

# 次回アクセスを保証する Web システムにおける アクセススケジューリング機構の改良

08T282 松浦 正尚（最所研究室）

本研究では次回アクセスを保証する Web システムにおけるスケジューリングの改良を行った．改良前後のシステムで比較実験を行い，提案手法の有用性を確認した．

## 1 はじめに

当研究室では，過負荷状態の Web サーバにおいて整理券を発行することで負荷を時間的に平滑化しつつ，サービスの提供が遅れることによるユーザへのストレスを軽減するために次回アクセスを保証する Web システム (NAP-Web)[1] を開発している．本研究では NAP-Web のアクセススケジューリング機構を改良し，より正確なスケジューリングの実現を目的とした．

## 2 NAP-Web

NAP-Web は Apache[2] のモジュールとして実装されるミドルウェアである．本システムは Web サーバが過負荷状態であることを検知すると，新たに到着したアクセスへのレスポンスを拒否すると同時に，そのアクセスを行ったクライアントに仮想的な整理券を配布する．整理券には Web サーバが過負荷状態から脱すると予測される時間 (次回アクセス可能時間) が記入されており，この時間が経過した後に再度リクエストを送信してきたアクセスを NAP-Web は可能な限り受け付ける．整理券を発行されたアクセスは次回アクセス可能時間を表示する Web ページヘリダイレクトされるため，ユーザは確実にサービスが提供される時間を知ることができる．

## 3 アクセススケジューリング機構

NAP-Web はアクセスを 4 つのグループに分類し，アプリケーションレベルでスケジューリングを行う．分類するグループは，通常通りサービスを提供する Run\_Ready，Run\_Ready に入りきらなかったものをキューイングする Wait，Web サーバが過負荷状態である際にレスポンスを拒否して整理券を配布する Next\_Wait，次回アクセスをキューイングして Wait に余裕が生じると移動させる Re\_Access の 4 つである．

本機構は次回アクセス可能時間を算出するための時間予測機能を持つ．時間予測機能は，Web サーバが過負荷状態から脱する時間を動的に予測し，これを次回アクセス可能時間として整理券に記録する．従来の予測式を式 (1) に示す．式 (1) では，統計情報として収集した各アクセスの平均処理時間  $t_{avg\_proc}$  に，Web サーバにキューイングされているアクセス数の合計

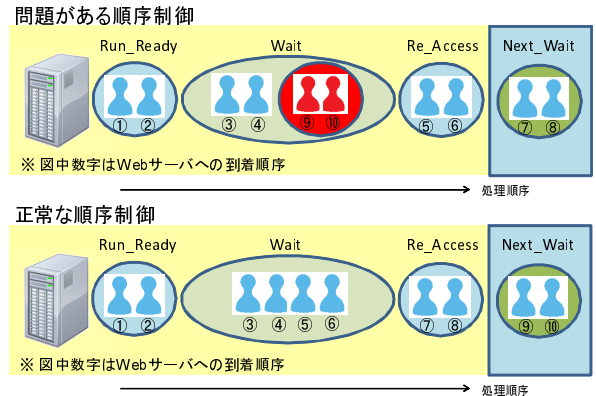


図 1: 順序制御の問題点

$NUM_{queue}$  を乗じ，これに余裕時間  $t_{safe}$  を加えたものを次回アクセス可能時間  $t_{next\_t}$  としている．

$$t_{next\_t} = NUM_{queue} \times t_{avg\_proc} + t_{safe} \text{ [sec]} \quad (1)$$

式 (1) に示した予測手法は，平均処理時間  $t_{avg\_proc}$  を安定的な値として得られないことや，将来的な次回アクセスの数が考慮されていないことにより，予測精度が低かった．そのため，式 (1) では Web サーバが過負荷状態から確実に脱している時間を通知するために余裕時間  $t_{safe}$  を加えており，実際に過負荷状態から脱するよりも長い時間を通知していた．このため，整理券を発行したアクセスの次回アクセスを待つ間，新たな初回アクセスを受け付けなければサーバの利用率が低下する．従って，NAP-Web では次回アクセス可能時間の経過を待っているアクセスが存在しても，新たな初回アクセスを受け付けている．このとき，順序制御に問題が生じる．この問題の概要を図 1 に示す．図 1 上段の⑨と⑩のように順序制御に問題があるアクセスを割り込みアクセスと称する．

また，従来のスケジューリングでは一定の期間毎に発行する整理券の枚数を制限していた．そのため，ある期間内で制限を超える枚数の整理券を発行する場合には，次の期間が始まるまでの時間を予測時間として通知していた．このとき通知される次回アクセス可能時間は非常に長く，ユーザに余計な時間を待たせることになる．

## 4 アクセススケジューリング機構の改良

本研究では順序制御の問題を改善することでサービスの公平性を確保し、さらに、ユーザの待機時間を可能な限り短くするためにアクセススケジューリング機構を次のように改良する。

- 平均処理時間  $t_{avg-proc}$  に代わり、値が安定的な平均スループット  $T_{avg-proc}$ [3] を予測式に利用する。
- 将来的な次回アクセスを考慮するために Wait キュー内のアクセス数と、有効な発行済み整理券の合計  $NUM_{queue\_ticket}$  を予測式に利用する。
- 異常に長い次回アクセス可能時間の通知を回避するために、一定期間内に発行できる整理券枚数を無制限とする。

新たに設計した予測式を式 (2) に示す。式 (2) を用いて算出した次回アクセス可能時間を  $t_{next\_T}$  とする。

$$t_{next\_T} = \frac{NUM_{queue\_ticket}}{T_{avg-proc}} \text{ [sec]} \quad (2)$$

## 5 評価実験

評価実験には同一のギガビット LAN 上に存在する二台のマシンを用いた。Web サーバとして Intel Core2Duo E8500 3.16 GHz, メモリ 2 GB, 負荷用マシンとして Intel Core2Duo E4500 2.20 GHz, メモリ 2 GB のマシンを利用した。負荷用マシンで擬似的なユーザを 1,000 人生成することで、特定のコンテンツを提供している Web サーバに対してアクセスが殺到する状況をシミュレーションし、改良前後における負荷分散の様子を調べた。

実験の結果、改良後の NAP-Web では割り込みアクセス数等に大幅な改善が見られた。しかし、NAP-Web が十分な統計情報を収集できていない過渡状態では、まだ予測精度が低くなり、千個以上の割り込みアクセスが発生していた。そこで、式 (2) に適当な係数を掛けることで、過渡状態におけるスケジューリングの誤差の軽減を試みることにした。新たな予測式を式 (3) に示す。

$$t_{next\_T-\alpha} = t_{next\_T} \times \alpha \text{ [sec]} \quad (3)$$

式 (3) は式 (2) に適当な係数を掛けたものである。 $\alpha = 1.0$  の場合が式 (2) に該当する。式 (3) の  $\alpha$  を 0.1 ~ 1.0 まで 0.1 刻みで変化させて同一の実験を行った。その結果、今回のシミュレーションでは  $\alpha = 0.3$  の場合に最良の結果が得られた。表 1, 図 2 に実験結果を示す。

表 1 より、上段 3 項目について大幅に改善していることが分かる。下段 2 項目については値に低下が見られたものの、その差は小さなものであった。図 2 より、 $\alpha = 0.3$  において Web サーバが定常状態となると割り込みアクセス数は零となり、完全な順序制御に成功していることが分かる。

表 1: 改良前後における比較表

平均値	改良前	$\alpha = 1.0$	$\alpha = 0.3$
整理券発行枚数	4452	25513	71311
次回アクセス可能時間 [sec]	40.28	5.82	1.71
割り込みアクセス数	5247.2	811.2	3.2
スループット [access/sec]	171.0	166.9	165.7
レスポンス待機時間 [sec]	3.75	3.79	3.91

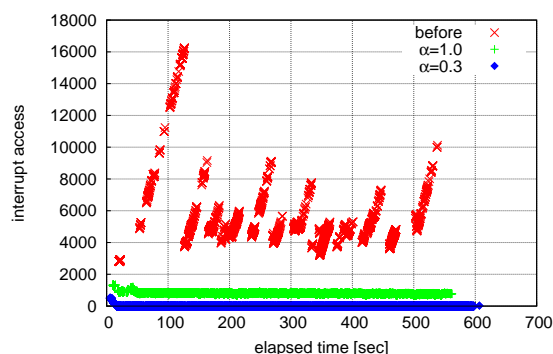


図 2: 改良前後の割り込みアクセス数

## 6 まとめと今後の課題

本研究では、NAP-Web のスケジューリングにおける問題点を解決するために、スループットを用いた時間予測式等を提案した。評価実験より、本研究における提案手法の有用性が示された。

しかし、図 2 より過渡状態において未だに割り込みアクセスが発生していることが分かった。そこで、更なる改良の手法として、運用前の Web サーバにおいてベンチマークテスト等を行い、その結果を用いたスケジューリングを実施すること等が考えられる。

## 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, pp.872-881, 2009
- [2] The Apache Software Foundation: “ Welcome! - The Apache HTTP Server Project ”, <http://httpd.apache.org/>
- [3] 山田茂和: “ 次回アクセスを保証する Web システム NAP-Web の性能評価と改良 ”, 香川大学, 学士論文, 2011