

## NAP-Web の次回アクセス補助機能における 次回アクセス待ちユーザのグループ化

加地 智彦<sup>†</sup> 最所 圭三<sup>‡</sup>

香川大学工学研究科<sup>†</sup> 香川大学工学部<sup>‡</sup>

### 1. はじめに

インターネットの高速化と普及は、Web 上で提供される様々なサービスを高度化させ、その結果、Web サービスを提供する Web サーバの負担は増している。サーバにアクセスが集中すると、十分なサービスが提供できない過負荷状態に陥る。過負荷状態に陥らないようにするために、主に利用されている技術として、負荷分散技術がある。しかしながら、必要な計算機資源の見積もりが困難であり、この見積もりに失敗すると、Web サーバの過負荷状態を解決できない。

我々は、アクセス集中による Web サーバの過負荷問題に対して、クライアントに対してチケットを配布し、次回アクセスを保証することで、ユーザの不満を抑えながらサービスを継続できる Web システム「NAP-Web」を開発している[1]。

NAP-Web はアクセスが集中した際、一部のアクセスを拒絶する。アクセスを拒絶されたクライアントに対しては次回アクセスのためのチケットを cookie の形で配布する。このチケットにはチケット開始時刻と終了時刻が書かれており、開始時刻から終了時刻までにクライアントがアクセスするとサーバは優先的に受け付ける。チケット開始時刻は、アクセスを処理できる時刻として予測したものである。

もし、チケットの開始時刻になっても、サーバが過負荷状態から脱しておらず、次回アクセスが失敗すると、そのアクセスを行ったユーザの不満は大きくなる。そのため、配布するチケットの開始時刻には余裕を持たせ、上記の状況を回避している。しかしながら、余裕を持たせているために、リクエストが集中しているのにサーバの手が空いていると言う無駄が発生してしまう。この無駄を無くすために、NAP-Web に次回アクセス補助機能を持たせることにした。次回アクセス補助機能は、クライアント・サーバ間で連携を行い、効率的に次回アクセスを行うための機能である。クライアントは次回アクセスが可能かどうかを調べるために、適当なタイミングでサーバの状態を確認する。

このタイミングを決定するために、NAP-Web は次回アクセス待ちユーザをグループ化して管理する。

本稿では、グループ化した次回アクセス待ちユーザを管理する、次回アクセス補助機能の実現について述べる。

### 2. 次回アクセスチケット発行機構

NAP-Web は次回アクセスに必要なチケットの発行を管理する機構を持つ。チケットの発行時に、それぞれのチケット開始時刻と終了時刻を算出しなければならない。この際に、スロットと呼ぶ概念を用いる。スロットとは、次回アクセスを可能とする時間を一定の間隔で区切ったものである。例えば、1 スロットの期間を 10 分間とすると、次回アクセスチケット発行機構はサーバの処理状態から、10 分間で処理可能なアクセスの数を算出し、その数を 1 スロットで発行できるチケットの最大枚数とする。同一スロットで発行されたチケットを持つ次回アクセス待ちユーザを同じグループとして扱う。

### 3. アクセススケジューリング機構

NAP-Web ではクライアントからの Web アクセスを Run\_Ready, Wait, Next\_Wait, Re\_Access の 4 つのグループに分け、スケジューリングを行っている。Run\_Ready は通常処理されるアクセスグループ、Wait は Run\_Ready が空くまでキューを作って待機するアクセスグループ、Next\_Wait は次回アクセスを指示され処理をリジェクトされたアクセスグループ、Re\_Access は Next\_Wait に入ったアクセスが次回アクセスを行った時に入り、Wait が空くまでキューを作って待機するアクセスグループである。これらに加え、Connected アクセスグループを、次回アクセス補助機能を実現するために追加した。Connected アクセスグループは、チケットの開始時刻の前に、5 節で述べる条件を満たした次回アクセスが入るキューであり、このグループに入ったアクセスは Wait や Re\_Access に空きが出来るのを待っている。

Grouping User who Wait Next Access, on NAP-Web  
Next Access Support

<sup>†</sup>Kagawa University Graduate School of Engineering

<sup>‡</sup>Kagawa University Faculty of Engineering

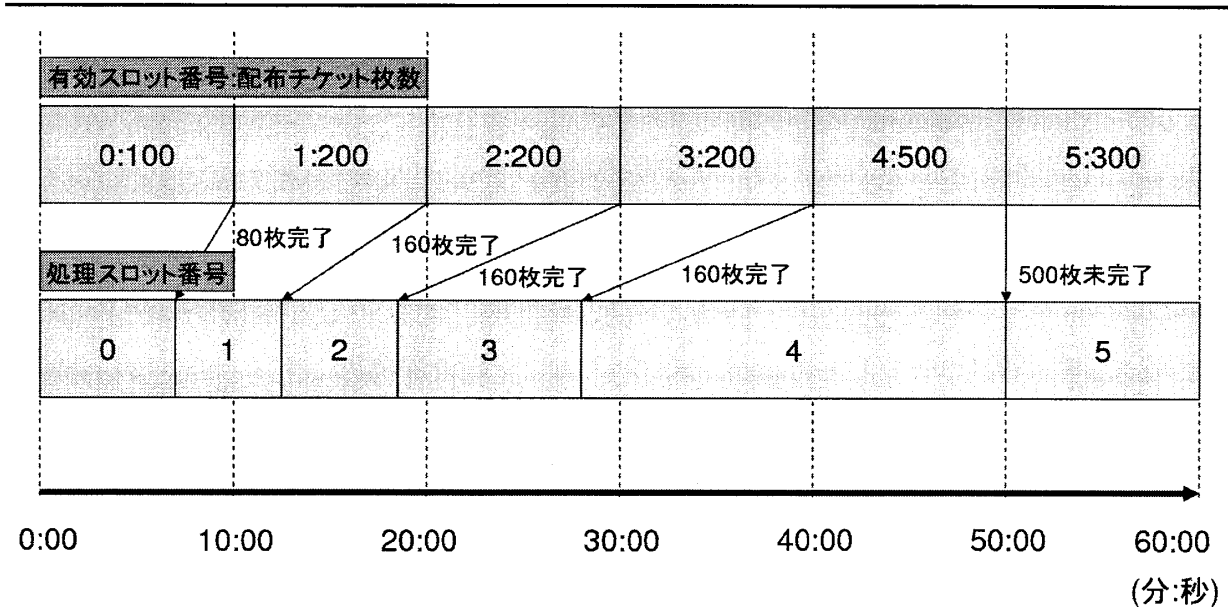


図 1：有効スロット番号と処理スロット番号の概要

#### 4. 有効スロット番号と処理スロット番号

サーバの状態を把握するために、有効スロット番号と処理スロット番号を用意する。有効スロット番号と処理スロット番号の概要を図 1 に示す。有効スロット番号とは、サーバの処理が予測どおりに進んだ場合に有効となるスロット番号である。図 1 において、経過時間が 10 分の時の有効スロット番号は 1 番である。また、処理スロット番号は、実際に処理しているスロットの番号であり、サーバの処理が予測より早く進んでいる場合は有効スロット番号よりも早く進む。処理スロット番号が進む条件は、スロットで配布したチケットの枚数のうち、何割かの次回アクセスが完了することである。この条件を 8 割とした時、図 1 では、0 番のスロットのチケットのうち、80 枚のチケットが完了した時に処理スロット番号は 1 番に進む。同様に、1 番のスロットのうち 160 枚のチケットが完了した時、処理スロット番号は 2 番に進む。この二つのスロット番号を比較することで、サーバの状態を把握する。有効スロット番号と処理スロット番号が等しい時、サーバでの処理は予測どおりに進んでいることになる。有効スロット番号が処理スロット番号より大きい時、サーバでの処理は予測より早く進んでいることになるとみなすことができる。

#### 5. Connected への分岐条件

Connected への分岐条件は、1) 有効スロット番号=処理スロット番号のとき、有効スロットと同じスロット番号のチケットを持つアクセスであること、もしくは、2) 有効スロット番号<処理スロット番号のとき、処理スロット番号と同じスロット番号のチケットを持つアクセスであること、

の二つである。

#### 6. クライアントの振る舞い

クライアントはサーバによってチケットを配布されると、その開始時刻より前にサーバの状態を確認する。例えば、予測時刻の半分の時にサーバの状態を調査するとする。図 1 において、スロット番号 4 番のチケットが割り振られ、40 分後に開始時刻が来るとすれば、まず、20 分後にアクセスを行う。20 分後にアクセスしたところ、有効スロット番号が 2 番、処理スロット番号が 3 番なので、Connected には繋がらず、もう一度再アクセスを行う。予測よりも処理が進んでいるので、次はそれから 10 分後にアクセスを行う。有効スロット番号は 3 番だが、処理スロット番号が 4 番なので、Connected アクセスグループにこのアクセスはつながり、そこで処理を待つことになる。上手く進めば、予測よりも 10 分早く処理が完了することになる。

#### 7. まとめ

本稿では、次回アクセスユーザをグループ化し、このグループに対してどのように次回アクセス補助機能を提供するかについて述べた。今後、この機能を実際に NAP-Web に実装し、その効果を確かめていく。

#### 参考文献

[1] 加地智彦、最所圭三：「過負荷時のユーザの不満を抑えるために次回アクセスを保証する Web システム」、情報処理学会論文誌、Vol. 50、No. 2、2009 (掲載予定)