

# Web サーバの負荷状況に応じたアクセス制御

01T217 加地智彦 (最所研究室)

ブロードバンドの普及に伴い、Web サーバの負荷は増大している。これに対応するために負荷状況に応じてアクセス制御を行うモジュールを設計し、そのうちの一部を実装した。システムを組み込み実験を行った結果、提案方式の有効性を確認出来た。

## 1. はじめに

現在、ブロードバンドの普及に伴い、インターネットのユーザ数が増え続けている。ブロードバンドならではのコンテンツも現れるようになり、その処理は複雑さを増している。コンテンツが置かれている Web サーバにはこれまで以上の大きな負荷がかかるようになった。この負荷を下げるために分散処理などの手法が取られているが、これらはハードウェアの導入が必要であり、導入と運用の面でコストが大きい。また、Web サーバに負荷がかかり要求された処理がなかなか終わらない、もしくは帰ってこない場合、ユーザにはサーバがどのような状況であるかを判断する情報がない。そのため、ユーザにはコンテンツをいつ閲覧できるのか分からないという不満が残る。本研究では、ソフト的に Web サーバへのアクセス数を制御することで Web サーバの負荷を下げ、またユーザに対してコンテンツの閲覧できる時間を伝えることの出来る Web サーバデーモンの開発を目指す。

## 2. 負荷についての議論

Web サーバは CPU パワーの不足、メモリ不足、ディスク I/O の不足といったハードウェアリソースの不足により過負荷状態になる。しかし、Web サーバの場合それらが複合的に絡んでくるので個々にモニタリングするのは非効率的である。本研究では「ユーザからリクエストが届き、Web サーバがレスポンスを完了するまでの時間」を負荷と見る。

## 3. 負荷実験

実際に Web サーバに掲示板プログラムを置き、Perl スクリプトを用いて短い文章を連続投稿することで負荷をかけてみる。どのような状況においてサーバのシステムダウンが引き起こされるかを実験により計測することにした。

Web サーバの環境は、下記の通りである。

- CPU: Athlon550MHz
- Memory: 512MB
- OS: FedoraCore2,
- サーバデーモン: Apache2.052

クライアントの環境は、下記の通りである。

- CPU: Pentium4 2.6C
- Memory: 512MB
- OS: WindowsXP SP2
- Perl: ActivePerl5.8.4

図 1、2 がその実験結果の一部である。

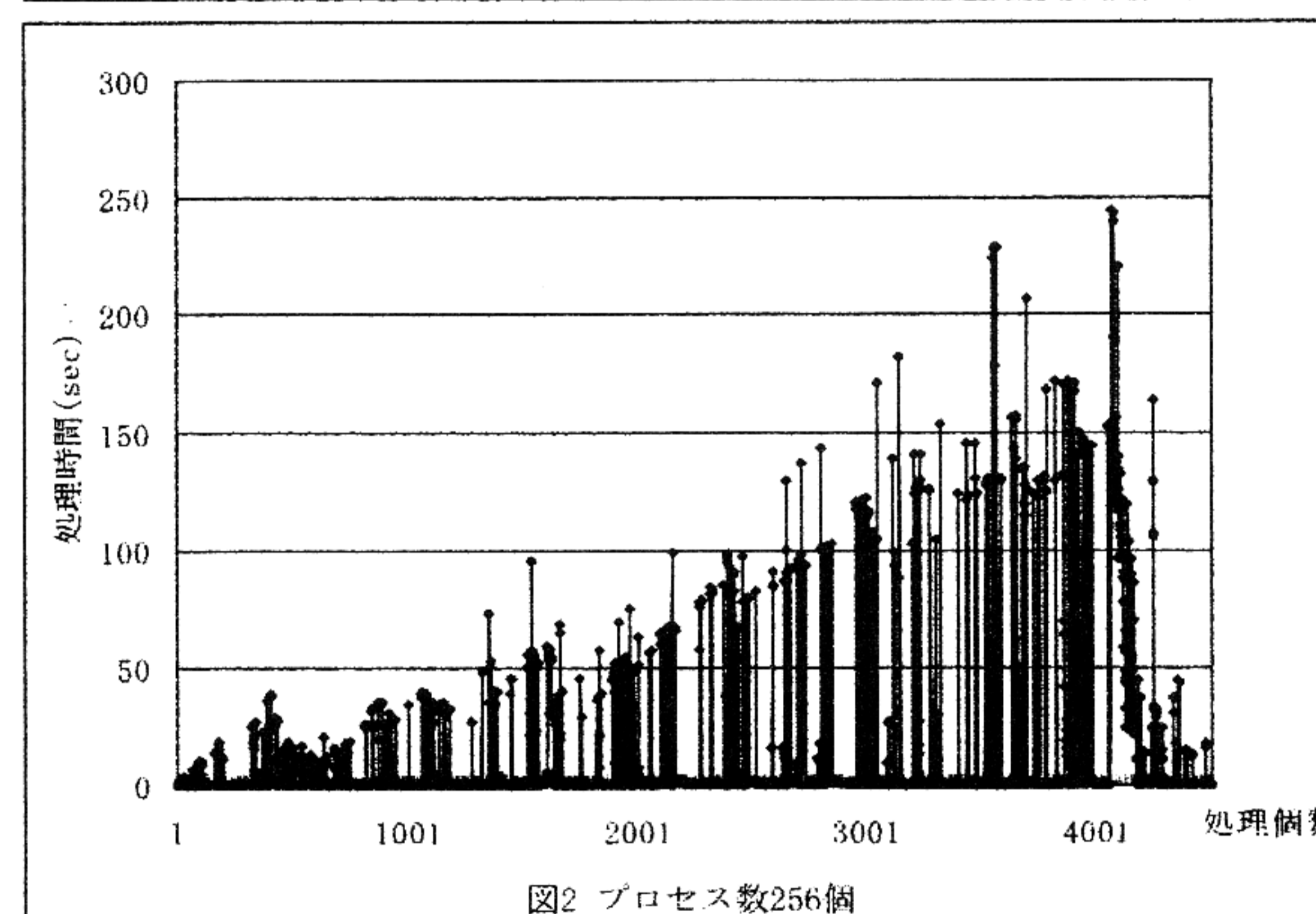
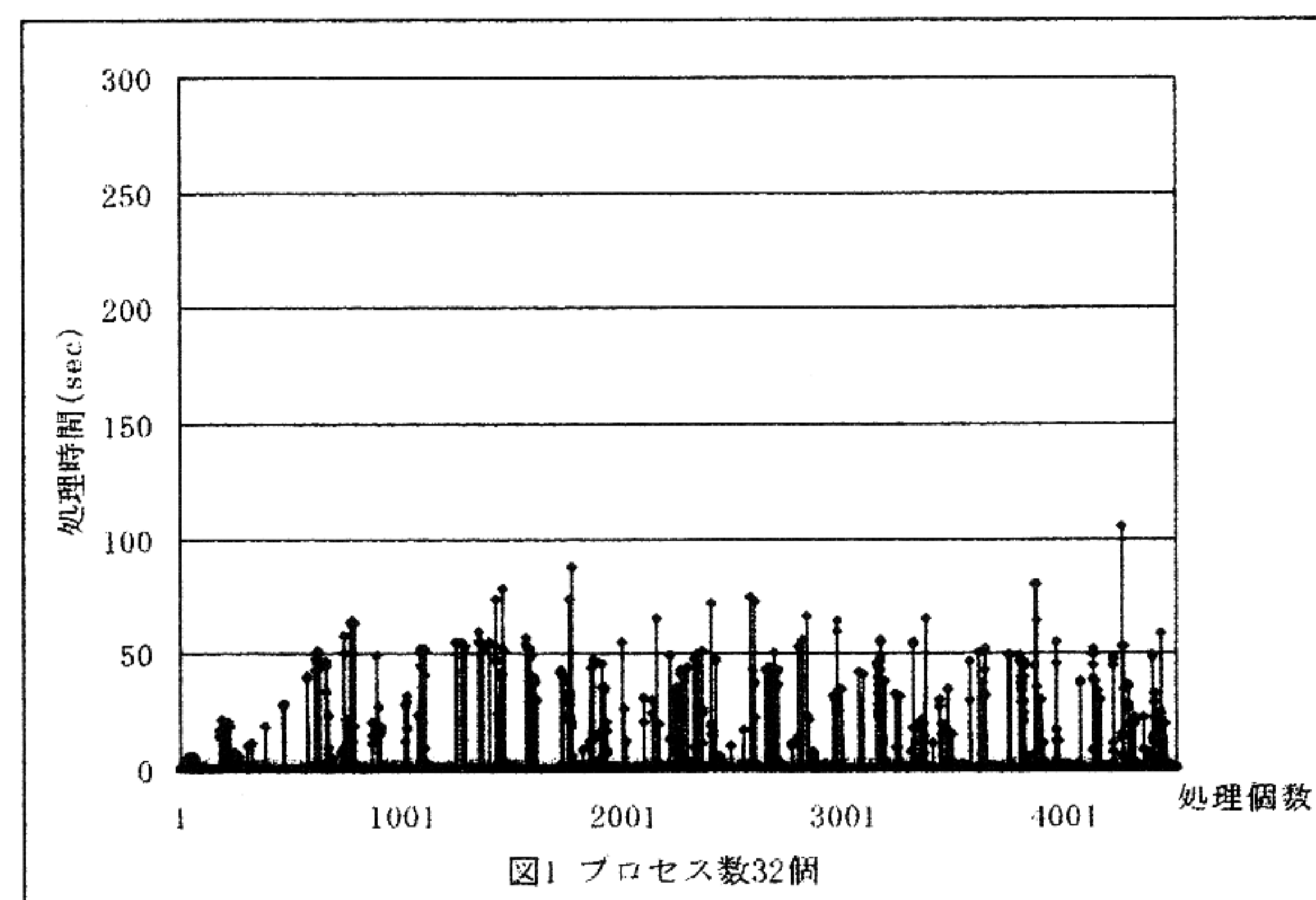


図1のように負荷が大きくないときは全体的に処理に時間のかかったプロセスがばらけているが、図2のようにシステムに負荷がかかると、処理がいつまで経っても終わらず貯まりに貯まって最後にはシステムダウンしてしまう、長時間いつまで経ってもレスポンスの返ってこないアクセスがある、といった結果が出た。

## 4. システム設計

以上の結果を踏まえて、次のような機能を持つシステムを考案した。

- サーバに負荷がかかっている場合アクセス制御を行う。
- アクセスを拒絶されたユーザに対して次に接続できる時間帯を保証する(整理券システム)

以上の機能により、負荷がかかっている時にはユーザには早いレスポンスを提供し、かつシステムダウンを予防できる。また整理券システムはアクセスを拒絶されたユーザの不満を減らすことが出来る。

## 5. 実装

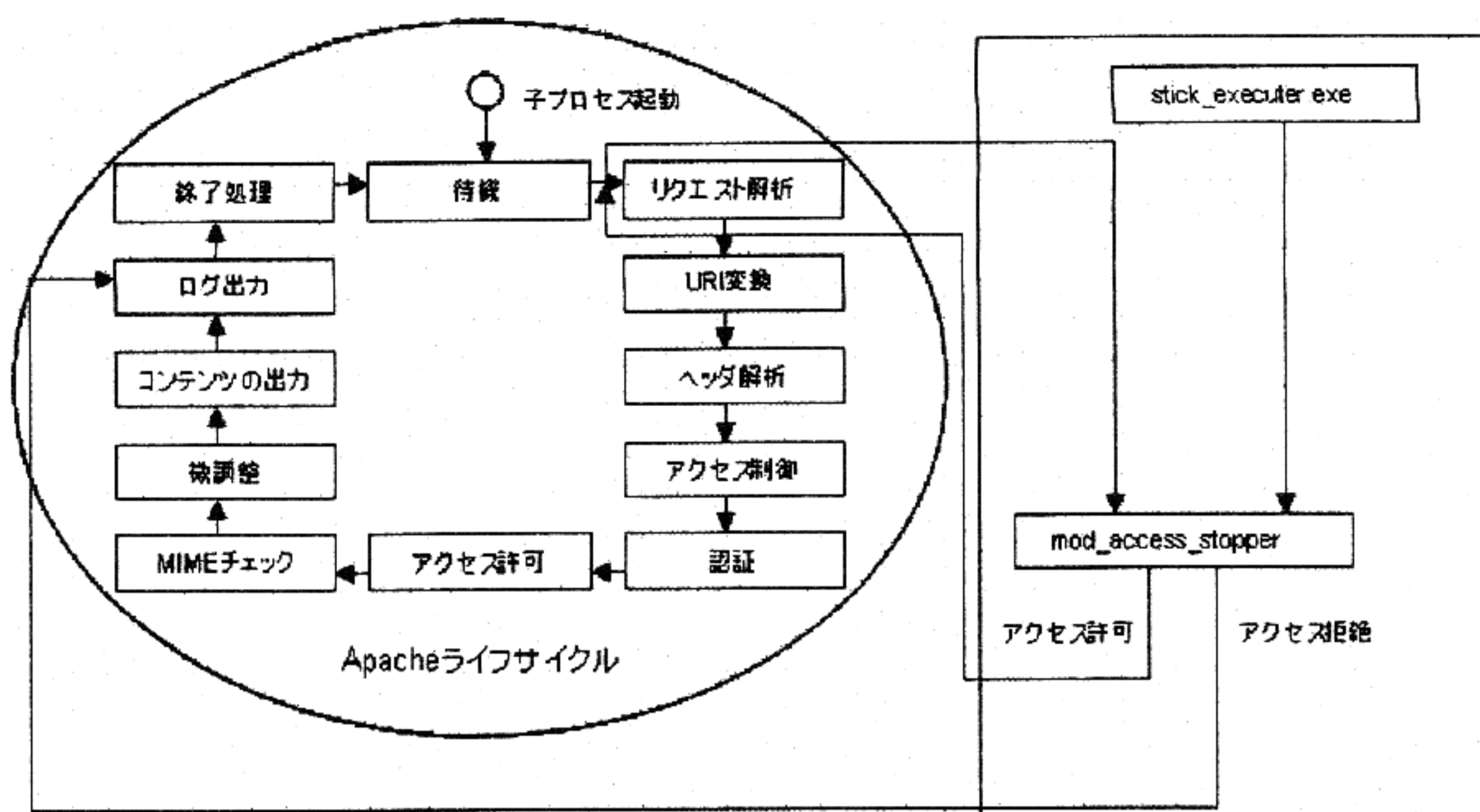


図3 実装したシステム

本論文では整理券システムの実装までは行ってない。図3の四角部分で囲んであるサーバに負荷がかかっている際にアクセス制御を行う Web サーバデーモン「Apache」[1]モジュールの開発までを行った。

Apache のリクエスト解析前にフックをかける形で実装している。外部プログラムにより負荷状況を診断する。負荷状況は Apache のログに現れる、各プロセスの処理時間を用い、一番最近の 100 アクセスの処理時間平均を取ることで計算する。そのデータをモジュールが受け取りアクセスの許可/不許可を行うものである。アクセスの不許可時には http エラーコード 503 をクライアントに返却する。

## 6. 評価実験

負荷実験と同じ環境で実装したシステムの評価実験を行った。負荷実験と同じく、掲示板に連続書き込みを行う。異なる点は、アクセス拒絶時に 10 秒間クライアント側が書き込みを待つ仕組みにしていることである。

図4, 5がその結果の一部である。

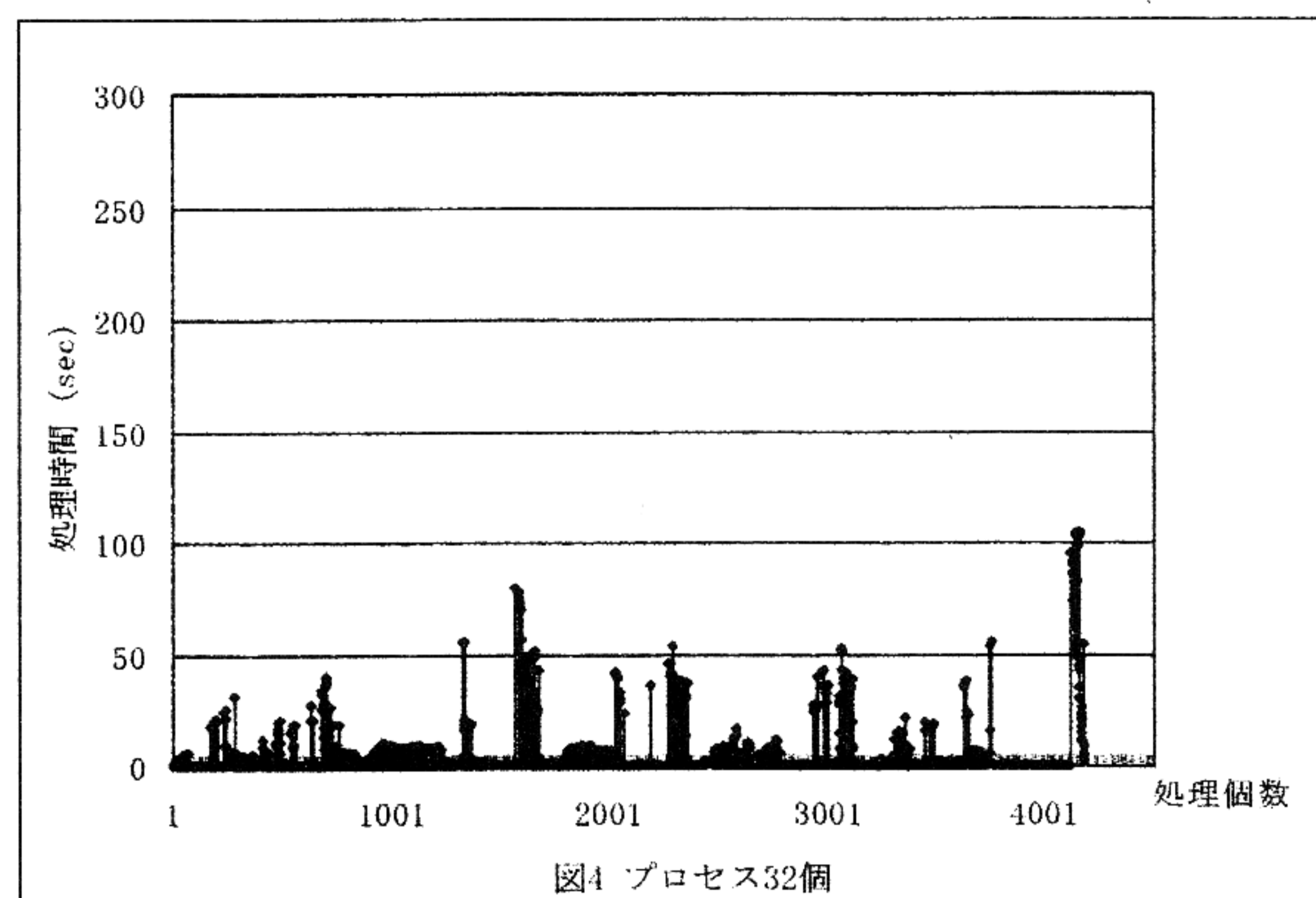


図4 プロセス32個

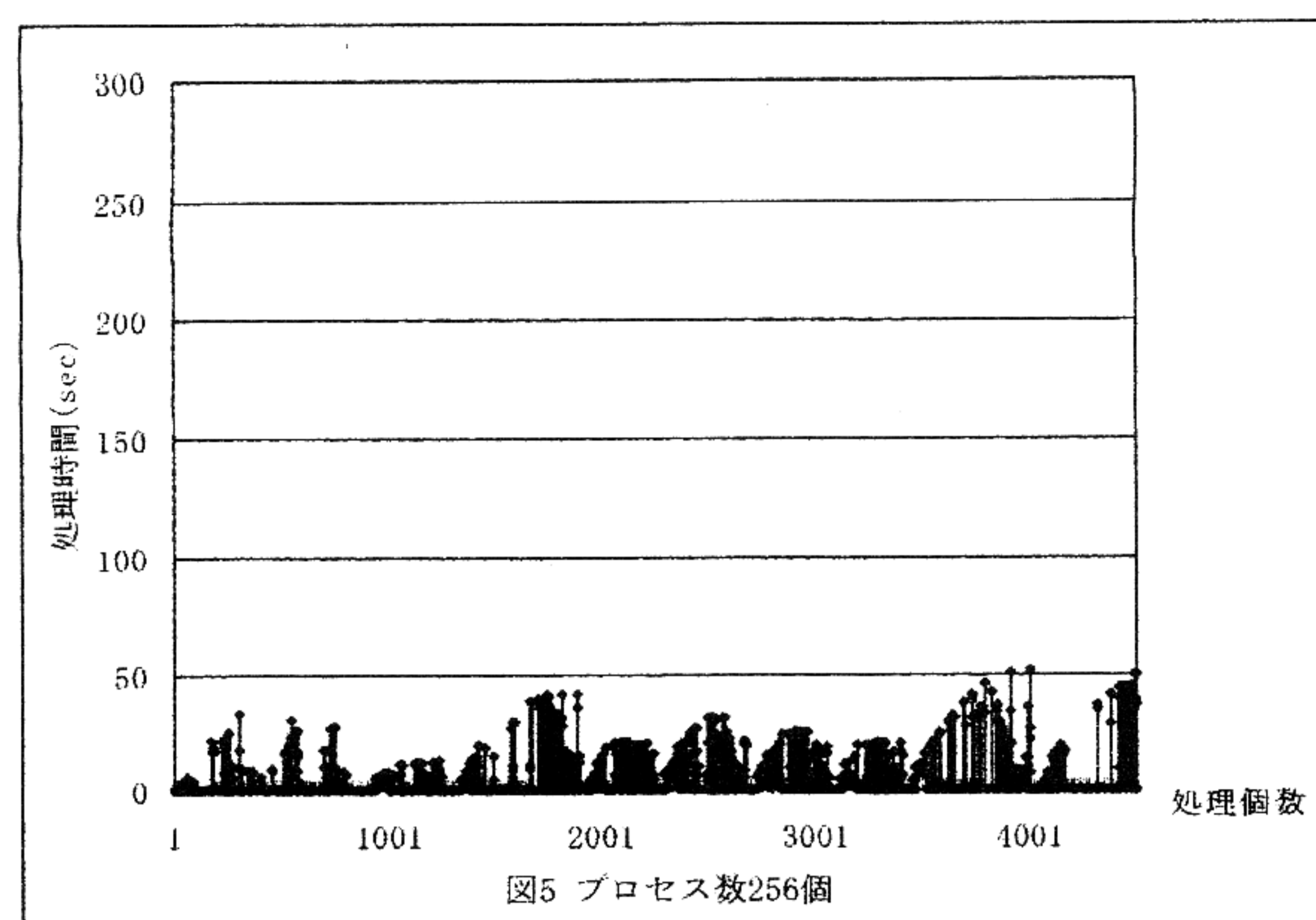


図5 プロセス数256個

図2と図5を比較しても解るとおり、仕事が貯まりすぎるといふこともなく、かつ図2のように4分を超えるほど処理に時間がかかるようなこともなくなった。

## 7. まとめ

Web サーバの負荷を制御することを目的として本研究を行った。負荷実験を行い、サーバに負荷がかかるとどのような状況になるのかを調べた。負荷を抑えるためのシステムを設計し、その一部を実装した。実装したシステムの負荷実験を行い、システムの有効性を確認した。現在のシステムの問題点は以下のとおりである。

- (ア) ファイル I/O を多用しているため、負荷判定をタイムリーに行うことができない
- (イ) 拒否したリクエストに対し一律に 503 エラーを返すため、モジュールによる拒否なのか他の原因による拒否なのか解らない。

今後の課題としては、次のことが挙げられる。

- 負荷判定用のデータを共有メモリ領域に置くことで判定速度の向上をはかる。
- 負荷判定用データの計算アルゴリズムの改善
- 並列処理による速度向上の可能性の検討
- 拒否したリクエストに対してモジュールによって拒否されたことが判断できるような専用のエラーページの作成。
- 整理券システムの実装
- さまざまな Web アプリでの本システムの有効性の確認

## 参考文献

- [1] 小山浩之著：Web エンジニアのための Apache モジュールプログラミングガイド、技術評論社、2003