

システム管理を体験させる学生実験に関する考察

木山 瑞基^{1,a)} 河野 真治^{2,b)}

概要：近年の企業や大学では事業や授業、研究などでコンピュータ上のサービスを活用している。これらサービスとはクラウドとオンプレサーバーやその上で稼働するシステム、利用者が持ち込む PC やモバイル端末、それらを繋ぐ回線を含む。システム管理者の仕事は常にサービスを安定したバージョンに更新することや不具合の修正、利用者からの要望・問い合わせに答える事、また、これらシステムを利用者に告知し使用してもらう事、さらにはシステムの情報を保護する事である。しかし、これを固定化された少数のメンバーで管理することはリソース的に難しく、教育的な観点からも望ましくない。本学科には学生や教員に対して提供しているシステムがある。そのシステムを運用するため教員と有志の学生からなるチームを作り対応している。このチームには3年次の学生実験からも参加するメンバーがおりそれぞれシステム管理に関する知識に差がある。そこで、システム管理に参加する敷居を下げることがある。本稿ではチャットツールを用いたシステム管理の手法を提案する。また、学生実験での活動内容や取り組みに関する考察について述べる。

1. 学生とスタッフが参加するシステム管理

企業や大学では組織名以外の連絡にネットワークを使用したり、業務や授業、研究活動において情報システムを使用することからコンピュータ上のシステムが重要である。これらはオンプレのシステムとクラウド上のシステム、さらに、従業員や教員、学生が持ち込む PC やモバイル、それらをつなぐ有線無線あるいは公衆回線を含む。これらシステムを安定して運用し、外部の脅威から守るのがシステム管理の仕事になる。この協調方法は日々最新技術が登場しており、それを積極的に導入すると同時に、利用者及び管理者がその技術をうまく

使えるようになることが望ましい。

これを固定された少数の管理者が行なうことはリソース的に難しく、利用者の教育的観点からも望ましくない。IT 技術は使って、改良していくことにより全体の生産性が向上し、組織的な協調あるいは進化が可能になる。つまり、利用者も積極的にシステム管理に関わることが望ましいと思われる。特に情報工学を専攻する学生が自身で利用するサービスを運用することに意味がある。それが結果的に管理者の負担を下げ、管理業務の意義を高めることにつながる。

琉球大学工学部では約 300 名の学生や教員に対してネットワーク接続サービスや仮想サーバーの貸し出しサービス、学内チャットシステムなどのシステムを提供するために、教員と学生のチームを作り対応している。これはシステム管理チーム(シスマン)と呼ばれている。システム管理活動は 2022 年度から演習科目の一つのテーマとなっており、三年

¹ 琉球大学大学院理工学研究科工学専攻知能情報プログラム

² 琉球大学工学部工学科知能情報コース

a) oruta@cr.ie.u-ryukyu.ac.jp

b) kono@ie.u-ryukyu.ac.jp

次の学生が参加している。演習科目の一環として参加する学生には様々な背景があり個人によってシステム管理に関する知識の差がある。そこで、システム管理演習への参加を効率的に進めるために敷居を下げる必要がある。本稿ではチャットツールである mattermost を用いたシステム管理の手法を提案する。

2. 本学で提供するサービス

本コースでは学生 240 人、スタッフ 20 人で構成されている。建物は二つに分かれていて、5 階分を占有している。ネットワーク環境は大教室と学生演習室、個々の研究室と教官室に有線無線の LAN を提供している。有線 LAN は Ruby on Rails で作成された Akatsuki と呼ばれるシステムを利用して管理されている。コース内の IP 割り当て、DNS 申請、LDAP 情報管理を提供するサービスを用いて対象の機器に IP アドレスを割り振ることで利用できる。また無線 LAN は本学科の職員や生徒を対象とした LDAP 情報を用いた認証が必要なものやイベント時に作成されるゲストを対象にしたものが存在する。

物理サーバーは 2U サーバーを 6 台導入している。このうち 4 台の 2U サーバーは仮想環境を提供する KVM や GPU リソースを使用できる singularity などを提供している。2 台の 2U サーバーはディスクサーバーとして稼働しており、分散ファイルストレージの ceph で管理されている。物理サーバー以外にクラウドサーバーも一台契約している。これは計画停電や台風時などでもサービスを提供できるようにするだけでなく、システムの冗長化のために各サービスのセカンダリサーバーとしても動作している。使用しているサーバー、Switch を表 1 に示す。

これらサーバー上で動作するサービスには学内の授業連絡やアナウンス等で使用されるオンプレミス型のチャットツールの mattermost がある。以前まではオンラインチャットツールの Slack を使用していた。しかし、メッセージの上限やアカウント管理などの観点からオンプレミスで管理できる mattermost へ移行した。

表 1: 物理ネットワークスペック

用途	機器名	台数
基幹サーバー	PowerEdge R740	4
ディスクサーバー	PowerEdge R740xd	2
クラウドサーバー	Poweredge R720xd	1
1U サーバー	PowerEdge R630	3
メインスイッチ	WS-C3850	1
UTM	FortiGate200D	1
サーバースイッチ	WS-C2960X-48TD-L	1
ルームスイッチ	WS-C2960X-48TS-L	21
ルームスイッチ (PoE)	WS-C2960X-48LPS-L	2
ルームスイッチ (PoE)	WS-C2960X-24PS-L	4
無線 LAN コントローラ	AIR-AP3802I-Q-K9	1
無線 LAN アクセスポイント	AIR-AP3802I-Q-K9	21

バージョン管理ツールには gitlab を使用している。学生の演習開発の成果物やシステム管理活動で作成したスクリプト、システムの情報を管理している。

知能情報コースの学科 web ページも管理しており、受験生向けの情報や在学生に向けた時間割、論文ページの掲載なども行なっている。

基幹サーバーのリソースの貸し出しには KVM, singularity, podman のそれぞれを採用し、ジョブスケジューラーの Slurm によってリソースの管理を行なっている。podman は実験演習だけでなく期間サービスの運用にも使用されており、先述した mattermost や gitlab, 学科 web などが podman を使用しコンテナで運用されている。

利用者の管理には LDAP を使用している。弊学の総合情報処理センターで発行された LDAP 情報と同期しており、この情報を使用して wifi へのアクセスやサーバーのログインを管理している。提供しているサービスの一覧を 2 に学科のシステムの構成を図 1 に示す。

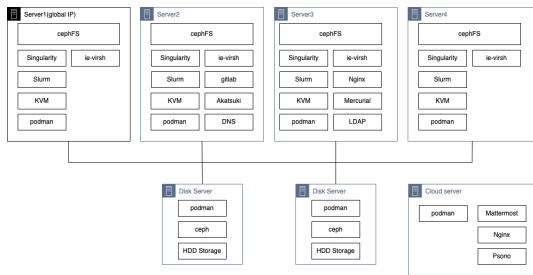


図 1: システム構成図.

表 2: 提供サービス

サービス名	概要
mattermost	学内チャットツール
gitlab	バージョン管理ツール
学科 web	受験生・在校生向けの情報や学科のイベントを掲載している
scrapbox	oss の共有ノートサービス. 学科ではドキュメントやログ管理に使用している
ie-virsh	ie 用の VM を管理するコマンド
Akatsuki	IP を管理する web アプリ
LDAP	職員, 学生のアカウント管理
DNS	学内外のドメイン管理
KVM	ハイパーバイザー型の仮想環境を管理する
podman	コンテナ型の仮想環境を管理する
singularity	コンテナ型の仮想環境を管理する
slurm	ジョブスケジューラ

3. システム管理チームの業務

シス管では週に一回のミーティングを行なっている. 基本的にオンラインミーティングツールの zoom を使用しての実施となっている. ここでは作業報告や緊急で対応する要件, 今後のスケジュールなどについて話し合っている. また, 停電時のサーバー停止と web サーバーの切り替えや年度末に行う KVM の使用状況を確認し削除など定期的に発生する業務も存在する.

上記で発生したタスクの管理には情報共有サービスの ScrapBox を使用している. タグを用いて情報を整理しており, 作業チケットや定期的に発生する作業のドキュメント, 緊急時の作業ログとして活

用している.

また, これらのタスクをこなす上で記述したコードは学科で管理している gitlab でバージョン管理しておりシステム管理チームに所属するユーザーのみがアクセスできるような権限になっている.

オープンソースのコントロールパネルやサービスへのログイン情報の管理には Psono を使用しており, こちらもシステム管理チームのメンバーのみがアクセス可能となっている.

ミーティング以外での利用者への告知, 問い合わせ, 質問, 会話などについてはオープンソースのセルフホスティング方式のチャットツールの mattermost を使用している. 緊急で対応する必要がある障害や依頼が来た際には mattermost 上からシステム管理チームに依頼が来る.

4. 学生実験

システム管理は 2022 年度から実践的なシステムの運用, 構築に関するスキルを習得することや継続してシステム管理に参加する生徒の確保を目的として演習科目のテーマの一つになった. 演習科目とは知能情報分野の専門的な知識を習得することを目的としており様々なテーマの中から生徒が一つ選択し履修する. 知能情報コースの三年次の学生を対象に開講されており, 1 学年 60 名ほどでテーマ数が 6 つほどなので 1 テーマ当たり 10 名ほどの受講生がいる. システム管理のテーマでは学生は基本 zoom を使用した遠隔授業を通じて活動を行う. 作業ログやドキュメントは普段システム管理チームが使用しているプロジェクトに招待されシステム管理における scrapbox の使い方の説明を受ける. サーバー室の見学や運用しているサービスのコントロールパネルの説明サーバーへの ssh config の設定や gitlab の syskan グループへの登録などを行う. また, システム管理のテーマの中からネットワーク, セキュリティ, web, サーバーのグループを作成しそれぞれで活動を行なっている.

以下に各グループでのシステム管理チームの実験演習での取り組みを示す.

4.1 サーバー班

サーバー班は基幹サーバー上の不具合の調査やその修正をおこなっている。取り組みのいくつかを以下に示す。

- バックアップの修正
- KVM テンプレートイメージファイルの修正

上記の二つの進め方として、代表者が一人サーバーにログインし、zoom で画面を共有する。その他の利用者は共有された画面を見て調査し、案を出したり scrapbox に作業のログを取った。

バックアップの修正

システムに障害が発生した際に復旧できるように rsnapsot で ceph で管理されているディレクトリや学科 web、生徒のホームディレクトリのバックアップをとっている。バックアップは daily、weekly をそれぞれ撮っていた。そのバックアップファイルの作成された日付を確認するとある期間から取られていなかった。まずは原因を調査するために実行ログ、エラーログの確認を行なった。するとバックアップファイルが大きすぎることから daily の処理が正常に走っていないことがわかった。そこで処理を weekly のみ変更することで解決した。

KVM テンプレートイメージファイルの修正

KVM を用いて計算機リソースの貸し出しを行う際に、四つある基幹サーバーで利用できる OS に差分があった。その為、OS のテンプレートファイルを ceph 上に置きそれぞれのサーバーで共有した上で ie-virsh から参照できるようにした。

4.2 セキュリティ班

セキュリティ班では学科システムへの脆弱性診断の構築を行なっている。システム更新前には脆弱性診断の環境があったが、現在の学科システムにはない。また、以降のしやすさや現在の基幹サービスがコンテナで配信されており、管理のしやすさからコンテナ環境での構築を行なう。基幹サーバーには一台グローバル IP を持つものがあつたり、生徒が利用する VM も申請した際にはグローバル IP を

付与できるため使用する脆弱性診断ツールはサーバーのスキャンが Vuls を選択した。

基幹サーバー上で作業するには admin 権限が必要だがシステム管理に慣れておらず操作ミスが考えられたことやコンテナで構築する為移行が安易なことから生徒それぞれに VM を構築させ基幹サーバーと同様の環境を作成した上で脆弱性診断の環境を構築する方針で進めた。生徒には学科システムと同じ ubuntu22.04、podman v4 系の環境を立てた。Vuls の構築には docker はあつたが podman はなかった為、podman command を使用できるようにスクリプトを書き換えた。実際に自身で構築した VM を Vuls コンテナから診断し結果を確認した。今後はこれらを基幹サーバーに向けて診断を行う為に設定を編集したり、生徒が利用する VM へ診断をする方法や頻度、診断結果の共有などを行う必要がある。

5. システム管理チームの問題点

弊学科のシステム管理チームは学生が主体となっている。その学生は毎年卒業しメンバーの入れ替わりがある。また管理するシステムは基本卒業した先輩が構築したものとなり、それらを scrapbox や mattermost にあるドキュメント、ログを確認しながら進めることとなる。これらの要因から以下の問題点がある。

システム管理チームに入ってくる人が少ないこと、実験演習をとっている人の定着率の悪さがある。システム管理活動に参加する方法として週に一回行なっているオンラインミーティングに参加するか管理者からシステムの管理権限を貰い作業するなどが挙げられるが複雑な操作が多い。2022 年から実験演習としてシステム管理が追加され、每期 10 数名ほど受講者がいるが現在定着しているメンバーはいない。その原因として現在のシステム管理チームは複雑な操作を必要とするため敷居が高いと考える。

次に、学科システムの gitlab は外部からの攻撃を受けたことがある。その際の原因としては gitlab の自動アップデートがメジャーアップデートに対応していなかったためである。サービスの脆弱性

は人力で調べるしかなかったため後手に回ったと考える。先行研究 [13] によって学科システムの脆弱性診断の自動化が構築されていた。しかし、2020年に更新された現行システムにはその手法は引き継がれていない。また、更新前は KVM を中心にシステムが動作していたが、現行の学科システム上のサービスはコンテナ上で動作している。その為、外部からの脅威に対して先行研究が示す脆弱性診断の自動化は有効であるが現行システムに対応した仕組みにしないといけない。

上記の脆弱性診断の自動化や先行研究 [11] [12] の mattermost のスラッシュコマンドを用いたシステム管理が存在していた。しかし、これらのシステムは現在のシステムでは動作していない。要因としてメンバーへの共有を行っていないことやドキュメンや使用方法を論文以外書いていないことが挙げられる。

6. メッセージングコマンドを使用した手法の提案

システムの管理をオープンソースのセルフホスティング式のチャットサービスである mattermost のスラッシュコマンドを通して実行する。スラッシュコマンドとは / から始まるコマンドを実行することで登録したサーバーに対しリクエストを送ることができる。また、スラッシュコマンドの後ろに空白くぎりで入力した文字列がリクエストボディとして認識される。また、スラッシュコマンドには空のメッセージボックスに / を入力すると、すべてのスラッシュ・コマンドのリストが表示される。特定の / コマンドを入力することでコマンドの引数やフラグの候補も表示される。本稿では学科で管理しているシステムの状態を確認できる機能を提供する http サーバーを立て、それに GET/POST リクエストを送信することで処理を行う。図 2 に構成図を示す。

学科システムは複数のサーバーから構成されており、それぞれで HTTP sever を立てるとサービス毎にリクエスト先を変更しないといけない為、統括して管理する HTTP server を一つ立てる。その HTTP server から他の基幹サーバー内で動作して

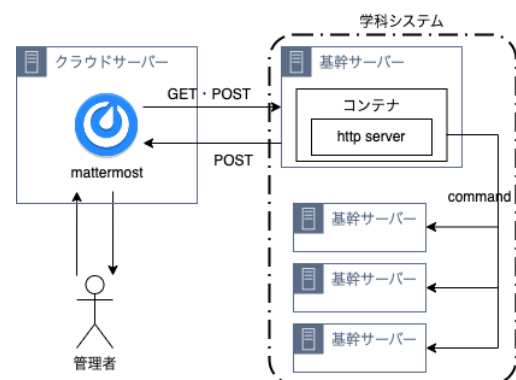


図 2: システム構成図。

いるサービスに対し処理を送り結果を mattermost から確認できる。以下に実装した操作やセキュリティ面での実装を示す。

7. プラグインの実装

mattermost はクラウドサーバー上で運用されているが今回実装した HTTP server は基幹サーバー内で運用している。mattermost から HTTP server にアクセスするためにグローバル IP を紐付け公開している。学科サーバーは在学中の生徒と教員のみがアクセス可能でありシステムの情報は秘匿すべきである。その為外部からのアクセスを制限し mattermost からのみ HTTP server へのアクセスを許可した。

本研究で実装したスラッシュコマンドは mattermost のシステム管理チームが管理するワークスペースのチャンネルから実行可能である。mattermost は学科に所属する全ての生徒がログインできシステム管理チームのワークスペースも制限をおこなっていない為全生徒がスラッシュコマンドの実行が可能である。しかし、学科のシステム利用者の情報は LDAP を用いて管理されており、サーバー上でシステム管理者として全ての操作を行えるのは syskan,sudo のユーザー属性が割り振られているアカウントとなっている。スラッシュコマンドでは一部 root ユーザーが実行する処理があることやシステム管理の操作に慣れていない学生には操作ミスやセキュリティの観点から制限をつけるのが適切だと判断した。その為、システム管理チームとそれ以外の利用者でスラッシュコマンド経由で実行できる操作に制限をかけた。mattermost はログインの際に gitlab 認証を使用しており、gitlab は LDAP 情報からログインを行う。その為 mattermost のアカウント名などは LDAP に登録されている情報から作成される。また、mattermost はスラッシュコマンドを実行する際にコマンド以外に実行したチャンネル情報や実行した操作者の情報をデータをして送信する。データの中にユーザー名があり HTTP server でユーザー名で LDAP 認証を行い sudo,syskan のユーザー属性を持つかを確認することによって実行できる操作に制限をかけ

ることができる。

8. 実例:filesystem の圧迫

学科システムでは生徒が授業や研究で利用する為の仮想環境を提供している。利用可能な仮想環境は VM,container が挙げられる。貸出をする VM のデフォルトのスペックは CPU1 コア、メモリ 1GB、ストレージ 10GB である。デフォルトのスペックでは不足する場合、要望に応じてスペックの変更を行っている。またコンテナも作成できるイメージの容量に制限は無く生徒が自由に基幹サーバーのリソースを使用できる。学生は自身で作成した仮想環境の削除は行える。しかし、毎年授業の一環で使用した VM や研究の実験環境として作成されたコンテナが作成されたまま放置されている。これによりストレージの容量不足が要因となる障害が発生する可能性がある。

これまで操作方法ではストレージの圧迫は定期的なディスク容量の確認は分散オブジェクトファイルシステムである ceph の状態確認によって調査してきた。

提案手法を用いることでサーバーのストレージの調査に有効な機能を提供できると考える。想定するコマンドを下記に示す。

サーバーの容量確認では df を基準のコマンドとする。よく確認するルート直下の容量はディレクトリの指定無しで表示できるようにする。

- /df サーバー名
対象のサーバーのルート直下のディスク容量が表示される。
- /df サーバー名 ディレクトリ
対象のサーバーの指定したディレクトリ配下の容量が表示される。

ceph の状態確認では ceph を基準のコマンドとする。クライアントのステータス表示によって異常がある場合はエラーメッセージの確認が可能である。

- /ceph health
ceph storage cluster の簡略的なステータスが表示される。
- /ceph health detail

ceph storage cluster のステータスが表示される。

- `/ceph status`
ceph 全体のステータスが表示される。

podman や VM などの仮想環境の情報を表示することでディスクの容量不足によるトラブルが発生する前に対応ができると考える。podman によって作成された仮想環境は podman を基準のコマンドに、VM によって作成された仮想環境は virsh を基準のコマンドとする。

- `/podman ps`
起動済みの podman コンテナを表示する。
- `/podman ps all`
停止しているものも含めて podman コンテナを表示する。
- `/podman logs コンテナ`
指定したコンテナのログを表示する。
- `/podman inspect コンテナ`
指定したコンテナの詳細を表示する。
- `/virsh list`
VM のリストを表示する。

9. 今後の課題

本稿で提案したメッセージングを使ったシステムの管理手法についての課題を挙げていく。

9.1 スラッシュコマンドの実装

本稿では提案手法によって管理者の負担を軽減し実現可能なコマンドを設計した。しかしまだ実装を行っていない為、これから実装を進めていく必要がある。また、実装と運用を通して必要な機能の追加や不要な機能の削除などを行う。

9.2 実行した処理の説明

現在運用している学科システムはさまざまなサービスが存在しており複雑である。その為管理に使用するコマンドも内部でどのような操作が行われているかを理解する必要がある。スラッシュコマンドで実装する処理について実行結果を表示する以外にどのような処理が行われているかを説明するオプションを追加する。

9.3 演習科目

システム管理を演習科目のテーマとして取り組む中で実際に手を動かし管理作業に参加する生徒が偏る傾向にあたり、生徒自身が作業している内容について理解していない様子があった。実際に稼働している学科システムの不具合や依頼などを作業内容としている為、構成が複雑であり使用するコマンドも多彩で授業内でこういった処理を行っているかを詳細に説明できないことが挙げられる。

10. まとめ

本稿では実験演習でのシステム管理チームの取り組みやシステムの保守・運用を円滑に行う為に学科システムにメッセージングを用いたシステム管理システムを構築した。多くの学生が学科システムの管理活動に参加することは教育的な観点から有効と考え演習科目のテーマでシステム管理を行なった。しかし、この活動では受講した学生がシステム管理活動に定着せず改善が必要となった。メッセージングを用いたシステム管理はサービスの多様化やシステムの高度化から初学者でも操作しやすく操作が統一されている為有効と考えた。今後、本システムを実際に運用する中で問題点を改善していく。

参考文献

- [1] Brazil, B.: 入門 Prometheus-インフラとアプリケーションのパフォーマンスモニタリング, O'Reilly Japan (2019).
- [2] Julian, M.: 入門監視-モダンなモニタリングのためのデザインパターン, O'Reilly Japan (2019).
- [3] Labs, G.: Grafana Loki.
- [4] Labs, G.: Grafana: The open observability platform | Grafana Labs.
- [5] Labs, G.: Promtail | Grafana Labs.
- [6] Prometheus.io: Alertmanager | Prometheus.
- [7] Prometheus.io: Prometheus - Monitoring system & time series database.
- [8] Mattermost Developers:
<https://developers.mattermost.com/>
- [9] GitLab Documentation:
<https://docs.gitlab.com/>
- [10] 木山 瑞基:ログ収集・管理をメッセージング経由で適切に設定する手法の提案, 琉球大学, 情報

処理学会システムソフトウェアとオペレーティング・システム研究会 (OS), 2022

- [11] 秋田海人:情報系学科における教育研究情報システムの運用改善と新規システムの構築に関する取り組み, 琉球大学, 修士論文,2020
- [12] 棚原理貴:mattermost 上でのスラッシュコマンドによる運用自動化の提案, 琉球大学, 学士論文,2020
- [13] 玉城裕太:情報系学科における教育研究情報システムの運用とセキュリティに改善に関する取り組み, 琉球大学, 修士論文,2019