

Webプロキシネットワークシステムにおける ノード自律協調型負荷分散方式

市木 良和[†] 鈴木 優[‡] 川越 恭二[‡]

[†]立命館大学大学院 理工学研究科 [‡]立命館大学 情報理工学部

1 はじめに

本研究では、Webプロキシネットワークにおけるノードの負荷を推定するための手法を述べる。近年、インターネットの普及により数多くのWebサービスが提供されているが、特定のサービスへの負荷集中により、十分なサービスを大量の利用者へ提供することが困難である場合がある。そこで、著者らはこれらの問題を解決するための一つの手法として、Webプロキシネットワークの開発を行っている[1]。Webプロキシネットワークとは、P2P技術を利用して複数のWebプロキシを接続することによって、単体のWebプロキシよりも大量のWebサーバ、利用者へ対応することを目的としたWebプロキシシステムである。

提案しているWebプロキシネットワークでは、それぞれのWebプロキシに対して均等なネットワーク負荷がかかるように、Webプロキシのネットワーク負荷に関する情報をWebプロキシ間で通信する。ところが、負荷分散を行うWebプロキシのネットワーク負荷が急に高負荷になった場合、Webプロキシ間の負荷に関する情報の取得効率が落ちるため、効率的にWebプロキシの負荷分散が動作しないという問題がある。そこで提案手法では、Webプロキシはファイルサイズから受け入れるコンテンツ取得要求の影響を概算する。提案手法を利用することにより、コンテンツ取得要求を受け入れる前に、Webプロキシは概算した情報を元にネットワーク負荷の負荷分散を行う。その結果、Webプロキシ間のネットワーク負荷が高負荷になる状況をWebプロキシ自身が回避することが可能になる。

2 提案手法

2.1 Webプロキシネットワークシステムの負荷分散構成

はじめに、Webプロキシネットワークシステムの負荷分散構成について説明する。まず、あらかじめプロキシサーバ間でP2Pネットワークを構築する。このネットワークを用いて、プロキシサーバ間で負荷情報の交

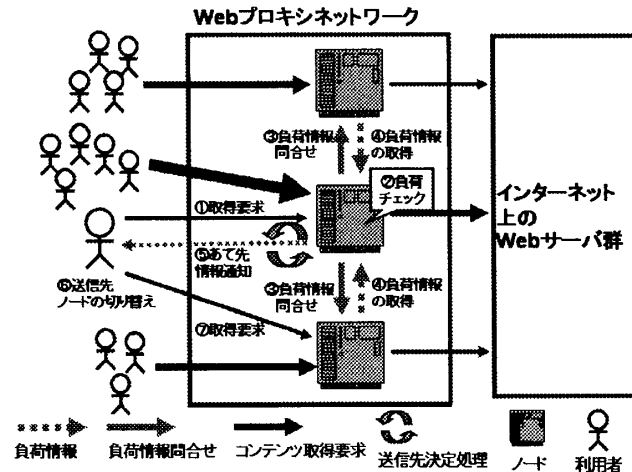


図1: ノード自律協調型負荷分散方式の手順

換を行う。以後、本稿ではP2Pネットワークを構成するプロキシサーバをノードと定義する。

次に、負荷分散を行う手法について説明する。Webプロキシネットワークシステムで行われる負荷分散手法の手順を図1に示す。まず、コンテンツ取得要求を受け取ったノードは負荷を測定する。測定した負荷情報を元に負荷分散の実行可否の判定を行う。判定する方法として、ノードごとに設定された、この閾値と負荷値を比較して判断する。そこで、負荷が低いと判断したノードは、そのノードがコンテンツ取得要求に基づき中継処理を行う。また、負荷が高いと判断したノードは、他ノードから負荷情報の取得を行う。負荷分散を行うと判断したノードは負荷情報を取得したノードの中から、負荷の少ないノードの導出を行う。中継処理を行うノード k を決定する方法を、式(1)に示す。

$$k = \begin{cases} \{l \mid L_l = \min(L_i/C_i), \\ 1 \leq l \leq m\} \text{ if } (C_n < L_n) \\ n \text{ otherwise} \end{cases} \quad (1)$$

コンテンツ取得要求を受け取ったノードを n とし、負荷情報の取得を行ったノードを $i(1 \leq i \leq m)$ 、各

ノードの負荷値を $L = \{L_1, \dots, L_m\}$ とする。そして l は負荷が少ないノードを表す。そして、 C_i は各ノードごとにあらかじめ設定されている閾値である。ノードは負荷値をそれぞれ設定した閾値で正規化を行った値を比較することによって、ノードの処理能力の差を考慮した負荷分散を行うことが可能と考えた。負荷分散を行うノードは負荷の少ないノードが決定できなかった場合は、エラーの通知を行う。そして、クライアントシステムは、コンテンツ取得要求を配信元サーバに送信を行う。これにより、利用者が本システムがコンテンツを取得できない状態を回避する。また、負荷の少ないノードが発見できた場合、負荷分散を行うノードは利用者のクライアントシステムに負荷が少ないノードのあて先情報を通知する。そして、クライアントシステムは通知されたあて先情報を元に送信先を負荷の少ないノードに切り替える。これにより、本システムの負荷を分散させることが可能となる。

2.2 Web プロキシネットワークで用いる負荷情報

ノードが負荷情報を他のノードから、可能な限り早く取得することが重要である。また、本システムの負荷分散において、データの転送速度にかかわる負荷情報の選択を行うことが重要である。まず、負荷情報としてノードが所持する接続の数を用いた場合が考えられる。しかし、ノード間のネットワーク帯域を考慮に入れてない。このため、ノードの通信速度が低下する場合がある。次に、負荷情報としてノード間の応答速度を用いた場合が考えられる。しかし、ノード間の応答速度が、利用者とノードの応答速度が同一とは限らない。このため、ノード間の応答速度という情報は本システムの負荷分散では利用できない。最後に、ノードが自分のネットワーク帯域を監視する手法の場合が考えられる。ノード自信がネットワーク帯域を厳密に監視することとは、監視すること自体が負荷になる。このため、この様な負荷情報を用いたとしても、ノード間の通信速度が低下する場合がある。そして、ノードは負荷情報を取得する速度が低下し、ノードの中継処理速度が低下する。

そこで、ファイルサイズが大きいコンテンツほどネットワーク帯域を使用する量が多いという特徴に着目した新しい負荷情報を用いる。ネットワークの帯域を最も使用するのが、ノードと利用者間のコンテンツを取得するための接続である。利用者とノードの接続は、取得目的のファイルサイズが大きい程、コンテンツを転送するまで時間がかかる。このことから、ファイルサイズが大きい接続を抱えているノードほど処理に時間

がかかると考えられる。そこで、コンテンツのファイルサイズからネットワーク帯域の影響を概算できないかと考えた。具体的には、利用者とノード間の接続に対して取得するファイルサイズを重み付けした値を負荷情報として用いる。この指標を式 (2) に示す。

$$L_i = \sum_{j=1}^t F_j \cdot W(c_j) \quad (2)$$

あるノード i ($i = 1, 2, \dots, t$) において接続されている Web コンテンツを c_1, \dots, c_t とし、コンテンツ c_j へのコンテンツ取得要求数を F_j 、ファイルサイズは $W(c_j)$ である。この負荷情報 L_i を元に負荷分散を行うノードは負荷の少ないノードを絞り込む。この指標によりノードの通信速度の低下を防ぐ。

また、式 (2) に示した提案した指標に加えて接続数の情報も考慮する。なぜなら、分散先が発見しにくい状況では、すべてのノードにおいて転送速度の低下は避けられない。そこで、接続数を均等にし速度の分割をはかる必要があると考えた。そこで、提案した指標に加えて、接続数の項目と、それぞれの項目ごとに設定した閾値と比較を行うことで解決を行う。

2.3 自ノードの負荷予測

ノード間の通信速度が低下する確率を抑えるために、ノードは自ノードの負荷予測を行い負荷分散を行う。本稿における負荷予測とは、コンテンツ取得要求を処理するノードを決定する前に、自ノードの負荷を中継する前に判断することである。実際には、ノードは受け入れる前の負荷値とコンテンツ取得要求を受け入れた場合の負荷を加算する。そして、加算した負荷値と閾値との比較を行うことで自ノードの負荷を予測できると考えた。負荷の予測を行うために、式 (2) に示した指標においてはコンテンツのファイルサイズを取得する手段が必要がある。そこで、ノードはキャッシュから目的のコンテンツのファイルサイズを取得する。また、キャッシュが存在しない場合は配信元の Web サーバから目的のファイルサイズを取得する。これにより、ノードは受け入れるコンテンツ取得要求の負荷がどの程度の負荷になるかを推測できる。結果、ネットワーク帯域を圧迫してしまうコンテンツ取得要求ほど別のサーバに分散する確率があがると考えられる。そして、ノード間の通信速度が低下する確率が下がると考えた。

3 評価実験

提案手法により、どの程度利用者がコンテンツを得る速度の低下を防げるかを評価するために実験を行った。

3.1 実験環境

本実験では、イベントドリブン型のシミュレータを開発し、シミュレーション実験を行った。各サーバや回線ごとにイベントを処理するスケジューラを設定した。コンテンツ取得要求や各中継処理などのタスクをすべてイベント化した。また、イベントの処理時間を1タイムスロットあたり1msecになるよう設定を行った。シミュレーションにおける評価環境を表1に示す。

3.1.1 ノードの接続状態

Webプロキシネットワーク内ではノードは必ず、一つまでの接続を持つ。そして、ノードは無作為なP2Pネットワークを形成する。本実験で用いた、論理ネットワーク構造を図2に示す。本実験中では、ノードの参加や脱退などノードが増減することがない。また、ノードは接続を切断や繋ぎ変えを行わない。これにより、ノードが形成する論理ネットワーク構造に変更がないものとする。

3.1.2 負荷情報取得手法

負荷情報の取得方法としてフラディングを用いる。フラディングとはノードが隣接ノードにメッセージを伝播していく手法である。伝達期限が切れたメッセージを受け取ったノードは、伝達元のノードに負荷情報を返信する。また、メッセージを送信したすべてのノードから負荷情報を受け取ったノードは伝達元のノードに負荷情報を返信する。これにより、メッセージを受け取ったすべてのノードから負荷情報を取得する。

3.1.3 クライアントシステムの動作

シミュレータでは自動的にコンテンツ取得要求を発生させた。シミュレーション開始前において、クライアントシステムのコンテンツ取得要求先は、無作為に決定する。次に、目的のコンテンツを自動的に決定す

表1: 評価環境

利用者数	40000~60000
ノード数	100
コンテンツの総数	1000
コンテンツの平均サイズ	100kbyte
ノード間の最大接続数	3
TTL	3
回線速度	10Mbps
接続数の閾値	15
提案指標の閾値	2.0×10^{20}

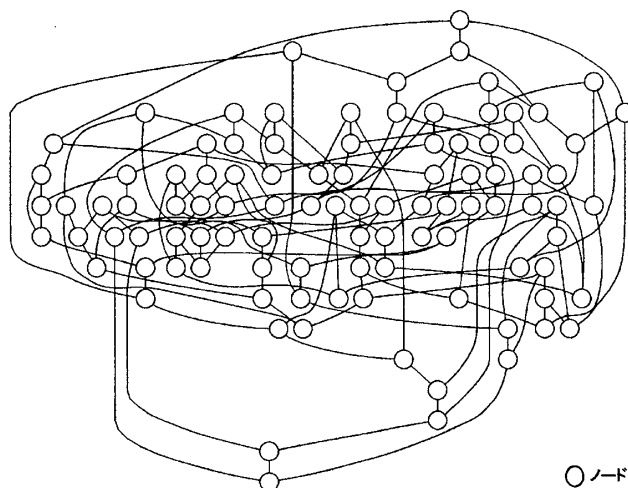


図2: シミュレーションにおけるノードの接続状況

る方法について説明する。本実験においては、コンテンツの人気により取得する対象の偏り具合を決定した。取得する対象のコンテンツはZipf-like分布[2]に基づくよう発生確率を決定した。最後にコンテンツ取得要求の発生頻度について説明する。利用者がコンテンツを取得してから再取得を行うまでの間隔は平均10[sec]とした。また、発生間隔の最小値を0.1[sec]とし、最大値を100[sec]とした。クライアントシステムはノードからあて先通知を受け取った場合においては、即座に変更先ノードにコンテンツ取得要求を行った。

3.2 実験結果

どの程度利用者がコンテンツを得る速度の低下を防げるかを評価を行う。また、ノードの転送速度にばらつきが発生しないかを評価する。比較対象を以下に示す。

負荷分散なし 負荷分散を行わなかったシステム構成
接続数のみ 負荷情報を接続数のみを用いた負荷分散手法

接続数と提案手法 負荷情報を接続数に加えファイルサイズで概算した情報を用いた負荷分散手法

これらの項目において比較を行った。

3.2.1 コンテンツを取得するまでの速度

データをどの程度の速度で取得できるかを比較する。評価指標として、平均転送速度を求める。まず、コンテンツ取得要求を発生してからコンテンツが取得完了する時間を測定した。そして、取得したコンテンツのファイルサイズと取得時間の商を取った値を転送速度とした。さらに、コンテンツ取得要求すべての転送速

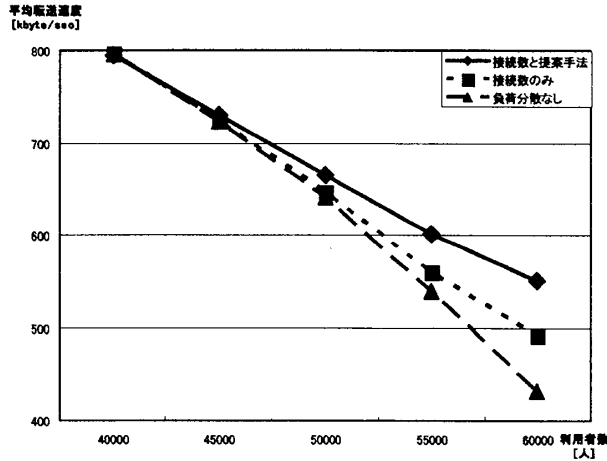


図 3: 平均転送速度の分布

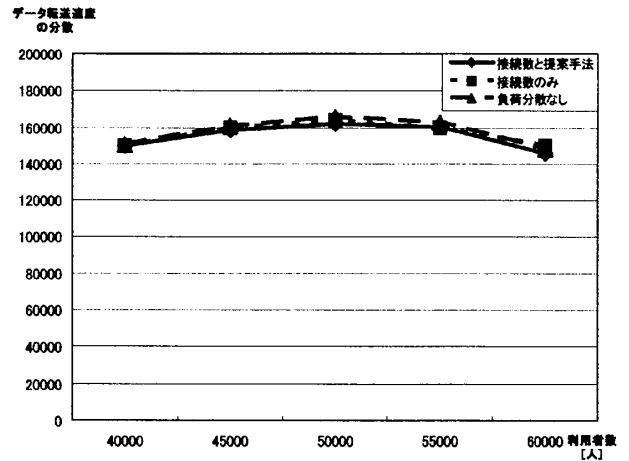


図 4: 転送速度の標本分散値の分布

度を平均したものを平均転送速度とした。転送速度は以下の(3)式に、また平均転送速度 \bar{D} を(4)式に示す。

$$D_r = \frac{W(c_r)}{T_{tr}} \quad (3)$$

$$\bar{D} = \frac{\sum_{r=1}^p D_r}{p} \quad (4)$$

それぞれのクライアントから得られたコンテンツ取得試行 $r (r = 1, 2, \dots, p)$ があり、コンテンツ c_t の取得時間を $T_{t1} \dots T_{tp}$ とする。

Web プロキシネットワークの利用者数を 5000 人ごとに変更し、この平均転送速度を測定した。平均転送速度の推移を図 3 に示す。利用者が増加すれば増加するほど平均転送速度の低下がどの手法においても確認できた。また、従来手法は負荷分散を行わなかったときと比べて平均転送速度が向上している。この結果から、提案手法は本システムの負荷が高く状態において、平均転送速度の低下を防ぐのに有効であることがわかった。

3.2.2 コンテンツを取得するまでの速度のばらつき

提案した手法が、転送速度の標本分散の値を導出し比較を行った。標本分散の値が小さくなるほど、転送速度が一樣に分布していることを示す。そのため、標本分散の値が高いほど、各ノードの転送にばらつきがある。転送速度のばらつきが発生してしまうと、特定のノードが他のノードに比べて、良い転送速度を得られない状況が発生する。そこで提案手法が、他の手法において転送速度がばらつかないかどうかを確認する。転送速度の標本分散の値 σ^2 を求める式を(5)に示す。

$$\sigma^2 = \frac{\sum_{r=1}^p (D_r - \bar{D})^2}{p} \quad (5)$$

3.2.1 節と同じ項目において、標本分散の推移を図 4 に示す。どの評価項目においても、分散値は全体的に同程度の数値を示していた。このことにより、提案手法は接続数を用いた負荷分散手法と比較しても、転送速度のばらつきが発生しないことを実証できた。

4 おわりに

負荷情報として接続数に加えてファイルサイズからネットワーク負荷を概算し負荷分散を行う手法を提案した。シミュレーションの結果、Web プロキシネットワークシステムの中継処理速度が向上することが確認できた。今後の課題として、負荷情報を高速に取得する P2P ネットワーク構造を検討を行う予定である。

参考文献

- [1] 飯間悠樹, 市木良和, 清田寛信, 鈴木優, 川越恭二. クライアント協調型機構を持つ負荷分散 Web プロキシネットワークシステム. 電子情報通信学会 第 18 回データ工学ワークショップ (DEWS), 2007.
- [2] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, and Scott Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. *Proceedings of the 3rd International Web Caching Workshop*, pp. 126-134, 1999.