。

7.1 実験の考察

実験によって OpenFlow によってトリガを操作することで、i3 ネットワークでの理想的な負荷分散が可能であることを示した。実験結果より、仮想障害よりもフロースイッチの方が負荷分散の効率は良い。

しかし、いずれの手法もクライアントの Web サーバへのアクセスを切りかえる際に遅延時間が発生するという問題がある。フロースイッチはクライアントからのアクセスを頻繁にふり分けるため、これにともなって遅延時間の発生が頻繁に起こる。このため、一定時間内の平均アクセス回数が仮想障害よりも少ないという問題がある。さらに、フロースイッチは OpenFlow コントローラに変更する IP アドレスを予め設定しておく必要があるため、仮想障害よりも拡張性が低い。このように、仮想障害とフロースイッチはトレードオフな関係にある。

7.2 システム基本機能の課題

2.3 節で述べたように、負荷分散を考える上で一連の処理をおこなうために同じ Web サーバと複数回通信する必要がある場合は、同じクライアントからの通信を常に同じ Web サーバに転送する機能が必要である。この機能はセッション維持機能と呼ばれる。

提案手法では、セッション維持機能がない。これは、送信元 IP を用いて経路制御をおこなうソースアドレスルーティングを利用することで実現できると考えられる。

7.3 ネットワーク規模の課題

本研究での実装環境は非常に小さなネットワークで構成されている。実際の運用を考慮すると、ネットワークの規模を拡大し、広範囲な負荷分散が可能な分散協調 OpenFlow ネットワークの実現可能性を実験で証明する必要がある。

8. おわりに

拡張性と可用性を考慮した Web サーバの運用管理の重要性にともなって、負荷分散の必要性が増加している。本研究ではコンテンツオリエンテッドネットワークを利用した負荷分散の手法を提案した。コンテンツオリエンテッドネットワークを利用することで既存の負荷分散技術の問題点を解決できる。提案手法を実装し、OpenFlow によってフローを制御することで効率的なサーバ選択による負荷分散が可能であることを示した。

参考文献

- [1] Internet Indirection Infrastructure
<http://i3.cs.berkeley.edu/>
- [2] I.Stoica, D.Adkins, S.Zhuang, S.Shenker, S.Surana, "Internet indirection in-frastructure", In ACM SIGCOMM Computer Communication Review (Vol.32, No. 4, pp. 73-86), ACM, (2002, August).
- [3] N.McKeown, T.Anderson, H.Balakrishnan, G.Parulkar, L.Peterson, J.Rexford, J.Turner, "OpenFlow: enabling innovation in campus net-works", ACM SIGCOMM Computer Communication Review, 38(2), 69-74,(2008)
- [4] OpenFlow スイッチング コンソーシアム
<http://www.openflow.org/>
- [5] GNU Wget
<http://www.gnu.org/software/wget/wget.html>