

NAP-Web へのページスケジューラの導入 Introducing Page Scheduler into NAP-Web

加地 智彦[†] 最所圭三[‡]
Tomohiko Kaji Keizo Saisho

1. はじめに

インターネットの普及により、Web サーバに対する処理要求が増大し続けている。アクセス集中により、Web サーバが、サービスを満足に行えなくなると、ユーザの不満が増大する。

我々は、アクセス集中の際に次回アクセスを保证することにより、ユーザの不満を緩和する Web システム NAP-Web[1]の開発を行っている。NAP-Web は Web サーバデーモン Apache の拡張モジュールとして動作する。

NAP-Web は、次回アクセスを保证するために、個々のアクセス (ファイル単位) レベルで処理の順序とタイミングを制御する Web アクセススケジューラを持つ。しかしながら、Web ページは複数のファイルで構成されることが多く、Web アクセススケジューラでは、Web ページを構成するいくつかのファイルの処理が後回しになり、ページの一部が表示されてから全体が表示されるまでに時間がかかる場合がある。このため、Web ページを構成するファイルを連続して処理を行うように Web アクセススケジューラを拡張した Web ページスケジューラを開発することにした。

なお、関連研究としてページ処理性能の向上を目的とした SQS[2]のページスケジューラなどがあるが、Web ページスケジューラは処理性能の向上ではなく、次回アクセスを保证する NAP-Web の仕組みにおいて、連続アクセスを行えるようにすることを目的としている点が異なっている。

2. Web アクセススケジューラ

Web ページスケジューラを説明する前に、ベースとなる Web アクセススケジューラについて説明する。

2.1 概要

Web アクセススケジューラは、ユーザからの Web アクセスを、Web サーバの負荷状況に応じて、以下の4つのグループに分けて処理順序を制御する。

1. Run_Ready
2. Wait
3. Next_Wait
4. Re_Access

Run_Ready に振り分けられた Web アクセスは、通常通りの処理が行われる。Run_Ready が一杯になると、Wait にアクセスが振り分けられる。Wait に振り分けられた Web アクセスは、Run_Ready が空くのをキューを用いて待つ。Wait からあふれた Web アクセスは Next_Wait に割り振られる。Next_Wait に割り振られた Web アクセスは、リクエ

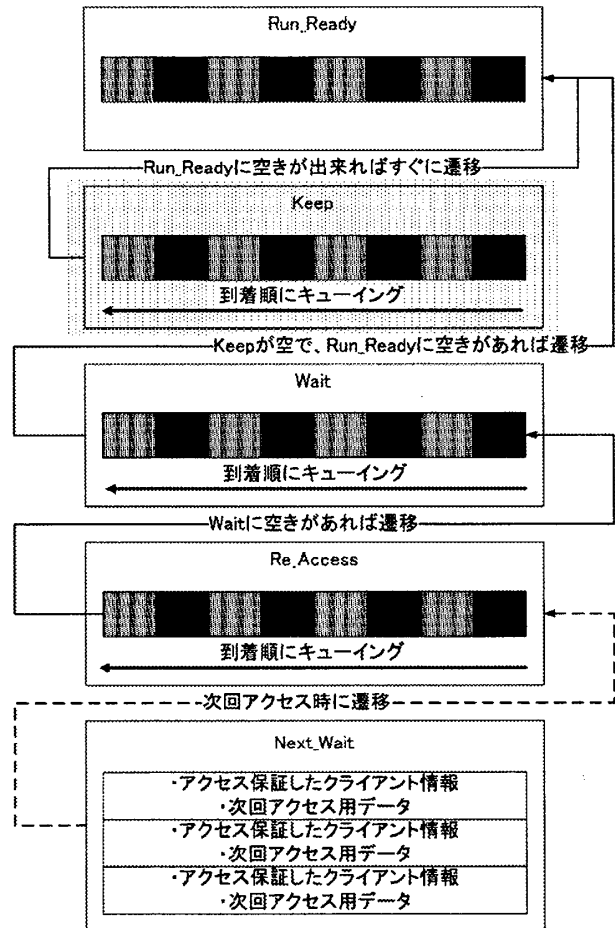


図1 Web ページスケジューラの概要

スト情報を記録された上で、処理を拒否される。ただし、その際に次回アクセス可能な時間帯を伝えられ、次回アクセスを保证するチケットを Cookie の形で渡される。Next_Wait に割り振られた Web アクセスは、伝えられた時間帯に、渡された Cookie を持って、次回アクセスを行うと Re_Access に割り振られる。Re_Access は、次回アクセス用のキューであり、Wait に空きが出来ると、Re_Access の先頭から Wait の最後尾に移動する。

2.2 問題点

Web アクセススケジューラはアクセスが行われるたびに、アクセスをグループに振り分けるため、連続アクセスが必要な場合、不都合が生じる。例えば、あるページが5つのファイルにより構成されるとき、3つのファイルへのアクセスは Run_Ready に入ったが、4つ目のファイルへのアクセスが Wait に入ると、そのファイルの取得をするために何秒か待たなければならなくなってしまう。また、もし5

[†] 香川大学大学院工学研究科 Graduate School of Engineering, Kagawa University

[‡] 香川大学工学部 Faculty of Engineering, Kagawa University

つ目のファイルへのアクセスが Next_Wait に割り振られてしまったとすると、次回アクセスを行わなければ Web ページに閲覧できない箇所が発生してしまう。

3. Web ページスケジューラへの拡張

2 節で述べた問題点を解決するため、Web アクセススケジューラを拡張することにした。拡張した Web アクセススケジューラを Web ページスケジューラと呼び、図 1 にその概要を示す。アクセスグループに、Keep アクセスグループを追加しており、スケジューラは Run_Ready に割り振られたアクセスを処理する際に、Cookie をアクセスに対して発行する。この Cookie を持って再アクセスが行われた場合、以前のアクセスから再アクセスまでの間隔（再アクセス間隔）が一定期間（再アクセス指定期間）内であれば、スケジューラは再アクセスを Keep アクセスグループに振り分ける。Keep アクセスグループはキューとなっており、Wait と Re_Access より優先して Run_Ready に回される。再アクセス指定期間は、短すぎると、本来再アクセスとすべきアクセスを再アクセスとして判定できなくなり、長くしすぎると、再アクセスとすべきではないアクセスまで再アクセスとして判定してしまう。そのため、サーバに対して実際に行われた、過去の再アクセス間隔の統計値を用いて再アクセス指定期間を動的に決定することで、最適化を行う。この Web ページスケジューラを NAP-Web に組み込むことで、1 度 Web ページへのアクセスが開始されれば、そのページを構成している全てのファイルをスムーズに取得できる。

4. 評価

提案の Web ページスケジューラを実装した NAP-Web の評価実験を行った。Web サーバとして、Xeon2.4GHz×2、1024MB を搭載した Linux2.6.24 の動作する PC を用いた。Apache のバージョンは 2.2.8 である。Web ページスケジューラを適用する場合としない場合に分けて、負荷実験を行った。負荷シナリオとして、クライアントは、Web サーバ上に置かれたページに対してリクエストを行うものとする。1 つのページを 5 つのファイルで構成し、各ファイルのサイズを 200KB とした。3 台のクライアント PC を用いて、負荷シナリオを 50 回繰り返すスレッドを 30 個、60 個、90 個、120 個、150 個と個数を変えて実行した。なお、NAP-Web の設定として、Run_Ready の最大個数は 10 個、Wait と Re_Access のキューは滞在時間の最大値が 4 秒になるように自動的に設定し、再アクセス指定期間を過去の再アクセス間隔の平均値の 2 倍とした。

評価の指標として、連続アクセス成功率と、ページ取得時間を用いた。連続アクセス成功率は、1 つ目のファイルを取得し始めてから、5 つ目のファイルを取得するまでに、ファイルを取得するためのアクセスが Wait や Next_Wait に割り振られることが無かったページ数が全体のページ数に占める割合である。ページ取得時間とは、クライアントが再アクセスのための待ち時間を終え、1 つ目のファイルの取得を開始した時間から 5 つ目のファイルの取得が完了した時間である。

図 2 に Web ページスケジューラを適用した場合としない場合の平均ページ取得時間を示す。Web ページスケジューラを適用した場合は、連続アクセスが成功した場合、

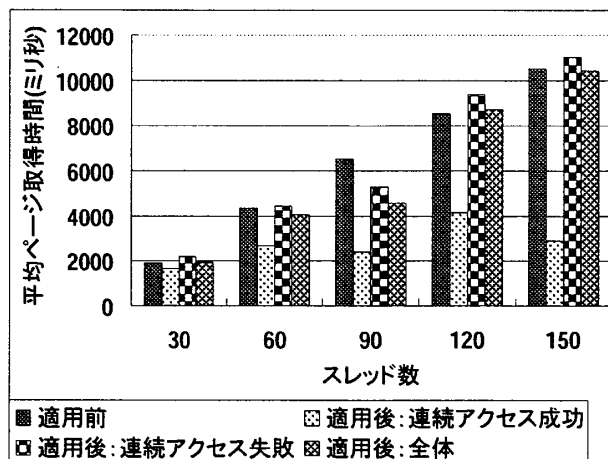


図 2 平均ページ取得時間の比較

表 1: 連続アクセス成功率

スレッド数	30	60	90	120	150
連続アクセス成功率(%)	43.0%	23.3%	25.1%	12.7%	2.0%

成功しなかった場合、および全体の平均ページ取得時間をそれぞれ示している。表 1 に連続アクセス成功率を示す。なお、Web ページスケジューラを適用しない場合は、連続アクセスに成功したことは無かった。

図 2 を見て分かるように、スレッド数が多くなればなるほど、Web ページスケジューラの適用効果は大きくなった。これは、一度ページの取得を開始さえすれば、Wait や Next_Wait に割り振られることなくページの取得を完了できることを示している。しかしながら、表 1 の結果は、スレッド数が増加すればするほど、連続アクセスの成功率が低下していることを示している。これは、スレッド数が増加すればするほど、再アクセス間隔のばらつきが大きくなり、動的に決定された再アクセス指定期間よりも再アクセス間隔が長くなるためだと考えられる。ただし、この原因がクライアント側の問題なのかサーバの問題なのかは未検証である。

5. まとめ

NAP-Web において、Web ページを取得するための連続アクセスを保証するために、Web アクセススケジューラを拡張した Web ページスケジューラを開発した。Web ページスケジューラを評価した結果、全てではないが連続アクセスが成功でき、失敗した場合と比較して高速にページを取得できることが確認できた。しかしながら、再アクセス指定期間の精度が低いため、スレッド数が多くなればなるほど連続アクセス成功率が低くなってしまふ。今後は、この原因の究明を行い、再アクセス間隔時間のヒストグラムと閾値を利用する、などの再アクセス指定期間の決定法について検討していく。

参考文献

- [1] Tomohiko Kaji, Keizo Saisho: Proposal of Web System "NAP-Web" to Promise Next Access, SICE Annual Conference2007, pp.959-964, 2007.9
- [2] 松沼正浩、光来健一、日比野秀章、佐藤芳樹、千葉滋: 過負荷時の Web アプリケーションの性能を改善する Session-aware Queue Scheduling, 日本ソフトウェア科学会論文誌「コンピュータソフトウェア」, Vol.23, No.2, pp.199-210, 2006