

Web 負荷ツール「WS2L」の構築

03T211 岡村尚紀（最所研究室）

種々のアクセスパターンで Web サーバに負荷をかけることができ、そのときの詳細なアクセスログを取得できる Web 負荷ツール WS2L を提案し、その設計および実装を行う。

1. はじめに

当研究室では、近年増大する Web サーバの負荷を軽減するため、Web サーバの過負荷に対し順序制御によりサーバのダウンを防ぐ Web サーバやミラーリングの一手法としての動的ミラーリング機構の研究を行っている[1][2]。これらの研究では、Web 負荷ツールと呼ばれる Web サーバの性能評価に用いるツールを使用している。しかし、多くの Web 負荷ツールでは Web サーバの性能を評価できるが、当研究室の Web サーバ開発に必要な情報を得ることができない。本論文では、当研究室の Web サーバ開発に必要な機能を持つ Web 負荷ツール WS2L の設計、実装および評価について述べる。

2. Web 負荷ツール

Web 負荷ツールとは、擬似的に複数のクライアントを生成し、これらに指定された順にリソースを取得させ、同時にリソースの取得で得られる情報を記録し、集計するものである。記録から Web サーバのスループット等を求め、Web サーバの性能を評価することができる。

しかし、一般の Web 負荷ツールの機能では、当研究室で開発中の Web サーバの開発に必要な情報を得ることができない。そこで、この開発に必要な下記の機能を持つ Web 負荷ツール「WS2L」を開発することにした。

- 情報を記録するログ機能。
- 設定を動的に変更できる機能。
- インターフェースに関する可視化機能。
- プログラムの分散配置。

3. WS2L の設計

3.1. システム概要

WS2L の機能を制御集計ツールと負荷実行ツールの 2 つに分割し、それぞれを独立に設計する。制御集計ツールは、負荷実行ツールを制御し、ログデータを集め、それらの集計を行う機能を持つ。負荷実行ツールは、スレッドを生成し、それらを Web サーバにアクセスさせ、情報を記録する機能を持つ。

また、それぞれのツールは、ツール間でメッセージを送りあうツール間通信機能を持つ。制御集計ツールのツール間通信機能は、負荷実行ツールに指示を出し、負荷実行ツールからログデータや報告を受ける。負荷実行ツールのツール間通信機能は、制御集計ツールからの指示を受け、ログデータや現在の状態を制御集計ツールに送る。WS2L の機能分割の

イメージを図 1 に示す。なお、図中のメイン機能は、それぞれのツールの機能をまとめ、調整するものである。

ツール間でやり取りされるメッセージのうち、制御集計ツールから負荷実行ツールに送られるメッセージを制御コマンドと呼ぶ。制御コマンドには、負荷実行ツールへの命令として、以下の 4 つがある。

- SR：負荷実行ツールの状態を問い合わせる。
- SS：スクリプトを設定する。
- TS：負荷テストを開始する。
- CC：負荷テストを中止する。

スクリプトとは、負荷テストの設定内容を記述したファイルである。

負荷実行ツールから制御集計ツールに送られるメッセージを応答レスポンスと呼ぶ。応答レスポンスは、制御コマンドに対する応答として、負荷実行ツールの状態や記録したデータを持つ。

本要旨では紙面の制限のため、負荷実行ツールについてのみ説明する。

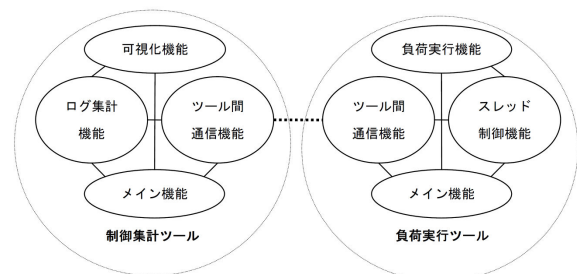


図 1 WS2L の機能分割

3.2. 負荷実行ツール

制御集計ツールから見た負荷実行ツールの状態を表すステータスを定める。制御集計ツールは、ステータスを基に次に行う処理を決定することになる。ステータスの種類は、以下の通りである。

- Wait : 制御集計ツールからの命令待ち状態。
- Set : スクリプトが設定されている状態。
- Running : 負荷テストの実行中状態。
(準備・変更・中止・終了を含む。)

ステータスは、制御コマンドの受け取ることにより遷移する。ステータスおよび制御コマンドと遷移の関係を図 2 の状態遷移図に示す。

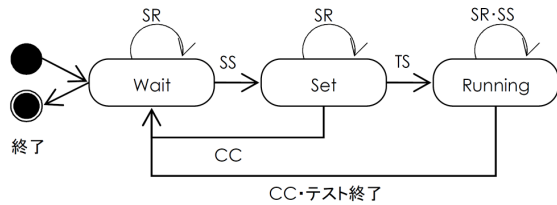


図2 ステータスの状態遷移

次に、負荷実行ツールが制御ツールからの指示を受け、テストを実行する流れを図3に示す。1スレッドは1クライアントに相当し、擬似クライアントに見立てたスレッドがスクリプトに従って複数のリソースに対して順番にアクセスを行う。図中のシナリオとは、複数のリソースに順番にアクセスを行う一連の処理、トランザクションとは、シナリオの中で一つのリソースにアクセスしデータを記録する処理を表す。

スレッド制御の機能は、スクリプトで指定されたスレッド数が現在生成しているスレッド数より多い場合にスレッドを生成する。各スレッドはシナリオを実行し、シナリオでは複数のトランザクションが順に実行される。各トランザクションにおいては、Webサーバにアクセスしデータを記録する。シナリオが終了するとスレッドは記録されたデータを制御集計ツールに転送し、生成されてからの一連の処理を終える。実際のログ転送は、指定された量のログ(ログ転送量)が溜まったときにまとめて送られる。一連の処理の後、スレッドを継続するかをチェックを行い、テスト終了やスレッド数を減らすなどの指示があれば自身で消滅し、指示がなければ繰り返し処理を行う。

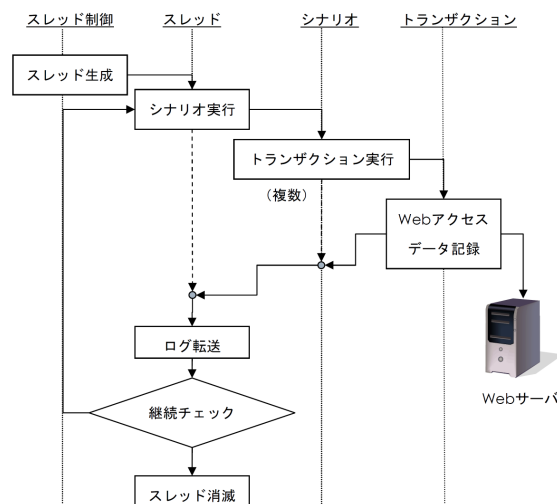


図3 テスト実行の流れ

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。制御集計ツールとして、実装した負荷実行ツールを制御するために最低限必要な機能を持たせたものを作成した。開発環境は、Windows XP 上で

Visual C++および.NET Framework 2.0 を使用した。

5. 評価

実装した WS2L を用いて簡単な機能評価を行った。評価実験は、同一 LAN 内のコンピュータを用い、スクリプトの項目を変え、1 分間の負荷テストを 10 回行い、アクセスシナリオ数とシナリオログ数を記録し、それらの平均を求めた。負荷量を 180 に固定し、ログ転送量 (forward_quantity) を変化させた場合の結果を図4に示す。図4より、ログ転送量が増加するとサーバへのアクセス数が増えていることが確認できる。これは、ログ転送量が小さいときに、性能が低下していることを表す。また、アクセスシナリオ数(実際のアクセス数)に比べ、シナリオログ数(記録されたアクセス数)が少ないことが分かる。この原因は分かっていないが、処理が中止されたスレッドのデータを転送させる機能を実装することで特定できると考えられる。

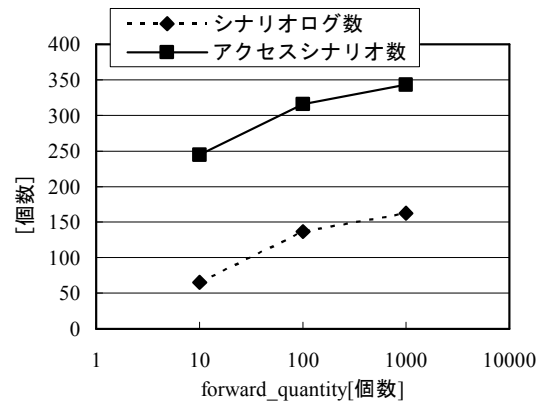


図4 forward_quantity を変化させた場合

6. おわりに

本研究では、当研究室で開発中の Web 負荷ツールの開発に必要な WS2L の開発を行った。WS2L に必要な機能をまとめ、その設計、実装および機能評価を行った。評価の結果、ログ転送量を増加させることにより、Webサーバへのアクセス数が増えることが確認できた。しかし、今回の実装に関して言えば、ログ転送機能で処理が滞り性能が低下する問題、シナリオログ数が減少する問題が発生した。

今後の課題として、制御集計ツールの完成、負荷実行ツールにおけるスレッド処理の効率化、処理が中止されたスレッドのデータを転送する機能の実装などを行う必要がある。

参考文献

- [1] 加地智彦, 「次回アクセスを保障する 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] 入江正行, 「Webサーバの負荷状況に応じた動的ミラーリング機構」, 香川大学工学部, 卒業論文 2005