CASとリバースプロキシを基盤とした学外システムの利用者制限とログ管理

中野 裕司[1], 久保田 真一郎[1], 松葉 龍一[2], 杉谷 賢一[1], 永井 孝幸[1],
田村 規雄[3], 八木 玲子[3], 西村 岳史[3], 中野 淳[3]

熊本大学 [1]総合情報基盤センター, [3]e ラーニング推進機構, [5]日経 BP 社

nakano@cc.kumamoto-u.ac.jp

概要：我々は、学外システムとして提供されるパソコンに関連した雑誌記事を、特定の科目を受講している学生にだけ提供するため、外部システムを利用者制限付きのシングルサインオンでアクセス可能にし、そのログを蓄積するシステムを構築した。学内の統合認証基盤である CAS を利用し、かつそこで認証された利用者をさらに制限した上で外部システムにアクセスさせる仕掛けを、学務情報システムから利用許可ユーザ情報を取得し、Apache httpd 上に、CAS 対応のための mod_auth_cas と、グループ制限を高速に行うための AuthDBMGroupFile を組み合わせることで実現した。また、外部システムへのアクセスをこのリバースプロキシサーバ経由でしかできなくしているため、外部システムへのアクセスログをほぼ完全に取得することが可能となった。

1 はじめに

近年、シングルサインオン(SSO)により、大学ポータルや学習管理システム(LMS)等の大学が提供する Web 上の各種サービスが学内外から利用可能になってきた。熊本大学においても、2006 年ごろから、SSO として CAS[1]を、大学ポータルとして uPortal を用い、LMS の Blackboard LS CE6/8、学務システム SOSEKI 等、20 程度のサービスを提供してきた[2]。

また、最近、クラウドサービスの普及に伴い、大学向けのサービスに関しても、SaaS による ASP 形式による外部サービスの提供が拡大しつつある。

本研究は、大学の提供する SSO を通った上で、さらに特定のグループに属するメンバーだけが、追加の認証等なく、外部のサービスを利用することのできるシステムの構築と、ユーザ情報を含むログ管理に関するものである。実際には、1,000 人規模の受講生で実施している e ラーニング形態中心の情報系科目の「情報処理概論」において、その受講者のみが、外部サーバ上の情報提供サービスの「日経パソコン Edu」のコンテンツを参考資料として利用しながら学習を進めることになり[3]、その実現のために本システムを構築した。

2 システム構築

2.1 システム構成

システム構成の概略を図 1 に示す。本システムは、学内外から LMS 上の本科目のページにアクセスしたユーザが、そのページ上のリンクから、追加認証なしに、シームレスに学外サービスを受けられることを実現する。

LMS の講義ページをアクセスしているユーザは LMS 利用開始時に既に CAS 対応認証が済んでいるため、本システムの CAS は通過できる。

その後、学務システムから本科目のユーザリス

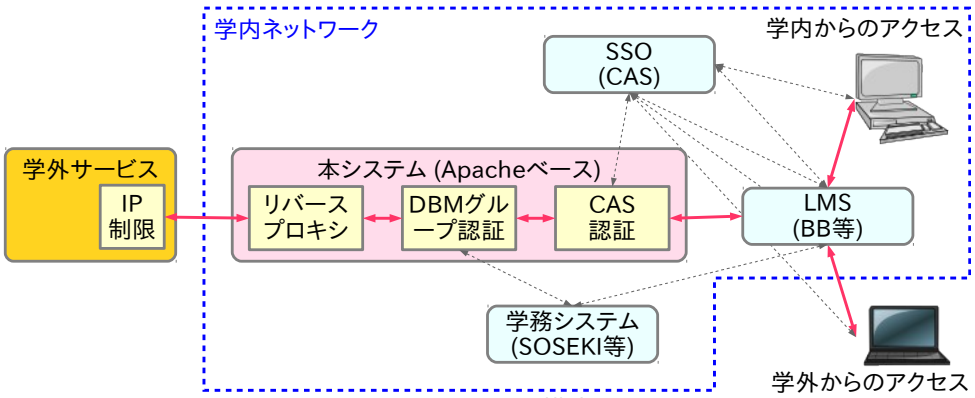

図 1. システム構成

トによるグループ認証を受け、それを通過した後にリバースプロキシにより外部サービスを受けることになる。

なお、全ての通信はSSLによる暗号化が施されている。以下、個々の実装に関して説明する。

2.2 Apache httpd サーバ

本サーバは、KVM仮想マシンとして動作しており、OSにCentOS release 6.3を使用している 。Apache httpdサーバは、同ディストリビューションに含まれるApache 2.2.15をそのまま使用している。

2.3 CAS認証

ApacheのCAS対応モジュールは、CASの提供元のJasigから提供され、最初に解説[4]に従ってRPM版(mod_auth_cas-1.0.9.1-1.el6.x86_64)のインストールを行い動作確認をしたところ、ユーザIDがCAS-User:XXXXというヘッダで外部サーバに渡り、設定ファイルのCASAuthNHeaderをNULLにしてもこの問題が解決しないことが判明したため、ソースからインストールした。

インストールしたバージョンは、1.0.10 (cf37626)で、本環境では、--with-apxs=/usr/sbin/apxs オプションをつけてconfigure, make installすると、/etc/httpd/modules/mod_auth_cas.so が作成される。このバージョンの規定値では、CASAuthNHeaderがNULLのため、特に設定しなくとも、ユーザIDの伝搬は避けられた。

auth_casの設定ファイルは、図 2に示す内容で/etc/httpd/conf.d/auth_cas.confに置いた。ここで、[A]はCASサーバを、[B]は後述のDBM形式のグループファイル名を示す。また、「require group ...」に関しても次節で説明する。

2.4 DBファイルを用いたグループ認証

Apacheで何らかの認証の後(今回の場合はCAS認証)、特定のグループに属するかどうかで、ディレクティブ単位のアクセスの可否を行う機能として、mod_authz_groupfileモジュール[5]が利用されることが多い。これは、テキストファイルにグループ名(:の後)に続いてそのグループに属するユーザIDを列挙することで容易に行える。

しかし、今回のように受講者が1,000名を超えるような場合は、極めて1行が長いものとなってしまうばかりか、Apacheのサイト[6]にあるように、「大きなファイルを探索するのは、非常に効率が悪いという点に 注意してください。そのような場合は、AuthDBMGroupFile の方がずっと良い性能を発揮します。」とあるように、このモジュールは使えず、AuthDBMGroupFileモジュール[7]を使うべきである。

```
CASCookiePath    /var/cache/mod_auth_cas/
CASLoginURL      https://[A]/cas/login
CASValidateURL   https://[A]/cas/serviceValidate
<Location />
   Authtype CAS
   AuthDBMGroupFile /etc/httpd/conf.d/[B]
   require group admin member
</Location>
```

図 2. auth_casの設定

```
# htdbm -cbpt [F] [A1] admin admin
# htdbm -bpt [F] [M1] member member
# htdbm -bpt [F] [M2] member member
# htdbm -l [F]
Dumping records from database -- [F]
   Username        Comment
   [A1]            admin
   [M1]            member
   [M2]            member
Total #records : 3
```

図 3. AuthDBMGroupFileの作成方法

そこで、今回はAuthDBMGroupFileモジュールを利用することにした。このDBMG形式のファイルはテキストではなく、ある種のデータベース形式になっており、高速なアクセスを可能としている。このファイルの作成及びグループへのユーザの追加方法を図 3に示す。ここで、[F]は書き込むファイル名、[A1]は「admin」という名前のグループに登録したユーザのユーザID、[[M1]、[M2]は「member」という名前のグループに登録したユーザのユーザIDである。ここで、示した2つのグループ「admin」及び「member」は、図 2の下から2行めの「require group admin member」に対応しており、これらのグループに属するユーザIDでであることがアクセスするために必須となる。

実際には、この登録は1,000名以上を日々更新することになるため、手動での登録は行わず、1日1回深夜にcronによるバッチ処理として、学務システムのデータベースから抜き出したCSVファイルを用いてシェルスクリプトで行なっている。

2.5 リバースプロクシ

本サーバをユーザがアクセスし認証が通った場合、あたかも本サーバが外部サービスを提供しているサーバかのように振る舞うことを実現するの

```
ProxyRequests Off
ProxyPass / http://[DN]/[DP]/
ProxyPassReverse / http://[DN]/[DP]/
ProxyPassReverseCookieDomain [DN] [SN]
ProxyPassReverseCookiePath /[DP]/ /
```

図 4. httpd.conf におけるリバスプロキシ設定

```
SSLProxyEngine on
ProxyPass / https://[DN]/[DP]/
ProxyPassReverse / https://[DN]/
```

図 5. ssl.conf におけるリバスプロキシ設定

```
LogFormat "%h %l %u %t ¥"%r¥" %>s %b ¥"%{Referer}i¥" ¥"%{User-Agent}i¥"" wref
CustomLog logs/ssl_access_log wref
CustomLog "|/usr/sbin/rotatelogs -lf ¥/var/log/httpd/ssl_access_log.%Y%m%d 86400" wref
```

図 6. ssl.conf におけるログ形式の設定

が、リバースプロキシである。

SSL 接続のリバースプロキシの設定は、一般的な解説にある通りであるが、一応示すと、Apache の設定ファイル http.conf の proxy directives に関する設定の最後の方に、図 4 に示す設定を、また、SSL の設定ファイル ssl.conf の最後に、図 5 の設定を加えた。

2.5 ログ設定

本サーバは CAS による認証を含んでいるため、ユーザ ID まで取得可能である(前述のようにユーザ ID は本サーバまでで、外部サービス提供者には届かない)。同一ユーザからのアクセスを 1 回と数えることが可能であるため、例えば、ページ毎のアクセス人数等正確に取得することが可能である。また、学習者の行動は、LMS 側にも詳細なログがあるため、併せて処理することによって、学習者の学習行動を詳細にトレース可能となる。

図 6 の設定を行い、86400 秒毎に新しい ssl_access_log. [年月日] というファイルに、アクセス元 IP アドレス、ユーザ ID、時刻、アクセスしたページ(ファイル)、ブラウザ情報等をアクセス毎に記録している。

3 運用と統計情報の取得

3.1 運用状況

本科目の期末試験を除く実施期間である 2012 年 10 月 1 日から 11 月 18 日の 7 週間がまだ完了していないが、現時点で既に 6 週間以上、一度も止まらず問題なく動作し、ログも取得できている。

3.2 統計情報の取得

前章で述べたように、本サーバのログはユーザ ID も含めた形式で 1 日 1 ファイルに分けて残している。対応する LMS の Web サーバ(WebLogic)のログもユーザ ID を含んでいるが、1 日 1 ファイルの形式ではないため 1 日分の関連情報を切り出して、当サーバに深夜転送し、2 つのサーバのログを併せて処理することで、ある程度の統計情報を日々自動取得している。

現在のところ、外部サーバに対しては、アクセス数、その中で TOP ページへのアクセス、LMS のフィードバック等からのアクセス、学内からのアクセス、学外からのアクセス、0 時から 24 時までの時間毎のアクセス等を、grep, sed, uniq, sort, wc 等のコマンドの組み合わせで取得している。また、MS 側に関しても、科目トップページへのアクセス、テストやレポートの提出数等を同様の方法で集計している。前述の通り、まだ講義期間が終了していないが、現時点でのいくつかの具体例を以下に紹介する。

図 7 は、学生の時間毎のアクセス数であり 、LMS 上の科目トップページと外部サービスへのアクセスが示されている。どちらも同様の動きを示しているが、科目トップへ入った学生の 1 割程度が外部サービスを参照していることがわかる。また、夜中に学習をしている学生の割合もかなり高いことがわかる。

図 8 は、同様に時間毎のアクセス数であるが、学内からのアクセス数と学外からのアクセス数を比較したものであり、学内からのアクセスは日中に綺麗な分布をしているが、学外からのアクセスは日中でもある程度あり、深夜になると日中の学内からのものと同程度もあることがわかる。

図 9 は、外部サービスを利用した学生数の計時変化であり、日によって、また週によって明らかに差異が見られる。これは、科目の取り扱う内容、教材の魅力、参考資料の魅力等、様々な要因が考えられるが興味深いデータであり、今後詳細に調べたい。

図 10 は、外部サービスを利用した学生の曜日による、また学内/学外からのアクセスによる違いである。毎週のオンライン確認テストの締め切りが日曜日深夜であることから、日曜日のアクセスが最も多く、その大半が学外からのアクセスであることがわかる。

4 まとめと今後の課題

今回、SSO とグループによる利用制限を加えた で形で外部サービスのアクセスが、ユーザ側では全く意識することなく可能になるシステムを Apache httpd 上の SSL リバースプロキシに 、mod_auth_cas と、AuthDBMGroupFile を組み合わせることで実現した。さらに、ユーザ ID も含む形での詳細なログを LMS のログと併用することで、学習者のアクセスに関するデータの自動集計をある程度実現した。

本システムを利用して実施している科目がもう少しで終了するため、期末試験も含めた形で今後、より詳細集計を行いたい。また、アクセスしたユーザのブラウザ情報等、また活用していないデータも多くあり、それらの集計・分析も進めたい。

参考文献

[1] Jasig Central Authentication Service (CAS) project http://www.jasig.org/cas (2012 年 11 月 16 日確認).

[2] 中野 裕司, 喜多 敏博, 杉谷 賢一, 根本 淳子, 北村 士朗, 鈴木 克明 : CMS を補完する学習ポータルの実装 - 教授システム学専攻ポータルを例として, 情報処理学会研究報告第 4 回 CMS 研究会, pp.55-60 (2006).

[3] 久保田 真一郎, 田村 規雄, 八木 玲子, 西村 岳史, 中野 淳, 松葉 龍一, 中野 裕司, 情報処理科目におけるオンラインの雑誌記事の活用, 第 2 回大学 ICT 推進協議会年次大会(発表予定) (2012).

[4] CAS RPM Module 解説ページ: https://wiki.jasig.org/display/CASC/RPM+Modules (2012 年 11 月 16 日確認).

[5] Apache Module mod_authz_groupfile: http://httpd.apache.org/docs/2.2/mod/mod_authz_groupfile.html (2012 年 11 月 16 日確認)

[6] AuthGroupFile ディレクティブ: http://httpd.apache.org/docs/2.2/mod/mod_authz_groupfile.html#authgroupfile (2012 年 11 月 16 日確認)

[7] Apache Module mod_authz_dbm: http://httpd.apache.org/docs/2.2/mod/mod_authz_dbm.html (2012 年 11 月 16 日確認)

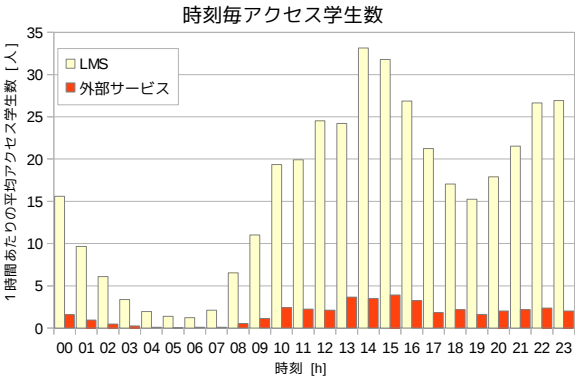

図 7. LMS 上の科目トップページ及び外部サービスへの時間毎アクセス数 (42 日分の平均)

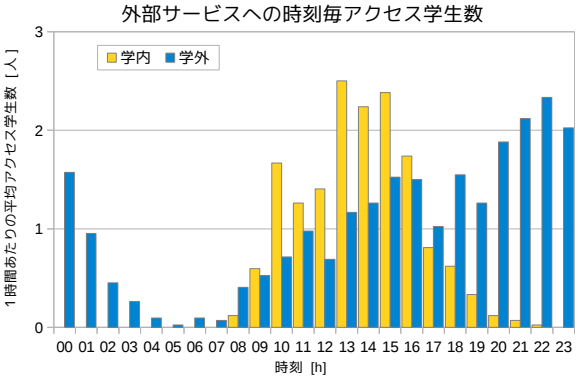

図 8. 外部サービスへの学内/学外からの時間毎アクセス数 (42 日分の平均)

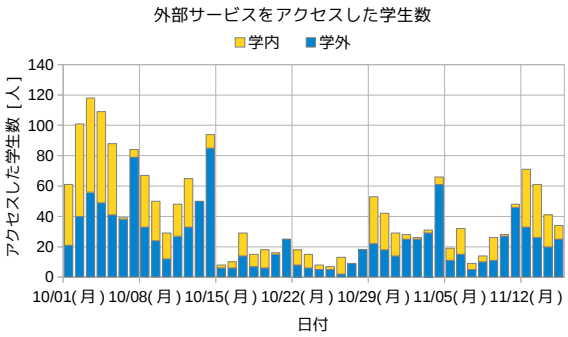

図 9. 外部サービスをアクセスした学生数の変化

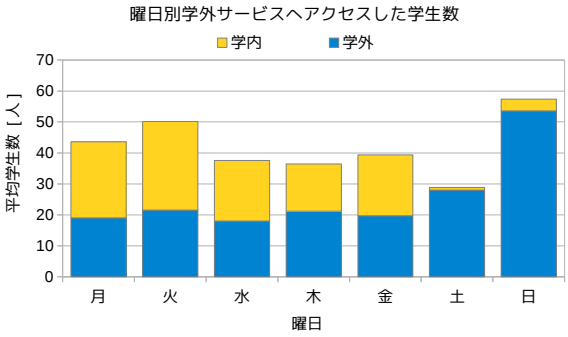

図 10. 学外サービスをアクセスした学生数の曜日による違い (42 日分の集計)