

# 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<sup>1,a)</sup> 河野 真治<sup>2,b)</sup>

**概要:** 琉球大学工学部では三年次の実験のシステム管理を体験するサーバー班を用意している。教員、技術職員から構成されたシステム管理チームに学生実験のメンバーが参加する。使用するツールとしては zoom, chat tool である mattermost、そして TODO 管理や知識データベースに scrapbox を用いている。今までの 2 年間の経験について考察する。学生の体験用に mattermost 上の chat システムとして管理コマンドを用いる方法を提案する。

## 1. 学生が参加するシステム管理

企業や大学では業務や授業あるいは研究などにコンピュータ上のサービスが重要である。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。これら全体を協調させ、故障や外部から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。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学生や教員に対してシステムサービス提供するために、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(シス管)と呼ばれている。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いる。そこで、システム管理に参加する時の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. 技術概要

### 2.1 Prometheus

Prometheus は [7] オープンソースのメトリクススペースのモニタリングシステムである。音声ファイル共有サービスを配信している SoundCloud 社によって 2012 年に開発されており、2015 年に一般に公開されている。対象サービスから監視サーバーに対し情報を取得する pull 型を採用しており、取得したデータは時系列データベースに保存される。特徴としてそれぞれのデータに付与されているラベルごとに情報をまとめることが出来る。また、PromQL という独自のクエリ言語を扱う事でアラート管理コンポーネントである Alertmanager にクエリを発行することができる。内蔵する式ブラウザからグラフ・ダッシュボードの作成やデータ検索ができるが作成したグラフは保存することができないことや汎用のダッシュボードシステムでは無い事から一般的には可視化ツールと組み合わせて運用される。

### 2.2 PromQL

Prometheus の時系列データを扱うことに特化したクエリ言語である。exporter によって付与されるラベルを用いることで柔軟な集計が可能である。グラフや Prometheus の式ブラウザを表示するのに使用したり、HTTP API を介して外部システムで利用することが出来る。

### 2.3 exporter

監視対象のデータを収集し Prometheus からのリクエストに応じて必要なデータを整形し Prometheus にレスポンスとして返すツールである。Prometheus に送信するデータに対しラベルを付与することができ情報の絞り込みが楽になる。サーバーの情報を収集する node\_exporter や通信

<sup>1</sup> 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

<sup>2</sup> 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

のエンドポイントを監視する blackbox\_exporter など公式が提供しているもの以外に多くのサードパーティがサービスの exporter を提供しており、自身で独自の exporter を作成することも可能である。

## 2.4 Alertmanager

Alertmanager [6] は Prometheus のコンポーネントであり、オープンソースソフトウェアとして公開されているアラート管理ツールである。アラートの重複排除、グループ化などによりアラートの送信を行うことができる。

## 2.5 Loki

Loki [3] は Prometheus に触発されたオープンソースのログ収集ツールである。特徴として Prometheus のようにログデータをラベル毎にまとめることができる。また、Prometheus と同様に LogQL という独自のクエリ言語を扱う事でアラート管理コンポーネントである Alertmanager にクエリを発行することができる。

## 2.6 Promtail

ログを収集して Loki に送信するツール。Prometheus の exporter のようにアプリケーションのログに対しラベルを付与することができる。

## 2.7 Grafana

収集されたデータ・ログをダッシュボードを用いてブラウザから可視化可能なツールである。自身でデータを収集をせずデータの可視化を行うためデータソースと組み合わせるの一般的なものである。データソースとして Prometheus と loki を対応しており、LogQL を用いてカスタムしたログ情報を表示することが可能である。

## 2.8 Mattermost

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。類似プロダクトに Slack があり、差別点としてはサーバーから自身で構築するためチャットに保存期間の上限が無い。また、組織や企業などの開発者向けに作成されており、様々なツールとの結合が可能である。

## 2.9 コンテナ型

仮想化技術の一つであり、他の仮想技術との相違点はカーネルはホスト OS と共用で利用する点である。これにより他の仮想技術よりリソースが節約でき、仮想環境の構築、削除が高速でできる。

## 2.10 Docker

Docker.Inc. が開発したオープンソースのコンテナ管理

ツールである。コマンドや専用の Dockerfile を用いて環境を構築することができ、また作成したイメージを登録することが出来る Docker Hub を用いることで環境を配布することもできる。

## 2.11 Podman

RedHat 社が開発した docker 互換のコンテナ管理ツールである。RHEL(Red Hat Enterprise Linux) の version7.6 以降からは docker のサポートが切れる事からコンテナ管理ツールとして採用された。また Golang で podman を操作できる関数群を提供しており、イメージやコンテナ、ネットワークなどの操作に使用できる

## 2.12 ハイパーバイザー型

仮想化技術の一つであり、ハードウェア上にハイパーバイザと呼ばれる仮想化ソフトウェアを動作させ、その上でゲスト OS を運用する。ホスト OS を不要とするがコンテナ型と比べて起動速度は低速となる。

## 2.13 KVM

KVM は (Kernel-based Virtual Machine) の略で linux カーネル上で動作する仮想化技術であり、カーネルをハイパーバイザとして機能させる。

## 3. 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。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 Akatsuki と呼ばれるコース内の 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 や知能情報コースの web サイト、gitlab や Akatsuki を稼働している podman 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。使用している 4 台の 2U サーバーについては表 1 でディスクサーバーは表 2 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連絡や授業等で使用されるチャットツールの mattermost やバージョン管理ツールの gitlab、実験演習や研究に使われる仮想環境の KVM、singularity、podman などがある。

表 1: 物理サーバスペック

CPU	Intel Xeon Gold 6238 (2.10GHz/22Core) x 2
GPU	Nvidia Tesla V100S
RAM	512GB
NVMe SSD	1.5TB
SAS SSD	5TB

表 2: ディスクサーバスペック

CPU	Intel Xeon Silver 4208
RAM	32GB
SAS HDD	300GB/15000rpm x 2
NLSAS HDD	4TB/7200rpm x 12

## 4. シス管の業務

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などは情報共有サービスの Scrap-Box 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 2022 年前期より実験演習にシステム管理が加わったことで受講している三年次の学生も管理に加わっている。

## 5. 学生実験

## 6. シス管の問題点

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、毎期 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 [10] 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 [8] [9] 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トや使用方法を論文以外書いていないことが挙げられる。

## 7. 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 1 に構成図を示す。

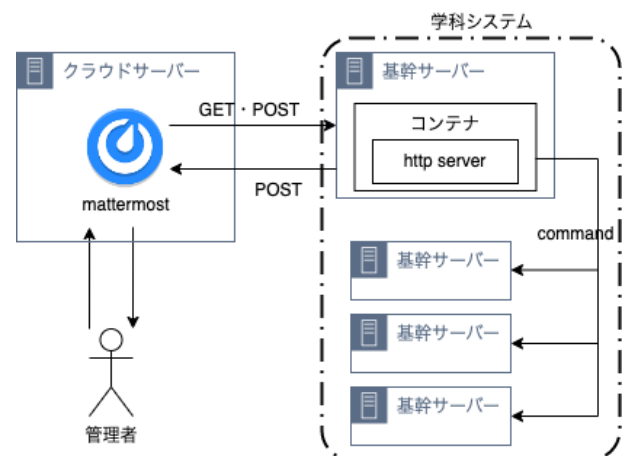


図 1: システム構成図。

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。

## 8. 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。

### 8.1 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。

### 8.2 セキュリティの問題

mattermost から呼び出す処理には一定の権限を持っていないと動作しない処理が含まれる。また、mattermost は学外ネットワークに接続されている為公開する httpserver は外部からアクセスできる必要がある。そうなった際に悪意のある第三者によって server に攻撃が仕掛けられることで任意のコマンドを実行される可能性がある。このようにセキュリティ上の問題が考えられる為、実装するにはセキュアなシステムにする必要がある。

## 9. まとめ

本稿ではシステムの保守・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。mattermost に参加している学生が処理できる想定なので削除の操作や変更の操作は検討しなかった。システム管理チームは教員と学生が中心となっている。その為、学生は学部一年から所属しても4年、長くても6年で卒業する。また、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メッセージングを用いたシステム管理は有効と考えた。今後、本システムを本番環境にデプロイし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] 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
- [9] 秋田海人: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
- [10] 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