

# 教育支援システムへの Web アクセスの負荷計測ツールの機能検討

花川直己<sup>†</sup> 大川昌寛<sup>†</sup> 蔵本幸司<sup>†</sup> 富永浩之<sup>†</sup>

近年、多くの教育機関で、大規模で汎用的な教育管理システムが導入されている。一方、ある目的に特化したり、先進的な機能を取り入れた小規模なシステムも必要とされている。本研究では、それらを対象にし、ユーザの集中アクセスにおいて、利用実態に応じたテストを行う負荷計測ツールを提案する。ツールは、まず、対象となるページの入力フォームを抽出し、送信される CGI データの構造を解析する。次に、負荷テストの実施要項や、アクセス時に用いる CGI パラメタの条件を指定し、実行レシピとして保存する。指定されたアクセス回数だけ、条件に合うデータセットを自動的に生成し、実際に負荷テストを実施する。負荷テスト中は、リクエストからの応答時間を計測したり、アクセス結果のステータスを記録する。必要に応じて、対象サーバ側でも、プロセスの状況や運用ログを記録する。現在、ツールの機能および実行レシピの仕様を検討し、試作版を実装している。

## Functions of A Load Measurement Tool of Web Access for Educational Support System

NAOKI HANAKAWA<sup>†</sup> MASAHIRO OKAWA<sup>†</sup>  
KOJI KURAMOTO<sup>†</sup> HIROYUKI TOMINAGA<sup>†</sup>

Recently, large scale and versatile learning management systems are adopted in many of educational organizations. On the other hand, small but advanced learning support systems are also required, which have various adaptive functions for the educational purpose and practical situation. However, these systems may have some weak points in scalability and security, because the developers tend to put emphasis on functional requirement. These systems may also have under-constructive modules as beta version. In this research, we propose a load measurement tool for these systems according to real usage situation. First, the tool analyzes input forms of a target Web page and extracts data format structure. Second, it prepares some data sets of user operation as access tests in high load capacity. The data sets are described in JSON format as an execution recipe. After execution of access tests, it records response time and status of access result. It also collects process situation and operation log in server side. In this paper, we introduce a prototype system. And we consider the recipe format and discuss the specification of each module.

### 1. はじめに

近年、Web テストや e-Learning によるオンライン学習、双方向マルチメディア通信を用いた遠隔学習など、ネットワークの教育への利用が盛んである。製品や Moodle などのプラットフォームを使って、大学や学部全体で運営する大規模なシステムの導入も進んでいる。

一方、研究室規模で、ある目的に特化したり、先進的な機能を取り入れたシステムも多い。しかし、サーバ過負荷状態における対応が十分でない場合がある。講義を補助するような教育支援システムでは、試験やレポートの締切など、負荷が集中する時期やサービスが予測しやすい、実際の運用にあたり、これらを想定して、システムのパフォーマンスに留意する必要がある。

また、他のサーバと連携し、一部の機能やデータベース管理を委託することがある。外部の Web サービスや API と連携している場合、状況次第でアクセスの規制や拒否といった措置がとられてしまう。そのため、想定している動作とは異なる挙動を行う可能性がある。そこで、必要な範囲で、事前にアクセスのテストを行い、その挙動を把握しておく必要がある。

### 2. アクセス負荷の計測ツールの概要

#### 2.1 ツールの目的

本研究では、複数の教育支援システムを開発・運用している。各システムを安定して運用していくために、リスク管理を行う必要がある。そのため、サーバダウンなどのクリティカルな障害が発生しないよう、起こり得る過剰なアクセスを事前に想定し、システムの挙動を把握しておく必要がある。

本研究では、主に教育支援システムを対象とし、ユーザの集中アクセスにおいて、利用実態に応じたテストを行う負荷計測ツールを提案する。集中したアクセスなどシステムにかかる負荷を計測し、負荷分散などの事前対策へのきっかけとする。本ツールで得られたデータを下に、過剰な負荷や応答の遅延の現状を把握し、要因を検討する。その上で、予算確保などの問題からハードウェア面での大幅な改善は困難としても、システムの実装方法などのソフトウェア面での改善や、運用指針の変更などの対策を十分に図っていく。

#### 2.2 ツールの特徴

本ツールの特徴を述べる。まず、テスト手法についてである。テストツールによる負荷のかけ方としては、Web ページへ過剰アクセスを行うものが一般的である。本ツール

<sup>†</sup>香川大学  
Kagawa University

では Brute Force Attack のような手法を採用する。具体的には、対象となるシステムにおいて、入力フォームを含んだページを指定し、フォームに様々なデータを入力する。そのデータを CGI パラメタとして送信し、システムが処理に要する時間を計測し、応答時間によって負荷耐性を判断する。手法上、テストの実行は、他に迷惑の及ばない状況のみに限定する。

また、負荷テストにおけるシステムへのアクセスでは、試験システムの解答パターンなど、利用実態に応じたデータが求められる。そこで、取得した入力フォームの情報から、負荷テストを行うために適したテストデータの生成を支援する。テストデータが満たすべき条件を指定し、大量のアクセスに用いる個々のテストデータをランダムに生成する。これによって、テスト実行者による負荷テストの労力を軽減し、作業の効率化を図る。

### 3. ツールの構成と機能

#### 3.1 ツールの全体構成

本ツールは、取得部、指定部、生成部、実行部、計測部、解析部の6つのモジュールから構成される(図1)。ネットワーク上では、ツールの本体が置かれるシステムエンド、テスト実行者がツールに対して指示を与えるユーザエンド、負荷計測のテスト対象となるターゲットエンドに分かれる。ユーザエンドは、Webブラウザからのアクセスとする。ターゲットエンドは、個々の教育支援システムを想定している。計測部は、許可された場合のみターゲットエンドの対象サーバ内に置く。

取得部は、ターゲットエンド内のWebページに対し、入力フォームを含んだHTMLの部分構造をDOM形式で取得する。指定部は、各フォームに入力が想定されるデータの型を推定し、テスト実行者が範囲などの条件をユーザエンドから指定する。また、実際にテストを行う日時や頻度なども策定する。これらの指定内容は、実行レシピとして、システムエンド内のDBに保存する。生成部は、実行レシピに基づいて、具体的なテストデータを乱数などを用いて自動で生成する。

実行部は、生成されたテストデータを用いて、ターゲットエンドに対し負荷テストを実行する。アクセスの応答時間や結果のステータスを取得する。これは、教育支援システムに対する、複数のユーザの利用アクセスを代行している。計測部は、ターゲットエンド内でリソースの負荷を計測し、システムエンドに通知する。解析部は、実行部で得られたアクセス結果や計測部で得られた運用ログを解析し、統計情報やグラフとして、テスト実行者に提示する。

#### 3.2 実行レシピの構成と記述

ツールの各モジュールは、スクリプト言語 Ruby で実装する。ユーザエンドのGUIには、スクリプト言語 JavaScript

のAPIやHTML5の機能も用いる。また、ツールでは、対象ページの構造、負荷テストの内容、実際のアクセス時に用いるテストデータなど、多くの情報を扱う。また、負荷テストの実行結果として計測したデータも扱う。これらのデータを一連のファイルにまとめ、実行レシピとして扱う(図2)。実行レシピは、ターゲット情報、実施要項、条件指定、データセット、応答結果、運用結果から構成される。実行レシピでは、データの種別が多岐にわたり、データの構造を統一しにくいいため、柔軟なデータ形式が求められる。また、ターゲットエンドとシステムエンドでの迅速なデータの通信も必要となる。そこで、Ruby や JavaScript でのAPIも整備されたJSON形式を採用する(図3)。

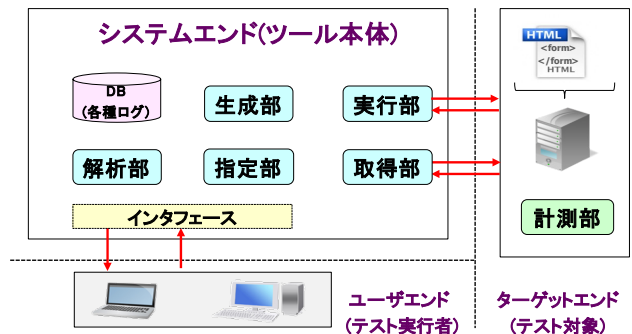


図1 Webアクセスの負荷計測ツールの全体構成

|   |   |
|---|---|
| <ul style="list-style-type: none"> <li>● ターゲット情報<br/>対象サイトURL<br/>対象ページフォーム部分</li> <li>● 実施要項<br/>実行日時<br/>アクセス回数<br/>アクセス頻度</li> <li>● 条件指定<br/>フォーム種別<br/>データ型<br/>データ範囲</li> </ul> | <ul style="list-style-type: none"> <li>● テストデータ<br/>データセット</li> <li>● 応答結果<br/>HTTPステータスコード<br/>応答速度</li> <li>● 運用結果<br/>対象サーバのCPU利用率<br/>対象サーバのメモリ使用率</li> </ul> |
|---|---|

図2 実行レシピの構成

```

{
  "target_url": "http://targetsystem.com/input.html"
  "target_form": "<form>...</form>"
  "start_date": "2013-12-14 12:00:00 +0900"
  "access_num": "2000"
  "access_interval": "20ms"
  "form_category": "text"
  "data_type": "int"
  "data_area": "10-99"
  :
}
    
```

図3 JSONによる実行レシピの記述

## 4. ツールの機能と処理手順

### 4.1 取得部

取得部の処理手順は、図4の通りである。ユーザエンドのGUIのデザインは、図5の通りである。GUI上で対象となるWebページのURLを入力すると、HTMLファイル全体をDOMとして取得する。取得したDOMをXMLパーサにかけて解析し、HTMLのformタグ内を解析して、入力フォームに関連する部分的なDOM構造を抽出する。具体的には、input, select, textarea, buttonなど、フォームの構成素のリストを生成し保存する。その際、サイズなどの属性、フォームの説明となるテキストやラベル、CGIデータとしての送信方法などもリストに含む。

例えば、教育支援システムにおいて、客観式試験システムでは各問の正誤や選択肢の個数、レポート提出システムではファイル名とファイル指定、アンケートシステムでは各項目と回答欄などの情報が相当する。GUI上では、元のDOM構造と個々のフォーム部品との対応関係を示し、ツールの解析で不十分な点は、ユーザからの入力で補完する。

### 4.2 指定部

指定部の処理手順は、図6の通りである。ユーザエンドのGUIのデザインは、図7の通りである。指定部では、負荷テストの実行レシピとなる生成条件と実施要項の2つの情報を扱う。まず、テストデータを自動生成するための条件を指定する。テスト実行者は、取得部で得られた対象ページのフォーム部分の各構成素に対し、入力データの型や範囲を指定する。テキスト入力では、整数/実数/文字列の区別、桁数や文字数、数値の上限と下限、文字列を構成する文字種(英字/数字)などである。具体的な候補を列挙してもよい。選択肢では、選択の範囲やパターンなどである。ファイルのアップロードでは、ダミーファイルを幾つか指定する。これらは、対象となる教育支援システムにおいて、利用者が起こし得る操作を想定して、適切な指定を行う。

次に、負荷テストの計画策定として、実施要項を指定する。実施要項は、テストの開始時間、アクセスを試行する回数と間隔、テストの打切の条件などを指定する。負荷テストは、対象システムだけでなく、ネットワークにも影響を与えるため、実施要項は十分な配慮の下に検討しなければならない。負荷テストの生成条件と実施要項は、実行レシピとしてJSON形式で保存される。

### 4.3 生成部

生成部の処理手順は、図8の通りである。ユーザエンドのGUIのデザインは、図9の通りである。生成部では、指定部で得られたテストデータの生成条件から、実際に負荷テストに用いるデータセットを自動的に生成する。データセットは、実施要項で指定されたアクセス回数だけ、生成条件に合う各データがランダムに設定される。生成されたデータセットは、必要に応じ、テスト実行者が吟味する。

不適切なテストデータを削除したり、一部のデータを修正したり、新たなテストデータを手動で追加したりである。

### 4.4 実行部

実行部の処理手順は、図10の通りである。ユーザエンドのGUIのデザインは、図11の通りである。実行部は、指定部でテスト実行者から指定された実行レシピと、生成部でツールが自動生成したデータセットに従って、対象ページに対する負荷テストを実施する。ユーザエンドのGUI上には、計画されている負荷テストの一覧が表示され、計画通りに実施するかどうかをテスト実行者が確認してから、実際の実行となる。

負荷テストの実行では、指定された回数だけ一定の間隔で、対象ページにアクセスする。アクセスごとに、各テストデータをフォーム入力として対象サーバに送信する。開始時間が指定された負荷テストの実施は、サーバの自動実行コマンドであるcronを利用する。本ツールは、対象ページへのアクセスごとに、送信したデータセット、リクエストから結果受信までの応答時間を記録していく。

### 4.5 計測部

計測部の処理手順は、図12の通りである。計測部は、可能な場合にのみ、ターゲットエンドの対象サーバ内に事前に設置しておく。負荷テストに関連する範囲で、対象サーバの挙動を監視したり、履歴情報を読み取ったりする。また、負荷テストの事前および事後に、システムエンドとも通信する。まず、該当する負荷テストの実行レシピを事前に設定しておく。負荷テストの実行中は、対象サーバの状態として、CPU利用率、メモリ利用率などを計測する。また、Webサーバの動作ログも取得する。ただし、セキュリティの問題もあるため、対象サーバの運用方針に即して、必要な情報のみを取捨選択して扱うように、モジュールのチューンナップをしておく。計測した結果は、JSON形式のログとして保存し、負荷テストの後で解析部に転送する。

### 4.6 解析部

解析部の処理手順は、図13の通りである。ユーザエンドのGUIのデザインは、図14の通りである。負荷テストを実行した後、システムエンドの実行部で得られたクライアント側の応答結果と、ターゲットエンドの計測部から転送されたサーバ側の運用ログを集約し、統計情報を加えて解析レポートを作成する。解析レポートの概要は、GUI上で表示し、詳細は文書ファイルのダウンロードで提供する。

負荷テストのアクセスは大量に行われるため、個々のテストデータに対する実行結果ではなく、割合、平均や分布、時系列変化など全体的な傾向が重要となる。例えば、応答時間の分布、HTTPステータスコードの割合、遅延状況の推移などである。これらのデータを見やすくするため、ユーザエンドのWebブラウザ上でグラフ化して表示する。これには、Rubyのライブラリであるgruffなどを用いる。詳細な解析には、統計解析用のプログラム言語Rも利用する。

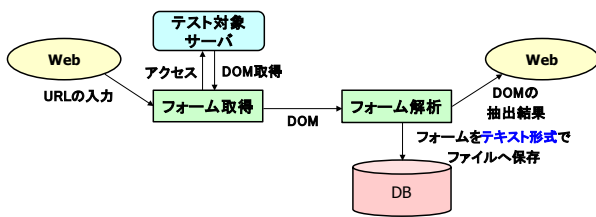


図4 取得部の処理手順

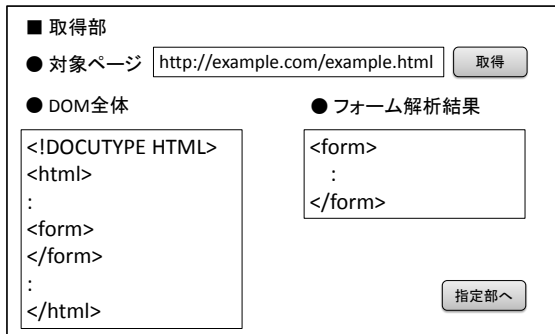


図5 取得部のGUI

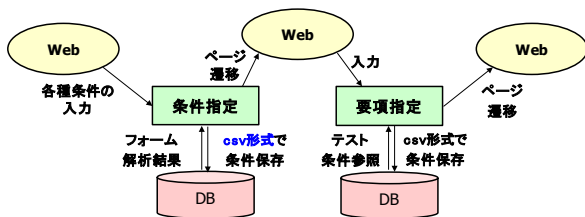


図6 指定部の処理手順

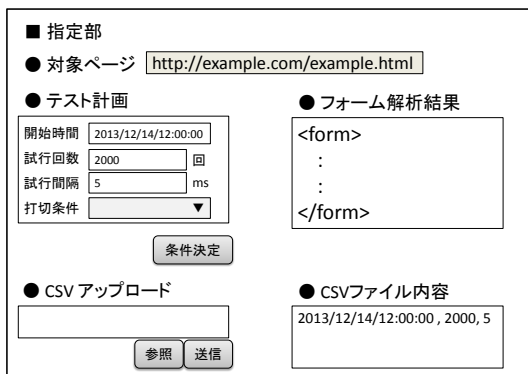


図7 指定部のGUI

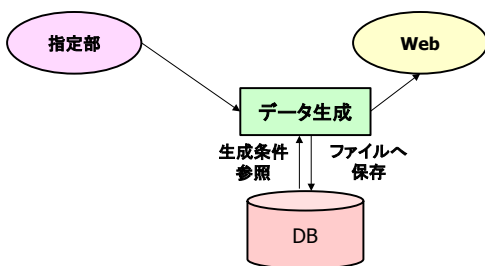


図8 生成部の処理手順

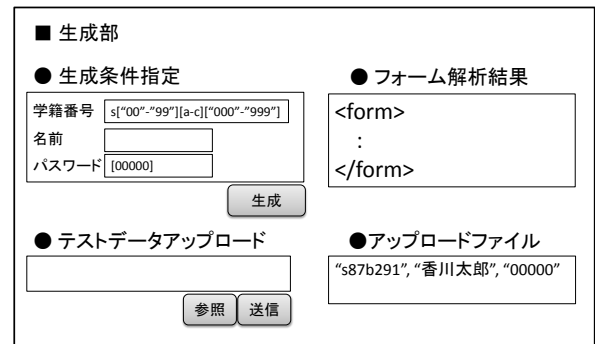


図9 生成部のGUI

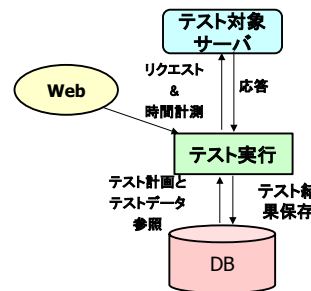


図10 実行部の処理手順

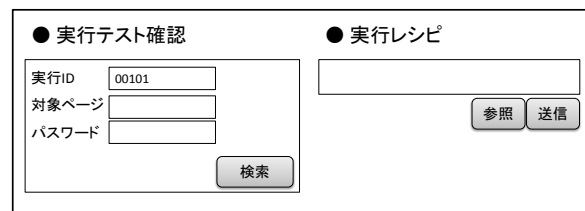
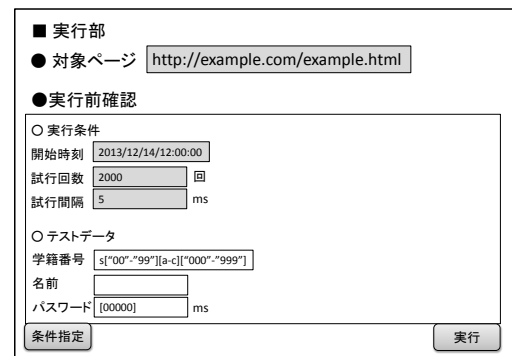


図11 実行部のGUI

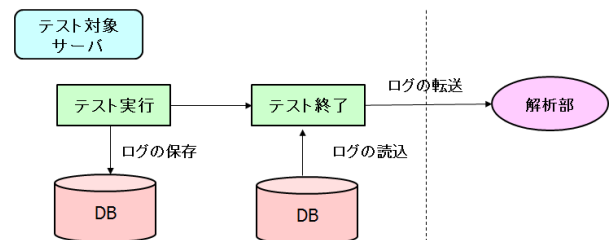


図12 計測部の処理手順

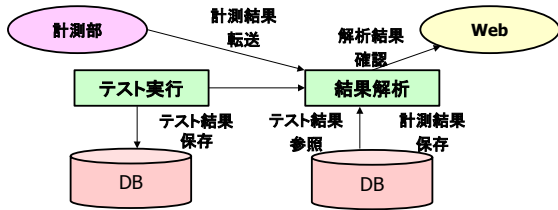


図 13 解析部の処理手順

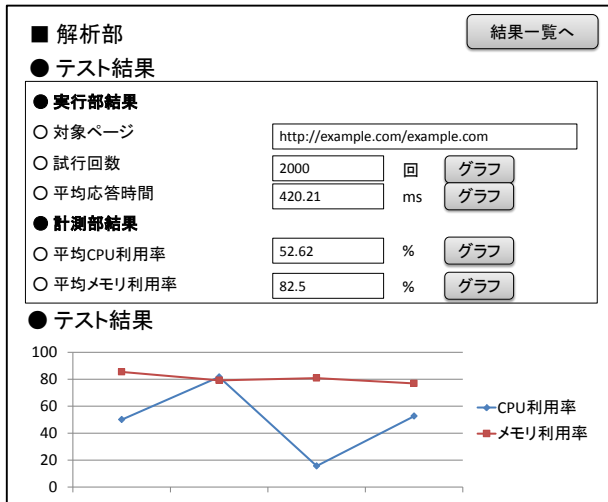


図 14 解析部の GUI

## 5. 教育支援システムへの適用

### 5.1 研究室規模の蔵書管理システムへの適用

蔵本[1], 吉田[2]では, 書籍関連サイトと連携した研究室規模の蔵書管理システムを提案している. また, 川鯉[3], 豊田[4]は, 文献共有による学習支援システムを提案している. 貸出や返却を扱うサーバの他に, 書籍の新規登録と情報検索は国立国会図書館の API を, また, 書籍の購入を支援する目的として Amazon の API を利用している. ただし, 後者については, 規約上の利用制限がある.

貸出や返却を扱うサーバは, 研究室で管理するものであり, 外部からのアクセスも受け付けられる. アクセスは日中に集中する. そこで, ツールによるテストを行うには, 深夜から早朝にかけての時間が適している.

### 5.2 初級プログラミング演習支援システムへの適用

倉田[5], 川崎[6], 西村[7]では, 大学情報系での初級 C 言語授業に競争型学習を取り入れ, 小コンテスト形式でのプログラミング演習を提案している. 大会運営サーバ tProgrEss を運用し, 提出されたソースコードの正誤を自動判定する. 部分配点, 時間調整点, 誤答減点を含めて得点を決める. コンテストは, 分単位の教室型, 日単位の宿題型のどちらにも対応する. 進捗状況を即時通知し, 順位公開して, 競争意欲を刺激する.

tProgrEss の運用場面には, 大学の講義や演習, 課外活動,

個人演習がある. これらは主に, 日中から夕方, あるいは深夜にかけて活動が活発に行われる. アクセスの集中する時間帯もほとんど同じになる. そこで, 負荷ツールによるテストを行うには, 深夜から早朝にかけての時間帯が適している.

tProgrEss に対する本ツールを利用した負荷テストとしては, ユーザのログインページ, ソースコードを提出する問題解答ページ, 問題解答の助けとなるヘルプページを探す検索ページなどへの適用が考えられる.

### 5.3 応用プログラミング演習支援システムへの適用

尾崎[8], 山田[9]では, 問題解決型の応用プログラミングとして, ボードゲーム戦略を題材とする対戦形式での Java 演習を提案している. また, ローカル側とサーバ側の支援環境 WinG を開発している. 五目並べを改良した, 五五ゲームを採用して授業実践を行ってきた. ローカル支援ツール WinG-LA では, ユーザの戦略作成を支援する. 大会運営サーバ WinG-CS では, 提出された戦略同士を対戦させる.

WinG を利用した授業実践の結果では, 大幅に提出が増えたが, 戦略の総当たり対戦の実行に時間がかかった. そのため, 順位の表示などに時間を要してしまう. 改善案としては, 対戦処理の効率化を検討するものがある. 内部処理の実装の違いによって, どの程度負荷が軽減されるのかを調査し, 負荷分散の技術導入も検討する.

### 5.4 記入式ドリル試験システムへの適用

萩原[10], 藤崎[11]では, 記入式解答の Web 試験 DrilLs-F を開発した. さらに, 国信[12]では, 初級レベルの C プログラムのソースコード読解問題への拡張を行った. DrilLs-F は, 大学理工系の情報基礎科目を対象とし, 情報数学の計算や C 言語の文法事項などのドリル練習に活用されている. C プログラムの簡単なソースコードを提示し, 入出力や実行過程を問う問題にも展開する. また, ソースコード読解問題の出題パターンを分類整理し, Web ベースのオーサリングツールを開発した. 実行可能なソースコードから, 出題部分を抜き出し, 入出力パラメータを変更した類題の作成も支援する.

DrilLs を運用していたサーバでは, 過去にアクセス過多などにより障害が起きている. そのため, DrilLs の実装面でもリファクタリングや処理方法を検討し, 少しでも負荷を減らす必要がある. ドリル式の試験であり, ユーザによる入力が多量に存在する. したがって, テスト計画を立てて各モジュール単位でのテストも行う必要があると考えている.

### 5.5 情報セキュリティ教育支援システムへの適用

中矢[13]では, ハッキング競技 CTF(Capture The Flag)を取り入れた情報セキュリティの導入教育の支援システム BeeCon を開発した. CTF は, サーバ上に隠された情報を旗(フラッグ)に見立てて, 攻撃側と防御側が競い合う競技である. ハッカー達の腕試しや交流の場として, 各地で CTF

大会が開催されている。余興ゲームとして、一般的に認知されていてゲーム性が高く、ルールも単純な神経衰弱を実装し、CTF 競技と余興ゲームによって観戦者を巻き込んだ大会を目指した。

BeeCon の特徴として、コンテストの開催時にアクセスが集中しやすい。そのため、不定期ではあるが、コンテストの開催時期が分かれば対策を立てることは比較的容易である。事前に、参加者などからコンテストの規模を想定し、リクエスト数やクエリ内容などを変化させることで、擬似的な運用も可能になり、本番でのアクセント対応もある程度対策を立てられる。

## 6. おわりに

実際の教育支援システムの運用においては、外部サイトとの連携において、アクセス頻度などの制約がある。そこで、実際の負荷やアクセス頻度などを計測するツールを試作した。フォーム入力を含み、CGI などが動作する Web ページに対し、予め指定されたパラメタに従って、ユーザからの入力を模倣し、アクセスを試みる。アクセスは、指定された頻度で行い、サーバに弱い負荷をかける。その際の反応時間などを計測し、強い負荷がかかったときの状況を予測する。また、サーバ側のツールでも Web のアクセスログを分析し、システムの状況を解析する。ツールは、取得部、指定部、生成部、実行部、計測部、解析部から構成される。サーバ側で、プロセスの状況や運用ログも記録する。負荷テストの情報は、実行レシピとして JSON 形式で管理する。現在、試作システムを作成中である。

本研究室では、複数の教育支援システムを開発している。これらは、授業や演習で利用するものであり、特定の時間帯にアクセスが集中しやすい。そのため、深夜から明け方などの時間帯で本ツールを適用し、システムの負荷耐性を調査する。調査結果に応じて、負荷分散などを行い、システム運用における信頼性向上に繋げる。

## 参考文献

- 1) 蔵本幸司, 川鯉光起, 富永浩之, "書籍関連サイトと連携した研究室規模の蔵書管理システムの登録と購入の機能", 教育システム情報学会 学生研究発表会 2013 年春, No.6, pp.1-2 (2012)
- 2) 吉田亜未, 富永浩之, "研究室規模の蔵書管理における履歴部と注釈部の機能", 教育システム情報学会 第 38 回全国大会, pp.197-198 (2013)
- 3) 川鯉光起, 中矢誠, 富永浩之, "研究室規模の蔵書管理と文献共有による学習支援 -コードレビューを取り入れた共時学習としての輪講の支援-", 情報処理学会 研究報告, Vol.2012-CE-119, No.20, pp.1-10 (2013)
- 4) 豊田竜也, 川鯉光起, 富永浩之, "情報系分野の解説書の輪講支援システムにおけるプログラム問題のコードレビュー機能の検討", 第 12 回情報科学技術フォーラム, Vol.12, No.4, pp.497-500 (2013)
- 5) 倉田英和, 富永浩之, 林敏浩, 垂水浩幸, "初級 C プログラミングの演習支援サーバ tProgrEss によるコード判定と授業実践", 情報処理学会 研究報告, Vol.2007, No.101, pp.11-18 (2007)

- 6) 川崎慎一郎, 西村智治, 富永浩之, "競争型学習を取り入れた入門的 C プログラミング演習における解答コードの提出履歴の分析", 教育システム情報学会 第 36 回全国大会, pp.200-201 (2011)
- 7) 西村智治, 青木辰徳, 富永浩之, "小コンテスト形式の初級 C 演習における教師支援 -解答プログラムの提出状況と得点推移によるモニタリング機能-", 情報処理学会 研究報告, Vol.2012, No.19, pp.1-8 (2012)
- 8) 尾崎浩和, 富永浩之, 林敏浩, 垂水浩幸, "ボードゲーム戦略を題材とする問題解決型プログラミング演習支援 -持続的な取り組みを促進する大会方式と運営サーバー" 情報処理学会 研究報告, Vol.2008, No.26, pp.1-8 (2008)
- 9) 山田耕平, 富永浩之, "ボードゲームの戦略プログラミングを題材とした Java 演習支援 -着手確認モジュールの導入と大会支援サーバの GUI の改良-", 電子情報通信学会 技術報告, Vol.111, No.473, pp.19-24 (2012)
- 10) 萩原秀和, 富永浩之, 松原行宏, 山崎敏範, "反復練習を重視する Web 型試験システム -情報処理教育への応用-", 教育システム情報学会 第 27 回全国大会, pp.369-370 (2002)
- 11) 藤崎博志, 衣笠裕, 富永浩之, 山崎敏範, "DrillS-F のマルチメディア出題機能によるプログラミング問題とアルゴリズム問題への適用", 教育システム情報学会 第 29 回全国大会, pp.389-390 (2004)
- 12) 国信真吾, 富永浩之, 林敏浩, "記入式 Web 試験 DrillS における C プログラムのソースコード読解問題 -類題生成を支援する Web オーサリングツールの開発-", 電子情報通信学会 技術報告, Vol.108, No.470, pp.217-222 (2009)
- 13) 中矢誠, 富永浩之, "初心者への情報セキュリティの教育機会としてのハッキングゲーム CTF", 電子情報通信学会 技術報告, Vol.112, No.66, pp.45-50 (2012)