

一時的アクセス権を用いたインターネット予約システムにおけるクライアントのスケジューリング

Scheduling of Clients in Internet Reservation System Using Temporary Access Right

三上 烈史[†] 吉田 祥真[†] 小林 良太郎[†] 加藤 雅彦[‡] 金岡 晃[‡]
 Tsuyoshi Mikami Shoma Yoshida Ryotaro Kobayashi Masahiko Kato Akira Kanaoka

1. はじめに

近年、ネットワーク技術の急速な発展により、Web サービスは社会で必要不可欠なものとなった。それと同時に、Web サービスが停止した際の社会的影響も非常に大きくなった。Web サービスが停止するケースとして、例えば以下が挙げられる [1]。

1. 人気商品の予約開始に伴う一斉アクセス

人気の高い新製品やチケット予約の開始時刻が予め告示されていた場合、この時刻にサービスを求める大量のクライアントからアクセスが殺到する。この結果、サービスの停止や応答速度の低下を招く。

2. マスメディアでの紹介に伴う一斉アクセス

オンラインショッピングサイトなどの場合、マスメディアで紹介されることで、大量のアクセスが同時に発生し、サービス停止の状態に陥る場合がある。

3. DoS 攻撃

悪意を持って、サーバに過剰な負荷を与えたり、サーバなどの脆弱性を悪用することで、サービスの運用や提供を妨げる。

本研究では、1 と 2 のようなケースを取り上げる。大量のクライアントが一斉にアクセスするため、Web サーバの負荷が増大する。その結果、クライアントはサービスを受けることが出来なくなる。

一般に、チケット予約などのサービスでは、複数の画面遷移を用いて、個人情報や支払い方法の入力を行う (以下、予約手続きと呼ぶ)。例えば、個人情報を入力する画面、支払い方法を入力する画面などを決められた順序で遷移しつつ、情報を入力していく。情報を入力している途中でアクセス不能になると、画面遷移はリセットされ、始めから入力しなおす必要がある。予約手続きを行なっている間は、クライアントは確実にアクセスできる状態にする必要がある。

そこで本研究では、インターネット予約システムにおいて一斉アクセスが発生した状況でも、予約手続きを保証する手法を提案する。

以下、第 2 章では従来の対策と関連研究を、第 3 章では提案手法を、第 4 章では予約手続きの成功率の理論式導出を、第 5 章では本論文のまとめを述べる。

2. 従来の対策と関連研究

一斉アクセスへの一般的な対処手法として、ロードバランサによる負荷分散や、事前のサーバ増強などが挙げられる [1]。ロードバランサとは、Web サーバなどに対するコネクションを、複数のサーバに振り分ける装置のことである。各 Web サーバの負荷に応じて、振り分け先を決定することが可能なため、Web サーバ 1 台あたりの負荷を低減することが可能となる。しかし、負荷分散だけでは、予約手続きの途中でアクセス不能になるリスクを回避できない。ロードバランサは、各クライアントが閲覧している画面遷移を考慮した優先制御は行わない。そのため、サーバの負荷が大きくなると、予約手続きの途中でアクセスが不能になる可能性がある。一方、サーバの処理能力を向上させる場合、一斉アクセスが発生していないときに余剰能力を持つことになり、システムのコストパフォーマンスが悪くなる。

近年では、クラウドサービスが急速に普及しており、インターネット予約システムをクラウドに移行する手段も考えられる [2]。クラウドサービスでは、必要なリソースを随時変更することができるため、アクセスが集中する時間だけリソースを多く借りることで、一斉アクセスに耐える手法が考えられる。しかし、リソースを多く借りるほど、コストがかかってしまう問題がある。一斉アクセスの想定次第では、膨大なリソースが必要になり、コストが跳ね上がってしまう。

加地らは、次回アクセス保証機能と呼ばれる機能を提案している [3]。初回アクセスを拒否されたユーザに対して、次回いつになればアクセスが可能かを保証する機能を提案している。これにより、満足なサービスを受けられなくなったユーザの不満を解消する手法を提案している。しかし、サーバにアクセスできるクライアント数が、厳しく制限されてしまうという問題がある。

[†] 豊橋技術科学大学大学院工学研究科
 Graduate School of Engineering, Toyohashi University
 of Technology

[‡] 筑波大学大学院システム情報工学研究科
 Graduate School of Systems and Information Engineering,
 Tsukuba University of Technology

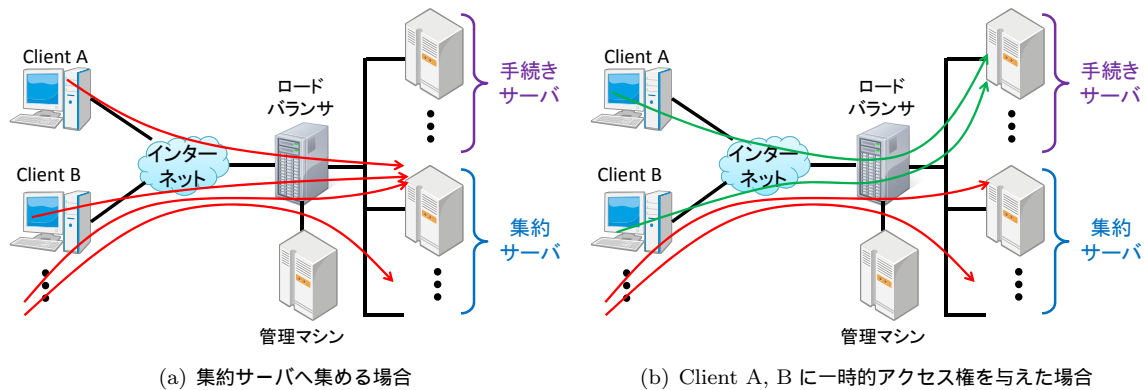


図 1 基本アイデア

3. 提案手法

3.1 基本アイデア

本研究では、インターネット予約システムにおいて一斉アクセスが発生した状況でも、予約手続きを保証する手法を提案する。本手法では、予約手続きを行おうとするクライアントに対し、一時的に低負荷なサーバへアクセスできる権利（一時的アクセス権）を与える。そして、一時的アクセス権を得たクライアントのアクセスを保証することで、手続き中にアクセスが不能になる事態を回避する。

基本アイデアを図 1 に示す。本提案を実現するために、集約サーバ、手続きサーバ、管理サーバを設ける。集約サーバは、一斉アクセスしてきたクライアントを集約する役割がある。手続きサーバは、一時的アクセス権を得たクライアントに、予約手続きを行なわせる役割がある。管理サーバは、集約サーバと手続きサーバのログの管理や、一時アクセス権を得たクライアントの振り分けを行なう役割がある。まず、図 1(a) のようにクライアントを集約サーバへ集める。普段、集約サーバは Web サービスを提供するが、同時に、クライアントからのアクセスを集約する役割を持たせる。そのため、一斉アクセスが発生している状況では、レスポンスは低下する。次に、一定時間 τ 秒ごとに、管理サーバによって、集約サーバへ集められたクライアントの中から、一時的アクセス権を与えるクライアントを決定する。一時的アクセス権を与えるクライアント数は、手続きサーバのスペックに応じて事前に決定する。一時的アクセス権を得たクライアントは、管理サーバによって、図 1(b) のように手続きサーバへ NAT され、予約手続きを行なうことができるようになる。なお、一時的アクセス権の有効期間は、 τ 秒とする。

次節では、一時的アクセス権の与え方について、詳細に述べる。

3.2 一時的アクセス権の与え方

集約サーバに集められたクライアント群から、一時的アクセス権を与えるクライアントを選択する方法を述べる。

まず、クライアントを保持する待ち行列を用意する。この待ち行列は FIFO とし、待ち行列から取り出したクライアントに、一時的アクセス権を与える。

集約サーバ上で、時刻 t を起点として τ 秒間のアクセスログを取得する。取得したアクセスログは、管理サーバへ送られる。管理サーバは、アクセスログの内容を元に、 τ 秒間においてアクセスしてきた時刻順に、クライアントを待ち行列に挿入する。なお、 τ 秒間で、あるクライアントが複数回アクセスしてきた場合は、もっとも早い時刻を、そのクライアントがアクセスしてきた時刻とする。そして、待ち行列の先頭から、事前に決定した数だけクライアントを取り出し、一時的アクセス権を与える。一時的アクセス権を与えられたクライアントは、手続きサーバ上へ NAT される。一時的アクセス権の有効期間は τ 秒間であり、クライアントが予約手続きを終了するかどうかには依存しない。つまり、予約手続きを終了した場合でも、有効期間中は、手続きサーバ上へ NAT される。また、予約手続きを終えることが出来なかったとしても、有効期間が経過すれば、一時的アクセス権を失う。

時刻 $t + \tau$ 以降においても、上記と同様にして、一時的アクセス権を与える。ただし、管理サーバは、アクセスログの内容に基づいて、クライアントを待ち行列に入れた後に、以下を行う：時刻 t から $t + \tau$ 秒間において、手続きサーバ上で予約手続きを終えることが出来なかったクライアントを、待ち行列に挿入する。

4. 予約手続きの成功率

本章では、提案手法の性能を見積もるために、理論式の導出を行う。具体的には、クライアントごとの予約手続きの成功率を導出する。ここで、予約手続きの成功率

とは、アクセス不能になること無く、予約手続きに必要な全ての画面遷移が完了する確率をさす。例えば、個人情報入力画面、支払い方法入力画面、確認画面、完了画面を順に遷移しながら、必要な情報を入力していくことで、予約手続きを遂行していく場合を考える。この場合、これら 4 画面を、連続で閲覧できる確率を予約手続きの成功率とする。

4.1 仮定

本研究では、ロードバランサを介したシステムをベースとし、以下の仮定を設ける。

- M_a 台の Web サーバと、1 台のロードバランサから構成されている
- 各 Web サーバの負荷は、ロードバランサにより常に均等に分散されている
- サーバの性能は全て同じである

上記のベースに対し、提案手法を適用する。

本手法を厳密に評価するには、クライアントのアクセス量、サービスにかかる時間、サーバの性能など、一意に決定する事が困難なパラメータが多く存在する。そこで、本稿では、複数の仮定を設けた上で理論式の導出を行う。

まず、クライアントに関する仮定は以下の通りとする。

- τ 秒間あたり λ 人のクライアントがアクセスし続ける
- このうち、最初の λ 人が評価対象である
- λ は集約サーバを高負荷にするのに十分大きい値である
- 予約手続きが成功したクライアントは、以後アクセスしない

次に、Web サーバに関する仮定は以下の通りとする。

- 手続きサーバも集約サーバもベースの手法と同じ性能を持つ
- 各サーバが受け入れ可能なクライアント数は既知で、 C_r 人である ($C_r < \lambda$)

4.2 提案手法を適用しない場合

提案手法を適用しない場合の予約手続きの成功率を求める。あるページを受け取ることのできる確率は、全クライアントに対するサーバの受け入れ可能なクライアント数の割合となる。クライアント数は λ 人、サーバの受け入れ可能な数は $C_r \times M_a$ 人である。よってあるページを受け取ることのできる確率は、 $C_r \times M_a / \lambda$ となる。

予約手続きに必要な Web ページ数を N ページとすると、予約手続きを成功させるには、 N ページを連続で閲覧できる必要がある。よって、クライアントごとの予約

手続きの成功率 P_{conv} は、式 (1) のようになる。

$$P_{\text{conv}} = \left(\frac{C_r M_a}{\lambda} \right)^N \quad (1)$$

4.3 提案手法を適用する場合

次に、提案手法を適用する場合の成功率を考える。本手法は、手続きサーバと集約サーバの 2 種類のサーバで構成されている。このうち、手続きサーバの台数を $M_r (< M_a)$ とする。各サーバの役割上、サーバの負荷は異なるため、各 Web ページを受け取ることができ確率も異なる。手続きサーバにアクセスする場合に、各 Web ページを受け取ることのできる確率を p_r 、集約サーバにアクセスする場合に、各 Web ページを受け取ることのできる確率を p_c とする。予約手続きの成功率 P_{prop} は、以下のようにして計算することができる。

$$P_{\text{prop}} = \text{手続きサーバにアクセスする確率} \times p_r^N + \text{集約サーバにアクセスする確率} \times p_c^N \quad (2)$$

次に、手続きサーバにアクセスする確率を求める。手続きサーバにアクセスする確率は、集約サーバにアクセスするクライアントのうち、一時的アクセス権を与えられる $C_r M_r$ 人に選ばれる確率となる。このとき、一時的アクセス権を与えるクライアントを選出する度に、集約サーバにアクセスするクライアント数は $C_r M_r$ 人ずつ減少するとみなすことができる。時刻 t に訪れた λ 人のうち、1 回目に一時的アクセス権を与えられる確率は、

$$\frac{C_r M_r}{\lambda} \quad (3)$$

となる。1 回目の一時的アクセス権を与える処理によって、集約サーバにアクセスするクライアントは $\lambda - C_r M_r$ 人となる。2 回目に一時的アクセス権が与えられる確率は、1 回目に一時的アクセス権が与えられず、かつ 2 回目に与えられる確率なので、

$$\left(1 - \frac{C_r M_r}{\lambda} \right) \frac{C_r M_r}{\lambda - C_r M_r} \quad (4)$$

となる。同様に繰り返していくと j 番目に一時的アクセス権が与えられる確率は

$$\begin{aligned} & \left[\prod_{i=1}^{j-1} \left\{ 1 - \frac{C_r M_r}{\lambda - (i-1)C_r M_r} \right\} \right] \frac{C_r M_r}{\lambda - (j-1)C_r M_r} \\ & = \frac{C_r M_r}{\lambda} \end{aligned} \quad (5)$$

となる。厳密には、時刻 t 以降もクライアントがアクセスするため、集約サーバには λ 人以上のクライアントがアクセスしている。しかし、時刻 t でアクセスしてきた λ 人のクライアントが、待ち行列に残されている。時刻

t 以降にアクセスしてきたクライアントは、最初の λ 人のクライアントの、さらに後ろに挿入される。待ち行列の先頭のクライアントから順に一時的アクセス権を与えるため、時刻 t 以降にアクセスしてきたクライアントに一時的アクセス権が与えられるのは、最初の λ 人に一時的アクセス権を与え終えた後となる。そのため、時刻 t 以降にアクセスしてきたクライアントは、無視して導出することができる。

同様にして、集約サーバにアクセスする確率も考えると、 j 番目に一時的アクセス権が与えられるまで、集約サーバにアクセスする確率は

$$\prod_{i=1}^{j-1} \left\{ 1 - \frac{C_r M_r}{\lambda - (i-1)C_r M_r} \right\} = \frac{\lambda - (j-1)C_r M_r}{\lambda} \quad (j \geq 2) \quad (6)$$

となる。なお、 $j = 1$ の場合は、最初から一時的アクセス権を与えられているため、確率は 0 となる。

式 (2), (5), (6) より、 j 番目に一時的アクセス権が与えられることで、予約手続きが成功する確率は式 (7) のようになる。

$$P_{\text{prop}j} = \begin{cases} \frac{C_r M_r}{\lambda} p_r^N & (j = 1) \\ \frac{C_r M_r}{\lambda} p_r^N + \frac{\lambda - (j-1)C_r M_r}{\lambda} p_c^N & (j \geq 2) \end{cases} \quad (7)$$

式 (7) より、提案手法を適用すれば、 $\frac{C_r M_r}{\lambda} p_r^N$ 以上の確率で、予約手続きが成功できる。

4.4 成功率の比較

提案手法を適用する場合と適用しない場合、それぞれの予約手続きの成功率を比較する。図 2 は、提案手法を適用する場合と適用しない場合の成功率を、グラフ化したものである。 $\lambda = 10000$, $M_a = 15$, $M_r = 5$ と固定して、 C_r を変化させている。提案手法については、式 (7) において $j = 1$ のときの成功率を比較対象とする。また、 p_r が既知であるとして、 $p_r = 0.9, 0.95, 0.99$ の場合をそれぞれ示している。図より、サーバ 1 台当たりの受け入れ可能なクライアント数が少ない間は、常に、提案手法のほうが高くなる。その後、徐々に差分が小さくなり、ベースの成功率のほうが高くなる。以上より、本提案手法を導入することによって、サーバの増強をすることなく、予約手続きの成功率を上げることができる。

5. まとめ

本研究では、インターネット予約システムにおいて一斉アクセスが発生した状況でも、予約手続きを保証する

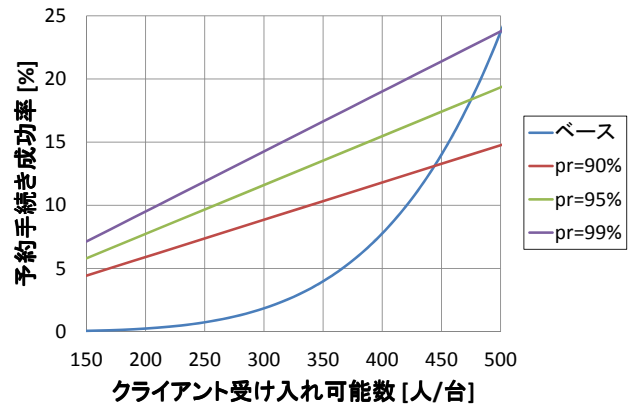


図 2 予約手続き成功率

手法を提案した。本手法では、予約手続きを行おうとするクライアントに対し、一時的アクセス権を与える。そして、一時的アクセス権を得たクライアントのアクセスを保証することで、手続き中にアクセスが不能になる事態を回避する。

また、予約手続きが成功する確率を理論式として導出し、考察した。その結果、本提案手法を導入することによって、サーバの増強をすることなく、予約手続きの成功率を上げることができる事を示した。

今後の予定として、3 つのことを検討している。1 つ目は、予約手続きにかかる時間を理論式として導出することである。本提案は、予約手続きを保証することを目的としているが、本手法によって、クライアント当たりの予約手続きにかかる時間が、短縮されている可能性がある。2 つ目は、仮定を充実させる事である。実際の Web サービスを想定した場合、待ち行列理論による Web サービスモデルの表現が有効であると考えられる。3 つ目は、評価実験である。実際に Web サーバやクライアントを設けて、予約手続きの成功率や処理時間を評価する事を検討している。また、これらの評価結果を元にすれば、より現実の環境を反映した理論式の導出が可能になると考えられる。

参考文献

- [1] 独立行政法人情報処理推進機構, "「サービス妨害攻撃の対策等調査」報告書," <http://www.ipa.go.jp/security/fy22/reports/isec-dos/index.html>
- [2] 経済産業省, "「クラウドコンピューティングと日本の競争力に関する研究会」," <http://www.meti.go.jp/press/20100816001/20100816001-3.pdf>
- [3] 加地智彦, 最所圭三, "NAP-Web における次回アクセスの効率化," 電子情報通信学会論文誌, Vol.J92-B, No.10, pp.1600-1610, 2009.