

平成20年度 卒業研究論文

コンテンツキャッシュプロキシの
動的負荷分散手法

指導教員

松尾 啓志 教授

津邑 公暁 准教授

名古屋工業大学 情報工学科

平成17年度入学 17115021 番

伊藤 雄一郎

目次

第1章	はじめに	1
第2章	コンテンツ配信システム	3
2.1	コンテンツ配信システムの形態	3
2.1.1	クライアント・サーバーモデル	3
2.1.2	P2P (Peer to Peer)	5
2.1.3	キャッシュ・プロキシサーバー	5
2.2	関連研究	7
第3章	既存手法	9
3.1	概要～キャッシュプロキシを利用した負荷分散～	9
3.2	従来手法詳細	10
3.2.1	キャッシュプロキシによるオーバーレイネットワーク	10
3.2.2	コンテンツの複製転送・探索	13
3.2.3	従来手法の問題点	18
第4章	提案手法	19
4.1	概要～送り手、受け手主導による負荷分散～	19
4.2	提案手法詳細	21
4.2.1	キャッシュプロキシによるオーバーレイネットワークの構成	21
4.2.2	コンテンツの複製・探索	21
4.2.3	負荷分散手法	22

第5章 実験と評価	25
5.1 実験1：高負荷時における負荷分散	25
5.2 結果	27
5.3 実験2：クライアントからのリクエスト数の変化	29
5.4 結果	31
5.5 考察	32
第6章 まとめ	33
謝辞	34
参考文献	35

第1章

はじめに

近年、インターネット、ブロードバンドの普及とともに、Web 利用者が増加している。それに応じて、大手サイトからのリンク等によって、クライアントから、通常では想定されていないコンテンツの大量のリクエストが Web サーバーに集中する、フラッシュクラウドと呼ばれる現象が起こる。フラッシュクラウドは、特定の Web サーバーに、通常の 10 倍から 100 倍の負荷を及ぼし、一時的に応答不能になる。この結果として、サーバーはクライアントによって、リクエストされたコンテンツを提供できないという事態に陥ってしまう。このため、フラッシュクラウド時におけるサーバーの負荷を減らし、クライアントにサービスを提供できるようにしなければならない。既存手法として、キャッシュプロキシがオーバーレイネットワークを構成し、多くのクライアントによって要求されるような人気コンテンツの複製をオーバーレイネットワークに配置する。このオーバーレイネットワークがクライアントからのコンテンツ要求に応答することによって、サーバーにかかる負荷を分散する手法がある。しかし、今度は、キャッシュプロキシオーバーレイネットワーク内での負荷の分散を考えなければならない。複数のサーバーの代理をキャッシュプロキシが行うような環境下では、各キャッシュプロキシの負荷状態は偏りが生じており、各キャッシュプロキシの負荷状態を考慮した負荷分散を行う必要がある。そこで、本論文では各キャッシュプロキシの負荷状態を考慮した負荷分散手法として、高負荷のキャッシュプロキシが自発的に、低負荷なキャッシュプロキシに対して負荷を配置する送り手主導方式、また、低負荷なキャッシュプロキシが自発的に、高負荷なキャッシュプロキシから負荷を貰い受ける

受け手主導方式を提案し、キャッシュプロキシオーバーレイネットワークにクライアントからの要求コンテンツを保持しているサーバーとは無関係の外部サーバーによる負荷を配置し、負荷状態に応じた、送り手主導方式、受け手主導方式の負荷分散法の有用性を評価する。

本論文の以降の構成を示す。第2章では、コンテンツ配信システム、関連研究について説明し、第3章では、フラッシュクラウド時、つまり、高負荷時における負荷分散手法の既存手法を説明する。第4章では、提案手法である、キャッシュプロキシの負荷状態を考慮した負荷分散手法を紹介し、第5章で、シミュレーション、評価を行う。

第2章

コンテンツ配信システム

クライアント（ユーザー）は、インターネットを利用して、情報やソフトウェアなど様々な Web コンテンツを受け取っている。インターネット上には、必ずコンテンツを提供してくれる相手が存在し、この相手にアクセスし、コンテンツを提供してくれるよう要求する。この相手は、要求されたクライアントに対して要求されたコンテンツを提供する。このようにして、クライアントはコンテンツを受け取ることができる。コンテンツ配信に関して、種々のコンテンツ配信システムがある。この章では、コンテンツ配信に関して基本システムであるクライアント・サーバーモデル、Peer to Peer システムについて説明する。また、コンテンツ配信の手助けになるキャッシュプロキシ、コンテンツ配信システムにおける負荷分散手法の関連研究を紹介する。

2.1 コンテンツ配信システムの形態

2.1.1 クライアント・サーバーモデル

現在、インターネット上のサービスモデルの主流になっているのが、クライアント・サーバーモデルである。（図 2.1）サービスやコンテンツを提供するサーバーと、サービス、コンテンツを要求するクライアントから構成されている。コンテンツを欲しいクライアントは、サーバーにコンテンツ要求を送り出して、コンテンツが自分に提供されるのを待つ。サーバーはコンテンツ要求を受け取ると、クライアントに対してコンテンツ要求に応答し、コンテンツを転送する。通常、サーバーはクライアントに対

して、「1対多」の関係にある。サーバーは複数のクライアントからのコンテンツ要求に応答しなければならず、サーバーにかかる負荷は大きくなる。また、現在インターネットの利用者の増加、ブロードバンドの普及による利用時間の増加により、クライアントからサーバーへの処理要求が増加している。このために、サーバーの負荷が増大し、クライアントはコンテンツを受け取るまでの時間が増大したり、コンテンツを受け取れなかったりすることがある。

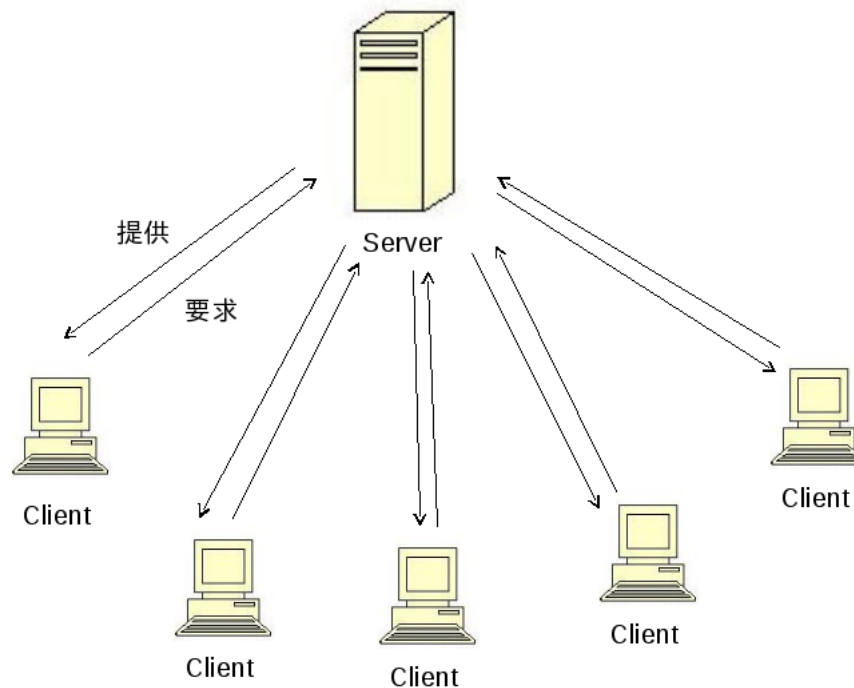


図 2.1: クライアントサーバーモデル

2.1.2 P2P (Peer to Peer)

図 2.2 に Peer to Peer の概略を示す。P2P ネットワークは、「各ノードが対等な立場で振る舞うネットワーク」であり、各ノードはサーバーとしてもクライアントとしても機能する。通常、P2P はネットワークを構成するため、P2P ノードというのは、サーバーのような大きな資源を持つ計算資源ではなく、小さい計算機資源によって構成される。クライアント・サーバーモデルにおいてサーバーなりうる計算資源数よりも、P2P ネットワークにおけるノードに成りうる計算資源数の方が圧倒的に多い。したがって、容易にこの資源を利用することができる。クライアント・サーバーモデルとの決定的な違いは、コンテンツを要求するノードと、応答するノード数が「多対1」であるのに対して、P2P ネットワークにおいては、コンテンツ要求するノードと、応答するノード数が「多対多」である。したがって、クライアントがコンテンツを要求するジョブがある単一ノードに集中せず、クライアント・サーバーモデルのような、特定のノードに負荷が集中することがない。

また、クライアント・サーバーモデルのように、クライアントやコンテンツ提供数の増加に対応するためのサーバーの補強・増設をする必要がないため、ノード数やサービス提供数の増加に対するスケーラビリティが高い。さらに、クライアント・サーバーモデルでは、サーバーもしくはその周辺のネットワークに障害が発生した場合、そのサーバーがコンテンツを提供するというサービス自体が停止してしまい、クライアントはコンテンツを受け取ることができなくなってしまう。一方で、P2P ネットワークでは、特定のノードや、ネットワークに障害が発生したとしても、別のノードや、ネットワーク経路によって、サービスを維持できるため、サービス全体としての耐障害性が向上する。

2.1.3 キャッシュ・プロキシサーバー

プロキシサーバーとは、企業などの内部ネットワークとインターネットの境にあって、直接インターネットに接続できない内部ネットワークのコンピューターの代理として、インターネット接続を行う。ネットワークに出入りするアクセスを管理し、内

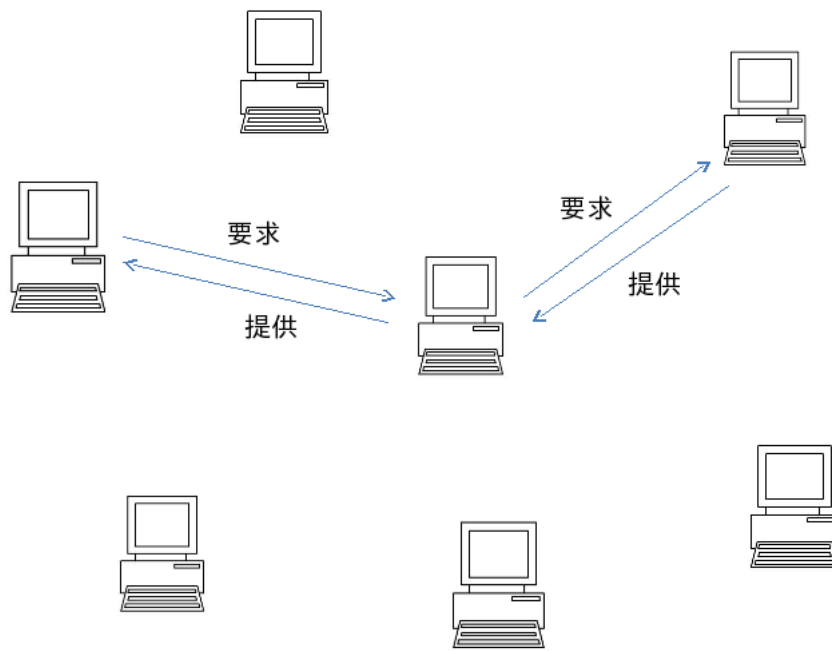


图 2.2: Peer to Peer

部からの特定の接続のみを許可したり、外部からの不正なアクセスを遮断するために用いられる。また、キャッシュサーバーとは、インターネット上で提供されている Web コンテンツの複製を蓄積することができ(キャッシュ)、クライアントから Web コンテンツの要求があった場合、本来のサーバーに代わり、コンテンツ要求に応じて、コンテンツを配信する。このことにより、ネットワークのトラフィックを減らす、サーバーの負荷の分散ができるといった利点がある。通常、プロキシサーバーはキャッシュサーバーの役割を兼ねていることが多い。負荷分散の役割として、キャッシュ・プロキシサーバーを利用するため、ここでは、主に、コンテンツの複製を配信する役割に着目する。(以後キャッシュプロキシ)

2.2 関連研究

本節では、フラッシュクラウド、負荷分散に関する関連研究を説明する。

フラッシュクラウド、サーバーの負荷の分散の手法として、Chenyu Pan, Merdan Atajanov らはキャッシュプロキシで構成されるキャッシュプロキシオーバーレイネットワークによってサーバーの負荷を分散させる手法, FCAN を提案している [1]。この手法では、サーバーが DNS を管理しており、クライアントからのサーバへのコンテンツ要求が上昇してきた際、DNS はクライアントからの名前解決において、サーバーの IP アドレスを返す代わりに、オーバーレイネットワークを構成するキャッシュプロキシの IP アドレスを返す。この手法によって、高負荷時にサーバーの負荷をキャッシュプロキシに分散している。しかし、この手法では、キャッシュプロキシはある特定のサーバーの代理の仕事をするを想定されており、キャッシュプロキシの負荷状態は考慮されていない。したがって、キャッシュプロキシが、複数の代理サーバーとしても機能する場合、キャッシュプロキシの負荷状態を考慮した上で、負荷分散を行う必要がある。

また、Content Delivery Network(CDN) についても研究されている。CDN では、コンテンツの複製を提供する複数のサーバーを地理的に分散し、クライアントのアクセス数を複数のサーバーに振り分ける。この手法では、クライアントのアクセスが特定の

サーバーに集中しないようにし、計算機や通信回線の性能の限界によってクライアントの待ち時間が増大することを防ぐ。浅原らによって Peer-to-Peer 技術を用いた CDN である ExaPeer を提案している [4]。ここでは、各コンテンツについて、どの地域のクライアントからアクセスが多いのかを動的に検出し、どの地域から多くアクセスが発生しているのかを把握する。得られた各コンテンツに対する需要分布に応じて、コンテンツの複製を動的に再配置する手法も挙げられる。この手法は、サーバーのリソースの増加によるコストが増加する。本論文では、負荷分散手法として FCAN についての改良を提案する。

第3章

既存手法

3.1 概要～キャッシュプロキシを利用した負荷分散～

本章では、キャッシュプロキシによるオーバーレイネットワークを用いた負荷分散手法FCAN [1]。この既存手法について説明する。この手法では、コンテンツ要求、応答には、クライアントサーバーモデルが想定されている。サーバーは、複数のコンテンツを保持しており、そのうち一部のコンテンツに対して、クライアントからの要求が急激に増加することを想定し、その際の、サーバーに負荷が集中することを防ぐものである。負荷平常時、コンテンツの要求、応答には、クライアント・サーバーモデルによりサーバーがクライアントからの要求に回答している。サーバは、クライアントからコンテンツ要求数（負荷数）を監視しており、クライアントからの要求が多くなりサーバーの負荷が上昇し、ある一定の閾値数を負荷が上回った場合、サーバーとクライアントの中間に位置するキャッシュプロキシ同士でP2Pオーバーレイネットワークを構成させ、サーバーのもつコンテンツをそのネットワークに保持させる。すると、そのキャッシュプロキシネットワークはクライアントからのコンテンツ要求に回答できるようになり、サーバーのコンテンツ応答負荷をキャッシュプロキシネットワークに分散することができるようになる。

3.2 従来手法詳細

3.2.1 キャッシュプロキシによるオーバーレイネットワーク

既存手法において、クライアントが Web コンテンツを利用したい場合、クライアントからサーバーにコンテンツ要求が出される。このクライアントからのコンテンツ要求の負荷分散として、キャッシュプロキシが利用される。キャッシュプロキシは、コンテンツの複製を保持し、サーバーの代わりにコンテンツを提供することにより、サーバーの負荷を減少させる。また、既に社会にはキャッシュプロキシは多く存在しており、このリソースを利用することにより新設コストを減らすことができる。

負荷が平常時、クライアント・サーバーモデルによって、サーバーはクライアントからのコンテンツ要求を受ける。クライアントからのコンテンツ要求が増加すると、サーバーの負荷は増大してしまう。サーバーは、クライアントからのコンテンツ要求数を監視しており、負荷が一定の閾値以上の要求数に達すると、複数のキャッシュプロキシに対して、P2P オーバーレイネットワークを構成するよう指示を出す。キャッシュプロキシは、クライアントとサーバーの間に位置し、クライアントとサーバーの間に、P2P オーバーレイネットワークを構成する。この構造を、「対フラッシュクラウド構造」と定義する(図 3.1)。これ以降、クライアントから送出されるコンテンツ要求は、DNS(Domain Name Server)によりキャッシュプロキシに割り振られる。DNSは、サーバーにより管理されており、平常時にはクライアントに対して、サーバーの IP アドレスを返す。しかし、クライアントからのコンテンツ要求が増加し、P2P オーバーレイネットワークを構成した場合には、サーバーの IP アドレスを返す代わりに、オーバーレイネットワークを構成するキャッシュプロキシの IP アドレスを返す。この時、各キャッシュプロキシに均等に要求が割り振られるよう、ラウンドロビン等により IP アドレスを返す。この P2P オーバーレイネットワークが、サーバーの保持するコンテンツの複製を保持することにより、クライアントからのコンテンツ要求は、サーバーの代わりに、P2P オーバーレイネットワークのキャッシュプロキシが応答することになる。このことによって、クライアントから送出されるコンテンツ要求の負荷は、サーバから P2P オーバーレイネットワークのキャッシュプロキシに分散されることに

なる。クライアントからのコンテンツ要求が少なくなり、ある一定の閾値以下になった場合、サーバーはP2P オーバーレイネットワークは必要ないと判断し、通常のクライアント・サーバーモデルを構成し直す。

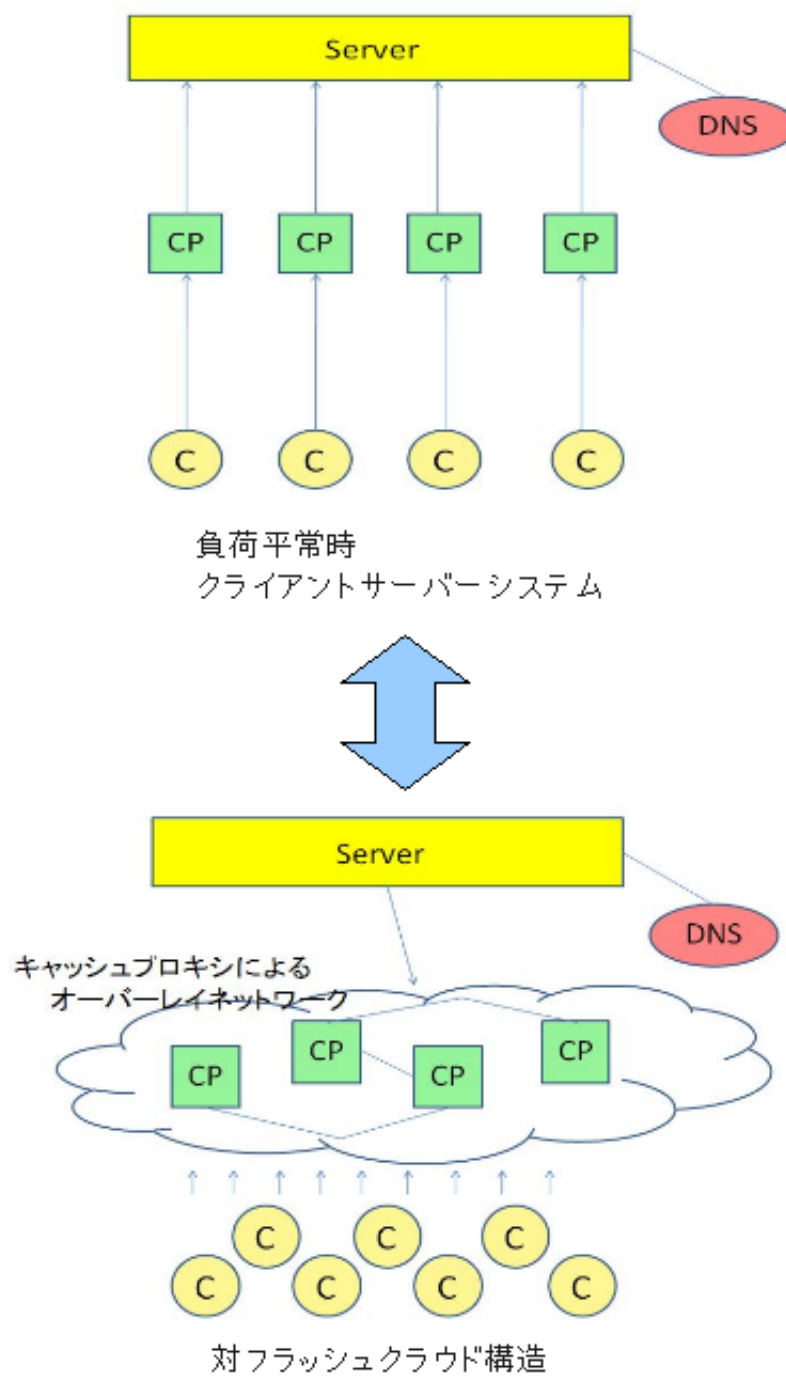


図 3.1: 状態遷移

3.2.2 コンテンツの複製転送・探索

前節では、クライアントからのコンテンツ要求がサーバーに集中するのを防ぐ方法として、オーバーレイネットワークを構成するキャッシュプロキシが代わりに応答することによりサーバーの負荷を分散していることを説明した。しかし、キャッシュプロキシがサーバーの代わりにするには、クライアントが要求するコンテンツを提供できなければならない。そのためには、クライアントが要求するコンテンツをキャッシュプロキシが保持していなければならない。また本手法では、キャッシュプロキシの P2P オーバーレイネットワーク内で、クライアントからコンテンツが要求され、キャッシュプロキシオーバーレイネットワーク内に割り振られた場合、このネットワーク内で探索が行われる。本節では、サーバーの保持するコンテンツの複製、探索法について述べる。ここで、コンテンツの複製法、探索手法 [5] について主要なものを挙げる。

検索手法

検索手法とは、目的のコンテンツを持つキャッシュプロキシを発見するために、データを要求するピアが送信するピアが送信する検索要求メッセージの転送手法を以下に示す。

- flooding

データを要求するキャッシュプロキシは、全ての隣接キャッシュプロキシに検索要求メッセージを転送する。検索要求メッセージを受け取ったキャッシュプロキシもまた全ての隣接キャッシュプロキシに同様に転送を行う。この検索要求メッセージには TTL(Time to Live) が設定されており、この値が各キャッシュプロキシを通過するごとに 1 ずつ減少する。TTL が 0 になった場合は失敗検索と判断される。

- Scoped-search

図 3.2,3.3 のようにデータを要求するキャッシュプロキシは、ある決められた数だけ、隣接キャッシュプロキシに検索要求メッセージを転送する。検索要求メッセージを受け取ったキャッシュプロキシもまた、ある決められた数だけ隣接キャッ

キャッシュプロキシに検索要求メッセージを転送する。この検索要求メッセージには TTL(Time to Live) が設定されており、この値が各キャッシュプロキシを通過するごとに 1 ずつ減少する。TTL が 0 になった場合は検索失敗と判断される。flooding が全ての隣接ノードに検索要求を出すのに対して、Scope-search は、送出するメッセージの数だけ送出されるのが大きな違いである。

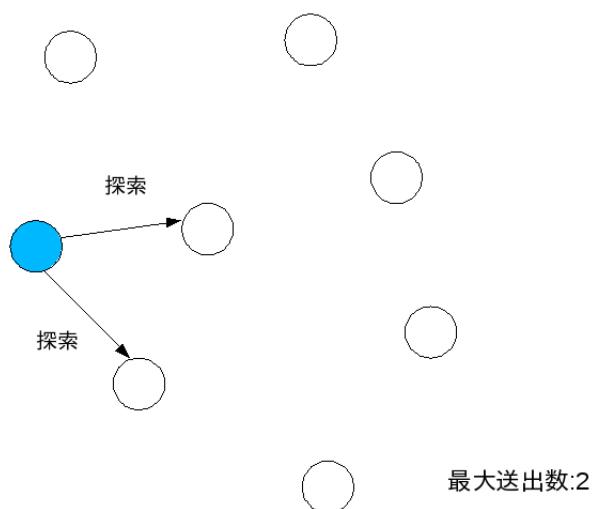


図 3.2: Scoped-Search 1 ホップ目

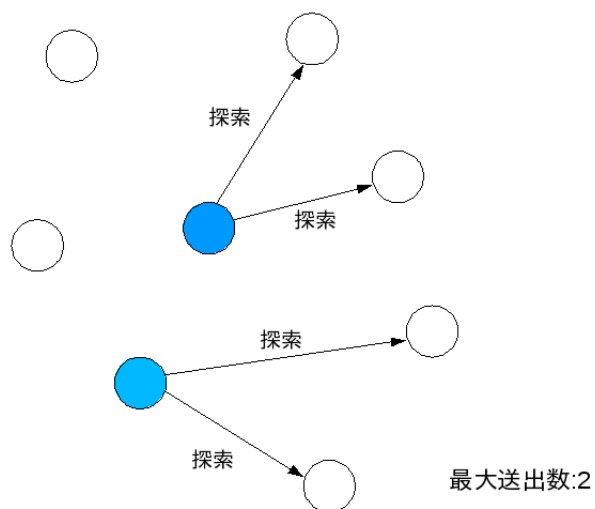


図 3.3: Scoped-Search 2 ホップ目

複製配置手法

複製配置手法とは、前節で挙げた検索の過程で複製する。主な複製方法として次のようなものが挙げられる。

- Owner Replication

図 3.4 , 3.5 のようにコンテンツを要求したキャッシュプロキシが受信したデータを複製として保持する。

- Path Replication

検索対象データを保持するキャッシュプロキシからデータを要求するキャッシュプロキシまでのパス上にあるすべてのキャッシュプロキシに複製が配置される。

- Random Replication

検索要求メッセージが通過した全てのキャッシュプロキシから Path Replication を適用した場合と同数のキャッシュプロキシと同数のピアからランダムに選択して複製を配置する。

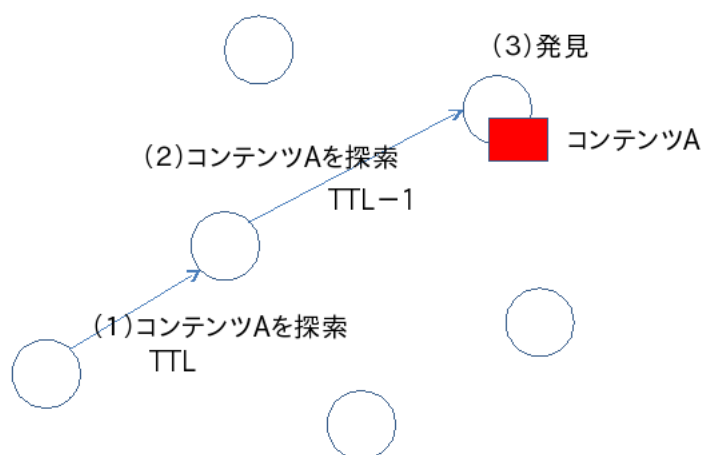


図 3.4: Owner Replication:目的コンテンツまでの探索

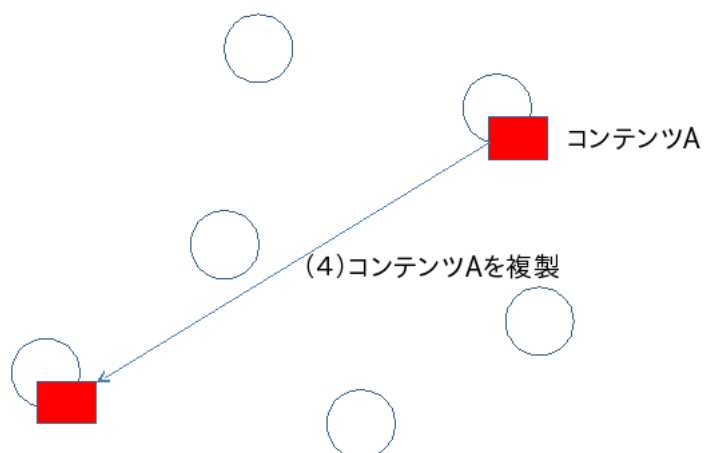


図 3.5: Owner Replication:探索元ノードに複製

本手法では、キャッシュプロキシは初期状態においてサーバーの保持するコンテンツの複製を保持していない。したがって、サーバーの保持するコンテンツをキャッシュプロキシに保持させる必要がある。またこの際、クライアントが要求する頻度が高い人気コンテンツを持つキャッシュプロキシが多く存在する複製法だとより良い負荷分散が期待できる。本手法において、サーバーは、クライアントにより要求されるコンテンツごとに、要求数を把握している。この時、ある一定の閾値以上のクライアントからの要求数があった場合に、サーバーはキャッシュプロキシに複製を行う。また、クライアントからのコンテンツ要求がキャッシュプロキシオーバーレイネットワーク内に割り振られた場合、割り振られた先のキャッシュプロキシが要求された該当コンテンツを保持していればクライアントに応答することができるが、保持していない場合は、オーバーレイネットワーク内で探索をする必要がある。この探索法として、Scoped-Searchによって探索される。見つからなければ、サーバーが応答し、見つかった場合、Owner Replication 法によって、探索元のキャッシュプロキシにコンテンツが複製され、探索元のキャッシュプロキシがクライアントに要求されたコンテンツを提供する。

3.2.3 従来手法の問題点

この手法では、キャッシュプロキシは、ある特定のサーバーの代理のみの役割をする。複数のサーバーの代理の仕事をキャッシュプロキシが行うような場合では、各キャッシュプロキシの状態を考慮せず、負荷分散を行っている。したがって、複数のサーバーの代理をキャッシュプロキシが行うといった条件下で、各キャッシュプロキシの状態を考慮した負荷分散をする必要がある。次章で、各キャッシュプロキシの負荷状態を考慮した負荷分散手法について提案する。また、既存手法では、キャッシュプロキシのリソース量が考慮されていなかった。しかし、実際、キャッシュプロキシには、コンテンツ保持できる量は有限であるために、提案手法では、キャッシュプロキシのリソースを有限とする。

第4章

提案手法

4.1 概要～送り手、受け手主導による負荷分散～

クライアントが Web コンテンツを利用しようとする場合、クライアントがコンテンツをサーバーに要求する。この際に、ある特定のコンテンツの要求数が増大することによる対策として、前章で紹介した既存手法では、サーバーの負荷をキャッシュプロキシによる P2P オーバーレイネットワークに負荷を分散していた。しかし、キャッシュプロキシが複数のサーバーの代わりの役割を果たすようなシステム (図 4.1) においては、P2P オーバーレイネットワークを構成している各キャッシュプロキシの負荷状態を考慮し、負荷を配置したほうが良いといえる。提案手法では、サーバーの負荷をキャッシュプロキシに分散させる際に、キャッシュプロキシの負荷状態を考慮して、負荷分散を行う。クライアントからのコンテンツ要求数が増加し、既存手法のように、キャッシュプロキシがオーバーレイネットワークを構成し、このネットワークに負荷が割り振られるようにする。その後、負荷状態の重いキャッシュプロキシは、負荷の軽いキャッシュプロキシに自キャッシュプロキシが持つ「クライアントにコンテンツを提供するという仕事」を代わりに請け負ってもらうことによって負荷状態に応じて負荷分散を行うという手法を導入する。高負荷状態のキャッシュプロキシが主導となり、軽負荷状態のキャッシュプロキシに自身の仕事を請け負わせようとする送り手主導方式と、軽負荷状態のキャッシュプロキシが主導となり、重負荷状態のキャッシュプロキシから仕事をもらうようにしようとする受け手主導方式を導入する [2]。

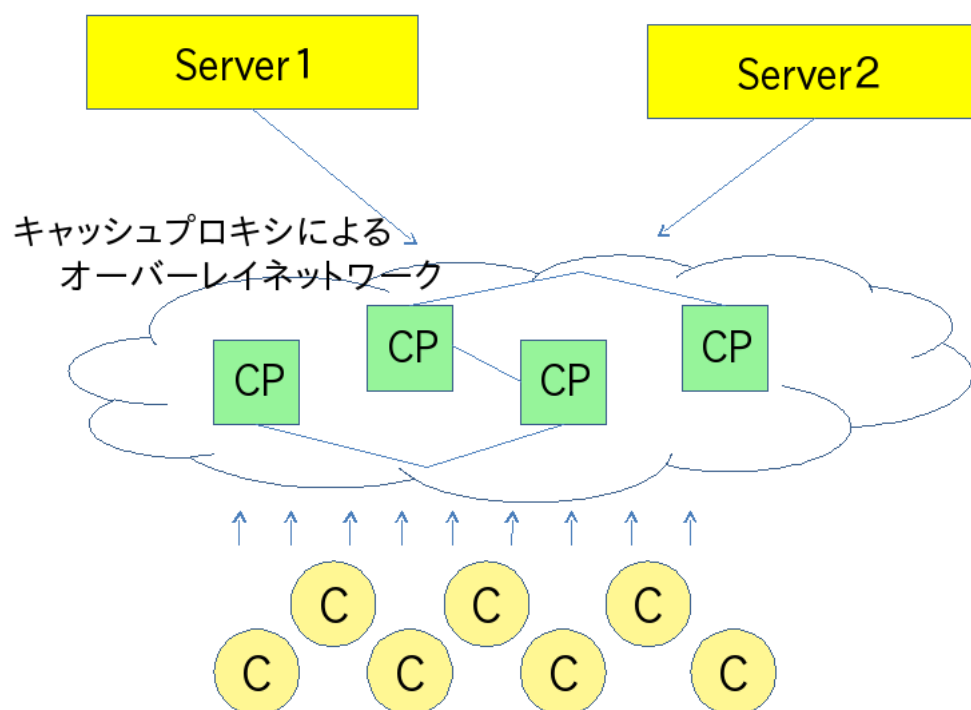


図 4.1: 複数サーバーが存在する状況

4.2 提案手法詳細

4.2.1 キャッシュプロキシによるオーバーレイネットワークの構成

既存手法と同様に、キャッシュプロキシは高負荷時にオーバーレイネットワークを構成する。クライアントからコンテンツのリクエストをキャッシュプロキシに割り振られるようにする。具体的には、DNSはクライアントからのコンテンツ要求の際に、サーバーの代わりにキャッシュプロキシのアドレスを返す。このことにより、キャッシュプロキシオーバーレイネットワークにコンテンツ要求を分散する。

4.2.2 コンテンツの複製・探索

クライアントからのコンテンツリクエスト負荷をサーバーからキャッシュプロキシへ分散するために、サーバー保持するコンテンツをキャッシュプロキシを持つように複製する必要がある。この手法としてサーバーに要求されるクライアントからのリクエスト数をコンテンツごとにカウントし、ある一定の閾値を超えた時に、キャッシュプロキシにコピーする。その後、Scoped-Searchによってコンテンツは探索され、Owner Replicationにより複製される。既存手法では、一台のキャッシュプロキシが保持できるコンテンツの制限数は想定されていなかった。しかし、実際には、キャッシュプロキシのリソースは有限であり、キャッシュコンテンツの上書きを行わなくてはならない。コンテンツの置換方法としては、主にFIFO(First In First Out),LFU(Least Frequency Used), (Least Recentry Used)LRUが挙げられる。TewariらによってピアP2P型を使用したキャッシュ置換アルゴリズムを用いた複製ファイルの最適配置について紹介されている。FIFOとは複製されたファイル順に削除する先入れ先出しアルゴリズムである。LFUとは使用頻度の少ない複製ファイル順に削除するアルゴリズムである。また、LRUとは過去最も長い期間使用されていないファイルを削除するアルゴリズムである。この研究を参考にし、今回はLRUを採用することとした。

4.2.3 負荷分散手法

本研究において前提とされるシステムは、クライアントからコンテンツのリクエストが出され、キャッシュプロキシがそのコンテンツ要求を請け負った場合、そのジョブは各キャッシュプロキシがもつコンテンツ応答待ち行列に入るものとする。その際に、そのキャッシュプロキシが要求されたコンテンツを保持していなかった場合、他のキャッシュプロキシへ要求する。つまり、Owner Replication における送信元ノードとなる。ここで、負荷状態の観測についての説明をする。負荷状態として、Heavy, Normal, Light の3状態からなる方法を採用し、その判断基準としてコンテンツ応答待ち行列の長さ L を用いた。閾値 L_1, L_2 により負荷状態は次のように決定される。

$$\begin{cases} 0 \leq L < L_1 \longrightarrow \text{Light} \\ L_1 \leq L < L_2 \longrightarrow \text{Normal} \\ L_2 \leq L \longrightarrow \text{Heavy} \end{cases}$$

次に、送り手主導方式、受け手主導方式の説明をする。送り手主導方式は、図 4.2 のように、そのキャッシュプロキシが高負荷時に、他のキャッシュプロキシに対してコンテンツ応答タスクの転送要求をする。具体的には、あるキャッシュプロキシに対して、クライアントからのコンテンツ要求が割り振られた場合、軽負荷時（上記で述べたうちの Light, Normal 状態に相当）には、コンテンツ応答タスクとして、そのキャッシュプロキシのコンテンツ応答待ち行列に入る。高負荷時（上記で述べたうちの Heavy に相当）には、他のキャッシュプロキシにコンテンツ応答タスク転送要求を出す。その要求が受け入れられればその受入先キャッシュプロキシにタスクの転送メッセージを送り、その受入先のキャッシュプロキシのコンテンツ応答待ち行列にタスクが入り、拒否されれば自身のキャッシュプロキシのコンテンツ応答待ち行列に入る。受け手主導の場合は、図 4.3 のように自身の負荷状態が軽負荷時（ここでは Light の時）には、他のキャッシュプロキシに対して、自身にタスクを送るキャッシュプロキシを見つけ、タスクを送ってもらうために、タスク転送要求メッセージを出す。タスク転送要求メッセージを受け取ったキャッシュプロキシは、負荷状態により重負荷時（ここでは Heavy

時)に応じて、タスク転送メッセージを出す。重負荷ではない場合には拒否し、何も行わない。

この間のタスク転送は、以下のようなメッセージを用いて行う。

Request:	タスク転送要求
Accept:	要求の受理
Reject:	要求の拒絶
Translate Task:	タスクの転送

Request とは、送り手主導 (受け手主導) 方式において、あるキャッシュプロキシが、高負荷時 (低負荷時) に他のキャッシュプロキシに対して出される、タスク転送要求メッセージである。要求を受け取ったノードは要求の種類と自分の状態から、Accept か Reject を返す。Translate Task とは、タスクそのものではなく、どのノードに対してコンテンツを転送するか、また、どのコンテンツを転送すればよいのかという情報メッセージである。

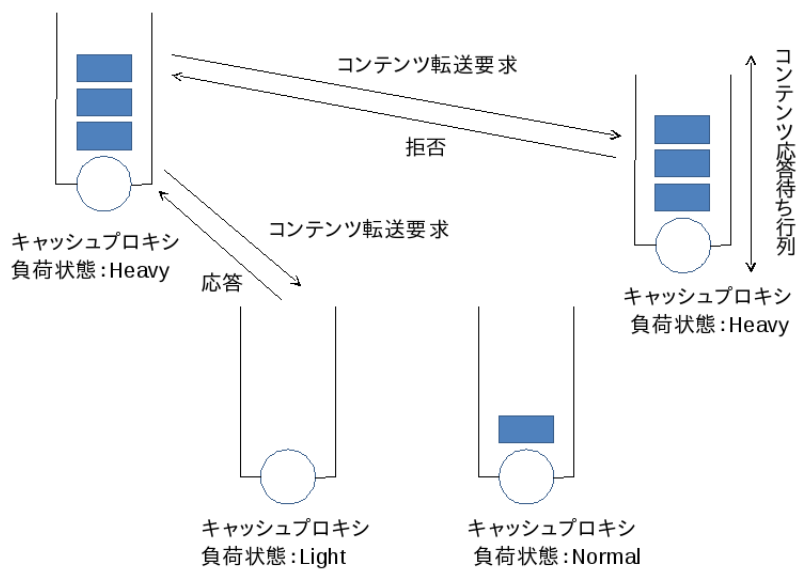


図 4.2: 送り手主導方式

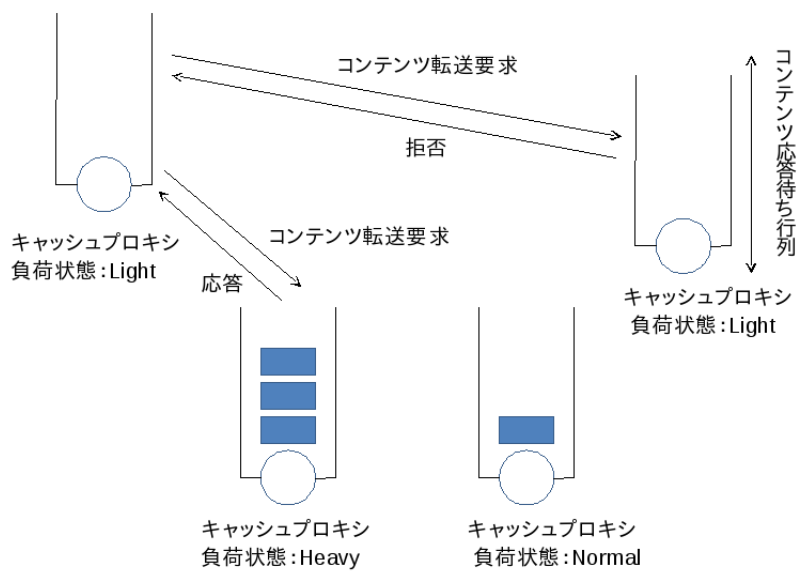


図 4.3: 受け手主導方式

第5章

実験と評価

従来手法では、キャッシュプロキシのオーバーレイネットワークは、ある単一のサーバーの代理の役割をすることを前提としていた。提案手法では、クライアントが要求するコンテンツを保持するサーバーとは無関係のサーバーが別に存在していて、キャッシュプロキシはそのサーバーの負荷分散も担当するために、各キャッシュプロキシの負荷状態が異なるという場面を想定している。この条件のもとキャッシュプロキシの負荷状態を考慮した負荷分散を行う。提案手法では、送り手主導、受け手主導を提案したが、これらの手法に関して実験をし、各手法の特徴、有用性を示す。

5.1 実験 1：高負荷時における負荷分散

キャッシュプロキシオーバーレイネットワークが、クライアントからのコンテンツ要求にどの程度応答できているのかという、コンテンツ応答数の評価を行う。本実験では、図のようにクライアントからの要求されるコンテンツを保持しているサーバーとは別に、クライアントからの要求されるコンテンツとは無関係なサーバーが存在する。キャッシュプロキシは、このクライアントからの要求されるコンテンツを保持しているサーバーの代理の仕事の他に、この無関係なサーバーの代理も行うとする。用意されたキャッシュプロキシのうち、クライアントからの要求に関係しないサーバーの仕事を代理するキャッシュプロキシ数を変化させたときの、送り先をランダムに決定する送り手主導方式、最も負荷が軽いキャッシュプロキシを送り先とする送り手主導方式、最も負荷が重いキャッシュプロキシに対してタスク転送要求を送る送り手主

導方式の3つの手法の比較をする。

シミュレーション実験は、自作のシミュレーションプログラムを用いて、次のように作成した。クライアントから送り出されるコンテンツとして、単位時間（秒）あたりにいくつのコンテンツ要求が送り出されるかを設定し、平均して単位時間あたりに設定したコンテンツ要求数が送り出されるように、順順に要求が送り出される。また、要求は同時に出されることはないものとする。また、DNSはサーバーに管理されており、DNSによって、負荷平常時には、クライアントからのコンテンツ要求が出された場合、サーバーのIPアドレスが返される。また、高負荷時の、対フラッシュクラウド構造の場合には、クライアントからのコンテンツ要求が出された場合、DNSによって、キャッシュプロキシによるオーバーレイネットワーク内のキャッシュプロキシのIPアドレスが返される。クライアントからの単位時間あたりのコンテンツ要求数、また単位時間あたりにキャッシュプロキシオーバーレイネットワークが応答するコンテンツ要求数をカウントし、どれだけのコンテンツ要求に対して、どの程度応答できているかを評価する。また、タスク転送要求メッセージや、要求応答メッセージ、要求拒否メッセージ等、タスク転送に必要なメッセージ数も考慮に入れるものとした。

シミュレーションを行う上でのパラメータは表 5.1 とおりである。

表 5.1: パラメータ

キャッシュプロキシ台数	10
1 コンテンツあたりの転送時間	50msec
サーバーが保持するコンテンツ数	10
人気コンテンツ数	2
不人気コンテンツ	8
メッセージオーバーヘッド	5msec
1秒あたりクライアントからのコンテンツ要求数	60
Fanout	1
閾値 L_1	3
閾値 L_2	4
別サーバーからのタスク転送頻度	40(Task/sec)

サーバーは10種類のコンテンツを保持しており、このうち2種類のコンテンツに対して、クライアントからのコンテンツ要求が集中する。メッセージオーバーヘッドとは、送り手主導、受け手主導の方式において、コンテンツ転送用要求、応答メッセージ、拒否メッセージといったメッセージを送出する際のオーバーヘッドである。Fanoutとは、Scoped-Searchにおいて、同時に送出手されるクエリ数である。 L_1, L_2 とはキャッシュプロキシの持つ、コンテンツ応答待ち行列長さの閾値である。この閾値によって、そのキャッシュプロキシの負荷状態が決定される。

5.2 結果

この実験においてのメッセージオーバーヘッドを考慮した場合の結果(図5.1)、オーバーヘッドを無視した場合の結果(図5.2)を示す。転送先をランダムに転送する送り手主導方式よりも、軽負荷なキャッシュプロキシに転送要求を出す送り手主導方式、高負荷なキャッシュプロキシに転送要求を出す受け手主導方式のほうがより多くコンテンツ要求に回答できるようになっていることがわかる。さらにここでは、軽負荷なキャッシュプロキシに転送要求を出す送り手主導方式と、高負荷なキャッシュプロキシに転送要求を出す受け手主導方式を比較する。システム全体として高負荷時において、送り手主導方式のほうが、受け手主導方式よりコンテンツ回答ができていないことが見て取れる。また、逆にシステム全体として軽負荷時には、送り手主導方式よりも、受け手主導方式の方が、わずかながら少ないコンテンツ回答数になっている。また、図5.2を見ると、受け手主導と送り手主導との差異はあまり見てとることができない。したがって、高負荷時には受け手主導が多く回答できていて、軽負荷時には送り手主導が多く回答できているという理由は、送出手されるメッセージオーバーヘッドであることがわかる。

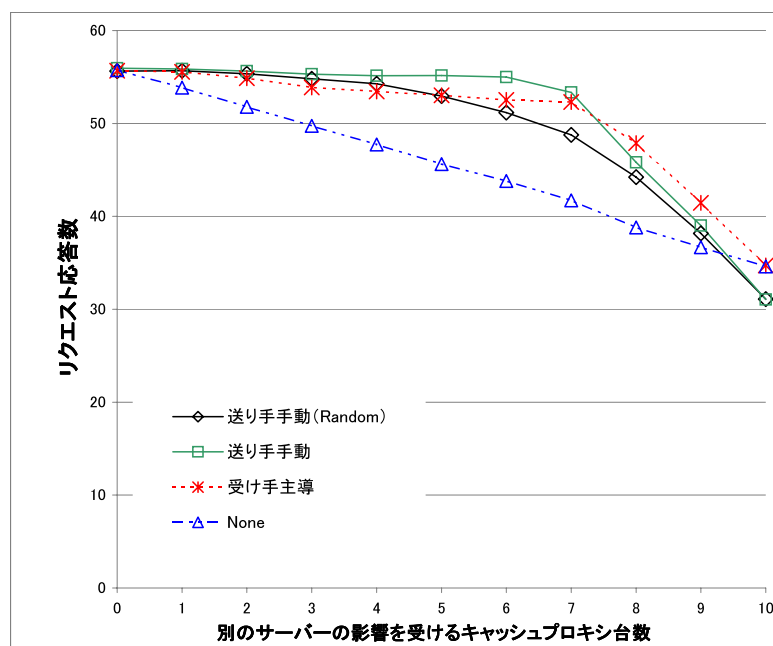


図 5.1: コンテンツ応答数の推移 (メッセージオーバーヘッドあり)

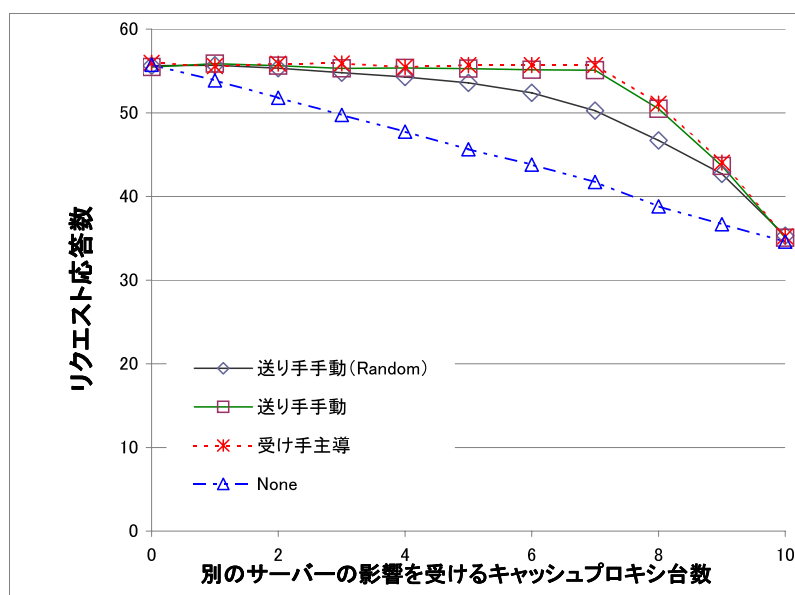


図 5.2: コンテンツ応答数の推移 (メッセージオーバーヘッドなし)

5.3 実験 2 : クライアントからのリクエスト数の変化

実験 2 では、キャッシュプロキシの負荷状態が違う場合、クライアントからのリクエストを増加した場合のコンテンツ要求応答の追従性について実験を行う。負荷変動として、10 台のキャッシュプロキシのうち、他サーバーからのタスク発生間隔は 40msec で、8 台に他サーバーからのタスクがいくようになっている。この上で、クライアントからのコンテンツ要求数が増加した場合のコンテンツ応答の追従性、コンテンツ応答数を評価する。キャッシュプロキシのパラメータは次のとおりである。シミュレーションを行う上でのパラメータは表 5.2 とおりである。

表 5.2: パラメータ

キャッシュプロキシ台数	10
1 コンテンツあたりの転送時間	50msec
サーバーが保持するコンテンツ数	10
人気コンテンツ	2
不人気コンテンツ	8
メッセージオーバーヘッド	5msec
Fanout	1
閾値 L_1	3
閾値 L_2	4
別サーバーからのタスク転送頻度	40(Task/sec)
別サーバーから影響を受けるキャッシュプロキシ数	8

またクライアントから送出されるコンテンツ要求は次のように変動するものとする。

(図 5.3)

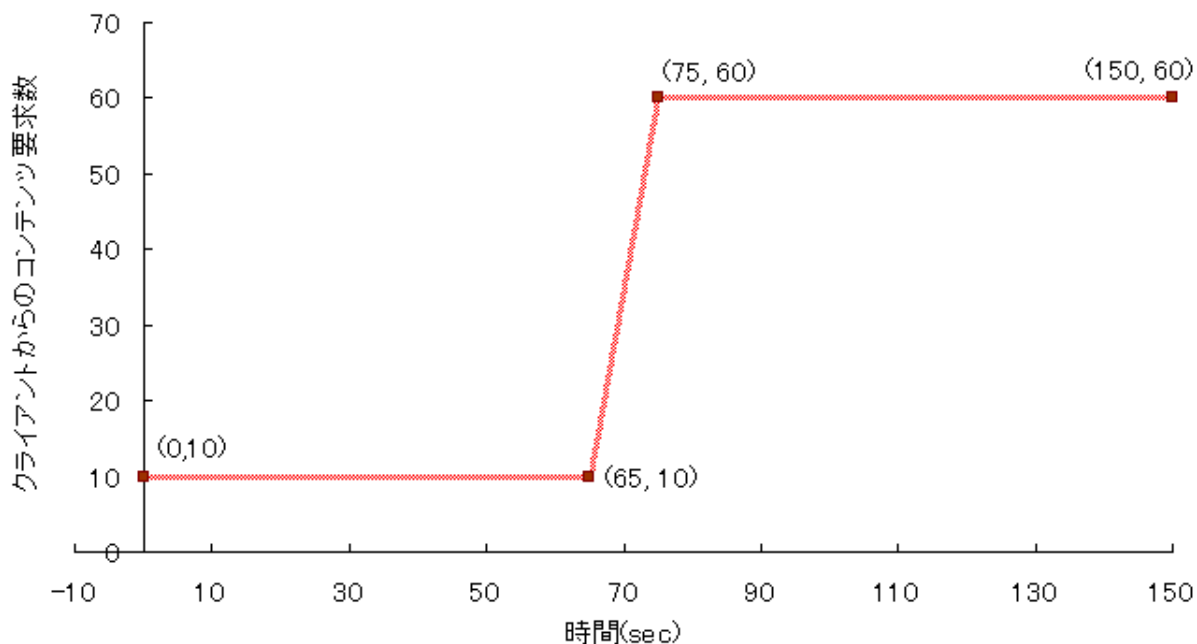


図 5.3: クライアントからの要求数の変化

クライアントからの要求数は一秒あたり 10 リクエストで、65 秒から 10 秒間かけて 60 リクエストに上昇。その後、1 秒あたり 60 リクエストを保つ。本実験では、このようにキャッシュプロキシの負荷状態が異なり、クライアントからのコンテンツ要求が上昇した場合の、コンテンツ応答数の追従性を比較評価する。比較する手法は、タスク転送先をランダムに決める送り手主導方式 (Random)、軽負荷なキャッシュプロキシにタスクを転送する送り手主導方式、高負荷なキャッシュプロキシのタスクをもらおうとする受け手主導方式の比較を行う。

5.4 結果

実験結果を図 5.4 に示す。今回の実験では、別サーバーからのタスクを受け取るキャッシュプロキシは 8 台とした。これは実験 1 において高負荷状態であり、今回も、メッセージオーバーヘッドの影響により、送り手主導方式は、受け手主導方式に比べ応答数が良くない。また、クライアントからの要求が増大した際、送り手主導方式 (Random)

は、適切なタスク転送先の決定が出来ていないぶん、送り手主導、受け手主導方式より一般的に応答数が良くないと考えられる。

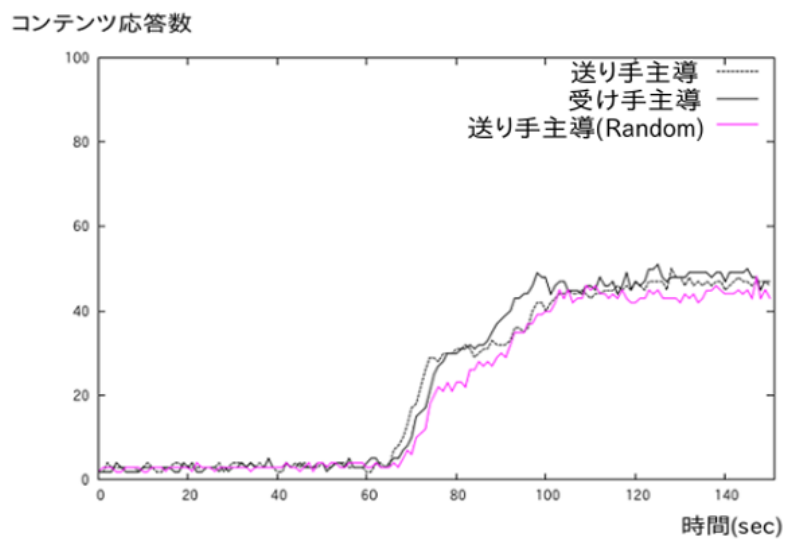


図 5.4: クライアントからの応答数の変化

5.5 考察

転送要求を送る先をランダムに決定する方式より、転送要求先を負荷状態により決定する方法では、転送要求先を負荷状態によって決定する方式のほうがコンテンツ応答数は多く応答できていると考えられる。これはランダムに転送要求を送るよりも、負荷状態を考慮して転送要求を送出したほうが、拒絶される確率が少ない。このため、タスク転送が多く行われていると考えられる。また、軽負荷なキャッシュプロキシに転送要求を出す送り手主導方式と、高負荷なキャッシュプロキシに転送要求を出す受け手主導方式を比較した場合、システム全体として高負荷時において、送り手主導方式のほうが、受け手主導方式よりコンテンツ応答ができていないことが見て取れる。また、逆にシステム全体として軽負荷時には、送り手主導方式よりも、受け手主導方式の方が、わずかながら少ないコンテンツ応答数になっていた。この大きな理由として、タスク要求メッセージ送出によるオーバーヘッドが挙げられる。システム全体として軽負荷時において、軽負荷状態であるキャッシュプロキシが多く存在する。このため、受け手主導方式においては、タスク転送要求を送出するキャッシュプロキシが多い。しかし、軽負荷状態であるキャッシュプロキシが多く存在するために、タスク転送要求を送出しても拒絶される可能性が高い。このため、メッセージ送出のオーバーヘッドが影響していると言える。システム全体として高負荷時において、高負荷状態であるキャッシュプロキシが多く存在する。このため、送り手主導方式においては、タスク転送要求を送出するキャッシュプロキシが多い。しかし、高負荷状態であるキャッシュプロキシが多く存在するために、タスク転送要求を送出しても拒絶される可能性が高い。このため、メッセージ送出のオーバーヘッドが影響していると考えられる。

第6章

まとめ

各キャッシュプロキシの負荷状態に応じて負荷を割り当てるために、高負荷キャッシュプロキシが主導となり、低負荷キャッシュプロキシに対して負荷を分散する送り手主導方式、低い負荷キャッシュプロキシが主導となって高負荷キャッシュプロキシから負荷を貰い、自分に分散させる受け手主導方式を提案した。オーバーレイネットワークにクライアントがコンテンツを要求するコンテンツとは無関係のサーバーの代理をさせる環境で、サーバーの代わりにさせるキャッシュプロキシ数を変化させ、送り手主導、受け手主導の有効性を評価した。

送り手主導方式は、受け手主導方式に比べ、高負荷時に応答数が少なくなり、受け手主導方式は、送り手主導方式に比べ、軽負荷時に応答数が少なくなるという結果が得られた。これは、送り手主導は高負荷時には、タスク転送要求に対して、応答できるキャッシュプロキシが負荷が高いが多いために存在しにくく、またタスク転送要求メッセージを出すオーバーヘッドが増大してしまう、また、受け手主導は軽付加時には、タスク転送要求に対して、システム全体として軽負荷のため、応答できるキャッシュプロキシが存在しにくく、タスク転送要求を出すオーバーヘッドによるものであることを示した。したがって、低負荷時には、送り手主導方式が、高負荷時には、受け手主導方式が有効である。このことから負荷状態に応じて手法を切り替えることが有効である可能性がある。

謝辞

本研究を進めるにあたって、多くの方々からご指導、御教授いただきましたことを感謝いたします。御多忙の中、研究方針を明確に御指導くださり、熱心に的確な助言を頂きました 松尾 啓志教授に、深く感謝いたします。中間発表や、研究経過にアドバイスをくださいました 松井俊浩 助教に深く感謝いたします。研究にも助言頂き、そしてまた、精神的にも支えになってくださった津邑・公暁 准教授に深く感謝申し上げます。 本研究を進めるにあたって、久野 友紀氏、加藤 伸哉氏、山城 穂高氏には、常日頃から熱い御指導を頂きましたことを心より深く感謝いたします。

参考文献

- [1] Chenyu Pan, Merdan Atajanov, Mohammad Belayet Hossain, Toshihiko Shimokawa and Norihiko Yoshida, "FCAN: Flash Crowds Alleviation Network Using Adaptive P2P Overlay of Cache Proxies,"
IEICE Transactions on Communications, Vol.E89-B, No.4, pp.1119-1126, (April, 2006)
- [2] 小池浩和, 松尾啓志, 岩田彰, "アクセス監視による再活性を導入した適応型負荷分散法" 名古屋工業大学卒業論文 (1996)
- [3] 棟朝雅晴, 高井昌彰, 佐藤義治, "確率学習による適応度評価を導入した遺伝的アルゴリズムに基づく動的負荷均衡", 情報処理学会論文誌 Vol.36 No.4 (1995.4)
- [4] 浅原理人, 島田明男, 山田浩史, 河野健二, "需要変動に応じたコンテンツの再配置を行う Content Delivery Network(コンテンツ配信)"
情報処理学会研究報告, Vol.2006, No.44, pp.85-92(2006.5)
- [5] 川崎 陽平, 佐藤 崇, 吉田 紀彦, "P2P ネットワークにおけるコンテンツの人気度を反映した複製配置", インターネットコンファレンス 2005 論文集, pp.106-113(2005.10)