

認証システム向け負荷テスト方式の提案

白木 宏明[†] 大松 史生[†]

三菱電機(株) 情報技術総合研究所[†]

1. 背景と課題

企業内には複数の業務システムが存在している。業務システムは、社員など企業内ユーザに対してアクセス制御を行う必要があり、認証機能を保有している ([1],[2])。この際、各業務システムを利用するごとにユーザが認証操作を行うことは非効率であることから、一度の認証操作にて複数の業務システムを利用可能とする統合認証システム（以降、認証システム）を導入する企業が増加している。

認証システムは、企業内において共通システムの位置づけにあることと、業務システムを利用する際に必ず利用されるため、負荷が集中する。そのため、認証システムを構成するサーバリソースの見積もりを行うためには負荷テストは必須である。負荷テストを実施する際には、リソース等の制限により小規模の負荷テスト結果を基に本番システム向けの負荷を算出していたが、システムの大規模化・仮想化により、単純な算出では問題が発見できなくなっており、本番システム相当の負荷テストが必要となっている。

2. 認証システムにおける負荷テスト

2.1.従来方式の負荷テスト方式

従来、認証システムの負荷テストを実施する場合、負荷テストツール ([2][3][4]など) を使用するケースが通常である。これらの負荷テストツールは、クライアントアプリケーション（例えば、ブラウザ）をエミュレートして、認証サーバに負荷をかけるものである（図 1）。この際、システムの負荷テスト要件にあった負荷をかけるためには、負荷テストツールのパラメータ調整が必要であるが、パラメータ調整については、テスト実施者の経験と実績から試行を繰り返すことで決定していた。

また、大規模な負荷テストを実施する際には、多数の負荷テストエージェントが必要なため、複数の負荷テストコントローラが必要となり、さらにコントローラの制御も必要であった。

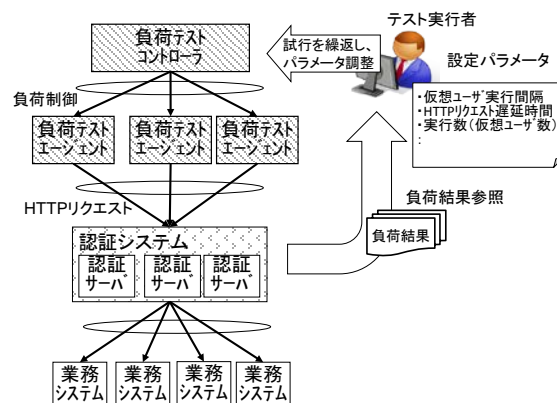


図 1 従来の負荷テスト構成

2.2.従来方式の課題

大規模な負荷テストを実施する際には、テスト環境を占有するため実施期間が限られる。しかし、従来の方式では負荷テスト環境を物理的に構築する以上に、負荷テストツールのパラメータ調整に時間がかかるという課題があった。

3. 認証システム向け負荷テスト方式の提案

前章の課題を解決する負荷テスト方式として、「負荷要件に合わせた負荷テストツールのパラメータ値の自動的な算出」が必要であり、これを実現するフィードバック式 負荷コントロールシステム（図 2）を提案する。

以下に特長を示す。

- ・ 認証システムのアクセスログからモデルを生成する仕組み：負荷モデル生成機能
- ・ アクセスログから抽出した負荷要件を基に負荷テストツールのパラメータ値を算出する仕組み：負荷パラメータ算出機能
- ・ 算出した負荷テストツールのパラメータを基に負荷テストツールを実行する仕組み：負荷テスト実行機能
- ・ 算出した負荷ツールツールのパラメータを基に実行した負荷テスト結果であるアクセスログから負荷モデルにフィードバックする仕組み：負荷モデルフィードバック機能

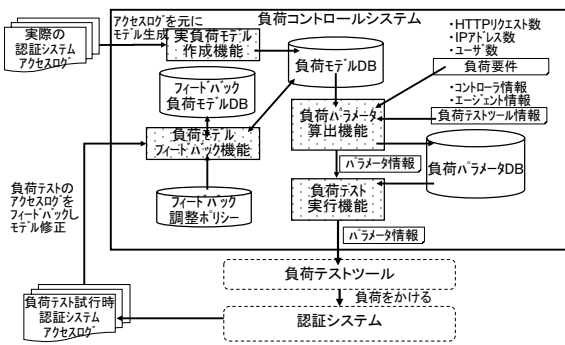


図 2 負荷コントロールシステムの構成

本方式における負荷テストツールのパラメータ算出の流れを以下に示す。

- ① 負荷テストで再現したいケースの実際の認証システムのアクセスログを取得し、負荷モデル作成機能が負荷モデルを作成（実負荷モデルと呼ぶ）
- ② 作成した負荷モデル、負荷要件、負荷テストツール情報を基に負荷パラメータ算出機能が負荷パラメータを算出
- ③ 算出された負荷パラメータを基に、負荷テスト実行機能が負荷テストツールを実行し、認証システムに負荷をかける（試行）
- ④ 負荷テスト試行時のアクセスログを基に、負荷モデルを作成

①の実負荷モデルと④の負荷モデルとが一致すれば、パラメータ値が正しいことは明確であるが、モデルが複雑であれば一致する可能性は低いと考えられる。パラメータ値を収束させるためには複数回の試行が必要となる。

初回の試行により生成した負荷モデルに対して、補正を加えた負荷モデル（フィードバック負荷モデル）を作成する。負荷モデルを補正するためのポリシーをフィードバックポリシーとして定義し、それによってフィードバック後の負荷モデルを調整する。負荷テストの試行、負荷モデル生成と補正を繰り返すことでパラメータ値を決定する。本方式の流れを以下に示す。

- 1) 負荷テスト結果からフィードバック負荷モデルを作成
 - 2) 前回のフィードバック負荷モデルと突合せ、差異とポリシーに従って負荷モデルを修正。
 - 3) 修正したモデルによりパラメータを再算出し、負荷テストを実施
- 1)～3)を負荷モデルの差が一定値内に収まるまで実施することで負荷テストツールのパラメータを決定する。

4. 負荷コントロールシステム適用結果

前章の負荷コントロールシステムを従来方式に適用したものが図 3となる。

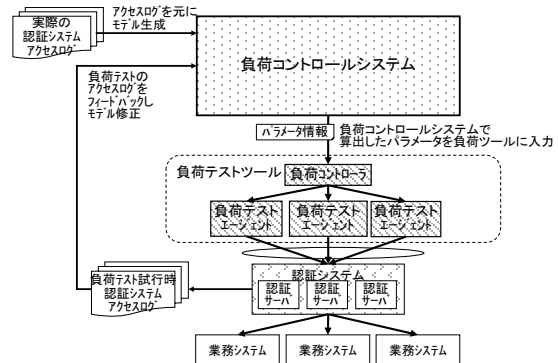


図 3 負荷コントロールシステム適用構成

この方式により、負荷要件に適したパラメータを短時間で自動算出することが可能になるとともに、負荷テストの結果をフィードバックすることにより、より最適なパラメータ値を算出することが実施可能なる。結果、負荷テスト時間の短縮が可能となり、一例ではあるが、端末数4万台、HTTPリクエスト数5000HTTPリクエスト/秒のモデルに対して、従来は約2日必要だったものを約半日に短縮可能であることを机上にて算出した。

4. まとめ

本方式では、複数の負荷コントローラサーバを制御することできめ細かな負荷テストを実施可能であること、モデル定義の追加によりさまざまな負荷テストへの対応が可能（ピークモデル、現実モデル：週初、週末、月初、月末など）であるという効果が得られる。

本稿では、現実システムのログから負荷テストツールのパラメータを自動的に算出する方式を提案した。ピークモデルに対しては、本方式が有効であることが確認できている。今後は、さまざま負荷モデルに対して、正しくパラメータ値が算出可能であることを示す予定である。

[参考文献]

- [1] 富士キメラ総研, 2014 ネットワークセキュリティビジネス調査総覧(上巻), 2014.
- [2] IPA, 情報セキュリティ白書 2014, 2014
- [3] IBM社 Rational Performance Tester
- [4] HP社 LoadRunner
- [5] オープンソース Jakarta JMeter