

WEB 負荷ツール「WS2L」の負荷実行モジュールの改良

04T287 小笹 光来（最所研究室）

当研究室の Web システムの開発に必要な情報を得るため、「WS2L」と呼ぶ負荷テストツールを開発している。本研究では WS2L の負荷テスト実行モジュールの改良及び機能拡張を行う。

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次回アクセス保証を行う Web システム (NAP-Web[1]) の研究を行っている。

これらの研究では、開発している Web サーバの評価および問題点の抽出のために Web 負荷ツールと呼ばれるツールを使用しているが、既存のツールでは開発に必要な情報を十分に得ることが出来ない。このため、当研究室で行っている Web システムの開発に必要な情報を得ることが出来る Web 負荷ツール、「WS2L[2]」の開発を行っている。

本研究では、WS2L の Web サーバに負荷を与えるモジュール (負荷実行モジュール) の改善及び機能の拡張を行う。

2 WS2L の概要

WS2L は図 2 に示すように 2 つのモジュールから構成されている。一つは負荷テストのシナリオ決定や結果を出力する制御実行モジュールと、もう一つは与えられたシナリオに基づいて Web サーバに負荷をかけ、その時の応答時間などを制御実行モジュールに返す負荷実行モジュールである。

WS2L の主な対象は、当研究室にて開発中の Web サーバシステム NAP-Web である。NAP-Web はクライアントからサーバへのアクセス時に、サーバが現在の負荷状況を判定し、クライアントのリクエストを許可/拒否するシステムである。拒否されたクライアントは指示された時刻でのアクセスを保証される。

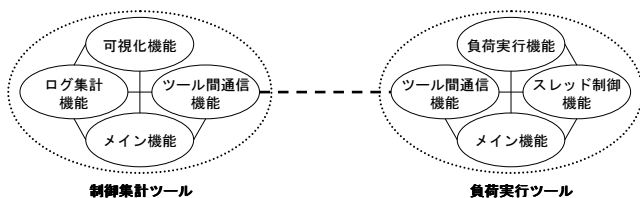


図 1: WS2L の構成

3 負荷実行モジュールの設計

NAP-Web の開発に必要な情報を得るために、負荷実行モジュールに以下の機能を追加した。

POST 機能

HTTP メソッドで POST を設定スクリプトで指定できるようにする。設定スクリプト内の <header> ノードの <method> 値に "POST" を指定し、<transaction_script> ノードの <meth_val> 値に送信する内容の値を指定する。

HTTP リダイレクト機能

ステータスコード「301」又は「302」がレスポンスとして返されると、「Location: 」ヘッダよりリダイレクト先 URL を取得し、リダイレクト先へ再度アクセスを行う。Location ヘッダにはリダイレクト先が相対パス又は絶対パス (URL) で表され、相対パスの場合はリダイレクト元と同一ホスト、絶対パスの場合は URL で示されるホストへアクセスする。

再アクセス機能

NAP-Web が再アクセス保証を行う場合アクセス拒否を行うが、その際にクライアントに対して HTTP ステータスコード 503 をレスポンスヘッダとして返し、再アクセスの権利であるチケット ID をクッキーとして配布する。待ち時間は本文に含めて渡す。負荷実行モジュールはレスポンス 503 を受信し、かつクッキーとして NAP-Web のチケット ID を受信した場合、受信した HTML 内にコメント形式で記される待ち時間分だけ待機した後に再アクセス処理に移る。再アクセスのためのトリガとなるクッキー名などのパラメタは設定シナリオから変更する事が出来る。

アクセスグループ機能

トランザクションを実行するスレッドをグループに分け、それぞれ再アクセス時の動作を変更させる機能である。グループには以下の 4 つがある。

- early : 指定された再アクセス時刻より早く再アクセスを行うグループ
- slowly : 指定された再アクセス時刻より遅く再アクセスを行うグループ
- just : 指定された再アクセス時刻丁度に再アクセスを行うグループ
- never : 指定された再アクセス時刻に関わらずアクセスを行わないグループ

アクセスグループには出現確率を設定する事ができる。図 3 にその例を示す。

図 3 中のグループノードの要素の値で出現確率を指定する。全体の負荷量を 1 とした時の割合であり、それぞれの値の合計は 1 でなければならない。

```

- <setting_script>
  <load host="all">1000</load>
  <runtime>00:00:20</runtime>
  <warmup>00:00:00</warmup>
  <cooldown>00:00:00</cooldown>
  <early time="5">0.2</early>
  <slowly time="10">0.2</slowly>
  <just time="0">0.3</just>
  <never time="0">0.3</never>
</setting_script>

```

図 2: アクセス時刻の分散の設定例

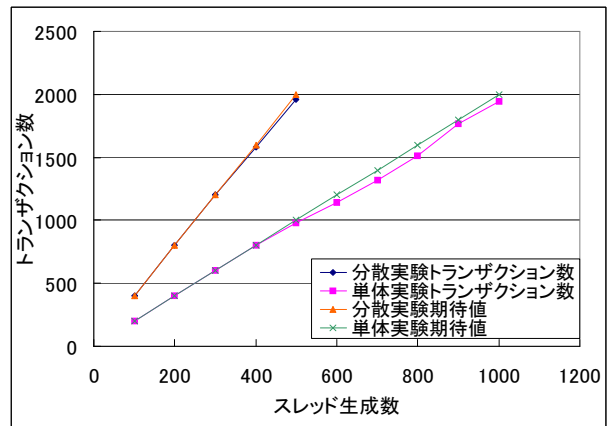


図 3: スレッド生成数とトランザクション数

アクセス保証時刻に対するユーザの振る舞いのシミュレート機能

NAP-Web サーバにより示された再アクセス指定時刻に対し、負荷実行モジュールが再アクセスを行う時刻を分散させ、ユーザの動作をシミュレートさせる機能である。

どの程度再アクセス時刻を分散させるかは、設定スクリプト内のアクセスグループの後に続くアトリビュート値で指定する。値は秒単位である。グループが early ならば、何秒の範囲で再アクセス指定時刻より早く再アクセスを行うか、slowly ならば、何秒の範囲で再アクセス指定時刻より遅く再アクセスを行うかを指定する。分散させる範囲は指定した秒数内でランダムに決定される。just と never に関しては现阶段では必要ないが、拡張のために値を設けている。

4 実装

以上の設計に基づき、負荷実行ツールの実装を行った。開発環境は、Windows XP 上で Visual C++および.NET Framework2.0 を使用した。

5 評価

一つのスレッド内で二つのトランザクションを実行するシナリオを用意し、100 から 1000 まで負荷量を変化させるテストを単体の場合と 2 台の場合について行った。

図 5 はスレッド生成数とサーバに送信されたトランザクションの関係を示している。スレッド数が 400 までは期待通りのトランザクションを実行できているが、それ以上スレッドを生成するとトランザクション数が低下している。2 台の負荷実行モジュールを用いた負荷テストの場合、単体で同数のスレッドを実行する場合と比較して期待値と実験値の差が小さく、安定して負荷をかける事が出来ている。

また、単体実験に用いた PC ではスレッドの最大生成数が 1294、1 秒間のアクセス数を平均すると 31.9 であり、最低値は 20 であった。このことから今回の環境では少なくとも 20 回以上の同時アクセスが可能であると思われる。また、分散実験では、台数に見合った負荷を発生させられる事が確認できた。

次に再アクセスの評価を行う。表 1 は再アクセスの際の各アクセスグループの出現比率を 2:2:3:3 に設定

表 1: 各アクセスグループの出現比率

再アクセス数	early	slowly	just	never
373	0.13	0.34	0.43	0.10
152	0.33	0.07	0.37	0.23
357	0.13	0.25	0.24	0.38
477	0.17	0.12	0.27	0.44
592	0.22	0.30	0.26	0.23
平均	0.20	0.22	0.31	0.28

した場合の出現比率である。各グループの出現比率を平均するとほぼ期待値通りとなることが確認できた。

6 おわりに

NAP-Web の評価を行う事ができるように WS2L の負荷実行モジュールの改良を行った。改良した負荷実行モジュールはスレッド数を増やすと多少性能は落ちるが、負荷テストを行えるだけの負荷をサーバにかけられることを確認した。分散配置での負荷テストでは、台数に見合った負荷をかけることに成功した。

今後の課題として以下の機能を追加及び改良を行う必要がある。

- TCP 同時接続数問題
- HTTP1.1 への対応
- 動的負荷変更機能の追加
- スレッド管理機能の強化

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 ”, 香川大学工学研究科, 修士論文 2006

[2] 岡村尚紀, “ Web 負荷ツール「WS2L」の構築 ”, 香川大学工学研究科, 学部論文 2006