

Web 負荷ツール「WS2L」の改良と評価

05t204 市田哲也（最所研究室）

アクセス制御を行う Web システムの開発やその評価を行うための、Web 負荷ツール WS2L の改良と新たに必要とされる機能追加、及びその評価について述べる。

1. はじめに

当研究室では、Web サーバの負荷を軽減するために、Web サーバのミラーリングによる負荷分散やアクセス制御による順序制御に関する研究を行っている。提案の Web サーバ[1]の評価および問題点の抽出のために WS2L と呼ぶ Web 負荷ツールの開発を行っている。

本研究では従来の WS2L に対して、これまで不十分であったマルチスレッドの対応や動的負荷量変更機能などの新たな機能を追加し、その評価を行った。

2. 概要

WS2L の構成を図 1 に示す。WS2L は 2 つのモジュールに分かれており、負荷のシナリオの決定や結果を出力する制御実行モジュールと、与えられたシナリオに基づいて Web サーバに負荷(リクエスト)をかけ、ログ情報を記録する負荷実行モジュールから成る。シナリオとは複数のリソースに順番にアクセスを行う一連の処理である。図 1 では各モジュールをそれぞれ 1 台ずつコンピュータに置いているが、全てのモジュールを 1 台のコンピュータに置くことや、負荷実行モジュールを複数台のコンピュータに置くこともできる。昨年度までの成果としては、負荷実行モジュールのいくつかの機能の実装と、制御実行モジュールのインタフェースについての設計、実装がされた[2][3]。しかし、

- ・ マルチスレッドへの対応
- ・ 制御実行モジュールの統一

の改良および、

- ・ 動的負荷量変更機能
- ・ 負荷量表示機能

の追加が求められていた。

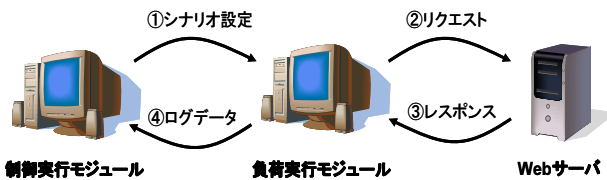


図 1. WS2L の構成

3. 機能の設計

上記で述べた、本研究で改良する仕様と追加する機能の説明と設計を行う

3. 1 マルチスレッドへの対応

WS2L では 1 スレッドが 1 クライアントをシミュレートしている。よって多くの負荷を与えるためには複数のスレッドで動作させる必要があるが、昨年までの実

装でも複数のスレッドの生成は可能であったが正しく動作しないことがあった。これは各スレッド間での同期がとれていなかったことが原因と考えられる。そこで、適当な箇所にミューテックスを入れることで、この問題を解決することにした。

3. 2 制御実行モジュールの統一

昨年度までの WS2L では、図 2 に示すようにログ転送を ControlTool で行っていた。つまり制御実行モジュールとして ControlModule と ControlTool の二つを必要としていた。ControlTool はログを受信し、ログ情報を書き込むだけの処理を行っており、この部分を別モジュールにする必要がないと考え、ControlTool の機能を ControlModule と統合することにした。統合を行うことでログの管理機能が統一され、効率的なログ管理を行うことができるようになる。これを実現するために ControlTool の通信機能をそのまま ControlModule 内に移動し、新しいスレッド上で動作させることにした。

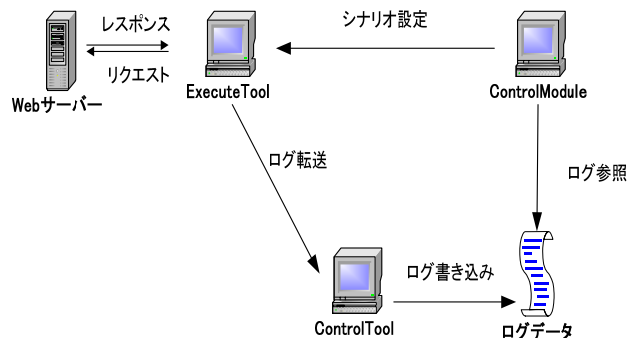


図 2. 昨年度までの WS2L の仕様

3. 3 動的負荷量変更機能

動的負荷量変更機能とは、テスト中に負荷量つまりスレッド数を変更する機能である。この機能は、リクエストの時間的な増加や減少をシミュレートするためのものである。このため、新たに負荷量を変更させる命令を追加する。さらに、その命令とともに変更させる値を制御実行モジュールから負荷実行モジュールに送信し、その命令を受信した負荷実行モジュールは受信した値に負荷量を変更できるようにする。また変更間隔も設定できるようにする。この機能は 1 つのスレッドを作成、終了するときに指定した時間、待機することで実現できる。変更間隔だけ待機することで負荷量の少ない変化でも長い時間をかけて少しずつ変化する状況をシミュレートすることが可能となる。

3. 4 負荷量表示機能

負荷量表示機能とはテスト中、現在の負荷テストの負荷量を表示する機能である。前節で述べた動的負荷量変更機能で負荷量を変更する際、現在の負荷量を把握して負荷量を変更するほうが基準が分かるので変更が容易になると考え、追加することにした。具体的には、負荷量の情報などを追加したログデータを負荷実行モジュールが送信し、受信した制御実行モジュールがその情報を読み出し、表示するというものである。

4. 実装及び評価

以上の設計に基づき実装を行った。開発環境はWindowsXP 上の Visual C++および.NET Framework2.0 である。以下では動的負荷量変更機能の動作確認とWS2L の評価テストについて述べる。

4. 1 動的負荷量変更機能の動作確認

30 秒間負荷量(スレッド数)1 でテストを行い、約 15 秒のときに負荷量5に変更したテストのグラフを 図3 の左に、30 秒間負荷量5 でテストを行い、約 15 秒のときに負荷量1 に変更したテストのグラフを 図3 の右にそれぞれ示す。どちらも変更間隔は 0[s] である。グラフは横軸がテスト時間、縦軸が応答時間を示している。どちらも 15 秒のときから約 2 秒で変化が起きており 図3 の左側では負荷量1 から 5 の値に変化し、右側では負荷量5 から 1 の値へ変化していることが分かる。またその変化は約 2 秒間で完了している。

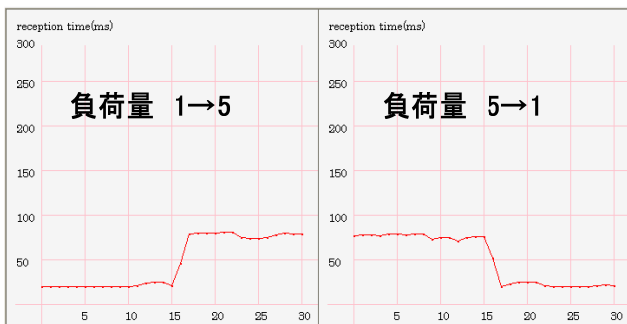


図3 変更間隔 0[s] で負荷量を変化させた場合

次に、変更間隔を 2[s] にしてそれ以外の条件を 図3 の場合と同一にして実験を行った結果を、 図4 に示す。変化に約 8 秒かかっており、変更間隔が反映されていることがわかる。よって動的負荷量変更機能は正しく実装されていると判断できる。

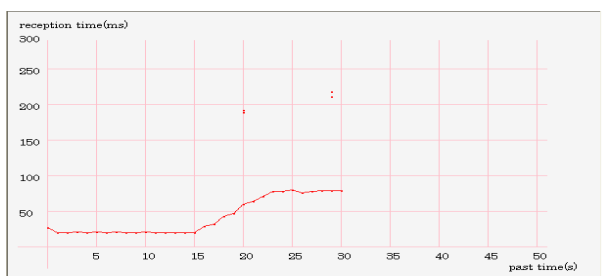


図4 負荷量1 から 5,変更間隔 2[s] で変化させた場合

4. 2 複数のクライアントの評価

テスト時間 30 秒でサーバに 1 台と 2 台の場合でテストを行い、負荷量を変化させたときの接続回数と応答時間を調べた。その結果を 図5 と 図6 に示す。

接続回数は 図5 に示すように負荷量 1 のときは、2 台のほうが 1 台のときより高いが、負荷量を増加させた場合、両方とも 1700 前後に収束していることが分かる。これは対象サーバの単位時間当たりの処理限界が 1700 前後であるためと考えられる。

応答時間は 図6 に示すように両方とも負荷量に応じて増加している。特に 2 台の場合は 1 台の場合に比べて約 2 倍の値になっている。これは対象サーバに対して 2 倍の負荷をかけることでサーバへの同時接続数が増えたため処理時間が増えたためと考えられる。よって WS2L は正常に動作し、複数台でも正しく動作していると言える。

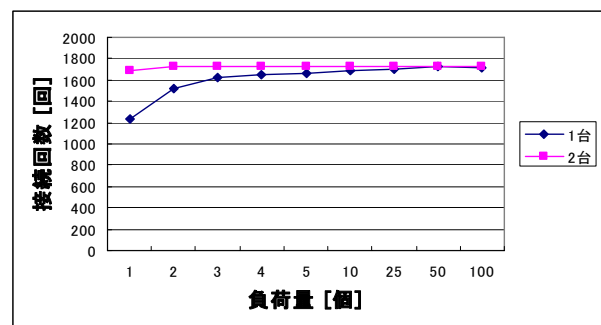


図5 負荷量を変化させた場合の接続回数の変化

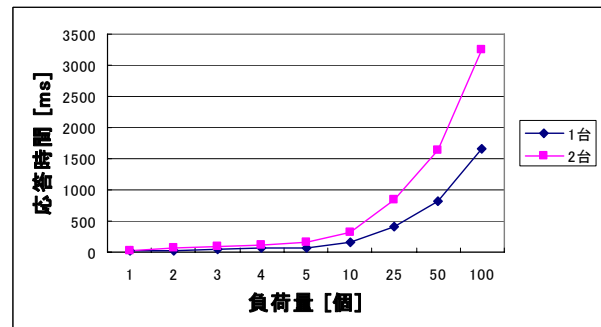


図6 負荷量を変化させた場合の応答時間の変化

5. おわりに

本研究では WS2L の機能の改良及び追加を行い、その動作確認と評価を行った。本稿では述べていない部分もあるが改良・追加した機能は全て正しく実装されていることが確認できた。今後の課題としてテスト中のリアルタイムグラフにおいてログデータの転送の遅延を解消し、より正確なリアルタイムグラフの表示などがある。

6. 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] 小笹光来, “Web 負荷ツール「WS2L」の負荷実行モジュール部の改良”, 香川大学工学部, 学部論文 2007
- [3] 熊代崇紀, “Web 負荷ツール「WS2L」の制御集計モジュールの開発”, 香川大学工学部, 学部論文 2007