Web サーバシステムにおけるコネクション受付制御方式*

佐竹 伸介^{†‡}・稲井 寛[†]・齊藤 智也[§]・荒井 剛[†]A Connection Admission Control Method
for Web Server Systems*Shinsuke SATAKE^{†‡}, Hiroshi INAI[†], Tomoya SAITO[§] and Tsuyoshi ARAI[†]

Most browsers establish multiple connections and download files in parallel to reduce the response time. On the other hand, a web server limits the total number of connections to prevent from being overloaded. That could decrease the response time, but would increase the loss probability, the probability of which a newly arriving client is rejected. This paper proposes a connection admission control method which accepts only one connection from a newly arriving client when the number of connections exceeds a threshold, but accepts new multiple connections when the number of connections is less than the threshold. Our method is aimed at reducing the response time by allowing as many clients as possible to establish multiple connections, and also reducing the loss probability. In order to reduce spending time to examine an adequate threshold for web server administrators, we introduce a procedure which approximately calculates the loss probability under a condition that the threshold is given. Via simulation, we validate the approximation and show effectiveness of the admission control.

1. まえがき

近年, サービス内容の多様化, 端末機器の多様化・低価格化, アクセス回線の広帯域化・低料金化などが進展し, 現在では, 非常に多くのユーザが WWW を利用している. このような状況において, 多くのユーザに対して有益な情報やサービスを提供しているサイトではリクエストが集中するため, 複数台のサーバを設置してクラスタ化し, 並列処理により実質的なシステム処理能力を向上させている [2]. また, サイト側のさらなる取り組みとして, リバースプロキシ [9] の導入, 効率的な Web アプリケーションの設計 [8], ホームページ記述法の開発 [12] などがある.

上述の Web サーバクラスタにおける並列処理やパケット交換網におけるパイプライン処理の特性を活かして, 多くのブラウザでは, 複数の TCP コネクション (以降, 単にコネクションという) を確立し, 複数のファイルを並列にダウンロードしている. たとえば HTTP1.1 [6] による通信では標準で 2 本のコネクションを使用している.

その一方で, サイトでは, 過負荷による応答時間の増大を防止するため, 接続を受け付けるコネクション総数を制限している [5]. しかし, 接続総数を制限すると, サイトへの接続を拒否されるユーザが増加する恐れがある. このことは, とくに電子商取引などを行っているサイトにおいて, 収益を減少させる要因になるため, 大きな問題となる.

そこで, 本論文では, サイト側において, 接続中のコネクション総数が少ない場合には新規クライアントからの複数コネクション確立要求をそのまま受け付け, 接続総数が多い場合は 1 コネクションのみ受け付けるコネクション受付制御方式を提案する. これにより, 可能な限り並列ダウンロードを有効にして応答時間の短縮を図り, かつユーザからの接続が棄却される確率 (1 本のコネクションも受け付けられない確率) の低減を目指す. 上述の参考文献 [2, 9, 8, 12] などの研究は応答時間の短縮を目指すものであり, 接続が棄却される確率の低減は考慮されていない.

* 原稿受付 2011 年 1 月 19 日

[†] 岡山県立大学 情報工学部 Faculty of Computer Science and System Engineering, Okayama Prefectural University; 111 Kuboki, Soja, Okayama 719-1197, JAPAN[‡] (株) アレス Aress Inc.; 39-11 Kori, Minami-ku, Okayama 702-8011, JAPAN[§] 島根大学 総合理工学部 Interdisciplinary Faculty of Science and Engineering, Shimane University; 1060 Nishikawatsu-cho, Matsue, Shimane 690-8504, JAPAN**Key Words:** web server system, connection admission control, queueing system, loss system.

以下, 2. でコネクション受付制御方式を提案する. 3. では提案方式の性能を評価するためのモデルを構築する. 4. では提案方式のパラメータであるしきい値の検討に要する時間を削減するための近似解析手法を示し, 近似の妥当性について検証する. 5. では性能評価を行って, 提案方式の有効性を明らかにする. 最後に 6. で本研究で得られた結果のまとめと今後の課題について述べる.

2. 提案方式

ホームページ閲覧の際の応答時間を短縮するため, 複数コネクション (HTTP1.1 では標準で 2 コネクション) による並列ダウンロードが一般的に行われている. 一方, サイト側では, 過負荷を防止するため, 最大受付コネクション数 S を設定している. ここで, すべてのクライアントが 2 コネクションずつ確立すると, 応答時間の短縮は見込めるが, 接続可能なクライアント数は最大 $S/2$ となる. 一方, すべてのクライアントが 1 コネクションずつ確立すると, 応答時間は大きくなるが, 接続可能なクライアント数は最大 S となる. よって, 応答時間と接続が棄却される確率の間にはトレードオフが存在すると考えられる.

そこで, 現在接続中のコネクション総数が少ないときには新規クライアントに 2 コネクション許可し, 多ければ 1 コネクションのみ許可する受付制御を考えた. 具体的には, 接続中のコネクション総数について, しきい値 T ($0 \leq T \leq S$) を設け, 新規クライアントからのコネクション確立要求が到着した時点で, 接続中のコネクション総数が T を超えているか否かにより, そのクライアントに許可するコネクション数を決定する. 接続中のコネクション総数が, しきい値 T 未満であれば 2 コネクション許可し, しきい値 T 以上の場合は, 1 コネクションのみ許可する. したがって, すべての新規クライアントに対して 1 コネクションのみ許可する場合は $T=0$, 2 コネクション許可する場合は $T=S$ とすればよい.

提案方式を実装するのはサイト側だけで, ブラウザには何ら変更を加える必要はない. 多くのブラウザは 2 本のコネクションを確立しようとするが, このうち 1 本しか受け付けられなかった場合は, その 1 本のコネクションを使って所望のファイルをダウンロードすることになる.

以降では, サイト内のサーバとあるクライアントの間に確立された 1 本または 2 本のコネクションを介して, そのクライアントが行う一連のページ閲覧行為をセッションとよぶことにする. よって, 新規クライアントに少なくとも 1 本のコネクションが許可されると, 新たなセッションが開始される. これに対して, 現在接続中のコネクション総数が S に達しているため, 新規クライアントに 1 本のコネクションも許可されなかった場合には, そのセッションは棄却される. 提案方式は, 現在接続中のコネクション総数に応じて, 新規クライアントに許可するコネクション数を決定するが, これにより, 可能な限

Fig. 1 Evaluation model

り並列ダウンロードを有効にして応答時間の短縮を図り, かつセッション棄却率の削減を目指す.

3. 評価モデル

本章では, 提案方式の性能評価を行うために, 待ち行列システムを用いた評価モデルを構築する. 本モデルでは, コネクション受付制御の有効性のみに焦点をあてて検討するために, ネットワーク上の遅延, コネクションの確立および解放, サーバ選択に要する手間や時間は無視している.

3.1 概要

評価モデルを Fig. 1 に示す. 単一処理施設で行列長に制限のない待ち行列システムが N 台並列に並んだもので, N 台の Web サーバで構成された Web サーバ群をモデル化した. それぞれの待ち行列システムに到着する客は, クライアントが送信するファイル転送要求である. 次節で述べるように, クライアントは, 1 ページ分の閲覧に要する時間 (指数分布に従うと仮定) が経過すると, 次に閲覧するページに対応する複数のファイル転送要求を送信する. そこで, これらの要求が集団的に到着することとした. また, ファイル転送要求のサービス時間はそのファイルサイズ (パレート分布に従うと仮定) に依存する. したがって, 1 台の Web サーバに対応する待ち行列システムのケンドール表記は $M^X/G/1$ となる.

簡単のため, 各サーバの処理能力は同一とし, クライアントからのファイル転送要求は, 負荷分散専用装置 (ロードバランサ) が各サーバに振り分けるものとした. また, セッション継続中は, 当該クライアントと特定のサーバとの間でコネクションが維持されているものとした.

本モデルでは, コネクションの確立および解放に要する手間と時間を省略しているため, あるセッションの最初のファイル転送要求が到着した際に, サーバ選択が行

われることとした．その際にロードバランサは提案方式に従い，1 コネクションのみ許可するか2 コネクション許可するか決定する．そして，コネクションを確立する1 台または2 台のサーバを選択する．ただし，接続中のコネクション数がすでに最大受付コネクション数 S に達している場合には，新規セッションは棄却される．

ロードバランサのサーバ選択方式は，接続中のコネクション数が最も少ないサーバを選択する Least-Connection 方式 [4] とする．同一セッションに属するすべてのファイル転送要求は，1 コネクションのみ許可されたクライアントの場合は同一のサーバに送られ，2 コネクション許可されたクライアントの場合は2 コネクションに均等に分配され，2 台のサーバに送られるものとする．Fig. 1 の例では，クライアント1, 2 はそれぞれ1, 2 コネクション許可されている．

新規セッションは，到着率 λ [client/sec] でサイトにポアソン到着するものとする．5. で行う性能評価では，到着率 λ を変えることにより，サーバ群に加わる負荷を変化させている．

3.2 クライアントにおけるページ閲覧要求の発生

クライアントは，サイトが提供している複数のページを閲覧することが多い．すなわち，あるセッションの継続中，そのクライアントは1 ページ分の閲覧に要する時間間隔において，次に閲覧するページに対応するファイル転送要求を送出するといったことを繰り返す．一つのページは，テキストファイルや画像ファイルなど複数のファイルから構成されていることが多いため，クライアントがあるページを閲覧する際には，接続中の1 台または2 台のサーバにはそのページに対応する複数のファイル転送要求が集団的に到着することになる．

以上のことから，本研究では，あるクライアントから寄せられたファイル転送要求が特定の1 台または2 台のサーバに集団到着するトラヒックモデルを考えた．なお，パラメータは過去の文献で用いられている数値を参考にして以下に示すものとした．1 ページあたりのファイル数は5~15 の一様分布 [3]，個々のファイルサイズは平均 21.07 [KByte] のパレート分布（分布係数 $\alpha=0.8037$ ，最小ファイルサイズ 1 [KByte]，最大ファイルサイズ 10 [MByte]）[7] に従うものとする．また，1 セッションあたりの閲覧ページ数は平均 12 の指数分布に従うものとする [3]．サーバ側で前回のページ閲覧に対するファイル転送の処理が完了しているか否かにかかわらず，クライアントは平均 25 [sec] の指数分布に従う時間間隔で次のページ閲覧を開始するものとする [3]．

3.3 Web サーバにおける処理

サーバは各クライアントからのファイル転送要求をラウンドロビンで処理するものとしている．その際のスロット長は，Linux2.6.7 の最小タイムスロットである 0.01 [sec] とした．また，サーバの処理時間は参考

文献 [11] より，64 [KByte] までは一律 0.04 [sec]，以降 35.5 [KByte] 増加するごとに 0.01 [sec] ずつ増加するものとした．

4. セッション棄却率の近似値の算出法

提案方式は，新規クライアントに許可するコネクション数を2 から1 に切り替えるためのしきい値 T を設定して，セッション棄却率の低減と応答時間の短縮を図ることを目的としている．したがって，様々な条件下において，適切なしきい値を検討する必要がある．その際に，評価モデルのシミュレーションを行うことになるが，セッション棄却率については，本章で述べる手法により，近似的に算出することが可能である．この近似解析を併用することにより，しきい値の検討に要する時間を削減することができる．

4.1 近似解析モデル

セッションが継続している期間の大半は，ユーザがページを閲覧している時間によって占められている．通常，各ファイル転送要求の Web サーバでの待ち時間やサービス時間は，ページ閲覧時間（平均 25 [sec] の指数分布に従うと仮定）に比べて非常に短いためである．そこで，セッション継続時間（25 [sec]）を12回で300 [sec]）をサービス時間とする即時式システム [1] と見なして，近似的にセッション棄却率を算出することが可能であると考えられる．

即時式システムに到着する客は，新規クライアントが確立しようとするコネクションである．新規クライアントはポアソン到着し，常に2本のコネクションを確立しようとするので，即時式システムに対する客の到着は大きさ2の集団到着となる．また，ファイル転送要求の Web サーバでの待ち時間やサービス時間を無視すると，即時式システムのサービス時間となるセッション継続時間は，指数分布のたたみこみ，すなわち，ガンマ分布に従う．さらに，最大受付コネクション数 S が設定されているので，即時式システムのケンドール表記は $M^X/G/S/S$ となる．ただし，接続中のコネクション数が T 以上のときには，同時に到着した2本のコネクションのうち的一方は棄却される．また，2本許可されたコネクションのサービスは同時に終了する．

4.2 即時式システムの解析

本節では，到着率 λ ，サービス率 μ として，前節で述べた即時式システムの解析を行う．この解析により得られたセッション棄却率が，前章で述べた評価モデルのセッション棄却率の近似値となるが，近似の妥当性については次節で検証する．

前節で述べたように，本即時式システムでは，サービス時間が指数分布に従わないため，マルコフモデルの通常の解析手法をそのまま適用するわけにはいかない．にもかかわらず，ここでは，マルコフモデルの解析手法を

用いてセッション棄却率を求めることとした．その理由は，マルコフモデルの解析手法により導出されたアーラン B 式（アーランの即時式システム M/M/S/S において呼損率を表す式）はサービス時間の分布にかかわらず成立することが知られており [10]，本即時式システムはアーランの即時式システムによく似ているためである．以下，即時式システムの解析によりセッション棄却率を求める手法について述べる．

1 コネクションのみ許可された接続中のクライアント数を i ，2 コネクション許可された接続中のクライアント数を j とする．したがって，接続中のコネクション総数は $(i+2j)$ となる．なお，あるクライアントにより同時に確立された 2 本のコネクションは同時に終了する．

まず， i と j の取り得る範囲について考える． $(i+2j)$ はコネクション数の上限 S を超えることはできないので，

$$0 \leq i+2j \leq S \quad (1)$$

である．また，接続中のすべてのクライアントが 1 コネクションしか許可されていない場合もあり得るので，

$$0 \leq i \leq S \quad (2)$$

となる．ここで，接続中の全クライアントが 2 本ずつコネクションを確立しているとする．このとき， $2j < T$ であれば，新規クライアントには 2 本のコネクションが提供されるので， $2j$ はたかだか $(T+1)$ となる．ただし，コネクション数の上限 S を超えることはできない．したがって，

$$0 \leq 2j \leq \min[T+1, S] \quad (3)$$

となる． i と j の取り得る範囲は (1)~(3) 式が同時に満足される範囲である．以降，状態 $(i, 2j)$ の定常状態確率を $p_{i,2j}$ として大域平衡方程式を立てるが， $i, 2j$ が範囲外の場合は $p_{i,2j} = 0$ と見なすものとする．

状態 $(i, 2j)$ への流入確率フロー速度について考える．まず，コネクション終了に伴う状態 $(i, 2j)$ への流入は次の二つが考えられる．状態 $(i+1, 2j)$ において単独で確立されたコネクションのうちのひとつが終了すると状態が $(i, 2j)$ に遷移するので，このときの流入確率フロー速度は $(i+1)\mu p_{i+1,2j}$ となる．また，状態 $(i, 2(j+1))$ において 2 本同時に確立されたコネクションが終了したときの状態 $(i, 2j)$ への流入確率フロー速度は $(j+1)\mu p_{i,2(j+1)}$ である．これに対して，コネクション確立に伴う流入については，次のように場合分けして考える必要がある． $i+2j < T+1$ のときは，新規クライアントに対して 2 本のコネクションが提供されるため，状態 $(i, 2(j-1))$ からの遷移となり，このときの流入確率フロー速度は $\lambda p_{i,2(j-1)}$ である． $i+2j = T+1$ のときは，2 コネクション同時確立に加えて単独コネクション確立も考えられるため，流入確率フロー速度は $(\lambda p_{i,2(j-1)} + \lambda p_{i-1,2j})$ と

なる． $T+1 < i+2j$ のときは，すでに接続数がしきい値を超えているため，単独コネクションの確立しかないので，流入確率フロー速度は $\lambda p_{i-1,2j}$ である．以上をまとめると，状態 $(i, 2j)$ への流入確率フロー速度は次式のように表される．

$$\begin{cases} (i+1)\mu p_{i+1,2j} + (j+1)\mu p_{i,2(j+1)} + \lambda p_{i,2(j-1)} & (i+2j < T+1) \\ (i+1)\mu p_{i+1,2j} + (j+1)\mu p_{i,2(j+1)} + \lambda p_{i-1,2j} & (i+2j = T+1) \\ (i+1)\mu p_{i+1,2j} + (j+1)\mu p_{i,2(j+1)} + \lambda p_{i-1,2j} & (T+1 < i+2j) \end{cases} \quad (4)$$

一方，状態 $(i, 2j)$ からの流出確率フロー速度については，単独コネクションが終了するとき $i\mu p_{i,2j}$ ，2 本同時に確立されたコネクションが終了するとき $j\mu p_{i,2j}$ である．また， $i+2j < S$ のときは新規コネクションの確立に伴う他の状態への遷移があり得るが， $i+2j = S$ のときは新規コネクション確立要求は棄却される．したがって，状態 $(i, 2j)$ からの流出確率フロー速度は次式になる．

$$\begin{cases} (i\mu + j\mu + \lambda)p_{i,2j} & (i+2j < S) \\ (i\mu + j\mu)p_{i,2j} & (i+2j = S) \end{cases} \quad (5)$$

(1)~(3) 式で規定された範囲で，すべての $(i, 2j)$ の組合せについて，(4) 式と (5) 式が等しいとおいた方程式と全確率の法則に基づく

$$\sum_i \sum_j p_{i,2j} = 1 \quad (6)$$

から構成される連立方程式が大域平衡方程式である．これを解くと，各々の定常状態確率 $p_{i,2j}$ を求めることができる．所望のセッション棄却率 p_B は次式のように与えられる．

$$p_B = \sum_{i+2j=S} p_{i,2j} \quad (7)$$

なお，しきい値 $T=0$ の場合は，常に $j=0$ であり，接続中のコネクション数とセッション数は等しくなる．したがって，この場合 p_B はアーラン B 式と等価となる．

4.3 セッション棄却率の近似の妥当性

3. で示した評価モデルにおけるセッション棄却率を近似的に算出する手法として，前節では，セッション時間をサービス時間とする即時式システムの解析を行った．評価モデルではネットワーク上の遅延，コネクションの確立および解放などに要する時間を無視しているため，セッション時間とは，ユーザのページ閲覧時間と Web サーバにおける応答時間の和となる．ここで，応答時間とは Web サーバでの処理待ち時間と処理時間の和であり，具体的には，閲覧ページに対応するファイル転送要

求が1台または2台のサーバに集団到着してから、それらすべての処理がサーバ側で完了するまでに要する時間である。通常は、応答時間に比べてページ閲覧時間が非常に大きいため、応答時間がセッション時間に及ぼす影響はほとんどない。しかし、サーバ群に加わる負荷が大きくなり、応答時間が増大すると、その影響は無視できなくなる。そこで、応答時間の増大によりセッション棄却率の近似の妥当性が保たれなくなる場合について検討することとした。

評価モデルでは、許可されたコネクション数にかかわらず、各クライアントはセッション中に平均12ページ閲覧するものとしている。したがって、1コネクションのみ許可されたクライアントと2コネクション許可されたクライアントでは、前者の方がコネクションあたりサーバ群に加える負荷が大きくなる。このことから、すべてのクライアントが1コネクションしか許可されないとき、すなわち、しきい値 $T=0$ のとき、他のしきい値のときよりも、接続中の全クライアントがサーバ群に加える負荷が大きくなるため、応答時間が増大する可能性が最も高くなる。後の5.2で示すように、新規クライアントの到着率が与えられると T が小さくなるほど応答時間が大きくなるので、以下、 $T=0$ の場合について近似の妥当性を検証する。なお、 $T>0$ の場合には、本節で示すものよりも近似の誤差は小さくなるため、妥当性の検証を省略するが、確認のため、次章で行う性能評価でセッション棄却率のグラフを示す際にシミュレーション値と近似解析値を併記している。

まず、アーラン B 式による解析結果を Fig. 2 に示す。ここでは、最大受付コネクション数 $S=500$ としている。セッション時間が増加するとセッション棄却率も増加するが、その影響は到着率が小さいほど大きくなっている。Fig. 2 で示した数値例では、サーバあたりの到着率 $\lambda/N=1.5$ [client/sec] のとき、セッション時間のわずかな増加でも、セッション棄却率に大きな影響を及ぼすことがわかる。しかし、到着率が小さいときには応答時間も小さいため、セッション時間はほとんど増加しない。本節の数値例では、後の Fig. 5 に示されるように、 λ/N が 1.7 [client/sec] を超えたあたりからセッション時間への影響が出はじめており、そのときにはセッション時間の増加がセッション棄却率の増加に及ぼす影響もやや小さくなっているため、比較的広い範囲で近似の妥当性が保たれることが期待できる。

以降で示すグラフでは、即時式システムの解析結果をライン、評価モデルのシミュレーション結果をポイントで示す。Figs. 3~5 にそれぞれ、セッション棄却率、平均応答時間、平均セッション時間とサーバあたりの到着率 λ/N の関係を示す。ここでは、サーバ数 $N=10$ 、しきい値 $T=0$ としている。Fig. 3 は評価モデルのシミュレーション値と即時式システムの解析値を比較しているが、この例では、最大受付コネクション数 $S=600$ のときに

Fig. 2 Session loss probability calculated from Erlang's B formula

Fig. 3 Comparison of the simulation result and the analytical one

近似の妥当性が保たれなくなっている。最大受付コネクション数 S を大きくすると、セッション棄却率は小さくなるが、到着率の増加に伴ってサーバ群に加わる負荷が高くなる。Fig. 4 に示されているように、最大受付コネクション数 $S=600$ の場合には、 λ/N が 1.6 [client/sec] を超えたあたりから応答時間が急激に増加している。これに対して、最大受付コネクション数 S が 550 以下の場合には、応答時間の急激な増加が抑えられている。Fig. 5 に示された平均セッション時間を見ると、最大受付コネクション数 $S=600$ の場合のみ、セッション時間が大きく増加していることがわかる。

以上の結果から、到着率が増加したときに、応答時間が大きく増加すると、近似の妥当性は保たれなくなる。しかし、実際のシステムでは、最大受付コネクション数を設定して、応答時間の急激な増大を防止する対策が採られているため、実用の範囲内においては、本章で検討したセッション棄却率算出法が有効に機能するものと考えられる。

Fig. 4 Mean response time via simulation of the evaluation model

Fig. 5 Mean session time via simulation of the evaluation model

5. 性能評価

本章では、セッション棄却率や応答時間の特性を調べて提案方式の有効性を明らかにするとともに、しきい値の決定法について検討する。

5.1 セッション棄却率の評価

まず、現行のシステム、すなわち、すべてのクライアントに対して2コネクションずつ受け付ける場合 ($T=S$) とすべてのクライアントに対し1コネクションのみ許可する場合 ($T=0$) のセッション棄却率の比較を Fig. 6 に示す。ここでは、最大受付コネクション数 $S=500$ とした。結果から、 $T=0$ の場合に比べて、 $T=S$ の場合に非常に大きなセッション棄却率を示している。これは、 $T=S$ のときに同時接続可能なクライアント数が実質的に $S/2$ となるためである。しかし、その一方で、 $T=S$ の場合には、ファイルの並列ダウンロードによる応答時間の減少が期待できる。そこで、接続中のコネクション総数が少ないときには、1クライアントあたり2コネクションを許可して、セッション棄却率特性を可能な限り Fig. 6 の $T=0$ の場合に近づけながら、小さな応答時間を実現することが、提案方式の目的である。

Fig. 6 Comparison of the session loss probability between the cases $T=0$ and $T=S$

Fig. 7 Influence of the threshold T on the session loss probability

サーバ数 $N=10$ ，最大受付コネクション数 $S=500$ で、しきい値 T を $0 \sim 500$ と変化させた場合のセッション棄却率を Fig. 7 に示す。図には、サーバあたりの到着率 $\lambda/N=1.5, 1.6, 1.7$ [client/sec] の場合を示している。しきい値 T が増加してもしばらくはほぼ一定のセッション棄却率を示しているが、最大受付コネクション数の付近では急激にセッション棄却率が増加している。しきい値 T の増加により、2コネクション許可されるクライアントの割合が増加し、接続中のコネクション総数は増加するが、(7) 式で与えられるセッション棄却率 p_B ，すなわち、 $i+2j=S$ となる状態確率の和に影響を及ぼすほどのものではないと考えられる。しかし、しきい値 T がさらに増加して最大受付コネクション数 S に近づくと、大半のクライアントは2コネクション許可されており、接続中のコネクション総数が大きくなっており、セッション棄却率は増大する。したがって、提案方式によりセッション棄却率を小さくするためには、想定される到着率ごとに $T=0$ の場合とほぼ同等のセッション棄却率を示す T の上限を見だし、その値以下にしきい値を設定しておけばよいことがわかる。たとえば、Fig. 7 でサーバ

あたりの到着率 $\lambda/N = 1.5$ [client/sec] の場合には 400 以下のしきい値を設定しておけばよいことになる．この際に 4. で示した解析を行うと，シミュレーションを実行した場合に比べて，検討時間を大幅に削減することができる．

つぎに，最大受付コネクション数がセッション棄却率に及ぼす影響について検討する．サーバ数 $N = 10$ として，最大受付コネクション数 $S = 450, 500, 550$ とした場合のセッション棄却率を Fig. 8 に示す．比較のため，グラフの横軸を各々の最大受付コネクション数で正規化されたしきい値 T/S とし，いずれもサーバあたりの到着率 $\lambda/N = 1.6$ [client/sec] としている．最大受付コネクション数 S が増加すると，より多くの新規セッションが受け付けられるようになるため，セッション棄却率は減少する．また，最大受付コネクション数 S が多いほど，セッション棄却率が急激に上昇を始めるしきい値が小さくなっている．このことは，最大受付コネクション数 S の増加に伴うセッション棄却率の減少により，システムに受け入れられる実質的な到着率が大きくなるため，早めにしきい値制御を行う必要があることを示している．

本節の最後に，サーバ数 N がセッション棄却率に及ぼす影響について検討する．サーバあたりの最大受付コネクション数 $S/N = 50$ と固定して，サーバ数 $N = 5, 10, 15$ とした場合のセッション棄却率を Fig. 9 に示す．グラフの横軸は各々の最大受付コネクション数で正規化されたしきい値 T/S とし，いずれもサーバあたりの到着率 $\lambda/N = 1.6$ [client/sec] としている．結果から，サーバ数 N が多くなると全般的にセッション棄却率は小さくなっているが，これは，複数処理施設の待ち行列システムで到着客を空き処理施設に順次割り振るようなサービスを行っている場合において，処理施設あたりの到着率が等しければ，処理施設数が多くなるほど利用率が向上する大群化効果 [1] によるものと考えられる．上述の Fig. 8 の場合と比較して対照的なことは，サーバ数 N が多くセッション棄却率が小さくなるほど，セッション棄却率が急激に上昇を始めるしきい値がわずかに大きくなっている点である．ここでは，サーバあたりの最大受付コネクション数が固定されているため，サーバ数 N が増加しても，セッション棄却率が大幅に減少しているわけではない．むしろ，サーバ数 N が増加することによって，新規セッション到着時点で最小の接続数となっていて割当てを受けるべきサーバの候補が増えたことによる影響の方が大きくなっているものと考えられる．

5.2 応答時間の評価

しきい値 T の設定によるセッション棄却率の制御については前節の議論のとおりであるが，平均応答時間については，しきい値 T を大きくするほど小さくなる．これは，しきい値 T が大きくなるほど 2 コネクション許可されるクライアントの割合が増加して，並列ダウンロード

Fig. 8 Influence of the maximum number of connections S on the session loss probability

Fig. 9 Influence of the number of servers N on the session loss probability

の効果が大きくなるためである．そこで，本節では，しきい値 T が平均応答時間に及ぼす影響について検討する．

Fig. 10 に，Fig. 7 と同様の条件下でサーバあたりの到着率 λ/N [client/sec] を変化させたときの平均応答時間を示す．Fig. 10 では結果の表示が重なって判別が付きにくくなっているところがあるため，Fig. 11 にサーバあたりの到着率 λ/N が 1 [client/sec] 以下の場合の拡大図を示している．しきい値 T が 400 以下の場合では，到着率が小さいときには小さな応答時間が得られているが，到着率が大きくなるにつれて， $T = 0$ の場合と同様の応答時間を示すようになる．これに対して，しきい値 T が 400 を超えると接続を許可されるクライアント数が制限されることによって，到着率が増加しても応答時間を小さく抑えることが可能となるが，このとき，Fig. 7 で示したように，セッション棄却率は増大している．

以上で検討したように，到着率，セッション棄却率，応答時間の許容範囲を想定して，しきい値 T を設定する必要がある．たとえば，サーバあたりの到着率 λ/N が 1 [client/sec] 以下の比較的軽い負荷を想定すればよいのであれば，しきい値 $T = 400$ とすることで，セッション棄

Fig. 10 Influence of the threshold T on the mean response time

Fig. 11 Enlarged drawing of Fig. 10 under $\lambda/N \leq 1.0$ [client/sec]

却率と応答時間の両方を小さく抑えることが可能となる。その一方で、たとえば、高負荷時においても応答時間を3秒程度に抑えたいのであれば、最大受付コネクション数 $S = 500$ とすることで、応答時間についての目的は達成されるので (Fig. 4 参照)、あとは所望のセッション棄却率を実現するしきい値 T について検討すればよいことになる。

6. あとがき

本論文では、Web サーバシステムにおけるコネクション受付制御方式を提案した。本方式は、新規クライアントが到着した時点で、接続中のコネクション総数がしきい値を超えているか否かにより、そのクライアントに対して許可するコネクション数を決定する。これにより、セッション棄却率低減と応答時間短縮の双方の実現を目指すものである。本方式を運用するにあたり、適切なしきい値を設定する必要があるが、その際の検討時間を短縮するために、しきい値が与えられた場合にセッション棄却率を近似的に算出する手法についても示した。

性能評価の結果、しきい値を最大受付コネクション数より少し小さい値以下に設定することで、セッション棄

却率を大幅に低減することが可能となることが示された。したがって、しきい値を最大受付コネクション数より少し小さい値に設定することにより、HTTP1.1 の標準である2コネクション確立が出来る限り許可され、応答時間を短縮することが可能となる。今後の課題としては、クライアントがさらに多くのコネクション確立を要求した場合の評価などが挙げられる。

参考文献

- [1] 秋丸, 川島: 情報通信トラヒック—基礎と応用—, 電気通信協会, オーム社 (1990)
- [2] T. Bourke: *Server Load Balancing*, O'Reilly (2001)
- [3] V. Cardellini, M. Colajanni and P. S. Yu: Redirection algorithms for load sharing in distributed web-server systems; *Proc. ICDCS '99*, pp. 528–535 (1999)
- [4] V. Cardellini, E. Casalicchio, M. Colajanni and P. S. Yu: The state of the art in locally distributed web-server systems; *IBM Research Report*, RC22209 (2001)
- [5] R. S. Engelschall: *Apache Desktop Reference*, Addison Wesley Longman (2001)
- [6] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach and T. Berners-Lee: Hypertext transfer protocol – HTTP/1.1; *RFC*, 2616 (1999)
- [7] Z. Genova and K. J. Christensen: Challenges in URL switching for implementing globally distributed web sites; *Proc. ICPP'00-Workshop*, pp. 89–94 (2000)
- [8] C. Henderson: *Building Scalable Web Sites*, O'Reilly (2006)
- [9] A. Luotonen: *Web Proxy Servers*, Prentice-Hall (1997)
- [10] 森村, 大前: 応用待ち行列理論, OR ライブラリー, 日科技連 (1975)
- [11] 佐竹, 稲井: Web サーバクラスタにおけるリクエストファイルサイズを利用したサーバ負荷推定の効果; 電気学会論文誌 C, Vol. 128, No. 10, pp. 1543–1549 (2008)
- [12] S. Souders: *High Performance Web Sites, Essential Knowledge for Frontend Engineers*, O'Reilly (2007)

著者略歴

さ たけ しん すけ
佐 竹 伸 介

1969年10月18日生。1988年岡山県立玉野光南高等学校情報電子科卒業。同年(株)メイトック入社, 1993年岡山電業(株)入社。2003年社会人学生として岡山県立大学大学院情報系工学研究科入学, 2005年同博士前期課程修了, 2008年同博士後期課程修了。2009年(株)アレス代表取締役, 現在に至る。情報通信システムの設計開発に従事。博士(工学)。電子情報通信学会会員。

いな い
稲 井 ひろし
寛 (正会員)

1961 年 11 月 17 日生．1985 年大阪大学基礎工学部情報工学科卒業，1987 年同大学院博士前期課程修了．1988 年神戸大学総合情報処理センター助手，1993 年岡山県立大学情報工学部情報通信工学科助教授，2000 年同教授，現在に至る．主とし

て計算機および通信システムの性能評価，通信プロトコル，広帯域通信網に関する研究に従事．工学博士．ACM，IEEE，電気学会，電子情報通信学会，情報処理学会の各会員．

あら い
荒 井 つよし
剛

1974 年 7 月 25 日生．1997 年茨城大学工学部情報工学科卒業，1999 年同大学院博士前期課程修了，2004 年同博士後期課程修了．同年岡山県立大学情報工学部情報通信工学科助手，現在，同助教．スペクトル拡散通信，無線パケット通信方式，通信

プロトコルに関する研究に従事．博士（工学）．IEEE，電子情報通信学会の各会員．

さい とう
齊 藤 とも や
智 也

1977 年生．1998 年津山工業高等専門学校情報工学科卒業，2000 年同専攻科修了，2002 年電気通信大学大学院情報システム学研究科博士前期課程修了，2005 年岡山県立大学大学院情報系工学研究科博士後期課程修了．同年津山高専情報工学科助手，

2008 年島根大学総合理工学部教務職員，現在に至る．主として通信システムの性能評価に関する研究に従事．博士（工学）．IEEE，電子情報通信学会の各会員．