

[招待講演] Moodle用オンデマンド型動画配信システム

檜垣泰彦^{†,††} 清水健一^{†††,††††} 藤本茂雄^{†,††} 松本暢平^{††}

[†] 千葉大学アカデミック・リンク・センター 〒263-8522 千葉市稲毛区弥生町 1-33

^{††} 千葉大学国際未来教育基幹スマートオフィス 〒263-8522 千葉市稲毛区弥生町 1-33

^{†††} 千葉大学大学院工学研究科建築・都市科学専攻 〒263-8522 千葉市稲毛区弥生町 1-33

^{††††} 株式会社ブルーリンクシステムズ 〒102-0071 東京都千代田区富士見 1-5-5 第二大新京ビル 3階

E-mail: †higaki.yasuhiko@faculty.chiba-u.jp

あらまし 多くの大学でLMSとして用いられているMoodleで、授業のためのオンデマンド配信を行なおうとするばあい、十分な機能をもった手ごろな仕組みがない。そこで、小規模サイトから利用でき、かつ十分な機能と性能を備えた動画配信システムを設計した。受講者に限定した配信、勝手なダウンロードからの保護、視聴管理機能、モバイル環境・データダイエットへの対応、Moodle本体との独立性、容易な動画の管理の機能を備えており、スケーラビリティに優れている。コロナ禍における総合大学のオンデマンド型メディア授業の中心的仕組みとしてこれを利用し、その結果、機能的にも性能的にも十分であることを確認できた。

キーワード HTTP ライブストリーミング (HLS), スコーム, メディア授業, マルチビットレート, コロナ禍

[Invited Talk] On-demand video delivery system for Moodle

Yasuhiko HIGAKI^{†,††}, Kenichi SHIMIZU^{†††,††††}, Shigeo FUJIMOTO^{†,††}, and Yohei MATSUMOTO^{††}

[†] Academic Link Center, Chiba University 1-33 Yayoi-cho, Inage-ku, Chiba-shi, 263-8522 Japan

^{††} Smart Office, Institute for Excellence in Educational Innovation, Chiba University

^{†††} Graduate school of Engineering, Chiba University 1-33 Yayoi-cho, Inage-ku, Chiba-shi, 263-8522 Japan

^{††††} Blue Link Systems Co.,Ltd. Daidai Shinkyō 3F, 1-5-5 Fujimi, Chiyoda-ku, Tokyo, 102-0071 Japan

E-mail: †higaki.yasuhiko@faculty.chiba-u.jp

Abstract Moodle, which is used as an LMS in many universities, does not have a fully functional and highly scalable mechanism for on-demand distribution for classes. Therefore, we designed a video distribution system that is affordable, has sufficient performance, and has the following functions; delivery limited to students, protection from arbitrary downloads, viewing management function, support for mobile environment and data diet, independence from Moodle itself, easy video management features. We used this as the central mechanism of on-demand media lessons at our university in COVID-19 pandemic, and as a result, we confirmed that it had sufficient functions and performance.

Key words HTTP Live Streaming (HLS), SCORM, media class, multi bit rate, COVID-19 pandemic

1. はじめに

千葉大学では「千葉大学グローバル人材育成“ENGINE”」^(注1)を策定し、2020年度より学部・大学院生の全員留学を実施することとし、その一環として留学中でも科目履修が継続できる教育環境整備を行っていた。そのためのインフラとして、

Moodleでオンデマンド動画配信を実現するための仕組みを構築し準備を進めていた[1][2]。

ところが、2020年に入ってからCOVID-19の広がりを受け、その位置づけは大きく変化し、開講数7,000にも及ぶ全ての授業をオンデマンド形式のオンライン授業で実施することとなった。本システムはその中心を担うこととなり、配信パラメータや構成を調整のうえ、2020年度前半の完全オンライン(オンデマンド)授業に臨んだ[3]。本稿では、本システムの設

(注1) : ENGINE, Chiba University, <https://www.chiba-u.ac.jp/engine/>

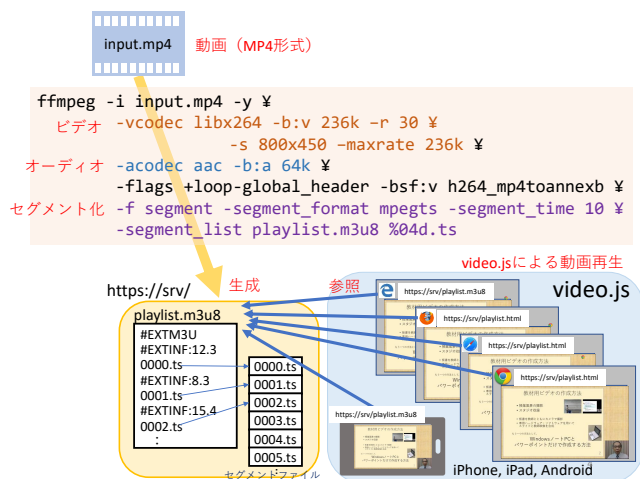


図1 HLS 動画配信基本形 [4]

計の概要 [4] とその後の最新の運用状況について述べる^(注2)。

2. 設 計

千葉大学では 2010 年度より全学 LMS(Learning Management System)として Moodle^(注3)を使用している。Moodleで動画配信を行うには、mp4形式の動画をファイルリソースとして掲載する、YouTube等の限定公開のURLを埋め込むなどの方法があるが、これらの方法では本システムが対象とするメディア授業用オンデマンド配信で必要となる機能を十分に実現できない。そこで、これを実現するための要件を定義し、それを満たすよう設計を行なった。

2.1 HLS 動画配信基本形

図1に示すように、本システムではHLS(HTTP Live Streaming)による動画配信方式[6]を採用している。入力となる動画(input.mp4)をffmpeg^(注4)を用いて複数の.tsファイルに分割(セグメント化)すると同時に、対応するプレイリストファイル(playlist.m3u8)を生成する。プレイリストファイル中には、セグメントファイルの位置情報とそのセグメントに収容されている動画(部分)の長さ(秒)が記述されている。動画配信サーバは通常のHTTPサーバであり、クライアントからのリクエストに応じてプレイリストファイルやセグメントファイルを送出というシンプルな構成となっている。EdgeやiOS、Androidによるモバイルデバイスでは、このプレイリストのURLを指定することでHLS形式をネイティブ再生できるが、その他のブラウザではHTML5プレイヤーフレームワークであるvideo.js^(注5)を用いて再生できる。Moodleとの連携はeラーニング教材用の仕組みであるSCORM[7]を用いて実現している。このよう

(注2)：本稿は、2020年 LOIS 研究賞受賞に伴う招待講演の予稿として、受賞した発表[3]に加え、その後の発表[4][5]の内容をもとに最新の情報を反映したものである。

(注3)：Moodle - Open-source learning platform, <https://moodle.org/>

(注4)：FFmpeg A complete, cross-platform solution to record, convert and stream audio and video., <https://ffmpeg.org/>

(注5)：Video.js - Make your player yours with the world's most popular open source HTML5 player framework, <https://videojs.com/>

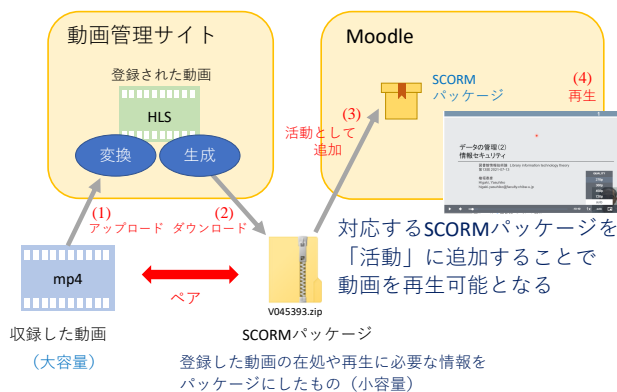


図2 動画登録から再生までの流れ

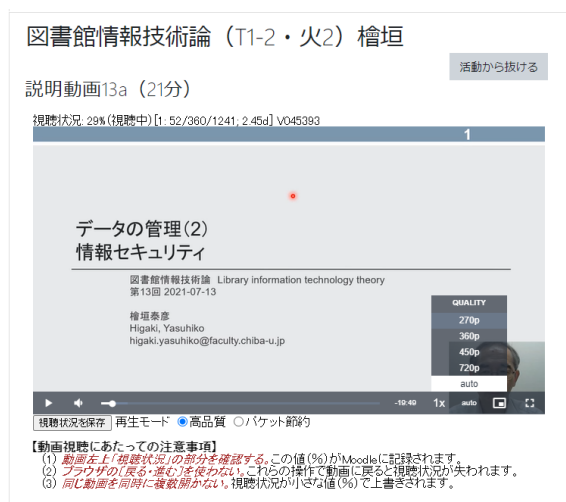


図3 Moodleにおける動画の再生

に、動画配信の中心となる部分は、シンプルなHTTPサーバであるため、研究室レベルから大学全学のような大規模な配信までスケラブルに対応できるという特徴がある。

2.2 機 能

図2に動画登録から再生までの流れを示す。Moodle本体との独立性を高めるため、Moodleとは別に動画管理サイトを設ける構成としている。収録またはパワーポイントからエクスポートして生成したmp4形式の動画を動画管理サイトに(1)アップロードすることで動画をHLS形式に変換・登録した後、対応するSCORMパッケージを生成・(2)ダウンロードし、それをMoodleの当該コースの(3)活動として追加することで(4)動画再生可能となる。「変換」では図1のセグメント化が行なわれる。生成されたSCORMパッケージには、プレイリストが含まれている。

図3に動画の再生の様子を示す。スライダーで再生位置を調整できるが、先送り再生(飛ばし再生)を禁止した動画では、未視聴部分への移動はできない。SCORMの設定により1倍速再生(1x)の他にn倍速再生(0.5x, 1.2x, 1.5x, 2.0x)を許可することができる。また再生モードの選択と画質の選択メニュー(QUALITY)を設けている。

2.3 動画配信の構成

図4にMoodleによるHLS動画配信の構成を示す。左半分

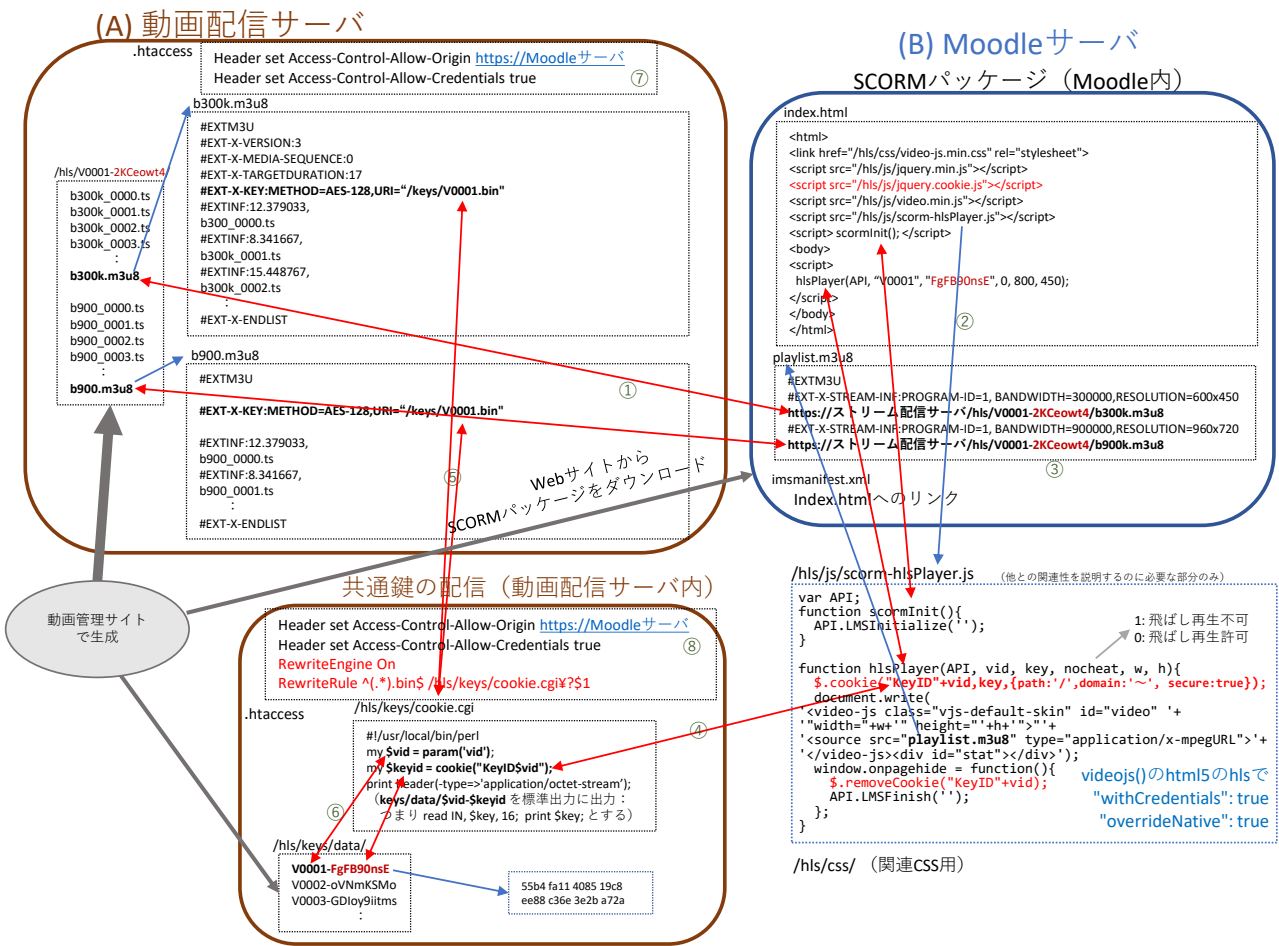


図4 MoodleにおけるHLS動画配信の構成[4]

(A)が動画配信サーバで、右半分(B)がMoodleサーバである。動画配信サーバにはプレイリストファイルとセグメントファイルが格納されている。

Moodleサーバでは、授業のコース内にSCORMパッケージが置かれ、その中にffmpegが生成したプレイリスト(サブプレイリスト)へのポイントとなるマスタープレイリスト(playlist.m3u8)が配置されている(①^(注6))。また、SCORMパッケージ中のHTMLファイルで、video.jsを利用した動画プレイヤー(scorm-hlsPlayer.js)を起動している(②)。

2.4 要件

2.4.1 受講者だけに限定した配信

授業のコース内に“Moodleの活動”として追加したSCORMパッケージにアクセスできるのは、そのコースの登録学生=受講者だけである。SCORM中のマスタープレイリストに記述された、動画配信サーバ中の対応するプレイリスト(サブプレイリスト)のURLの一部にはマジック文字列(例では“2KCeowt4”)が含まれており(③)、受講者に限定した動画再生を実現できる。

さらに、この文字列が漏洩したばあいの対策として、HLSのセグメントファイル暗号化オプションを利用している。図5に示すように、各セグメントファイルをAES共通鍵暗号化で暗

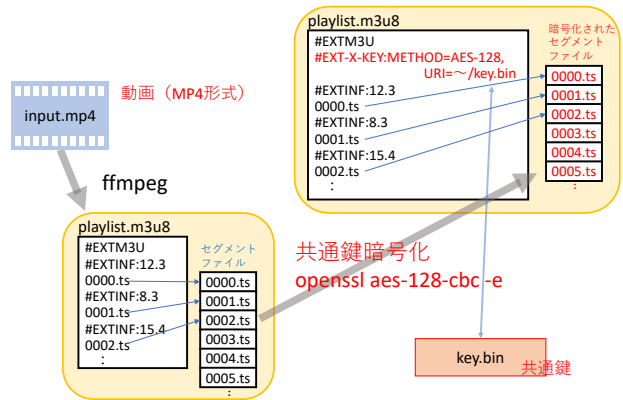


図5 HLSのAES共通鍵による暗号化[4]

号化し、その共通鍵の所在をサブプレイリスト中で指定することができる。図4の左下部分が共通鍵を保持している部分であり、これを動画配信サーバ内に置いている。tsファイルの所在が漏洩しても、この共通鍵の所在がわからなければ、暗号化されているため動画を再生することはできない。共通鍵の所在はサブプレイリスト中に記述されているため、サブプレイリストの所在がわかれば、動画を再生できてしまう。

そこで、さらに共通鍵の所在を隠蔽する仕組みを設けている。SCORMパッケージ中で別のマジック文字列(例では

(注6): 以下、丸囲み数字は図4内に対応する

説明動画13a (21分)

情報		レポート			
姓/名	学生証番号	受験	開始日時	最終アクセス日時	評点
千葉 太郎	2020001	1	2021年 06月 28日(月曜日) 14:24	2021年 07月 14日(水曜日) 20:28	29.01
稲毛 弥生	2020002	1	2021年 07月 12日(月曜日) 02:31	2021年 07月 12日(月曜日) 02:45	100

図 6 Moodle における視聴状況の確認

“FgFB90nsE”) の値を Cookie に設定し、動画配信サーバの共通鍵配信部分の CGI(cookie.cgi) でそれを受け取り (④), 実際の鍵の在処を特定するようにしている. この CGI はサブプレイリストの共通鍵指定に対応している (⑤). 実際の共通鍵は別の場所にファイルとして格納されており, それを CGI で取り出して利用する (⑥). つまり, Moodle のコース中の SCORM で設定する Cookie が通る位置関係でないと, AES 共通鍵は実際には渡らない. Moodle サーバから動画配信サーバに Cookie を通すため, CORS の設定が必要である (⑦⑧).

2.4.2 動画のダウンロード禁止

HLS 形式の動画のダウンロードは, ffmpeg 等のツールでプレイリストファイルを指定することで可能である. 本方式では, 共通鍵の取得に Moodle サーバとの間で Cookie の受け渡しを行うように設計したため, 動画のダウンロードをより困難としている. 通常のダウンロードツールは, ブラウザ外で動作するため, Cookie が渡らずダウンロードできない結果となる.

2.4.3 視聴管理

動画教材を SCORM として実装したため, SCORM の LMS API 機能を利用して, 教材側から Moodle へ視聴情報 (スコア=動画の何% を視聴済みか) を渡すことができる. 各動画の視聴状況は, Moodle のレポート (図 6) で確認できるほか, トラック詳細のページで cmi.core.total_time の値として総視聴時間の確認も可能である. ログから詳細な視聴状況の確認も可能である.

実運用で視聴管理の信頼性が問題となった. 一度視聴して 100% となった視聴履歴がクリアされる場合があるという報告があり調査した結果, 同時に同じ動画を開いて再生した場合, 最後に閉じた動画の視聴履歴で上書きされる, ブラウザの“戻る”や“進む”操作で動画のページに移動すると視聴履歴がクリアされる場合があることが分かった. これらに対する根本的解決は困難であるため, 現在は, 動画視聴にあたっての注意事項として, 図 3 に示すようにプレイヤーの下に『ブラウザの [戻る・進む] を使わない. これらの操作で動画に戻ると視聴状況が失われます.』『同じ動画を同時に複数開かない. 視聴状況が小さな値 (%) で上書きされます.』を記して対応している. ユーザの誤操作に対しても十分な視聴管理機能の信頼性を確保するためには動画プレイヤー部分の再設計が必要である.

2.4.4 モバイル環境・データダイエットへの対応

モバイル環境のためには, 通信状況に応じてビットレートを変えられることのできるマルチビットレート構成が有効である [2]. また, 意図的に低いビットレートを選べるようにすることで,

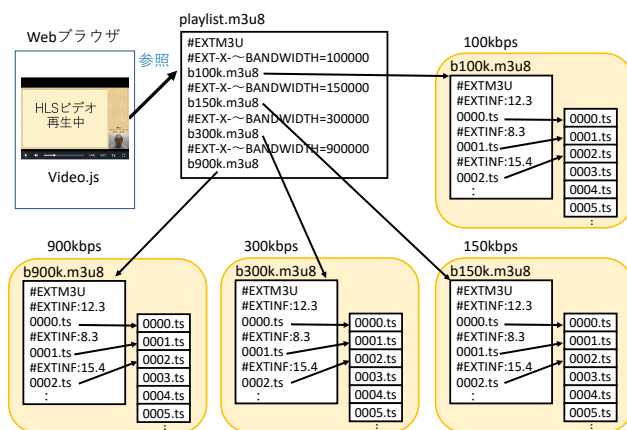


図 7 マルチビットレート HLS の構成

パケットを節約するための“データダイエット”に対応できる. 図 7 に示すように, HLS では異なるビットレートのセグメントファイル群を用意することで, マルチビットレートへの対応が可能である. マスタープレイリスト中でビットレートに応じたサブプレイリストへの振り分けを記述できる (①).

2.4.5 Moodle 本体との独立性

Moodle で HLS 動画を利用する方法としては, Moodle 本体に組み込む, プラグインとして実装する, SCORM パッケージとして実装する等の方法が考えられる. ここで採用した SCORM パッケージとして実装する方法は, Moodle との独立性が高く, Moodle の進化の影響を受けにくいのが特徴である. その反面, 動画を利用する Moodle とは別のサイト (動画管理サイト) に動画をアップロードして, SCORM パッケージを生成し, それを Moodle に追加して利用するという一連の手順が利用者にとってはわかりにくいと感じられる場合がある.

動画を SCORM パッケージとして管理することで, 容量の大きな動画本体 (HLS の場合はセグメントファイル群) そのものに代わって SCORM パッケージで管理を代行できる. そのため, 動画の管理が容易となる. 教材動画のコレクションを構築するためには, 容量の小さな SCORM パッケージをデータベースで管理することで実現できる.

さらに, 誰が作成した SCORM パッケージでも利用できる, Moodle のアカウントとは独立に管理できる点が利点としてあげられる. 動画教材の登録者と利用者は別で構わないため, 動画教材の共有に便利である. 複数の授業で同じ動画を利用しても実態 (セグメントファイル) は一つであるにもかかわらず視聴管理は別々にできる.

2.4.6 動画コンテンツの管理

Moodle の他に動画管理サイトを設けて ffmpeg によるトランスコードや共通鍵の生成, 暗号化を行い, 動画配信サーバに転送する仕組みとしている. コンテンツの流れを図 8 に示す. 利用者は mp4 形式等の動画ファイルを動画管理サイトにアップロードする. ffmpeg でセグメント化され, セグメントファイルは AES 暗号化される. サブプレイリストには共通鍵サーバの URL を記入し, それらを動画配信サーバ群に転送する. マスタープレイリストファイルをもとに, SCORM パッケージジェ

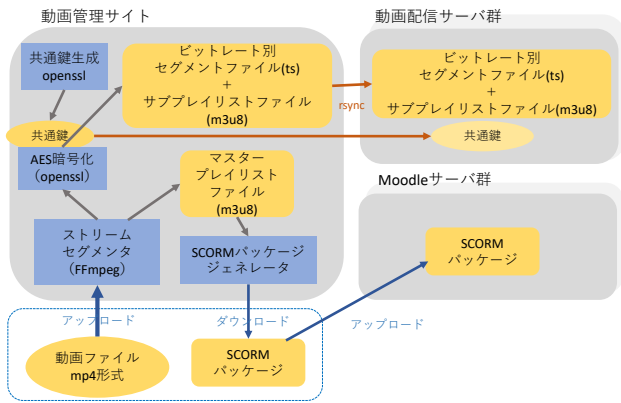


図 8 HLS 動画配信コンテンツの流れ [4]



図 9 動画管理サイト (トップページ) [4]

ネレータで生成された SCORM パッケージは、ユーザによってダウンロードされる。SCORM パッケージは、必要なファイルが入った ZIP アーカイブ形式である。これを、Moodle の活動として追加して利用する。

図 9 に動画管理サイトでログイン後に表示されるトップページの例を示す。アップロード (登録) 済みの動画のサムネイルが表示される。「動画の新規登録」のリンクがトップに設けられている。

このリンクから、動画をアップロードすると、処理待ち状態となる。処理待ち状態では、処理中の件数と、処理待ちの件数が表示される。処理中の状態では、現在の処理内容 (音量調整処理中、動画変換処理中、ポスター生成中、暗号化処理中、ファイルの転送中) やトランスコード (動画変換処理) の経過パーセント、終了予定時刻が表示される。

変換が終了すると、動画ビューアが表示される。アップロードした動画のプレビュー再生ができる。このページから、SCORM パッケージのダウンロードが行える。どぼし再生 (自由な位置からの再生) の可否や、ポスター (サムネイル画像) の指定が可能である。

Moodle 以外の LMS からの利用もできるように、いわゆる「限定 URL」発行の機能も備えている。この機能をつかうこと

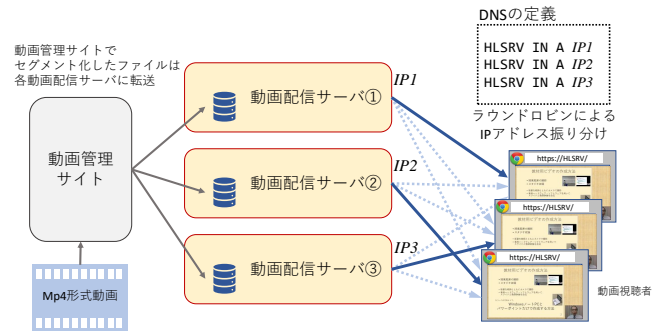


図 10 動画配信サーバの構成

で、メールによる教材動画の配布も可能となる。複数発行することができ、必要な期間を過ぎたら無効にすることができる。

管理機能として、動画変換ジョブの管理 (モニタリング) 機能を設けている。同時に実行できる変換ジョブは、ロードアベレージで制御している。

動画管理サイトを Moodle から完全に独立させたことで、動画管理サイトが停止している場合でも動画の配信が行なえる、Moodle や配信サーバが停止しているばあいでも、動画の登録 (セグメント化処理) が行なえる、等のメリットがある。

3. 運用

コロナ禍の 2020 年度から、在籍学生数約 14,000 人の大規模な大学において、本システムは効果的に機能し、結果的にその有効性を実証することとなった。配信サーバの運用と稼働状況について述べる。

3.1 動画配信サーバ

図 10 に動画配信サーバの構成を示す。コロナ禍におけるハードウェア増設であったため、即納可能なタワー型のサーバ仕様のもの^(注7)を購入し利用した [3]。同じ内容で IP アドレスが異なる 3 台のサーバで構成されており、それらを管理する DNS において、DNS ラウンドロビンとして定義されている。動画管理サイトでセグメント化した動画と共通鍵は、動画登録時に各サーバに転送される。各動画視聴者にはこの 3 つの IP アドレスのどれかが返され、そのサーバと通信する。

この運用において、どれかの配信サーバが HTTP に応答しないとき、視聴者側の video.js はそのサーバとの接続を諦め別の IP アドレスへ接続する。そのため、サーバのメンテナンス時は、そのサーバの apache を停止するだけで利用者に影響することなく動画配信の継続したままでの作業が可能である。このことは、配信サーバのメンテナンスを容易にしている。また、配信サーバの増設は容易であり、規模に応じた台数を用意することで、高いスケーラビリティを実現できる。

3.2 稼働状況

図 11~14 に 2020 年度前半・後半と 2021 年度前半 (主な前半の授業期間を終了した本稿執筆時点 7/26 まで) の本システム稼働状況のグラフを示す。

(注7) : NEC Express5800/T110j Xeon E-2124, メモリ 32GB, HDD 4TB × 4 raidz1, FreeBSD 12.2 × 3 台 (2021 年 7 月現在)

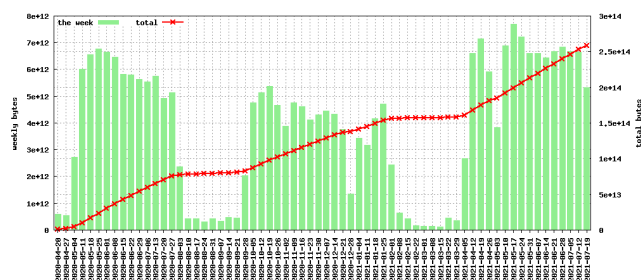


図 11 動画配信サーバ配信バイト数

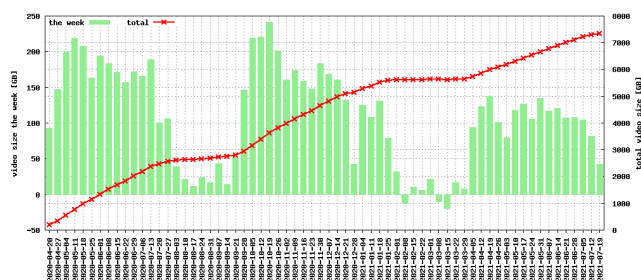


図 14 登録動画総量

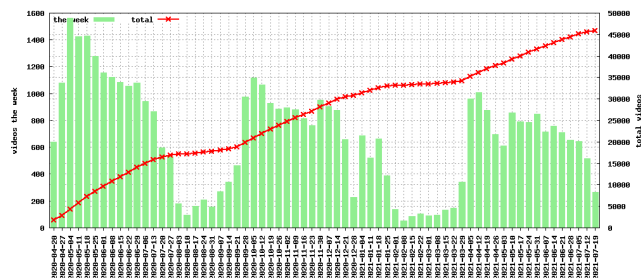


図 12 登録動画数（登録された動画数）

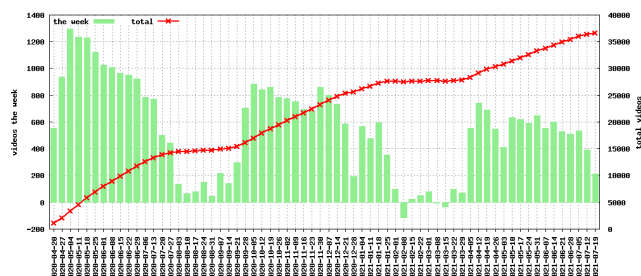


図 13 登録動画数（登録されている動画数）

3.2.1 動画配信サーバ配信バイト数

図 11 は動画配信サーバの配信バイト数（3 台合計）である^(注8)。2020 年度前半（第 1・第 2 ターム期間）と 2021 年度前半を比べると、2021 年度のほうが週ごとの配信量が増加している。配信開始以来、もっとも配信量が多かったのは、2021 年 5 月 17 日からの週で 7.70TB/週（平均 1.10TB/日）である。2020 年度前半の総配信バイト数は 81TB、後半は 77TB、2020 年度 1 年間の総配信バイト数は 158TB である。2021 年度前半については本稿執筆時点（7/26）で 100TB である。

3.2.2 登録動画数

図 12、図 13 は登録動画数の推移である。図 12 は登録操作が行なわれた動画の数であり削除は反映されない。それに対し、図 13 は各時点で実際に登録されている動画の数である。期間中、授業開始直前の 2020 年 5 月 4 日の週に最大登録数 1560 件/週（平均 223 件/日）の動画が登録された。2020 年度 1 年間に登録された動画数は 33,336 件であった。

図 13 の赤の実線はその時点で登録されている動画の数の推移を表している。緑の棒グラフはその週に登録された動画数から削除された動画数を引いた値である。2020 年度で利用した

動画のうち必要なくなったものを削除したため、2021 年 2 月 8 日の週はマイナスとなっている。2020 年度前半終了時点で 15,524 本、後半終了時点で 27,752 本の動画が登録されている。2021 年度前半は本稿執筆時点（7/26）で 36,559 本の動画が登録されている。

3.2.3 登録動画総量

図 14 に動画配信サーバにおける登録動画総量の推移を示す。赤の実線が各時点で 3 台のサーバに蓄積されている動画サイズの推移であり、緑の棒グラフが週ごとの変化量である。2 年目である 2021 年度のほうが 2020 年度よりも増え方が少ないのは、1 年目の動画を再利用している授業があるためであると考えられる。2020 年度終わり時点の総量は 5.6TB、2021 年度前半の本稿執筆時点では 7.2TB である。

4. まとめ

Moodle でオンデマンド型動画配信が行えるシステムを構築した。メディア授業用として必要な機能を備えており、コロナ禍の 2020 年度から 2021 年度前半にかけての千葉大学における運用において有効性を実証する結果となった。シンプルな構成であり、研究室レベルの配信から総合大学における配信まで対応できる。

文 献

- [1] 檜垣泰彦, 池田晶子, 大木勇人, 藤本茂雄: e ラーニングを指向した Moodle 動画配信方式, MoodleMoot Japan 2020, 2020-02-28 <https://moodlejapan.org/mod/data/view.php?id=36&rid=2071> (2021 年 7 月 26 日 閲覧)
- [2] 原田明成, 張徳鵬, 檜垣泰彦, “メディア授業に適した授業動画配信方式の検討,” 信学技報, vol. 119, no. 477, LOIS2019-67, pp. 63-67, 2020 年 3 月.
- [3] 檜垣泰彦, 藤本茂雄, 高瀬浩一, 武内八重子, 松本暢平, 池田晶子, 清水健一, 岡田聡志, “メディア授業用動画配信の実施～ COVID-19 への緊急対応 ～,” 信学技報, vol. 120, no. 93, LOIS2020-3, pp. 1-6, 2020 年 7 月.
- [4] 清水健一, 檜垣泰彦, 藤本茂雄, 須貝康雄, “e ラーニングを指向した Moodle 動画配信システムの設計,” 信学技報, vol. 120, no. 149, LOIS2020-7, pp. 13-18, 2020 年 9 月.
- [5] 清水健一, 檜垣泰彦, 藤本茂雄, 須貝康雄, “オンデマンド型動画配信システムの配信性能の検証,” 信学技報, vol. 120, no. 417, LOIS2020-50, pp. 18-23, 2021 年 3 月.
- [6] HTTP Live Streaming, IETF, rfc8216, August 2017 <https://tools.ietf.org/html/rfc8216> (2021 年 7 月 26 日 閲覧)
- [7] Advanced Distributed Learning Initiative, SCORM 1.2, <https://www.adlnet.gov/projects/scorm-1-2/> (2021 年 7 月 26 日 閲覧)

(注8) : 訂正 : [5] の図 7 のグラフは配信サーバ 3 で配信したバイト数が正しく反映されていなかった。本稿の図 11 が正しい。