

教員支援型クリッカー “tap4D” の開発

浅野 真之¹ 隅谷 孝洋^{1,a)}

概要：2000年初頭から、“クリッカー”と呼ばれるARS(Audience Response System)が学校の授業で活用されている。しかし、クリッカー利用には課題も多く、その一つにクリッカー導入や管理の手間の煩雑さがある。本研究では、導入、管理が平易であり、なおかつ教員の授業設計の意思決定の支援が可能なクリッカー“tap4D”を開発し、簡易的な評価実験を行った。

1. はじめに

アクティブラーニング導入が高等教育機関を中心に進んでいる昨今、授業における学生と教師の双方向なコミュニケーションの活性化を手軽に支援するツールの一つとしてクリッカーがある。クリッカーはARS(Audience Response System)の一種で、教員が示した択一式の質問に対して、学生は手元の端末を操作して回答する。回答状況は主に棒グラフや円グラフとしてリアルタイムに集計、表現される。2000年初頭にアメリカより持ち込まれたクリッカーは、北海道大学での導入を契機に全国の高等教育機関に広まっていった。しかし、集計データ、またはクリッカーそのものがシステムとして独立しており、LMSなどの教育システムとの連携が取れない等、昨今の潮流に反している点がある。我々は、教育機関へのLMS(Learning Management System)の普及に着目し、LMSとWebアプリケーションを関連付ける規格であるLTI(Learning Tool Interoperability)バージョン1.0に対応したWebクリッカー“tap4D”の開発を行っている[1][2]。本研究では実装されたtap4Dを授業内で取り入れ、簡易的な評価実験を行った。

2. tap4D

本章では実装されたtap4Dの特徴、及び工学的設計について論じる。

2.1 既存のクリッカーとの比較

本節では、既存のクリッカーとtap4Dの異なる点について述べる。我々は既存のクリッカー利用の報告を元に、現状のクリッカーの弱点を補う目的でtap4Dを開発してい

る。既存のクリッカーの弱点を以下に簡単にまとめた。詳細な議論については[2]を参考されたい。

- アプリケーションではなく専用ハードウェアを利用するクリッカーの場合は、管理や維持のコストが大きい
- 主要な学習支援システムと連携が取れない
- 質問内容や結果を教員間でシェアしづらい
- ログイン処理などが煩雑であり、利用に時間がかかる

2.1.1 LMS 上で利用

tap4Dの特徴として最も顕著なのがLMS上で利用することが可能であるという点である。前章で述べたようにtap4DはLTIに対応しており、LMSに機能を追加する。このように、LTIによってLMSと連携し、LMSに追加されたツールのように扱うことができる外部webアプリケーションはLTIツールプロバイダと呼ばれる。一般的にはLMS上でWebクリッカーのような機能を追加する場合は、各LMSに対応したプラグインなどを利用しなければならない。そのため、Moodleでは利用できるのにBlackboardでは利用できないなどの、教育資源の共有問題が発生する。しかしLTIを利用する場合は、一種類のツールプロバイダで複数のLMSから利用することが可能になる。これはWebクリッカーのような、不特定多数が多機関で利用する性質のツールと親和性が高い。

しかし、LTIには、ユーザIDがツールプロバイダ側に渡ってこない等の問題点も多く存在する。基本的な機能は単純なLTI接続で利用可能だが、たとえばwebクリッカーで出席を取る等のtap4Dの一部の機能については、LMS側のツールを追加することで利用するという形になる。多くの機能をツールプロバイダ側で実装できれば、LMSツール側の機能は単純になり、多くのLMSに対応しやすくなる。しかし、導入がやや煩雑になり、LMSのバージョンアップごとにLMSツールの導入やバージョン管理が必要になるため、運用にコストがかかる。そのため、前述のように、

¹ 広島大学
Hiroshima University
^{a)} sumi@riise.hiroshima-u.ac.jp

ツールプロバイダ開発のベストプラクティスとして、できる限り多くの機能を LMS ツールなしに利用できるようにすることがあげられる。

2.1.2 質問の記憶・共有機能

既存のクリッカーの多くは、一度登録した質問を再利用できなかつたり、そもそも選択肢のみで質問を入力できないタイプが主流である。しかし、例えばリメディアル教育では毎年毎学期、学生に発問する内容に大きな変化がないように、問題の入力の煩雑さから、クリッカー利用に積極的でないケースも考えられる。さらに、クリッカーを必要とするような数十人から数百人が受講する授業では、課題の採点や出席管理など、教員には大きな負担がかかる。したがって、クリッカーを利用すること教員の手間を増やすことは避けなければならない。そこで tap4D では、教員が明示的に質問の削除を行わない限り、質問内容を記憶し続ける。ここで示す質問内容とは以下のようなものがあげられる。

- 質問文・及び選択肢
- 集計結果
- 質問実施におけるその他のログ(日時等)

また、記憶した質問文に関しては、利用者が指定した公開範囲に合わせて他の利用者が検索し複製して利用することができる。注意すべき点は質問の複製であって、集計結果やログのマージはできないということだ。詳しいインターフェースについては後述する。

2.1.3 可視化

tap4D は、択一式質問のみを現在サポートしている。前節の記憶機能にも関連するが、集計結果を棒グラフで出力している。これは、現行のクリッカーが円グラフでの集計を行っているものが多いが、比率ではなく、実際の回答数を示すことで、回答人数の規模を示す意図がある。また、質問時の詳しい状況を調べる機能として、教員が質問の回答の経時変化を閲覧する機能がある。詳しいインターフェースについては後述する。

2.1.4 一般的な Web クリッカーとしても利用可能

tap4D は LMS からではなく、ツールに直接アクセスすることで通常の Web クリッカーと同様に利用可能である。教員に関してはユーザ登録が必要となるが、LMS 上で利用するときと同じように、問題の検索や共有が可能である。学生に関しては登録することで過去の自分の回答が参照できたり、出席機能が使えるようになるが必須ではない。詳細な設計については後述する。

2.2 システム構成

tap4D の構成及び開発環境について、下記に示す。

- Web サーバ : Apache, Node.js
- 使用言語 : PHP, JavaScript
- 使用ライブラリ : D3.js, Socket.IO, Bootstrap, LTI 用ライブラリ

- データベース : MySQL
- 開発環境の OS : CentOS 6

なお、bootstrap はカスタマイズしていないデフォルトデザインを利用している。クリッカー部分の Web サーバとして Node.js を採用しているのは、apache と比べて単純なアクセスが集中してもレスポンスが落ちにくいためである。この構成に類似した先行研究として [3] がある。

2.3 データベース設計と LTI

前述のように、LTI は、例えば moodle のログイン時に入力するユーザ名等の、LMS 利用者がログイン時に入力する ID をツールプロバイダ側に提供しない。ユーザの氏名は提供されるため、ユーザの苗字もしくは名前にユーザ ID を入れる利用法も可能だが、一般的ではない。そこで、tap4D 内で採用したデータベース構成についてここでは議論する。ただし、LTI はそもそもユーザの識別はサポートするが、特定をサポートしないことを規格の中で明示的に示しているので、これはイレギュラーな利用であることに注意されたい。

2.3.1 基本的な注意点

大前提として、LTI は「ゆるい規格」であることを認識する。LTI が定める LMS 上の各データ名とそれに対する値のフォーマットは厳密に定義されているわけではなく、プログラム上のデータ型も LMS によって異なる。そのため、以下の点について注意する。

- データ型などを把握するために、データの内容をさまざまなロールやコースで確認する
- データベース上のテーブルに LTI データを直接入力する場合は、すべて文字列型にしておく
- データの挿入時に、データ型やデータが定義されているかを確認する

最後の点は、通常の Web アプリケーション開発においても重要なことだが、入力欄を自作する Web アプリケーションとは異なり、ツールプロバイダは LMS から一方的に送られてくる多様なデータを処理する必要があるため、より繊細に注意を払う必要がある。

2.3.2 ユーザデータとリンクデータ

LMS からツールプロバイダに渡されるデータはユーザデータとリンクデータの二つに大別できる。ユーザデータはアクセスしてきたユーザの LMS 上のデータの一部である。具体的には以下のデータが渡される。

- LMS 上のユニークなユーザ ID(ユーザのログイン ID ではない。以下 UID と呼ぶ)
- 氏名(名前と苗字それぞれ)
- e メールアドレス
- ロール(学生、教師、TA 等)

リンクデータは LMS からツールプロバイダへのリンクを識別するための情報である。具体的には以下のデータが渡さ

表 1 LUID テーブルの例

LUID	UID	SID	LID
1	moodle15	hoge.ac.jp	link1
2	Bb17	hoge.ac.jp	linkA3
3	moodle15	tako.ac.jp	link6

表 2 GUID テーブルの例

GUID	LUID	CONNECT_DATE
hoge@ac.jp	15	2016-04-11
ika@ac.jp	26	2016-04-18
tako@ac.jp	35	2016-05-11

れる。

- サーバ ID(LMS を識別する ID)
- LMS 名
- LMS の種類
- LMS のバージョン
- コース ID(コースを識別する ID)
- コース名
- リンク ID(LMS 上のリンクを識別する ID)
- リンク名

注目する点は UID, サーバ ID があればユーザを一意に識別できるという点である。実際の tap4D のテーブル構成の一部を表 1 に示す。ここで示す UID, サーバ ID, コース ID, リンク ID に振られる ID を LUID(Local User ID) と呼ぶ。なお, 表 1 では紙面の都合上サーバ ID を SID, リンク ID を LID としている。同じドメイン上に存在しても LMS が異なれば識別が可能であり, 同じ種類, 同じ名前の LMS を複数利用していても, ドメインの違いから識別が可能なのが表 1 の例から確認できる。

2.3.3 ユーザ登録と共有

前節のように, LUID のみでは各 LMS からアクセスするユーザを識別できても, 実際の利用者を特定できない。従って多機関にまたがって講義を行う非常勤講師や, 複数の LMS を単一機関が利用している場合は, tap4D 上に蓄積されたリソースを円滑に共有できないという問題が残る。この問題を解決するために, ツールプロバイダ側でもユーザ登録を行う。

ユーザ登録を行うことで, 利用者が設定したログイン ID(tap4D の場合はメールアドレス) と LUID を紐付けることができる。テーブルの構造は表 2 のようになる。このログイン ID を LUID と対比して, GUID(Global User ID) と呼ぶ。この構成は, LUID が LMS から渡ってくるデータに変更がない限り常に一定であることから, 1 度ある LMS から tap4D へアクセスし, tap4D 内でログインすると, 2 回目以降の tap4D 内ログインが不要となる。本来, LTI を用いることでツール内のユーザ登録などの処理を行う手間が省けるというメリットがあるが, この構成ではユーザ登録を利用者に要求しているという点で, LTI を利用するメ

リットを損なっている。この問題への対応策として以下のような点に注意する必要がある。

- ユーザ登録をなるべく簡潔なものにする
- 多機関連携機能が必要なユーザのみがユーザ登録するような構成にする
- 利用ごとにログインする必要のないように LTI から渡るデータを活用する

2.4 LMS を介しない利用

前節で触れたように, tap4D は LMS から tap4D へ送られる LTI 形式のデータを元にユーザの識別を行なっている。しかし, LMS を前提とした限定的な利用では, LMS のない環境との連携が取れない, LMS は利用しているが LTI に非対応である等の問題が残る。そこで tap4D では LMS を介さない利用に対応する。

2.4.1 LTI 形式のデータ

もし教員利用者がユーザ登録を行っている場合, tap4D は過去に LMS 経由で利用したデータを保存しているので, GUID を利用してログインすれば, 利用者が作成したリソースにアクセスできる。教員利用者が LMS を介してユーザ登録をしていない場合は, 本来利用者を識別する LTI 形式のデータが tap4D 側にはないため, 前節で論じたユーザ登録を行う必要がある。

2.4.2 クリッカーの利用

クリッカー機能を利用する際, LMS 経由と異なる点が 2 点ある。

- (1) LMS から学生の LTI 形式のデータが送られてこない
- (2) LMS 上にリンクを生成できないため, 独自にリンクを生成する必要がある

この相違点を解消するために, 次のような仕様とする。

- (1) 教員が問題を作成すると, クリッカー用 URL が発行される
- (2) 学生は 教員が発行したクリッカー用 URL にアクセスする
- (3) 学生の「はじめての利用」の場合は LUID が発行され, 学生の PC に Cookie として保存される
- (4) 学生の「2 回目以降の利用」の場合は, 設定された値を LUID として扱う
- (5) 学生の「2 回目以降の利用」の場合で値の設定が確認できなかった場合は「はじめての利用」と同じように扱う

この仕様の欠点は Cookie に学生の LUID を記憶しているため, 端末やブラウザを変更してしまうと学生は回答履歴を参照できない点である。

2.5 インターフェース

tap4D の主要な画面遷移を図 1 に示す。本節では, [1] や [2] で論じられていない質問検索と大きな変更があった

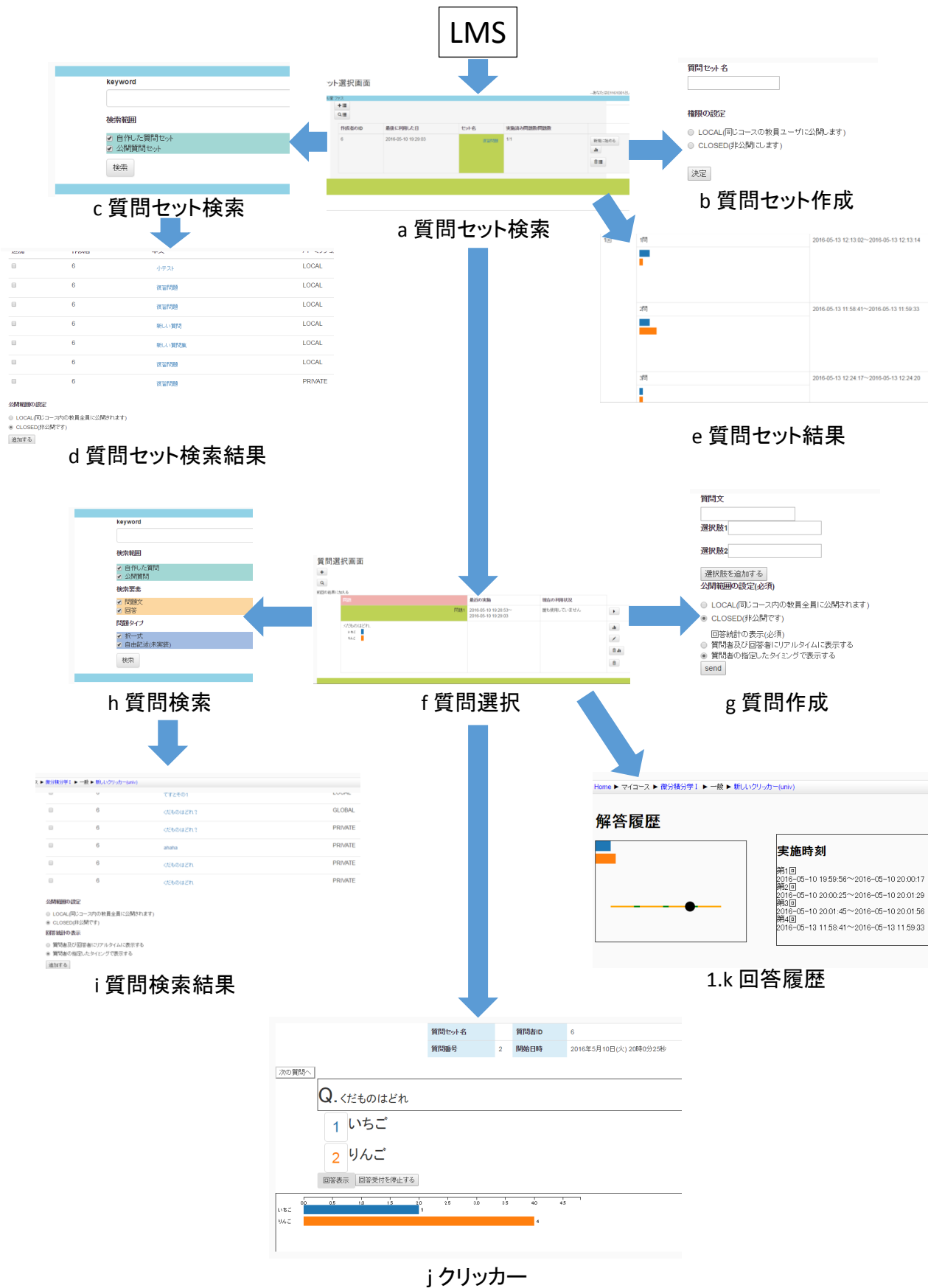


図 1 主要な画面遷移

図 2 質問検索画面

図 4 問題詳細画面

質問ID	質問文	公開範囲
6	くだものはどれ?	GLOBAL
6	くだものはどれ?	PRIVATE
6	ahaha	PRIVATE
6	くだものはどれ	PRIVATE
6	くだものはどれ	PRIVATE

図 3 質問検索結果画面

図 5 クリッカー画面

クリッカー画面のインターフェースについて論じる。

2.5.1 質問作成と公開範囲

既存のクリッカーと異なり, tap4D は質問を利用者間で共有できる。そのため, 作成した質問の公開範囲を設定する必要が生まれる。そこで, 問題作成及び, 問題を検索する場面で公開範囲を設定するインターフェースを追加した。検索については次節で論じる。問題作成画面を図 1.b に示す。設問の選択肢は作成者がボタンをクリックすることで入力フォームを適宜増やす。公開範囲については表 3 の 3 レベルを用意した。

表 3 公開範囲の設定

公開レベル	詳細
GLOBAL	tap4D 利用者全員に公開する
LOCAL	同じコース内で公開する
PRIVATE	公開しない

このレベル設定により, 問題作成者は共有したい質問のみを公開することができる。

2.5.2 質問検索

検索画面を図 2 に示す。質問検索には質問セット検索と質問検索の二つがある。質問検索は, 参照可能な質問群から, キーワード検索ができる。検索対象は問題文, 質問, あ

るいはその両方から設定できる。検索範囲は過去に教員が自作した質問か, 他の教員が作成し, 公開した質問, あるいはその両方である。現在の質問形式は択一式のみだが, 今後, 自由記述などの他の問題形式に対応していくことを見越して, 問題形式による絞込みもできる。

検索結果は図 3 のように表示され, 問題文をクリックすると, 図 4 のように問題文や選択肢などの問題の詳細情報を参照できる。現在は作成者の UID がそのまま表示されているが, ユーザ登録をすることで, ユーザネームに変更する予定である。本来なら LTI から渡される名前を表示させるべきだが, 前述のとおり, 苗字や名前の欄に学生 ID を登録しているタイプの利用も見られるため, tap4D ではプライバシー保護の観点から LMS から送られてくる名前の利用を避けた。

利用したい質問は図 3 の左側のチェックボックスにチェックを入れて, 公開設定を行うことで利用できる。

2.5.3 クリッカー画面

tap4D のクリッカー画面を図 5 に示す。教員側には学生の画面を制御するためのインターフェースや 現在の質問の状況を示す表が追加されている。表には現在時刻, 実施者の UID, 質問番号 (何番目の質問か), 質問セット名が表示される。その他の教員用のボタンのそれぞれの機能を表 4

表 4 教員用ボタンの機能

ボタン名	機能	備考
次の質問へ	教員学生ともに、同じ質問セット内の次の質問へ遷移する	
回答表示	学生に集計結果を表示	問題作成時の設定で「教員のタイミングで結果を表示する」に設定した場合のみ表示
回答受付を停止する/再開する	学生の回答ボタンを無効にする/有効にする	ボタンを押すことで停止すると再開するが切り替わる

に示す。学生と教員で共通なのは、問題文、回答ボタン及び選択肢、集計画面である。教員が回答ボタンを押した場合には該当選択肢に対応する棒グラフが赤くマーキングされる。教員は学生に各選択肢について説明するときには有効である。その他の機能は既存のクリッカーと同じである。

2.5.4 回答履歴画面

回答履歴画面を図 1.k に示す。回答履歴画面では学生がどのような間隔で回答しているかを確認することができる。画面左下の黒い丸を線に沿ってドラッグすると、回答がどのようなタイミングで集中したかを感覚的に知ることができる。線の色は質問開始から終了までを表しており、黄色は回答受付中、緑は回答終了を示している。回答履歴画面の詳細については [1] を参照されたい。

3. 評価実験

開発した tap4D の工学的な性能を測るために簡単な実験を行った。具体的には以下の項目についてログ収集ツールを用いて調査した。

- メモリーの占有率
- 入出力にかかる負荷の遷移
- CPU にかかる負荷の遷移

3.1 実験条件

以下の表 5 に実験条件を示す。また実行環境については、仮想環境 KVM を利用している。以下、ホストサーバと仮想マシンのスペックを示す。

3.2 実験内容

計算機室での実習中に、端末を通じて問題を配信し、解答

表 5 実験条件

被験者人数	教員 1 名 1 年生 66 名
使用ツール	sysstat, vmstat, iostat, netstat
日時	2016 年 05 月 13 日
実験場所	広島大学西図書館演習室
授業科目名	情報活用基礎
講義形式	計算機室での実習

表 6 ホストサーバ環境

OS	CentOS 6.7
CPU	Xeon E3-1200 v3 3.1GHz (4 cores)
RAM	16GB
HDD	2TB

表 7 仮想マシン環境

OS	CentOS 6.7
CPU 割当	1
RAM 割当	2GB
Storage 割当	64GB

表 8 学生利用端末の状況

Win	Lin	持込 (Win)	持込 (Mac)	iOS	Android
18 人	1 人	35 人	11 人	0 人	1 人

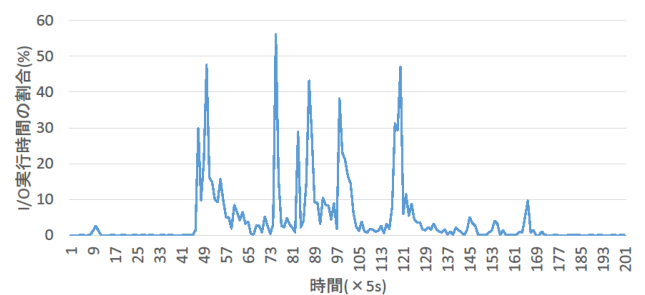


図 6 CPU 総実行時間当たりの I/O 実行時間の割合

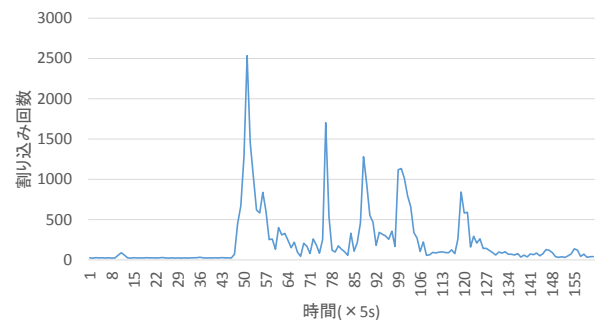


図 7 割り込み処理の様子

する。本実験で用いる端末は計算機室のものと、学生の持ち込み端末である。広島大学では学生の持ち込み端末について特に機種、OS 等の制限を設けていない。具体的な OS や機種の分布は表 8 に示す。Win は Windows 系 OS, Lin は Linux 系 OS, Mac は Macintosh 系 OS, Android は携帯端末を、持込は演習室の端末ではなく、学生の持ち込み PC を利用している事を表す。

3.3 実験結果

実験結果を図 6～図 10 に示す。図 6 に表れているように、

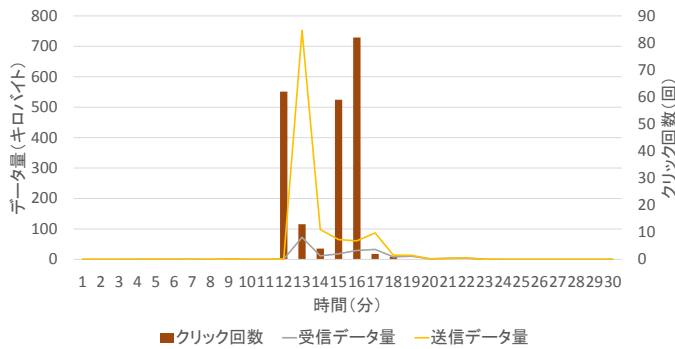


図 8 通信データ量とクリック数の関係

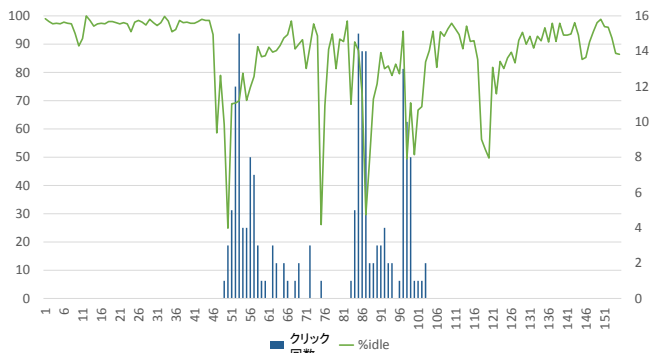


図 9 CPU アイドル率とクリック数の関係

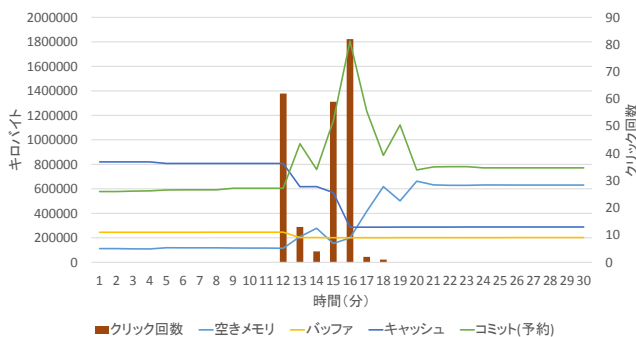


図 10 メモリとクリック数の関係

クリッカーというシステムの性質上, CPU は入出力処理に大きなリソースを割いていることがわかる. 従って, 入出力処理の負荷は図 9 で示している CPU のアイドル率に大きく影響することがわかる. 図 9 から, CPU の最小アイドル率はおよそ 27% 程度まで下がっていることがわかる. CPU アイドル率が 27% まで落ち込むと他のソフトウェアの処理の遅延が発生する可能性がある. これは瞬間的な減少であるものの, アプリケーションレベルでの入出力処理の回数を減らすなどの工夫が必要であることがわかった.

図 7, 図 8 において, 利用者のセッション確立時 (学生がその日初めて tap4D にアクセスするとき) に大量の処理を行っていることがわかる. セッションは web アプリケー

ションでは, 個々のユーザ固有の情報を, サービスを利用している間保持するために利用される. tap4D の場合, 個々のユーザに対してセッション領域を割り当てる処理が学生の一回アクセスという形で集中的に発生している. また, 図 10 のメモリの占有量の推移をみると, 一部開放されていないことがわかる. これは未開放の変数がプログラム上に存在する可能性が考えられる.

また, クライアント上でのレスポンスは工学的測定は行っていないものの, 非常に良好であり, 学生のクリックに対して, 遅延が発生したことは認められなかった. しかし, 一部の学生が何度も回答してしまい, 後の集計と差が出てしまう現象が発生した.

教員側のインターフェースにおいて, 問題を作成するとき誤って選択肢を増やしすぎてしまったが, 選択肢を削るインターフェースがなく, 初めから作り直しになってしまったという問題が発生した. また, 学生側は使い慣れている LMS から利用することから, アクセスの仕方がわからない等のアクセサビリティの問題は発生しなかった.

3.4 考察

前節の事項から, tap4D には次のような改善点があると考えられる.

- 教員側のユーザインターフェースの改修
- 学生側の処理を司るプログラムの変数やセッションのリファクタリングが必要
- 学生の多重回答を防ぐ仕掛けが必要

また, 評価点として次のことが挙げられる.

- 学生のほぼ全員が tap4D にアクセスできて, なおかつアクセスまでの時間が短かった
- クライアントの画面応答が非常に良好

学生のアクセサビリティは LTI の利用により格段に向上した. しかし, 他のクリッカーとの比較を行っていないため, 今後は比較実験などを行う必要がある.

4. まとめ

教員支援型クリッカー tap4D を開発した. このクリッカーは問題の共有や記憶, 再利用などの機能を有し, 教員間の資源の共有を円滑にする効果が期待できる. またソフトウェアの安定性評価のための実験を行った. 結果は CPU の占有率が瞬間的に膨大になる等の問題があるものの, クリッカーの応答は良好だった.

5. 今後の方針

今後の課題として, 考察で述べたように, 他のクリッカーとの比較評価や教員に対する使用感のヒアリングに基づく評価がある. この使用感は, 今回測定できなかった問題の共有や記憶に関する点を含む. また, 出席管理機能の拡張を LMS 別に行い, 授業中での出席確認をスムーズに行うこ

とを目指す。この出席管理機能に限らず、次世代のクリッカーとして盛り込まれることが切望される機能を、教員からアンケートなどで広く取り入れていきたい。

6. 謝辞

本実験に先立って、デバッグ作業に貢献していただいた広島大学西図書館職員の皆様や、実験に協力していただいた広島大学1年生及びティーチングアドバイザー様には厚く御礼申し上げます。

参考文献

- [1] 浅野真之, 隅谷孝洋: 時間に着目した LTI クリッカー”tap4D”の開発, 情報処理学会研究報告 Vol.2015-CE-131 No1, pp.1-7(2015)
- [2] 浅野真之, 隅谷孝洋: LTI を用いた web クリッカーの開発, 日本教育工学会第31回全国大会講演論文集, pp. 719-720(2015)
- [3] 大見嘉弘, 水谷正大, 永井保夫: オープンソースによるクリッカーシステムの提案と試作, 情報教育シンポジウム 2014 論文集 pp201-206 (2014)