NAP-Web のためのページ情報取得システムの設計と実装

05T251　　新田 淳二 （最所研究室）

次回アクセスを保証する Web システム「NAP-Web」のアクセス待ち時間の予測精度を向上させるための情報を取得する、ページ情報取得システムの設計と実装について述べる。

1　はじめに

当研究室では、過負荷時にサービスの提供を受けられないユーザの不満を、サービスの提供が可能になる時間を知らせ、その時間のアクセスを保証することで緩和する Web システム「NAP-Web」の開発[1]を行っている。NAP-Web は、ユーザからのアクセスを制御することにより、Web サーバへの過負荷を回避する。このため、ユーザがいつになればアクセスが可能になるかを示すアクセス待ち時間の予測を行っている。しかし、1 つのコンテンツは複数のリソースより構成され、各コンテンツでアクセス時間が異なる。そのため、処理時間のばらつきが大きくなり、予測精度が悪化するという問題点がある[2]。この問題点を解決するため、ページ情報取得システムの開発を行うことにした。このシステムは、提供している Web ページの構成と処理時間を把握する。本稿では、ページ情報取得システムの設計と実装について述べる。

2　アクセス待ち時間の予測

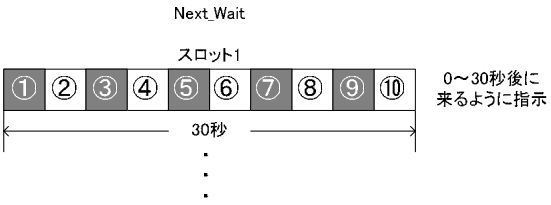

図 1　アクセス待ち時間予測に使われるスロット

NAP-Web は、アクセス待ち時間を予測するために、図 1 に示すスロットと呼ぶ概念を用いている。現在のアクセス待ち時間を予測する手法は、式 A、式 B に示すようにリソースアクセス処理時間の平均値を用いているだけで、各コンテンツの処理時間とページ構成を考慮していない。

$$\text{スロット最大値} = \frac{\text{スロット期間}}{\text{平均アクセス処理時間}} \quad \text{A}$$

$$\text{アクセス待ち時間} = \text{スロット開始時刻} + \frac{\text{チケット番号} \times \text{スロット期間}}{\text{スロット最大値}} \quad \text{B}$$

よって、処理時間のばらつきが大きいアクセスが集中すると、アクセス待ち時間の予測精度が悪化するという問題点がある。また、コンテンツの構成を把握していないため、複数のリソースからなるコンテンツでは連続アクセスを必要としているかがわからず、そのコンテンツに対しての正確なアクセス待ち時間を予測することができない。

この問題に対して、サーバ側で提供するコンテンツのページ構成と処理時間を把握することにより、アクセス待ち時間の予測精度を上げることができる。表 1 に示すようなページ構成情報をサーバ側が把握しているとすると、式 C を用いて容易にアクセス待ち時間を予測することが可能である。チケット番号とはアクセス待ちのユーザの順番である。

$$\text{アクセス待ち時間} = \text{スロット開始時間} + \sum_{K=1}^{N-1} \text{チケット番号Kの合計処理時間} \quad \text{C}$$

また、ページ構成と処理時間、コンテンツサイズといった情報の統計を取ることにより、新規にアップロードしたコンテンツに対しても同様な性質を持つコンテンツが存在するならば処理時間の見積もりができ、アクセス待ち時間の予測が可能となる。

表 1　提供しているコンテンツのページ構成

コンテンツ	リソース	サイズ	処理時間	合計サイズ	合計処理時間
Samp1.html	Samp1.html	1000 KB	1000 ms	5000 KB	5000 ms
	img1.jpg	2000 KB	2000 ms		
	img2.jpg	2000 KB	2000 ms		
Samp2.html	Samp2.html	500 KB	500 ms	1500 KB	1500 ms
	img3.jpg	1000 KB	1000 ms		

3　ページ情報取得システムの概要

ページ情報取得システムは以下の図 2 に示すように 4 つの機能より構成されている。図 2 の右側の部分が本研究で開発するものである。

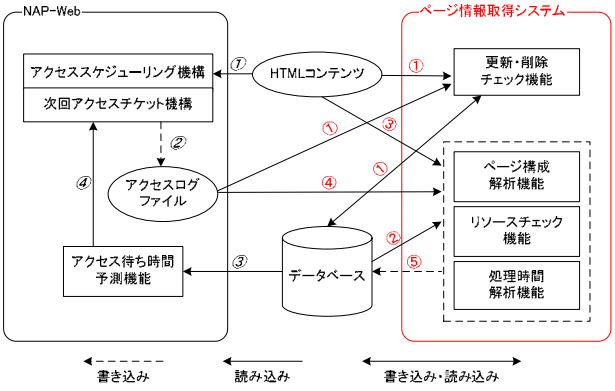

図 2　NAP-Web とページ情報取得システムの関係

NAP-Web ではユーザからのアクセスに対して、アクセススケジューリング機能によって、アクセスが処理されるか否かが決定される。HTML コンテンツをユーザに提供したとき（①）、アクセスログをとる（②）。アクセスが拒否されたユーザには、アクセス待ち時間がかかれたチケット渡す。このアクセス待ち時間を予測するのがアクセス待ち時間予測機能である。アクセス待ち時間予測機能は、DB に格納されたコンテンツ情報を参照し（③）、アクセス待ち時間を予測する（④）。次回アクセスチケット機構は、予測されたアクセス待ち時間を書いた仮想的なチケットをユーザに発行する。

ページ情報取得システムは、**更新・削除チェック機能**により HTML コンテンツがあるディレクトリとアクセスログファイルを検査し、HTML コンテンツの更新と削除の結果をデータベースに情報を書き込む（①）。**ページ構成解析機能**は、更新されたコンテンツのリソース構成を解析し（②）、DB に書き込む（⑤）。**リソースチェック機能**は、更新された各リソースのファイルサイズを求め（③）、更新されたリソースの対象となるコンテンツの合計サイズを計算して DB に書き込む（⑤）。**処理時間解析機能**はデータベースに書き込まれた更新・削除情報を取り出し（②）、アクセスログファイルから各リソースの処理時間を計算して（④）DB に書き込む（⑤）。

4　性能評価

本システムは NAP-Web で使用するにあたってサーバが過負荷になったり、処理時間がかかりすぎたりすることは避けなければならない。このため、各機能の CPU タイムを評価する。開発言語として PHP、データベースに、PostgreSQL を用いた。実験では、解析対象であるログデータを 1,000 行、10,000 行、100,000 行と増加させて計測を行う。今回の実験では、本システムを初めて動作させた場合を想定しているため、削除処理は発生しない。以下の 3 つのテスト結果を図 3、図 4、図 5 に示す。

- テスト 1
  コンテンツ数が固定で、構成するリソース数が変化した場合の処理時間を計測する。
- テスト 2
  コンテンツ数が変化し、構成するリソース数が固定の場合の処理時間を計測する。
- テスト 3
  コンテンツ数と構成するリソース数が変化する場合の処理時間を計測する。

テスト 1(図 3)では、更新コンテンツが固定でリソース数が増加であるため、更新・削除チェック機能を除いて、線形に処理時間が増加していることがわかる。テスト 2(図 4)では、コンテンツ数の増加に伴い処理時間が増加しているのは、更新・削除チェック機能のみである。テスト 3(図 5)では、コンテンツとリソース数が増加するため、全ての機能で処理時間が線形に増加していることがわかる。

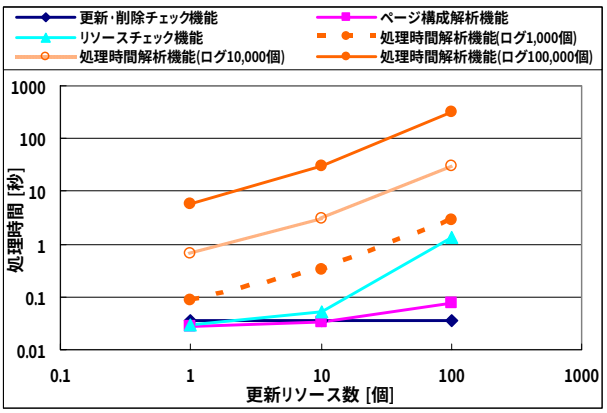

図 3　テスト 1 での各機能の処理時間

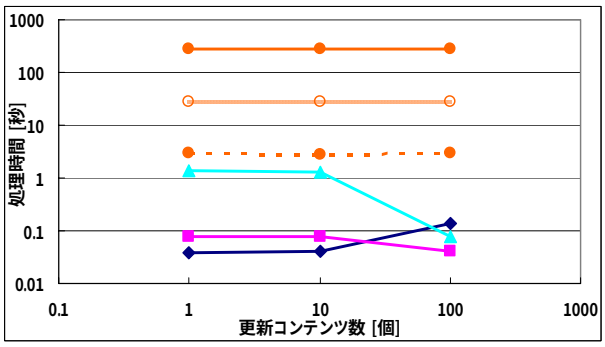

図 4　テスト 2 での各機能の処理時間

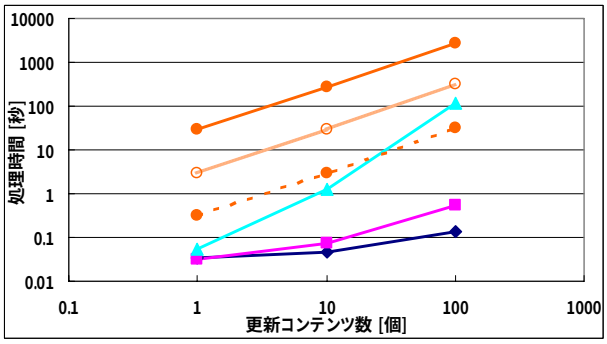

図 5　テスト 3 での各機能の処理時間

以上のテスト結果より、処理時間は、ログ数と更新リソース数の変化により大きく変動することがわかる。現段階では、リソース数が 1,000 とログ数が 100,000 行のときでは、処理時間が約 9 時間もかかってしまう。実用化のためには、アルゴリズムを見直し、処理時間の大幅な改善が急務である。

5　おわりに

NAP-Web のアクセス待ち時間の予測精度を向上させるために Web ページの構成と処理時間を取得するページ情報取得システムの設計･実装を行った。予稿には載せていないが、ページ構成を把握するために必要な情報が取得できていることは確認できた。しかし、処理時間に関しては、とても実用化できるレベルではない。よって以下にこれからの課題と改善点を示す。

- アクセスログを処理するアルゴリズムの改善
- ファイル更新情報取得の高速化
- DB を変更してのアクセスタイムの比較
- Flash、CGI などの動的コンテンツへの対応
- NAP-Web との連携によるアクセス待ち時間の予測精度向上の実証

参考文献

[1] 加地智彦、「次回アクセスを保証する Web システム『NAP-Web』の開発」、 香川大学大学院工学研究科、修士論文、2006

[2] 加地智彦 最所圭三、「NAP-Web へのページスケジューラの導入」、 第 7 回情報科学技術フォーラム講演論文集(FIT 2008)、 L-031、 Vol.4、 PP.159-160、2008