

東大 LMS のオンライン授業対応

岡田 和也¹⁾, 関谷 貴之¹⁾, 田中 哲朗¹⁾, 品川 高廣¹⁾, 柴山 悅哉¹⁾, 工藤 知宏¹⁾, 田浦 健次朗^{1),2)}, 蘆田 隆行³⁾, 友西 大³⁾, 郡司 彩³⁾, 平田 肇⁴⁾, 松井 章良⁴⁾, 斎藤 知毅⁴⁾

1) 東京大学 情報基盤センター

2) 東京大学 情報理工学系研究科

3) 東京大学 情報システム部

4) キヤノン IT ソリューションズ株式会社

okada@ecc.u-tokyo.ac.jp

Lessons Learned: LMS Operations for University-wide Online Lectures under COVID-19

Kazuya Okada¹⁾, Takayuki Sekiya¹⁾, Tetsuro Tanaka¹⁾, Takahiro Shinagawa¹⁾, Etsuya Shibayama¹⁾, Tomohiro Kudo¹⁾, Kenjiro Taura^{1),2)}, Takayuki Ashida³⁾, Dai Tomonishi³⁾, Aya Gunji³⁾, Tsuyoshi Hirata⁴⁾, Akiyoshi Matsui⁴⁾, Tomoki Saito⁴⁾

1) Information Technology Center, The University of Tokyo

2) Graduate School of Information Science and Technology, The University of Tokyo

3) Department of Information Systems, The University of Tokyo

4) Canon IT Solutions Inc.

概要

東京大学では、新型コロナウィルス感染症（COVID-19）の拡大を受け、3月18日に授業のオンライン化の推奨を、さらに3月31日には完全オンライン化を決定した。そして、学事暦通りに4月3日に授業を開始した。大部分の教員がオンライン授業の経験値がない状態から始めて、非常に短期間でオンライン授業の準備を行った。

学習管理システム（LMS: Learning Management System）では、各種連絡通知、資料配布、レポート回収など、従来は一部の授業でしか利用されていなかった機能が、多数の授業で利用されるようになった。それにともない、LMS の負荷が増大した。昨年度は、授業期間中の比較的よく使われる時間帯で、同時ログイン数はせいぜい2,000人程度であったが、今年度は11,000人以上になった。そのため、大幅な性能強化が必要となった。

3月中にソフトウェアのチューニングを行い、4月1日にはサーバの増強を行ったが、学期開始からしばらくの間は、高負荷にともなう性能劣化が頻発した。また、利用者の増加に伴い、従来から存在したバグの顕在化など様々な問題への対応が必要となった。

本稿では、授業の完全オンライン化に対応するために、高負荷による性能劣化やバグの顕在化にともなう障害などに如何に対応してきたかを報告する。また、LMS の性能監視の難しさ、今後の LMS のあり方について議論する。

1 はじめに

東京大学では、COVID-19 の流行に伴い 2020 年度夏学期（4 月～9 月）の授業を全てオンラインで実施した。全学的な授業オンライン化の決定に先行して、情報基盤センターと大学総合教育研究センターでは、オンライン授業に必要な各種情報をワンストップで提供するポータルサイト utelecon^[1] を立ち上げた。大学アカウントの設定方法、Zoom/Webex のアカウント作成・使い方、学習管理システム（LMS: Learning Management System）の使い方、オンライン授業のノ

ウハウ、FAQ などの情報を提供している。ポータル開設ではスピードを重視し、協力者が GitHub 上で順次コンテンツを更新する体制とした。授業の配信には、Zoom, Webex, Google Meet を全学で利用できるよう整備した。

オンライン授業では、Web 会議用 URL の共有、レポート回収、資料配布、テストなどをオンラインで完結させる必要があり、LMS の重要性が高まった。しかし、昨年度までの東京大学の LMS (ITC-LMS) は、授業期間中の同時ログイン数が最大 2,000 人程度であり、全授業を ITC-LMS で実施するには大幅な性能強

化が必要となった。

本稿は、授業の完全オンライン化にあたり ITC-LMS で実施した性能強化、発生したトラブルについてその原因と対応をまとめた。また、オンライン授業に適した今後の LMS に必要な機能についても議論する。これらの知見が他大学でも、LMS を全学規模で運用する際の一助になれば幸いである。

本稿では、まず第 2 節で ITC-LMS の構成について説明する。次の第 3 節では、4 月以降に発生した問題とそれらに対する対策を中心に、ITC-LMS の全学規模でのオンライン授業への対応をまとめた。第 4 節では、第 3 節で述べた全学規模での LMS 運用で得られた教訓をまとめた。最後に第 5 節では今後の LMS に求められる要素を整理する。

尚、本稿は 2020 年 9 月末時点での情報に基づき執筆されている。

2 ITC-LMS の概要

東京大学では、情報基盤センターが 2004 年から LMS のサービスを全学に提供している。当初は、CFIVE というオープンソースの LMS の開発・運用・サービス提供を行っていた。その後、必要な要件を検討した上で入札を行い、2014 年には キヤノン IT ソリューションズ（キヤノン ITS）が開発・運用する商用の LMS を導入した。2018 年には、LMS としての機能強化や可用性の向上を要件に加えた入札を行い、再度キヤノン ITS が落札した。その結果 2019 年 3 月から稼働しているのが、LMS パッケージシステム in Campus^[3] をカスタマイズした現在の ITC-LMS である。

先代の LMS と比べた現在の ITC-LMS のシステム構成上の特徴は、1) パブリッククラウドの利用、2) マイクロサービス・アーキテクチャ風のシステム構成の 2 点である。先代までの LMS は、情報基盤センター内に専用サーバを設置して運用していた。一方、現在の ITC-LMS の導入時点では、将来需要がどれくらい伸びるか予測が難しかったこと、限られた人手を機器の保守より利用者のサポートに回したかったことから、パブリッククラウドを採用した。

NTT 東日本のクラウドゲートウェイ サーバーホスティングサービス^[2] を利用し、学外のデータセンタで仮想マシン、ロードバランサ、DB を動作させている。大学とデータセンタの接続には、SINET L2VPN を利用し、東京大学に割り当てられた IP アドレスをクラウド内に持ち込み割り当てている。L2VPN の帶

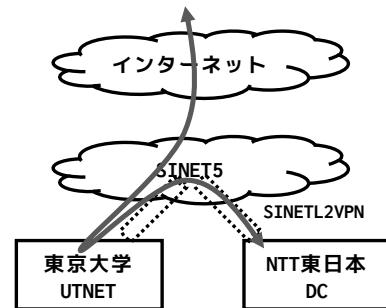


図 1 SINET L2VPN を利用した大学・データセンタ間接続

域は、1Gbps ベストエフォートを利用している（図 1 を参照）。今回のオンライン授業対応では、パブリッククラウドで運用していたことにより、オンプレミスと比べてサーバの構成変更や追加が比較的容易かつ短時間で実施できた。

先代の LMS では、モノリシックな構成のシステムをオンプレミスで提供していた。一方、現行システムでは、LMS の機能毎に異なるサーバに分割した疎結合アーキテクチャを採用している。機能毎の独立性が高い、個別にメンテナンス可能、スケールアウト・スケールダウンしやすい、冗長化による障害対策が可能ななどの利点は、マイクロサービスアーキテクチャと同様である。ただし、各サーバが同じデータベースサーバを共有しているため、データベースのメンテナンスは独立に行うことができず、スケールアウトにも限界がある。サービス基盤には Kubernetes^[4] を採用しており、仮想マシン（VM）4 台で Kubernetes クラスタを構成している。データベースには 2 台（active-standby 構成）の VM を別途割り当てている。

図 2 は、LMS 内部のシステム構成図である。利用者からのリクエストはロードバランサにて複数の nginx (ingress nginx) へ分散させる。そして、nginx から利用者毎の状態を管理するフロントエンドのサービスに転送される。その後、課題登録・提出、テスト登録・受験、資料登録・ダウンロードなどの機能に応じたバックエンドのサービスが利用される。各サービスは負荷状況に応じて 3-6 つのインスタンスがクラスタ内で動作しており、Kubernetes の機能で自動的に負荷分散される。例えば、試験期間中などテストのサービスの負荷が高くなる時期には、テストのインスタンスを増やすことで負荷耐性が高くなる。

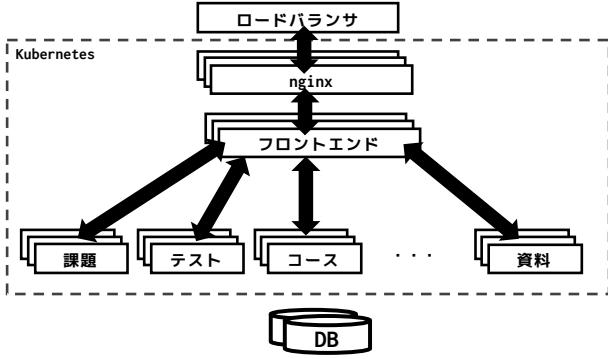


図 2 マイクロサービス型 LMS の構成

3 LMS でのオンライン授業対応

3.1 オンライン授業対応の時系列と体制

表 ①¹ は、東京大学におけるオンライン授業対応と ITC-LMS において発生した障害と対処、各種メンテナンス・増強を時系列にまとめたものである。3月に入ると全学的にオンライン授業に関する検討が行われ、3月18日の総長メッセージで「対面での講義は最小限とし、オンライン化を奨励し推進する」という方針が示された。さらに、3月31日には理事・副学長から完全オンライン化の通知が行われた。3月上旬の段階で、ITC-LMS の利用が増大することは既に予想されていたが、その後の需要予想は膨らむ一方であった。

利用者増に備え、情報基盤センターの情報メディア教育研究部門とキヤノン ITS で対応を協議した。そして、3月中にデータベースのチューニングを行い、4月1日には仮想マシンの CPU・メモリの追加、各種パラメータの変更、そして負荷試験を実施した。

しかし、教養学部前期課程（学部1, 2年生）の授業が始まった4月6日には、アクセスが集中する授業開始時刻前後の負荷に耐えられず、利用者が ITC-LMS にアクセスできない状況を招いてしまった。その後は各種問題に対応するために、2020年6月末まで、東大とキヤノン ITS の関係者で2-3日に1回のオンラインミーティングを開催し、修正作業を進めるサイクルを回した。障害の原因是、ネットワークの帯域不足、基盤システムのバグ、システムのパラメータ設定、ソフトウェアのボトルネックなど多岐に渡っていた。より迅速に問題の切り分けを行うため、運用を担当するキヤノン ITS だけでなく、東大の教職員もシステム内のスレッド数、各種負荷、アクセス数、転送データ量などの監視・データ収集・解析を行った。さらに、キヤノン ITS から特別に許諾を得て、東大の一部教職員が ITC-LMS のソースコードを閲覧可能とした。そして、ソースコードレベルで問題のある箇所の特定、改

善策の検討を行い、キヤノン ITS の改修作業にフィードバックした。

3.2 発生した障害と対応

3.2.1 nginx での不均等な負荷分散

事象の概要： ITC-LMS で利用者からのリクエストを振り分ける nginx は、次段のポッド^{*1}を構成するコンテナ4つに対して均等にリクエストを分散する仕組みになっている。しかし、4月の授業開始から nginx による負荷分散の不均等が明らかになり、特定のサービスへアクセスが集中してしまう事象が発生した。その結果、リクエストが集中したサービスが高負荷になり落ちてしまった。

事象の原因： 負荷分散に利用していた ingress-nginx での cookie を利用した負荷分散機能のバグが原因であった。

対応： 既にバグが修正された ingress-nginx が提供されていたが、基盤の Kubernetes 自体のアップデートも条件となっておりシステムの停止時間が長くなることが想定されたため、nginx の設定変更にて応急処置を実施した。その後、夏休み期間中の9月3日に実施したメンテナンスにて、Kubernetes のアップデートに合わせて、ingress-nginx のイメージ更新を実施し根本解決を行った。

3.2.2 静的コンテンツの帯域圧迫

事象の概要： ITC-LMS の画面を表示させるためにはメインの html ファイル以外に、多くの javascript ファイル、css ファイル、png ファイルが必要になる^{*2}。

事象の原因： Google Chrome のデベロッパーツールで見たところ、静的なファイルを GET した時の Response の中で、Cache-Control: no-cache としているために、ブラウザがこれらのファイルをキャッシュできずに、毎回 Web サーバまで取りにいっていた。静的ファイルの処理もフロントエンドでおこなうため、フロントエンドの負荷（および使用スレッド数）が増加するとともに、通信帯域を消費することになっていた。

対応： 該当ページの閲覧時に、ITC-LMS のシステム更新などがない限りは変更されないものなの

^{*1} ポッド (Pod) とは、Kubernetes 上でのアプリケーションの実行単位であり、1Pod が一つ以上のコンテナから構成される

^{*2} 筆者の一人が2020年9月25日に出講表を表示した時は、javascript ファイル9個、css ファイル11個、png ファイル13個だった

表1 ITC-LMSにおける障害対応の経過表

東大のオンライン授業対応	ITC-LMSで発生した障害	ITC-LMSの対応
2020/03/11 オンライン授業・web会議ポータルサイトを開設(utelecon)		
2020/03/18 オンライン授業を推奨する総長メッセージ		
2020/03/19 オンライン授業説明会・新入生へのオンライン授業準備を要請		
2020/03/26 オンライン授業説明会		
2020/03/31 授業の完全オンライン化を全学に通知		
2020/04/01		授業開始に備えてサーバのメモリ/CPUを増強
2020/04/06 教養学部前期課程 授業開始	ファイルのアップロードでエラーが発生 サービスへの負荷分散に偏りが発生	スレッド数の変更
2020/04/07		スレッド数を変更
2020/04/09		スレッド数を変更
2020/04/10		DBのコネクション数を変更
2020/04/13	データベースの緊急メンテナンスにより 日中にサービスを2時間停止	
2020/04/14		スレッド数の暴走に対して一時対応として定期的なサービス再起動を実施するよう変更
2020/04/15		スレッド数・コネクションプール数などの各種監視を強化
2020/04/16		
2020/04/17		コネクションプールのパラメータ変更
2020/04/18		コネクションプールのパラメータ変更
2020/04/19		
2020/04/20	ウェブサーバのメモリ不足が発生 nginxにおけるセッションの不均等問題 が発覚	
2020/04/21		
2020/04/22		テスト・負荷試験環境を準備
2020/04/24		サーバ増強
2020/04/25		nginxによる負荷分散の偏り問題
2020/04/27		pcbfのTomcatのスレッド数を変更
2020/05/01		
2020/05/07		API呼び出しの修正・大規模テストに向けた負荷試験実施
2020/05/12		
2020/05/17	Kubernetes基盤ネットワーク障害	L2VPN帯域の増強
2020/05/18		API呼び出しへのキヤッショ追加
2020/05/22		API呼び出し回数の削減
2020/05/24		Googleアナリティクス導入
2020/05/28 教養学部前期課程の大規模試験およびその予行演習	DBのメモリ不足によるシステム停止 学期末試験に向けた大規模テストリハーサルにて高負荷が発生	3000人テスト対応 コネクションプール/DBパラメータの変更 コネクションプール/DBパラメータの変更
2020/05/29		
2020/05/30	アップロードに対してバックグラウンドのウイルススキャンが追い付かない問題 が発生	メモリ割り当てを変更 コネクションプールのパラメータ変更
2020/06/17		更新通知性能改善リリース
2020/06/21		L2VPN帯域の増強
2020/09/03		Kubernetes基盤のアップデート L2VPN帯域の増強 Kubernetes基盤のアップデート

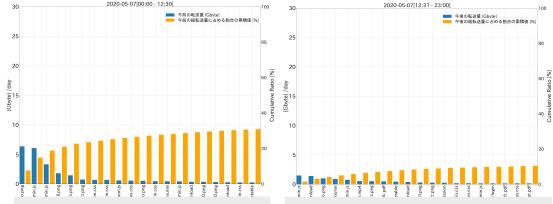


図 3 静的コンテンツの対応前後での通信量の多かった URL 上位 (2020 年 5 月 7 日)

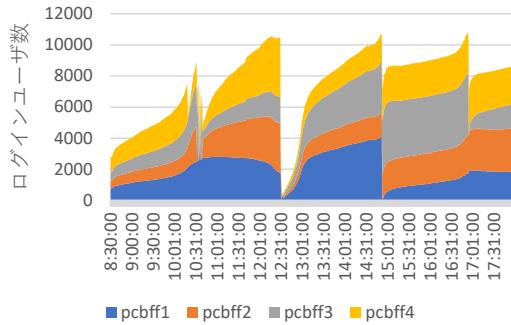


図 4 ログインユーザ数の推移 (2020 年 4 月 21 日)

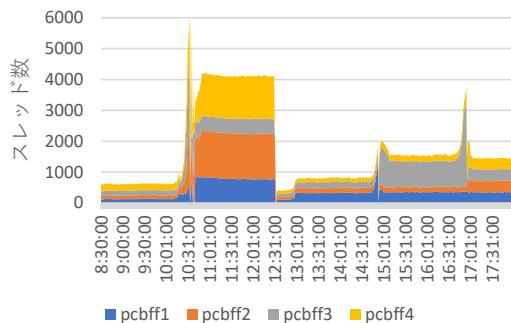


図 5 pcbff スレッド数の推移 (2020 年 4 月 21 日)

に、Google Chrome のデベロッパーツールで見たところ、1 回のセッションの中でも毎回 Web サーバから転送していて、それぞれ数十 ms から数百 ms かかっている様子が観察された。デバッグ等の目的以外では、積極的に no-cache とするべき理由はないので、静的ファイルについては Cache-Control: no-cache を外す変更をおこなった。図 3 は通信量が多かった URL をグラフにしたもので、対応をおこなった日（昼休みに対応）の左側が午前（対応前）、右側が午後（対応後）になっている。各グラフの横軸の URL は URL の末尾のみを残している。対応前は、png, js, css などの静的コンテンツの URL が上位を占めていたのに対して、対応後は他の URL が増えてきていることから、キャッシングにより不要な通信の削減につながっていることがわかる。

3.2.3 フロントエンドサービスのスレッド暴走

概要： 授業開始時などにログインするユーザの数が急増すると、セッション管理やページ生成などを担当している pcbff と呼ばれるフロントエンドサービスのスレッドの数が異常に急上昇する現象が頻発し、実質的にユーザのリクエストに答えることができなくなってしまった。

事象の原因： Kubernetes における最小実行単位は Pod と呼ばれ、1 つまたは複数のコンテナで構成されている。pcbff は 4 台のコンテナで分散してフロントエンドサービスを提供していたが、図 4 に示すようにログインユーザ数がピーク時には 10,000 人を超えるようになり、pcbff に以前よりも大きな負荷がかかるようになった。その結果、何らかきっかけで 4 台のいずれか、あるいは複数の pcbff のスレッド数が通常時は多くても 500 程度であったものが数分のうちに 2000~4000 以上程度まで急上昇し、その Pod に振り分けられていたユーザからのリクエストに応答できなくなる状態が発生した。図 5 では、10:25 や 16:50 などの授業を開始するタイミングで、pcbff のスレッド数が急増する様子が分かる。

対応： 当初はスレッドプールのサイズを増やすなどの対策をとったが、いったんスレッド数が上昇し始めると十数分程度で上限に達してしまったため、解決に至らなかった。根本的には pcbff のバグ（負荷が高くなると処理の待ち時間が長くなりページの生成に時間がかかりすぎる）と思われたが、すぐに原因を突き止めて修正することが困難であった。応急措置として授業開始前などのタイミングで pcbff を再起動する予防策を実施した。副作用としてログイン中のユーザが強制的にログアウトされることになったが止むを得ないと判断した。図 4 では、pcbff の再起動により定期的にログインユーザ数が急減している様子が見て取れる。その後、サーバ資源の増強などをおこなって処理能力に余裕をもたせた状態で運用し、その間に pcbff のプログラムの修正をおこなってスレッド数の急上昇が起きにくくなるようにした。

3.2.4 コネクションプールの設定間違い

概要： フロントエンドとバックエンドサービスの間で確保されているコネクションプール数が少なく、リクエストが正しくバックエンドに流れず処理能力が大幅に低下していた。

事象の原因：マイクロサービスアーキテクチャでは、サービスを分割して動かすためにサービス間で RESTful API 等を利用してデータのやりとりを行う。この際にコネクションが必要となり、特定のサービスへの接続でコネクションを張りすぎないように制約を設けている。原因是、フロントエンドとバックエンドサービスの間で確保されているコネクションプール数が接続数に対して大幅に少なく、リクエストが正しくバックエンドに転送できず、キューが溢れてしまったのが原因である。

対応：コネクションプールのサイズ割り当てを再設計し、変更を実施し対応した。その後、解決したかに見えたが、割り当ての間違いによりコネクションプールを想定通りに利用できない状況になっていた。再度割り当ての検討と設定の修正を実施し、バックエンドのサービスに処理が適切に割り当てられ流れるようにした。

3.2.5 SINET L2VPN 帯域の逼迫

概要：L2VPN の帯域が狭く、各種ファイルのダウンロード・アップロードに時間を要していた。

事象の原因：学期開始前の準備段階の負荷試験時に帯域が想定よりも狭くなっている可能性が判明した。SINET の L2VPN を利用しているため学内のネットワーク設定、SINET 側の VPN 設定、クラウド内部の VPN 設定を見直し、クラウド内部の帯域が設計よりも狭いことを発見した。

対応：キヤノン ITS とパブリッククラウド事業者と協力し、L2VPN の帯域を段階的に改善し、帯域がボトルネックとなる障害は解消された。

3.2.6 大規模一斉オンラインテスト

概要：学期末が近づくと、期末テストのオンライン化が問題となった。従来のようにテストを一斉に開始するなら、1 年次必修科目では、3,000 人以上の学生がほぼ同時にテスト開始ボタンを押し、ほぼ同時に解答を提出することになる。これが実施不可能であることは容易に想像できる。公平性を保てる範囲で、解答の開始時刻に幅を持たせた実施方法を模索する必要があった。まずは、負荷テストの実測値からシステムのスループット（単位時間あたりに処理できるリクエスト数）を推定し、英語のテストの場合、10 分間に 3,000 人強がある程度分散して受験を開始すれば、処理可能と判断した。5 月末に、実際に約 3,000 名の学生を動員

してリハーサルを行ったところ、レスポンスが極度に悪化し実用的な性能が得られなかった。

事象の原因：ログを調べたところ、最初の 10 秒間に約 600 名の学生が受験を開始していた。さらに、開始時刻前後に学生が連打し、大量のリクエストが発生していた。

対応：リハーサル実施担当者にシステムの性能特性を正確に伝えられず、学生向けの指示が足りなかつたのが反省点である。その後、情報科目でも同様のリハーサルとさらに本番を行ったが、性能上の大きな問題は発生しなかった。この科目的本番では、学生を 10 グループに分割し、各グループが 2 分間隔で解答を開始した。また、テスト問題を指定する際にはダイレクトリンクを用い、他のページへのアクセスを減らす工夫も行った。

4 オンライン授業対応の教訓

本節では、ITC-LMS のオンライン授業対応と各種障害への対応を経て得られた教訓をまとめる。

4.1 マイクロサービスアーキテクチャの難しさ

第 2 節で述べたように、ITC-LMS ではマイクロサービスアーキテクチャを採用し、Kubernetes 上でサービスを提供している。マイクロサービスとして LMS の機能を分割し独立させることで、開発の迅速化と保守性が向上した。しかし、今回のオンライン授業対応や各種障害を経験してマイクロサービスアーキテクチャで運用していく上で第 3.2.4 節のコネクションプールに起因する問題が明らかになった。

サービスが独立しているため、サービス間のやりとりは RESTful API などをを利用して行われる。そのため、高負荷やバグにより応答処理に時間を要する場合は、特定のサービス間のセッション数が増加し上限に達すると、他のサービスへの新規接続ができない状況が発生してしまう。こうした問題を避けるために、サービス間のセッション数に上限を設けるコネクションプールと呼ばれる仕組みが利用されている。ただし、各サービスへのアクセス数は均等ではなく、頻度や傾向により発生する偏りを考慮して上限を設定しなければならない。LMS の場合は、LMS の同時利用者数、利用される機能の傾向を把握した上でコネクション数の上限を割り当てることが肝要である。コネクションプールの割り当てが必要な数よりも少ないと、サービスの負荷は上昇しないがリクエストが詰まり、システム全体の処理遅延やタイムアウトを招いてしま

う。一方で上限を高く設定した場合には、コネクション自体は円滑に張られる一方で、データベースへのアクセスが過負荷になってしまい処理速度の低下やタイムアウトを招いてしまう可能性がある。データベースへの接続時にもリクエストをプーリングする設定があるため、各種データベースに応じた適切なプーリングパラメータを設定しなければならない。

ITC-LMS では、第 3 節で述べた通り、2020 年 4 月から 7 月に至るまでにサービス間のコネクションプール、データベースのパラメータ最適化を継続して実施してきた。パラメータの最適化に時間を要したのは、システム内のプーリングが多段になっており、繰り返し試験と修正が必要であったためである。また、システム稼働時にどの程度プールが消費されているかを把握するための監視が不足していたことも性能劣化の原因特定が遅れた要因である。マイクロサービスアーキテクチャは昨今多くのサービスで採用されるようになっているが、システム性能を十分に出すには、マイクロサービスに適したサービスの設計やコネクションの最適化手法を十分に把握し、設計・構築・テスト・運用が可能な技術を備えている必要がある。

4.2 オンプレミス型 LMS vs クラウド型 LMS

ITC-LMS は、2019 年度の更新で Infrastructure as a Service (IaaS) 型のパブリッククラウド上に構築して運用している。従来のように大学内にサーバを設置し、オンプレミスで運用している場合と比較していくつかの長所があった。2020 年 3 月に急遽 LMS の性能強化が必要となった際に、調達期間を短縮し柔軟にサーバの強化を実施できた。3 月以降もメモリ・CPU の変更、テスト環境の構築など短期間で実施できた。オンプレミスで運用している場合は、サーバの調達や設置・構築といった時間が必要となり、短期間で増強するのは困難であった。特に、COVID-19 の影響で製造・物流の能力が低下していた状態では、通常よりも長い時間を要したと想定される。LMS 内部の障害などが多くかったものの、IaaS 型のクラウドを採用していたことが幸いし、継続したデバッグと性能強化につながったと考える。今後もオンライン・オフラインを問わず LMS の活用機会が増えていく中で、各種要望に柔軟に対応していくにはパブリッククラウドでの運用が適していると考える。

最近では Service as a Service (SaaS) として Blackboard 社の Blackboard, Google 社の Google Classroom などでクラウド型の LMS サービスが提供されている。これらのサービスでは、LMS が各社のクラ

ウドサービス上で提供されるため、大学側でインフラの運用コスト、開発コストを下げることができる。特に Google Classroom は、G Suite for Education を導入している大学であれば大学ドメインの Google アカウントを持つ利用者（教職員や学生）に限定してクラスを提供できる。一方で、SaaS として提供されているため、提供されている機能以外の利用や拡張は難しく、それぞれの LMS に大学側の授業形態を合わせる必要がある。

4.3 オンライン授業に適した LMS の機能

週に 1 回教室で直接顔を合わせて、教員は授業内容を説明、学生は必要に応じて質問、というやり取りは対面授業なら当然簡単である。このような教員と学生とのやり取りに利用可能な、遠隔会議ツール、電子メール、チャットなどの各種のツールやサービスは多数存在するものの、オンライン授業の実施に当っては、当該授業で教員が用意したツールやサービスの情報—具体的には「オンライン授業 A を受講する人は <https://zoom.us/j/xxxxxxxx> にアクセスしてください」といった文字列など—といったオンライン授業に関する情報を、教員が記入して学生が参照可能な仕組みが必要であり、授業の開講期間中に頻繁に利用される LMS で提供することが望ましい。

このような情報の編集や参照を可能なインターフェースを授業毎に作成すること自体は然程難しくないと考えられるが、授業開始前に多数の学生が同時にアクセスしても耐えられるシステムとする技術や、そもそも情報の入手先を報せる方法等を考慮すると容易な問題ではなく、2020 年度のオンライン授業の開始時には実装できなかった。ようやく 2020 年度後期開始直前の 9 月中旬になって、ITC-LMS では「オンライン授業情報」という仕組みを新たに設けたが、当該情報を抽出して学生に提供する専用のウェブアプリケーションを LMS 本体とは別に提供している。

以上を踏まえると、科目に対する属性情報の追加や、追加した属性情報の外部から参照などを柔軟に実装できることが望まれる。

一方、ITC-LMS の既存の機能がオンライン授業に役立つこともある。本学のある部局では、オンライン授業のメリットを生かして LMS 上で学生が最大限に聴講可能（学務システムで登録しない授業でも、授業を聞き、教材等を見ることできる）取り組みを行った。元々 ITC-LMS では、コース内の履修者を教員が独自に登録したり、LMS のユーザが履修者として自由にコースに参加可能にするなど、コース内の履修者

の範囲を教員が柔軟に設定可能にしていましたことから、このような取り組みがシステム上は少なくとも容易に実現できたと考えられる。

オンライン授業になって各講義で出題される課題が増えたと言われる。ITC-LMS では出題された課題も含めて LMS 上で作成されたコンテンツに関する更新情報を、ウェブインターフェース上に表示する他、前日の更新情報を毎日まとめて LINE/メールで利用者に通知している。しかし、学期中は通知対象となる更新情報が格段に増えたことから、逐次送信する LINE/メールによる通知機能は、全ての対象となる利用者に送信が完了するまで、丸一日以上掛ることもあった。かくして学生は自分が履修中の講義に関して、LMS 上で出題されている課題を見落とすことの無いよう、頻繁にログインして個々のコース(講義)のコンテンツを確認する必要が生じている。これに対して学生から、履修中のコースで出題されている課題を一括して確認する機能が欲しいとの声が上がり、2020年9月現在該機能を実装する方向で検討が進んでいる。

5 これからの LMS のあり方

LMS の機能をさまざまな側面から眺め、LMS のあるべき姿について述べる。

アクセス制御の基盤： LMS にログインしたユーザは、参加しているコースとそのコース内での役割等に応じて、アクセスできるコンテンツと利用できる機能が変わってくる。たいしたことではないように聞こえるかもしれないが、これが LMS の基盤的機能であり、教材配布やレポート提出などの一般的な LMS の機能は、この基盤上にアプリとして実現できるものである。学務システムが有する公式の情報と担当教員等が付加した非公式の情報(e.g. グループワークの所属グループ情報)を統合して管理し、LMS の内外のサービスがこれら的情報を利用して一貫性のあるアクセス制御を行えるようにできることは望まれる。

コミュニケーションツール： 完全オンライン授業化が実施された今年度前半は、対面での教員と学生や学生同士のコミュニケーションが不可能となつたため、LMS のコミュニケーション機能が見直された。プッシュ型の情報提供については、学生が日常的に使っているチャットやメールに自然な形で情報を流すことが望まれる。

ポータル・ダッシュボード： 一方、フル型の情報提

供は、ワンストップサービスになっていることが望まれる。大学や教員からの連絡事項の確認、自分の過去の学習状況把握、ToDo リスト管理などを一箇所でシンプルに提供することが望まれる。

学習データの収集装置： 学生が自分の学習状況を把握したり、教員が担当クラスの状態を把握したりするためには、そのための基礎となるデータの収集・分析・可視化などが必要となる。細かな学習行動データの収集装置としての役割が望まれる。

サービスプロバイダ： LMS はさまざまなサービスを Web のインターフェースを介して提供する。学生の中にはスクリエイピングを行って自分用の独自アプリケーションを作成しているものもいる。こういう学生のためには、API を提供することが望まれる。そうすることで、より良いインターフェースの開発に拍車がかかることが期待できる。

6 おわりに

本稿では、東京大学の LMS における 2020 年夏学期のオンライン授業対応を通して実施した性能改善の過程、発生した障害とその原因と対応、またそれらの経験を通して得られた教訓を報告した。

本稿で報告した対応により夏学期を終えたが、ITC-LMS には多くの課題や改善が必要な箇所が判明した。秋学期までには、オンライン授業情報を明示的に登録し、学生が自分のオンライン授業情報(会議システムの URL など)を取得できるように機能を追加する。他にも課題の一覧機能やページ遷移の改善など様々な改善・機能追加を継続して実施する。

参考文献

- [1] 東京大学 オンライン授業・Web 会議 ポータルサイト, <https://utelecon.github.io/>.
- [2] クラウドゲートウェイ サーバホスティングサービス、東日本電信電話株式会社, <https://business.ntt-east.co.jp/service/serverhosting/>.
- [3] in Campus, キヤノン IT ソリューションズ株式会社, https://www.canon-its.co.jp/products/webapps_incampus/.
- [4] Kubernetes, <https://kubernetes.io/ja/>.