

優先アクセス機構を持つ Web サーバの開発およびその評価 Development and Evaluation of a Web Server with Priority Access Mechanism

溝渕 久哲† Hisanori Mizobuchi
最所 圭三 Keizo Saisho

1 はじめに

一時的にサービス需要が高まりアクセスが集中した場合に Web サーバが過負荷状態に陥ることを回避する方法の 1 つとして、負荷分散技術が用いられている。その一例として、キャッシュサーバやミラーサーバを用いた Web システムがある。本論文では、このような Web システムのことを分散 Web システムと呼ぶ。

当研究室では、Web サーバとキャッシュサーバ群からなる分散 Web システムを開発している^[1]。このシステムでは、Web サーバに対して一般のアクセスとキャッシュサーバ群からの更新のためのアクセスが入る。このとき一般のアクセスの増加によりサーバが過負荷になるとキャッシュサーバの更新が滞ってしまう。そのため本研究ではキャッシュサーバ群からのアクセスを優先して処理できる優先アクセス機構の開発を行っている。

本稿では、優先アクセス機構の設計、実装、及びその評価について述べる。

2 優先アクセス機構

2.1 設計

図 1 に本研究で開発する優先アクセス機構の概要を示す。優先アクセス機構は本研究で開発した NAP-Web^[2]に優先アクセス機能を追加することによって実現する。NAP-Web は、過負荷時にアクセスを行ったユーザに対し整理券を配布することで次回アクセスを通知し、通知した時間に行われたアクセスを処理することを保証する。このため図 1 の左側に示すキューを用いて、スケジューリングを行っている。Run_Ready, Wait, Re_Access の 3 つのキューと、チケット情報を記録する Next_Wait を持ち、アクセスがあった場合、負荷状態に応じて各場所にキューイングされ、順次処理を行う。Run_Ready には実際に処理をしているアクセスが入り、Wait には実行待ちになっているアクセスが入る。また Re_Access には再アクセスを行ったアクセスが入る。過負荷時の初回アクセスについては、通常処理を行わず代わりに再アクセスの時刻を含んだチケットを発行する。チケットの情報は Next_Wait に記録する。チケットを持って、通知した時間に再度アクセスしてきた場合は、Re_Access キューへと格納し処理を行う。

本稿では、キャッシュサーバからのアクセスを優先アクセス、一般のアクセスを通常アクセスと呼ぶ。優先アクセス機構は優先アクセスを処理するもので、図 1 の右側に示す優先アクセス専用のキューを追加することで実現する。Pri_Run_Ready には実際に処理をしている優先アクセスが入り、Pri_Wait には実行待ちになっている優先アクセスが入る。従来の NAP-Web の機構 (以下通常アクセス機構)

と優先アクセス機構のスケジューリング機構を並列に置くことで、一般のユーザからのアクセスが集中している場合でも、それとは独立して処理することが出来る。

Run_Ready で同時に実行する数と Pri_Run_Ready で同時に実行する数を調整することで優先アクセスをどの程度優先するかを調整出来る。

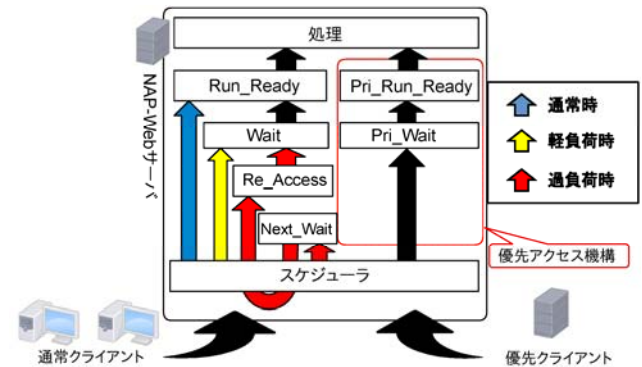


図1 NAP-Web に組み込んだ優先アクセス機構

2.2 簡易機構

優先アクセス機構が加わる事により、通常アクセスの処理にどのような影響があるかを調べる必要があると考えた。そのため実装が容易な、キューを用いない優先アクセスを直接処理する機構を実装し、その影響を調査した。本稿ではこれを簡易機構と呼ぶ。

優先アクセスかどうかはアクセス元の IP アドレスにより判定している。アクセスがあると最初に IP アドレスを確認し、優先クライアントであれば直ちに処理を開始し、そうでなければ通常アクセス機構で処理する。そのため、簡易機構では Run_Ready のキューの数と優先クライアントからのアクセス数の合計が、同時に処理されるアクセスの数となる。またその割合が、通常アクセスと優先アクセスに割り当てられる計算機資源の割合となる。

簡易機構を用いて、通常アクセスに対する優先アクセスの影響を調査した。

実験手法

サーバとして、Intel Core2 Duo E7500 2.93GHz, メモリ 4GB, CentOS 5.5, Apache2.2 を使い、クライアントとして、優先アクセスを行うマシンと通常アクセスを行うマシンを用いる。それぞれ指定した数の同時アクセスをサーバに対して行う。アクセス先は掲示板(ASKA BBS)への投稿である。各クライアントマシンでは、指定した数のスレッドを生成し、それぞれのスレッドがサーバに対しアクセスを繰り返し行う。なお Run_Ready のキューの数は 30 に設定している。

† 香川大学, Kagawa University

通常クライアントの同時アクセス数(通常アクセス数)は 3,000 に固定し, 優先クライアントの同時アクセス数(優先アクセス数)を変えて実験を行い, 各アクセスのスループット及び応答時間を調査する. 優先アクセス数を 0,1,5,10 と変えて, 通常アクセス開始直後の応答時間の乱れの影響を避けるため, 通常アクセス開始後 200 秒経過してから優先アクセスを開始する.

実験結果

図 2 に応答時間, 図 3 にスループットを示す. それぞれ 20 秒の移動平均をとっている. 以降のグラフでは全て 20 秒の移動平均を用いている.

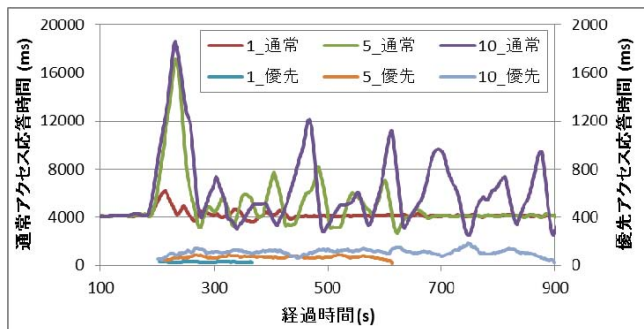


図 2 各同時アクセス数の場合の応答時間

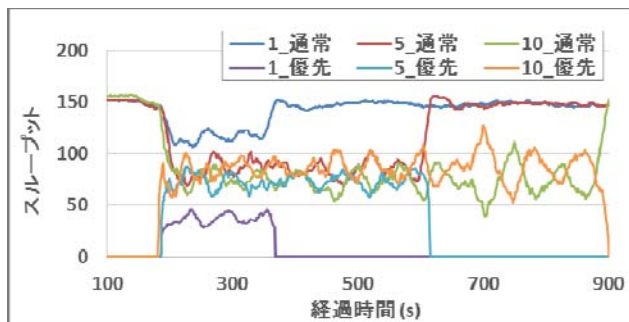


図 3 各時アクセス数の場合のスループット

応答時間について見ると, 通常アクセスの応答時間が優先アクセスの開始直後に極端に遅くなっていることがわかる. 優先アクセス数が 1 の時はそれほど影響を受けていないが, 5 と 10 の場合は開始時に大きく影響され, 80 秒程度で一旦優先アクセス開始前の応答時間になった. しかし, その後もピークは少しずつ減少しているが周期的に遅くなっている. 優先アクセスに対しては, アクセス数にほぼ比例して応答時間が遅くなっているものの, 高速に回答している. この結果は, 優先処理は実現できているが, 通常アクセスに悪影響を及ぼしていることを示す.

スループットについて見ると, 優先アクセスと通常アクセスのスループットの総和はほとんど変化しないが, 優先アクセスが増えるに従って優先アクセスのスループットが増加している. しかし Run_Ready のキューの数が 30 にもかかわらず, 優先アクセスが 5 と 10 のときで, ほぼ半分の計算機資源が優先アクセスで使われており, 通常アクセスと優先アクセスに割り当てるサーバの計算機資源の制御が不十分であった. この原因としてキュー操作を行うこと

によるオーバーヘッドや, OS のスケジューリングが影響しているものと考えられる.

3 優先アクセス開始時の影響の除去

優先アクセス開始時の通常アクセスの応答時間の遅れの原因は, 優先アクセスが入ったことにより通常アクセスに割りてられるリソースが減少し, そのため再アクセスしてきたリクエストを, チケット発行時に予測した時間よりも処理が長くなったためであると考えられる. そこで同時に処理する優先アクセス数を徐々に増やすことで, 通常アクセスへの影響を小さくすると考えた.

そこで, 優先アクセスの単位時間あたりの処理数を徐々に増やしていくことで, 通常アクセス応答時間への影響を抑えることができるかどうかを調査した.

実験内容

2.2 節と同じ実験環境で, 優先クライアントの同時アクセス数を徐々に増加させる実験を行った. 通常クライアントの同時アクセス数を 3,000 に固定し, 優先アクセス数を一定間隔(t 秒)毎に 1 ずつ追加し 5 まで増加させた. t は 0, 20, 40, 60, 80 の 5 通りを試した.

結果

図 4 に通常アクセスの応答時間を示す. t が 20 と 40 の時は, t が 0 の時と比べ, 応答時間の遅れが半分程度に収まった. t が 60 と 80 の時は, 影響はさらに小さくなり, 図 2 の同時アクセス数 1 の場合とほぼ同じ値となった.

以上のことから, 優先アクセスが入ることによる通常アクセスの応答時間への影響を抑える方法として, 優先アクセスの同時処理数を徐々に増やす方法が有効であることが確認できた.

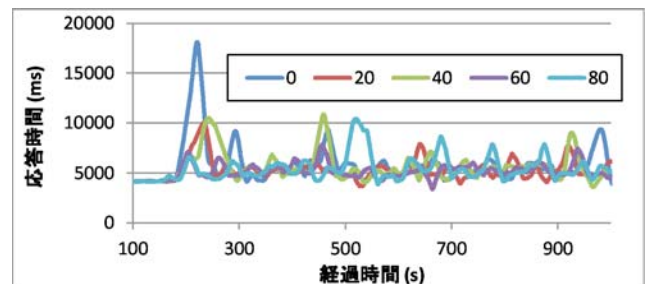


図 4 同時アクセス数を一定間隔ごとに増加させた場合の通常アクセス応答時間

4 キューを用いた優先アクセス機構

簡易機構により優先アクセスを優先して処理できることを確認できた. これを踏まえキューを用いた優先アクセス機構を実装することにした.

従来の NAP-Web のスケジューリング機構から Next_Wait と Re_Access を除いた機構を流用した. アクセスがあると最初に IP アドレスを確認し, 優先クライアントであれば優先アクセスのスケジューリング機構で, そうでなければ従来の NAP-Web のスケジューリング機構で処理する. 繰り返しになるが, これにより本機構では Run_Ready のキューの数と Pri_Run_Ready のキューの数の合計が, 同時に処理されるアクセスの数となる. それらの割合が通常ア

クセスと優先アクセスに割り当てられる計算機資源の割合となる。

4.1 キューの制御によるオーバーヘッドの影響

実装した機構では、優先アクセスについてもキューを用いている。そのため簡易機構よりも計算量が増加する。このオーバーヘッドが無視できる程度であるかを調査した。

実験内容

実験は2.2節と同じ環境で行った。キューを用いた機構を新機構、簡易機構を旧機構と呼ぶ。キューの設定値を

表1に、優先アクセスと通常アクセスの同時アクセス数を表2に示す。

表1 キューの設定値

	Pri_Run_Ready	Run_Ready
旧機構	(なし)	30
新機構	5	30

表2 同時アクセス数

	同時アクセス数
優先アクセス	5
通常アクセス	3,000

実験結果

優先アクセスのスループットと応答時間を図5と図6にそれぞれ示す。スループットを見ると、両方の機構の差はみられなかった。応答時間についても同様である。

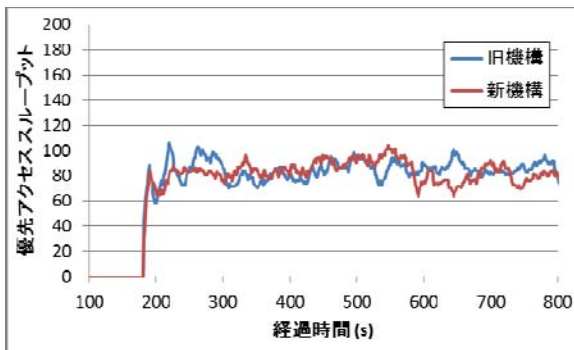


図5 優先アクセスのスループット

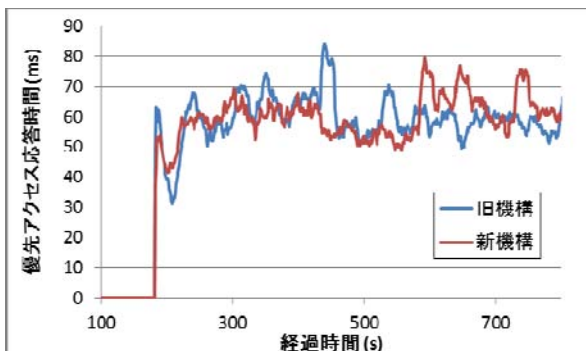


図6 優先アクセスの応答時間

図は省略するが、通常アクセスについても両方の機構での差は見られず新機構は2.2節の結果とほぼ同様となった。

以上の結果から、キューを用いて優先アクセスを処理するためのオーバーヘッドは十分に小さいといえる。

4.2 キューの効果

新機構では優先アクセスもキューを用いて処理をすることで、リソースの切り分けを行っている。優先アクセスの同時アクセス数に変化してもRun_Readyで処理される数が変化しないので通常アクセスの処理に影響を及ぼさないのである。本稿ではこのことを確認する。

実験内容

2.2節と同一の環境でPri_Run_Readyのキューの数を2, Run_Readyのキューの数を20にした状態で、優先クライアントからPri_Run_Readyのキューの数よりも多い同時アクセス(5,10,20の3パターン)を行った。

実験結果

優先アクセスの応答時間の変化を図7に、通常アクセスの応答時間の変化を図8に示す。旧機構と異なり、優先アクセスの同時アクセス数にかかわらず、通常アクセスの応答時間がほぼ同じ値となっていることがわかる。また優先アクセスの応答時間は、ほぼ優先アクセス数に比例して長くなっていた。

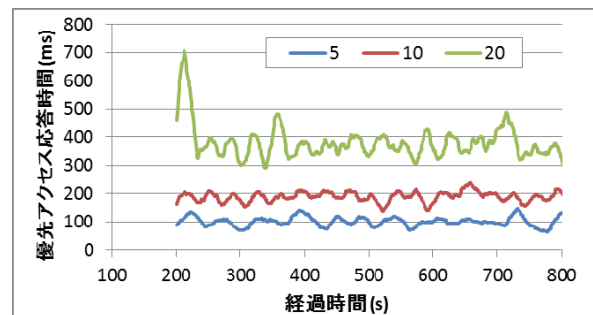


図7 優先アクセスの応答時間

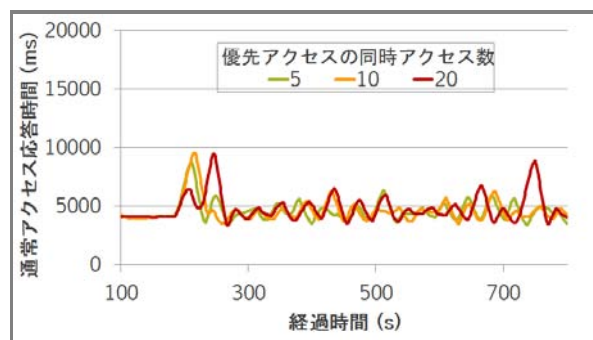


図8 通常アクセスの応答時間

通常アクセスのスループットを図9に示す。同時アクセス数を5に設定した場合、それ以上の結果と比べて、わずかにスループットが上がっている箇所が見られる。この部分の優先アクセスのスループットを確認したところ、スループットが下がっていた。この原因として、クライアントプログラムのタイミングによりPri_Run_Readyを埋めることができなかったと考えられる。以上のことから、優先

アクセス数が増えたり減ったりしても通常アクセスの処理には影響を及ぼさないことが確認できた。

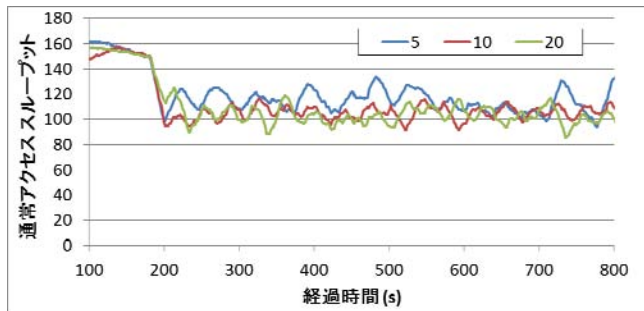


図 9 通常アクセスのスループット

4.3 優先アクセスへの計算機資源の割当量の調整

Run_Ready と Pri_Run_Ready のそれぞれのキューの数の割合を変更することによって、優先アクセスへの計算機資源の割当量を調整することができるかどうかを調査した。いくつかのキューの数の組み合わせにおけるスループットを測定した。

実験内容

実験は2.2節と同じ環境で行った。優先アクセスのキューの個数と通常アクセスのキューの個数を、表 3に示す組み合わせ(Pri_Run_Ready 数, Run_Ready 数)で行った。

表 3 実験を行ったキューの数

		Pri_Run_Ready	
		1	2
Run_Ready	10	(1,10)	(2,10)
	20	(1,20)	(2,20)

実験結果

全体のスループットに対する通常アクセスのスループットの占める割合を図 10に示す。また、Run_Ready と Pri_Run_Ready のキューの数の合計に占める Run_Ready のキューの数の割合と 200 秒から 800 秒間のスループット割合の平均値を表 4に示す。なお、通常アクセスと優先アクセスのスループットを合計した値は、全ての組み合わせで差はなく 160 程度となっていた。

キューの数の割合とスループットの割合が同程度の値となると考えていたが、全ての結果でキューの数の割合よりスループットの割合が低くなっていた。さらに、キューの数の割合が同じであればスループットの割合も同じになると考えていたが、こちらも想定と異なった。一方、(1,20)と(2,20)を比較すると、通常アクセスのキューの数の割合が大きい(2,20)のほうの通常アクセスのスループットが大きくなっていた。(1,10)と(2,10)の場合も僅かではあるが(1,10)のほうが大きく、キューの割合の大小関係と、スループットの割合の大小関係についてはかろうじて維持されていた。

スループットがこのような結果となった原因については現在調査中であり、今後の課題である。

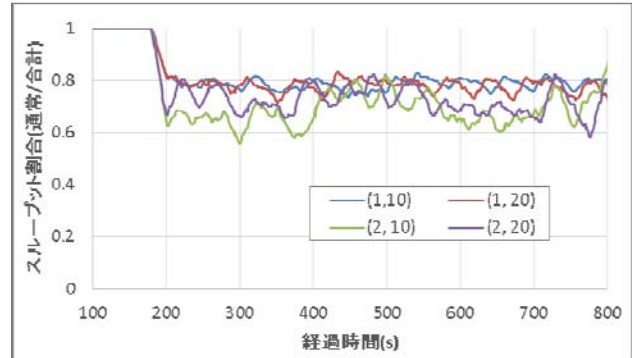


図 10 各キュー数時におけるスループットの割合

表 4 キューの数の割合とスループットの割合

	キューの数の割合	スループットの割合
(1, 10)	0.909	0.786
(1, 20)	0.952	0.780
(2, 10)	0.833	0.687
(2, 20)	0.909	0.722

5 おわりに

特定のアクセスを優先的に扱うため、NAP-Web に優先アクセス機構を追加した。簡易な機構で通常のアクセスに対する影響を評価した。優先アクセス機構には優先アクセスが開始された直後に、通常アクセスの応答時間が極端に遅くなる問題点が生じた。その解決策として、優先アクセスの処理を徐々に増やすことで、影響が 1 箇所に集中しないようにする方法を検討し、実験によりその解決策の有効性を確認した。

またキューを用いた優先アクセス機構を実装しキューの数を変えて実験を行った結果、キューの数に応じたリソース割り当てが行えていないことが判明した。また、キューを追加したことにより優先アクセス数が増加しても通常アクセスの処理には影響を及ぼさないことを確認した。

今後の課題として、キュー数に応じたリソース割り当てが行えていない原因を調査する。また Pri_Run_Ready 数を動的に変更し、優先アクセスの同時処理数を段階的に増加させる機構の追加、優先アクセスの判定を IP アドレス以外の方法への対応などがある。

謝辞 本研究は JSPS 科研費 25330082 の助成を受けた。

参考文献

- [1] 堀内晨彦, 最所圭三, "Development of Scaling Mechanism for Distributed Web System", SNPD, 2015
- [2] 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", 情処学論, Vol.50, no.2, pp.872-881, 2009