卒業課題のレポートの未完了である章の数にもとづくアラートチケットの並び替えによる調査開始までの時間の短縮

月森 陽太[1]　平尾 真斗[2]　串田 高幸[1]

概要：Cloud and Distributed Systems Laboratory（以下，CDSL とする）で稼働する Prometheus は，監視対象から取得したメトリクスが監視ルールを満たすとアラートを作成する．Alertmanager が通知したアラートは Redmine にチケットとして作成され，担当者は調査を行う．課題は，作成日時が早いチケットから調査を開始すると，学生の「卒業課題 II」の進捗が考慮されず，進捗が遅い学生が使用したサーバの調査が遅れる．提案では，学生の「卒業課題 II」をサポートする大学院生が記録した，テクニカルレポートの未完了の章の数を学生ごとに集計する．その後，各学生が使用したサーバで検出されたアラートに対応するチケットを，テクニカルレポートの未完了である章の数が多い学生から順に並び替え，先頭のチケットを担当者に通知する．1 人の学生が使用したサーバで検出されたアラートに対応するチケットが複数ある場合は，作成日時が古い順に並べる．評価実験では，2025 年 9 月 1 日から 12 月 1 日までに作成された未着手のチケット 26 件を対象に，提案適用後のチケットの順位と比較対象との順位相関係数を比較した．1 つ目の比較対象は，テクニカルレポートの初版が未提出の学生が使用したサーバで検出されたアラートに対応するチケット 16 件を作成日時が早い順に並べ，その後に残りの 10 件を作成日時が早い順に並べた順位である．2 つ目の比較対象は，CDSL の「Cadence」「論文輪講会」「勉強会」の欠席数の合計が多い学生から順に，その学生が使用したサーバで検出されたアラートに対応するチケットを，作成日時が早い順に並べた順位である．順位相関係数は，提案適用後の順位と 1 つ目の比較対象から約 0.85，提案適用後の順位と 2 つ目の比較対象から約 0.99 であった．提案によるチケットの並び替えは，テクニカルレポートの初版が未提出の学生が使用したサーバで検出されたアラートに対応するチケットよりも，欠席数が多い学生のサーバで検出されたアラートに対応するチケットを優先している．また提案適用前と適用後のチケット作成から調査開始までの時間を計測した．提案適用前の 2025 年 9 月 1 日から 12 月 1 日に調査が開始されたチケット 23 件の調査開始までの時間は最短で約 1 分，最長で約 4 日 18 時間 27 分であった．提案適用後の 2025 年 12 月 11 日から 12 月 12 日に調査が開始されたチケット 2 件は，テクニカルレポートの第 2 版が未提出の学生が使用したサーバで検出されたアラートに対応するチケットであり，調査開始までの時間は約 8 分と約 38 分であった．チケット作成から調査開始までの最短の時間は，提案適用前で約 1 分，適用後は約 8 分であり，提案手法なしと比較して約 7 分増加した．また最長の時間は，提案適用前で約 4 日 18 時間 27 分，適用後は約 38 分であり，提案手法なしと比較して約 4 日 17 時間 49 分減少した．

1. はじめに

背景

クラウドコンピューティングの時代において，テクノロジー企業が提供するサービス上で発生するインシデントは，顧客満足度とビジネス収益に対する潜在的な脅威である．インシデントは，サービスの標準的な運用に含まれない，提供されるサービスの品質やレベルの低下を引き起こす，または引き起こす可能性のある事象である [1]．サービスプロバイダーは，ユーザの不満や経済的損失を回避するために，発生したインシデントをチケットとして管理して迅速なサポートを提供する [2]．

チケットは，システム障害や誤動作が発生したサービスやインスタンス名，またテクニカルサポートスタッフによって行われた操作の履歴を記録するものである．システムの応答速度の低下やソフトウェアのバグ，システムの設定ミスを改善し，信頼性を向上させるために広く利用されている [3, 4]．

監視システムは，監視対象のシステムの状態が事前に定

[1] 東京工科大学コンピュータサイエンス学部
クラウド・分散システム研究室
〒 192–0982 東京都八王子市片倉町 1404-1
[2] 東京工科大学大学院バイオ・情報メディア研究科
コンピュータサイエンス専攻
クラウド・分散システム研究室
〒 192–0982 東京都八王子市片倉町 1404-1

義した条件を満たした場合にアラートを作成し，E メールやチャットツールでオンコールエンジニアに通知する [5]．作成されたアラートからチケットが作成されると，オンコールエンジニアは文書化された手順にもとづき障害の調査を行う．さらなる調査が必要な場合に，より高度なスキルと知識を持つ 2 次または 3 次の担当者にエスカレーションされる [6]．

オンコール対応は，大規模なサービスの運用を行うエンジニアリングチームにとって，サービスの信頼性と可用性を維持するために必要である [7]．

インフラストラクチャやアプリケーションの監視のツールの 1 つに Prometheus がある．Prometheus は，監視対象から CPU 使用率やメモリ使用量，ディスク使用量，ネットワーク統計のメトリクスを時系列の形式で収集し，メトリック名のキーと値のペアを保存することで，多次元データモデルを提供する [8,9]．また，Prometheus から送信されるアラートの重複排除やグループ化，プロジェクト管理ツールへの通知を行うために，Alertmanager がある [10]．

タスク管理やプロジェクト管理のためのソフトウェアのうち，オープンソースソフトウェアである Redmine がある [11]．Redmine では，タスクをチケットとして管理することで，タスクの進捗状況の把握や，過去の対応履歴の確認と検索を行うことができる [12]．

東京工科大学コンピュータサイエンス学部の Cloud and Distributed Systems Laboratory（以後，CDSL とする）では，障害が発生したサーバやアプリケーションに対して，あらかじめ決められた担当者が調査と対応を行う体制が構築されている．障害対応を円滑に行うために，サーバやアプリケーションから取得されるメトリクスの監視や，ログの収集と検索，アラートの通知，およびチケットの管理を行うシステムを運用している．これらのシステムは，K3s による Kubernetes クラスタ上で稼働している．Kubernetes は，複数のホストにまたがるコンテナ化されたアプリケーションのデプロイ，スケーリング，管理を自動化するためのシステムである [13]．K3s は，Kubernetes の公式ディストリビューションであり，実行に必要なメモリやバイナリサイズが小さいことが特徴である[*1]．CDSL で運用しているシステムの構成を図 1 に示す．

図中の Mint は，CDSL で運用している 10 台のサーバのうちの 1 台の名称である．10 台のサーバには，ハイパーバイザーとして Broadcom 社の VMware 製品である VMware ESXi が導入されている．10 台のサーバのうち Mint をはじめとした 7 台は，CDSL に所属している学生がアプリケーションを稼働させたり実験や作業を行う目的で，Ubuntu がインストールされた仮想マシンを作成し，使用することができる．Logging クラスタ上では，サーバやアプリケーションから収集したログの構造化を行うアプリケーションや，検索や分析機能を提供するアプリケーションが Pod として稼働している．Pod とは，1 つまたは複数のコンテナのグループであり，Kubernetes の最小かつ最も基本的なデプロイ可能なオブジェクトである．Mint をはじめとした 10 台のサーバのハイパーバイザーである VMware ESXi のシステムログ (/var/log/syslog.log) は，Logging クラスタ上の Logstash に送信される．Monitoring クラスタ上で稼働している Filebeat は，Monitoring クラスタ上の Pod のログファイルを Logging クラスタ上の Logstash に送信する．Logstash は，取得した非構造化データを構造化して Elasticsearch に送信する [14]．Elasticsearch は，送信されたデータを保存し，検索や分析機能を提供する [15]．Kibana は，Elasticsearch 上に保存されたデータの分析や検索を行うダッシュボード機能をブラウザ上で提供する [16]．Monitoring クラスタ上では，システムの監視やアラートの通知，およびチケットの管理を行うアプリケーションが Pod として稼働している．Prometheus は Blackbox Exporter や，VMware Exporter，cAdvisor からメトリクスを取得する．Blackbox Exporter は，VMware ESXi や仮想マシン上の OS（Operating System）やアプリケーションに対して HTTP（Hypertext Transfer Protocol）や ICMP（Internet Control Message Protocol）を経由したエンドポイントの外形監視を行い，その結果をメトリクスとして Prometheus

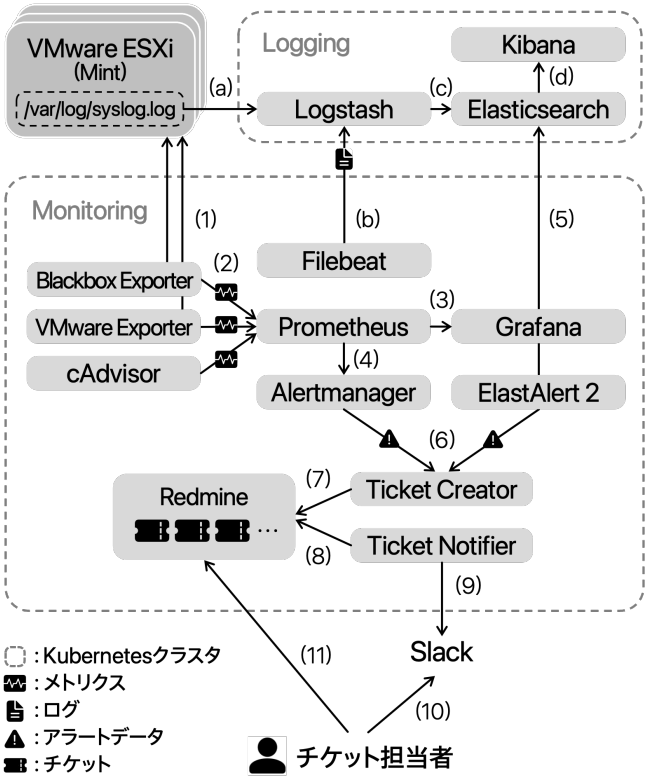

図 1: CDSL で運用しているシステムの構成

[*1] https://github.com/k3s-io/k3s

に送信する[*2]．VMware Exporter は，VMware ESXi のホストや仮想マシンのメトリクスを取得し，Prometheus に送信する[*3]．cAdvisor は，実行中のコンテナのメトリクスを取得し，Prometheus に送信する [17]．Grafana は，Prometheus 上の時系列データベースにクエリを実行したり，ダッシュボードにグラフを表示しデータを可視化したりする機能を提供する [18]．Prometheus は，取得したメトリクスがあらかじめ指定したルールを満たしたときにアラートを作成し，Alertmanager に送信する．また ElastAlert 2 は，Elasticsearch に収集されたログがあらかじめ指定したルールを満たしたときにアラートを作成する[*4]．Alertmanager と ElastAlert 2 はアラートを Ticket Creator に送信し，Ticket Creator は Redmine 上にアラートの内容を記載したチケットを作成する．Ticket Notifier は，Redmine に作成されたチケットを Slack でチケット担当者に通知する．チケット担当者は Redmine にアクセスし，チケットに記載されたアラートの調査と対応を行う．チケット担当者は CDSL に所属する有田 海斗，岡田 京太郎，坂井 萌桜，佐藤 健斗，月森 陽太，手塚 雄星，ムハンマド アクラム，山﨑 拓海の計 8 名の学生であり，チケットに記載されたアラートの調査と対応を行う．月曜日から金曜日の 9 時 00 分から 10 時 29 分，10 時 30 分から 12 時 29 分，12 時 30 分から 14 時 59 分，15 時 00 分から 17 時 29 分で，それぞれチケット担当者が決められており，作成されたチケットはチケット担当者に割り振られる．チケット担当者が複数のチケットを保有している場合は，作成日時が古いチケットから調査を行う．また調査と対応は，最大で 3 次までエスカレーションされる．

東京工科大学コンピュータサイエンス学部では「卒業課題 II」の科目があり，CDSL では「Cadence」が週 1 回行われている．「Cadence」では，「卒業課題 II」の進捗状況や，CDSL で提出が求められる「卒業課題 II」の成果を記述した「テクニカルレポート」の執筆状況について，CDSL の他の学生に説明を行い，フィードバックを受ける．学生は「Cadence」の開始時刻までに，スライドとテクニカルレポートを作成する．これらはそれぞれ章立てと記載内容が指定された統一のフォーマットがあり，学生はそのフォーマットを準拠する．「Cadence」の開始前には，スライドとテクニカルレポートを Google ドライブ上の指定のファイルにアップロードする．また，学生が取り組む「卒業課題 II」をサポートする大学院生（以後，学生のメンターとする）がテクニカルレポートの複数の項目について「追加更新が必要か」を判定し，スプレッドシートに記録する．

[*2] https://github.com/prometheus/blackbox_exporter
[*3] https://github.com/pryorda/vmware_exporter
[*4] https://github.com/jertel/elastalert2

課題

課題は，1 人の担当者が複数のチケットを担当している場合は，作成日時が古いチケットから調査を行っているが，このとき「卒業課題 II」の進捗が遅れている学生が使用したサーバの調査に時間がかかることがある．図 2 に，課題の状況を示す．

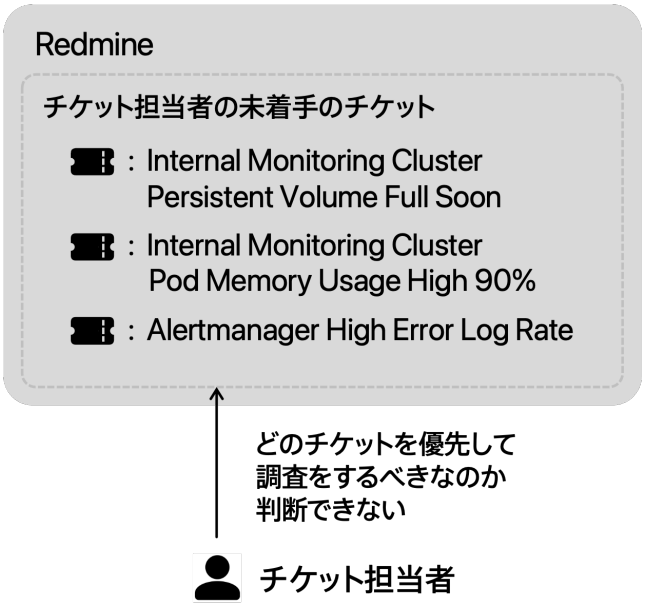

図 2: 課題の状況

チケット担当者は，3 件のチケットを担当している．チケットの内訳は「Internal Monitoring Cluster Persistent Volume Full Soon」と「Internal Monitoring Cluster Pod Memory Usage High 90%」，「Alertmanager High Error Log Rate」である．「Internal Monitoring Cluster Persistent Volume Full Soon」は，Monitoring Cluster にある Prometheus の Pod が使用する Persistent Volume の容量が 100%に近づいていることを示すアラートのチケットである．「Internal Monitoring Cluster Pod Memory Usage High 90%」は Monitoring Cluster にある Redmine の Pod のメモリ使用率が 90%以上であることを示すアラートのチケットである．「Alertmanager High Error Log Rate」は，Alertmanager から出力された「Error」の文字列を含むログが 1 分間に 1 件以上出力されていることを示すアラートのチケットである．これらのチケットは優先度が決められていないので，どのチケットを優先して調査をするべきか分からない状況を示している．

各章の概要

第 2 章の関連研究では，関連する既存研究について述べる．第 3 章の提案では，課題を解決する提案とユースケース・シナリオについて述べる．第 4 章の実装では，提案をもとに作成したソフトウェアの実装方法について述べる．

第 5 章の評価実験では，実験環境について述べる．第 6 章の議論では，提案の議論を述べる．第 7 章では，全体のまとめを述べる．

2. 関連研究

膨大な数のチケットに記録されたテキストにもとづいてチケットをクラスタリングし，それらにキーワードを生成し付与することで，新しく作成されたチケットに対する最適なアクションガイドラインを迅速に適用させることを目的とした研究がある [19]．一方で，チケット作成時のリアルタイムのメトリクスや障害の状況を考慮していないことに関して改善の余地がある．

機械学習と自然言語処理技術をもちいて，人間が介入することなくチケットを迅速にクラスタリングおよび分類する手法を提案し，チケットの手動分類にかかる時間や解決プロセスの非効率性の課題を解消することを目的とした研究がある [20]．この研究では，クラスタリングおよび分類の精度に対する評価は示されているが，インシデント対応にかかる時間の短縮について言及されていないことに改善の余地がある．

過去のチケットの内容と処理履歴からチケットが解決に至るルートを学習する生成モデルを構築し，新しく作成されたチケットを最終的な解決グループへルーティングすることで，チケットが複数の部門を行き来する回数を減らし，解決までの時間短縮を目的とした研究がある [21]．この研究では，どのチケットから優先して処理すべきかについて言及されていないことに改善の余地がある．

3. 提案

提案では，CDSL の学生が執筆したテクニカルレポートの未完了である章の数を「卒業課題 II」の進捗として，チケットの順位を決定する．

提案方式

学生のメンターが記録した各学生のテクニカルレポートの各章が「完了」または「未完了」であるかの判定の数を学生ごとに集計する．その後，学生が使用したサーバを特定し，そのサーバで検出されたアラートに対応するチケットを，テクニカルレポートの未完了である章の数が多い学生から順に並び替え，先頭のチケットを担当者に通知する．学生が使用したサーバで検出されたアラートに対応するチケットが複数件ある場合は，作成日時が早いチケットから並べる．理由は，古い障害が放置されると，それが原因で新しい障害が発生することがあるためである．例えば，ディスク容量不足のアラートが放置されると，その後にアプリケーションが起動できない障害が発生する．図 3 に，提案方式の概要を示す．

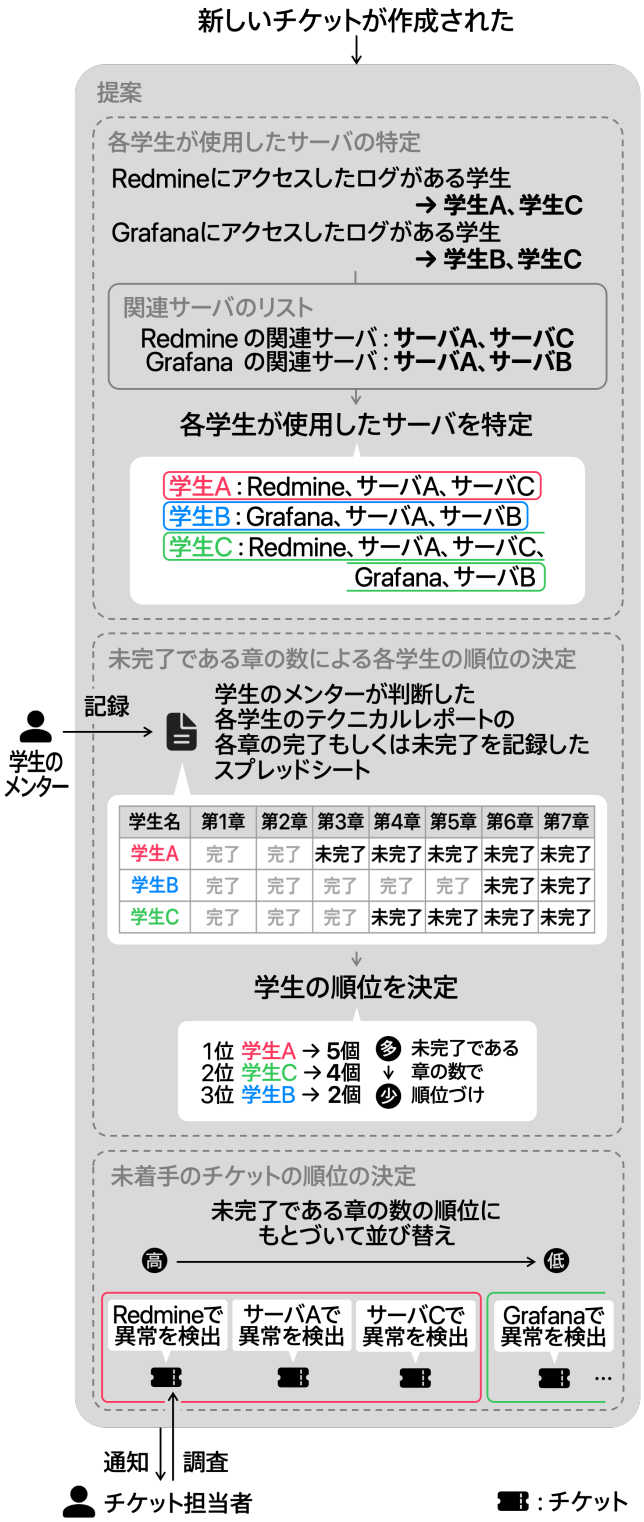

図 3: 提案方式の概要

新しいチケットが追加されたことを検出すると，各学生のテクニカルレポートにおける学生のメンターが判定した未完了の章と，各学生が使用したサーバを取得する．

テクニカルレポートにおいて未完了である章の数の合計を学生ごとに集計し順位をつける．各学生のテクニカルレポートの未完了である章の総数を降順に並び替えた順位に

おいて，未完了の章の総数が 5 個の学生 A は 1 位，4 個の学生 C は 2 位，2 個の学生 B は 3 位である．チケット担当者の未着手のチケットに記載されたアラートについて，テクニカルレポートの未完了である章の数の順位にもとづいてチケットを降順に並び替え，先頭のチケットをチケット担当者に通知する．図の場合，未完了である章の数の順位が 1 位の学生 A が使用した Redmine，サーバ A，サーバ C で異常を検出したことを示すアラートのチケットについて，作成日が古い順に並べたセットを，チケット順位の先頭に配置する．その後，2 位の学生 C が使用したサーバで異常を検出したことを示すアラートのチケットを，作成日時が古い順に並べたセットを，先に配置したセットの後ろに配置する．同様に，3 位の学生 B が使用したサーバで異常を検出したことを示すアラートのチケットを，作成日時が古い順に並べたセットを，先に配置したセットの後ろに配置する．未完了である章の数の合計が同じ学生がいる場合は，その学生が使用したサーバで検出されたアラートに対応するチケットを，作成日時が古い順に並べたセットを，先に配置したセットの後ろに配置する．先頭のチケットをチケット担当者に通知し，チケット担当者は，通知されたチケットに記載されたアラートの調査を開始する．

仮想マシン上に Ubuntu をインストールしたサーバを使用した学生を特定する方法

仮想マシン上に Ubuntu をインストールしたサーバを使用したと判定する基準を以下に述べる．

- チケットが作成された日時に VMware ESXi 上に登録されている仮想マシンであること．
- 仮想マシンに登録されている名前が「{ 学籍番号 }-{ ホスト名 }」の形式である仮想マシンを対象とする．
- 仮想マシンに登録されている名前が「{ 学籍番号 }-{ ホスト名 }」の形式ではない仮想マシンは，2025 年 11 月 2 日時点で 79 台中 6 台あった．対象外とする 6 台の仮想マシンを以下に示す．
  - Lily 上の「blackbox-m1」
  - Lily 上の「esxi-backup」
  - Iris 上の「koyama-elasticsearch」
  - Iris 上の「kushida-nisl1」
  - Iris 上の「kushida-nisl2」
  - Iris 上の「kushida-nisl3」
- 仮想マシンに記載された学籍番号と対応する学生の名前から，その学生が仮想マシンを使用したと判定する．

この方法は，学籍番号と学生の名前の対応付けを記録した CSV ファイルがあることを前提とする．CDSL では，学生が作成した仮想マシンの名前の先頭に学生の学籍番号をつけるルールがある．監視対象の各 VMware ESXi の管理コンソールの UI に表示される VMware ESXi に登録されている仮想マシンの名前から，学籍番号を取得する．学籍番号を取得する方法は，仮想マシンの名前が「{ 学籍番号 }-{ ホスト名 }」の形式であるため，最初の「-」以前に入力されている文字列を取得する．仮想マシンの名前の先頭にある学籍番号と，学籍番号と学生の名前の対応付けが記録された CSV ファイルから，仮想マシンを作成した学生を特定することができる．チケットが作成された日時において，監視対象の VMware ESXi をインストールした物理サーバごとに，仮想マシン上に Ubuntu をインストールしたサーバを作成した学生を特定し，その学生を VMware ESXi をインストールした物理サーバを使用した学生とする．

VMware ESXi をインストールした物理サーバを使用した学生を特定する方法

VMware ESXi をインストールした物理サーバを使用したと判断する基準を以下に述べる．

- VMware ESXi をインストールした物理サーバであること．
- 仮想マシン上に Ubuntu をインストールしたサーバを使用した学生を特定していること．
- VMware ESXi 上にあるすべての仮想マシンについて，それらの仮想マシン上に Ubuntu をインストールしたサーバを使用したと判定した学生を，VMware ESXi をインストールした物理サーバを使用したと判定する．

この方法は，仮想マシン上に Ubuntu をインストールしたサーバを使用した学生を特定した後に行う．仮想マシン上に Ubuntu をインストールしたサーバを使用した学生を特定した後，監視対象の VMware ESXi 上にある仮想マシンを作成した学生を特定し，その学生を VMware ESXi をインストールした物理サーバを使用したと判定する．

ユーザアカウントでログインを行う必要があるアプリケーションが動作する Pod を使用した学生を特定する方法

ユーザアカウントでログインを行う必要があるアプリケーションが動作する Pod を使用したと判定する基準を以下に述べる．

- アプリケーション内にユーザアカウントが登録されていること．
- アプリケーションが動作する Pod において，アプリケーションのログに記録されたユーザアカウント名と対応する学生の名前から，その学生がその Pod を使用したと判定する．

ユーザアカウントでログインを行う必要があるアプリケーションが動作する Pod の例として，Grafana がある．このような Pod においては，学生個人のユーザアカウントでログインを行うことで，アクセスしたユーザアカウントの名前がログに出力される．アプリケーションのログに記録さ

れたユーザアカウント名から，使用した学生を特定し，その学生が Pod で動作しているアプリケーションを使用したと判定する．ログを検索する期間は，後期の開始日である 2025 年 9 月 20 日からチケットが作成された日時までの期間である．理由は，CDSL では後期で行われた「卒業課題 II」の成果をテクニカルレポートを執筆し提出することが求められており，テクニカルレポートにもちいる実験のデータを収集するために作成するサーバは，後期の開始日以降に作成されるためである．

ユーザアカウントでログインを行わないアプリケーションが動作する Pod を使用した学生を特定する方法

ユーザアカウントでログインを行わないアプリケーションが動作する Pod を使用したと判定する基準を以下に述べる．

- 「ユーザアカウントでログインを行う必要があるアプリケーションが動作する Pod」の稼働や運用に必要となるサーバの名前を，「ユーザアカウントでログインを行う必要があるアプリケーションが動作する Pod」の関連サーバとして事前に入力してあること．
- 学生が使用したと判定した「ユーザアカウントでログインを行う必要があるアプリケーションが動作する Pod」の関連サーバに「ユーザアカウントでログインを行わないアプリケーションが動作する Pod」が含まれるとき，学生は「ユーザアカウントでログインを行わないアプリケーションが動作する Pod」も使用したと判定する．

ユーザアカウントでログインを行わないアプリケーションが動作する Pod の例として，Prometheus がある．このような Pod が，ユーザアカウントでログインを行うアプリケーションが動作する Pod の関連サーバに含まれていれば，その関連サーバを使用したと判定する．例えば，Prometheus が Grafana の関連サーバに含まれているとき，Grafana にログインを行うと Prometheus も使用したと判定する．関連サーバのリストは YAML で記述されており，階層構造になっている．1 階層目にはシステムの名前が指定される．2 階層目には 1 階層目で指定したシステムの稼働に必要な ESXi や VM，Pod の名前が指定される．システムの稼働に必要な ESXi や VM，Pod は，監視システムを構築する担当者が判定する．

ユースケース・シナリオ

CDSL の環境において，チケット担当者が担当しているチケットに記載されている，アラートが通知されたサーバとは別のサーバからアラートが通知され，新しいチケットが作成された状況を想定する．図 4 にユースケース・シナリオを示す．

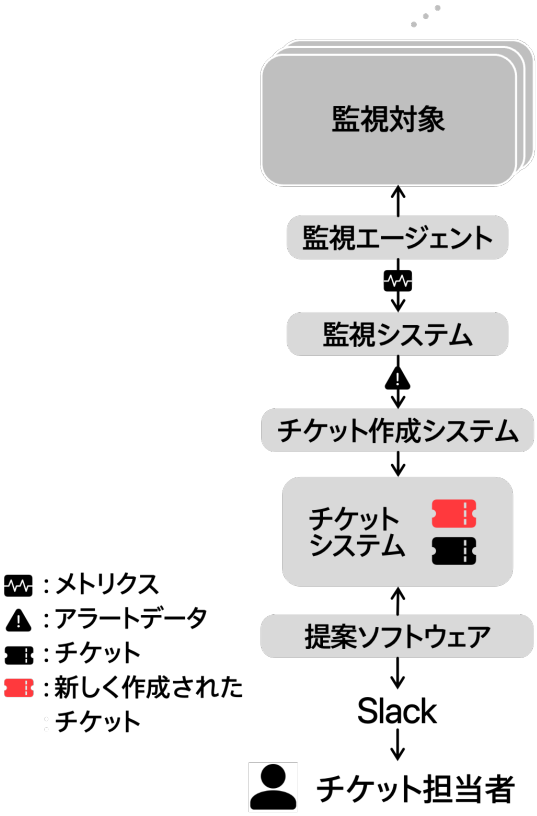

図 4: ユースケース・シナリオ

監視対象は CDSL で運用されているサーバを示す．監視エージェントは監視対象からメトリクスを取得し，監視システムに送信する．監視システムは取得したメトリクスが，あらかじめ指定したしきい値より大きい場合にアラートを作成する．監視システムはアラートをチケット作成システムに通知し，チケットシステム上にチケットを作成する．提案ソフトウェアは既存のチケットと新しく作成されたチケットを検出し，テクニカルレポートの未完了である章の数が多い学生が使用したサーバで検出されたアラートに対応するチケットを，未完了の章の数が多い学生から順に並び替え，先頭のチケットをチケット担当者に通知する．

提案ソフトウェアを適用した後，チケット担当者が複数のチケットを担当した場合に，「卒業課題 II」の進捗が遅れている学生が使用したサーバで検出されたアラートに対応するチケットが優先して調査や対応が行われる．そのため，学生の作業が停止する時間が短縮される確率が高くなる．

4. 実装

本提案を実装したソフトウェアである Progress-Scored Prioritizer は，Python 3.12.3 で作成した．使用した Python ライブラリは，Google の各種サービスとの連携や認証処理を行う google-api-python-client，google-auth-httplib2, google-auth-oauthlib, googleapiclient.discovery,

googleapiclient.http，google.oauth2.credentials，google.auth.transport.requests, google_auth_oauthlib.flow，システム操作やデータ処理を行う os，subprocess，re，json，datetime である．図 5 に，Progress-Scored Prioritizer の概要を示す．

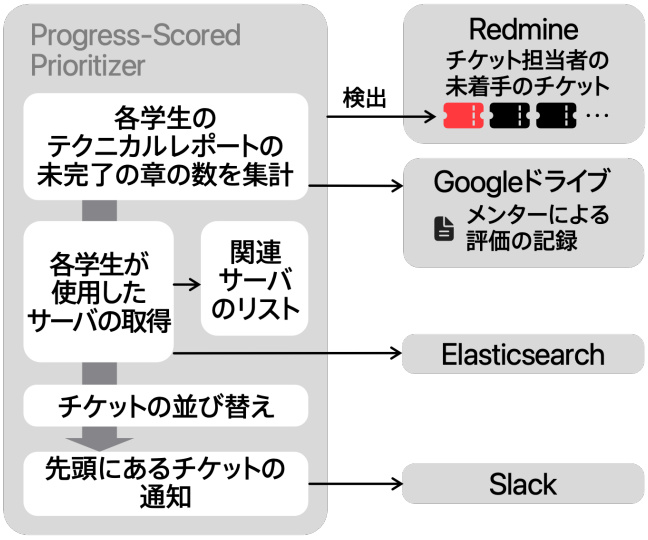

図 5: Progress-Scored Prioritizer の構成

Progress-Scored Prioritizer は，Redmine 上に新しくチケットが作成され，チケット担当者の未着手のチケットが追加されたことを検出する．「学生のスライドとテクニカルレポートで未完了の章の特定」では，Google ドライブ上の学生のメンターによる評価が記録されたスプレッドシートを取得する．「学生が使用したサーバの取得」では，Elasticsearch に保存されているログのうち，Redmine と Grafana のユーザアカウントでログインしたことを示すログの検索を行い，事前に定義した「関連サーバのリスト」にもとづいて，学生が使用したサーバを取得する．その後，「チケットの並び替え」では，チケット担当者の未着手のチケットに記載されているアラートが通知されたサーバについて，各学生のテクニカルレポートの未完了の章の数で降順に並べる．「先頭にあるチケットの通知」では，並び替えたチケットの先頭にあるチケットの名前と ID，チケット担当者の名前，URL を Slack に通知する．図 6 に，提案ソフトウェアによって Slack に通知されたチケットを示す．

図 6: 提案ソフトウェアによって Slack に通知されたチケット

Slack App である「Progress-Scored Prioritizer」は，チケットの名前と ID を含めた「1 次調査 #27287: Internal Host CPU UsageHigh-Usage100%」と，チケット担当者の名前である「山崎 拓海」が表示されたメッセージを，「山﨑 拓海」にメンションをつけて通知する．「1 次調査 #27287: Internal Host CPU UsageHigh-Usage100%」のテキストで表示されたリンクを押下することで，「http://monitoring-master-ml:32300/issues/27287」の URL に遷移することができる．

学生が使用したサーバの取得

VMware ESXi ごとに，使用した学生の名前を取得する方法について述べる．新規にチケットが作成された日時にそれぞれの VMware ESXi 上で作成されている仮想マシンのホスト名を取得する．学籍番号と学生の名前との対応付けが記録された CSV ファイルをもちいて，仮想マシンの名前に記載された学籍番号と対応する学生の名前から，その学生が仮想マシンを作成したと判定し，VMware ESXi ごとに集計を行う．

次に，ユーザ認証を必要とするアプリケーションである Redmine と Grafana の Pod においては，学生個人のユーザアカウントによるログインを行うことで，アクセスしたユーザアカウントの名前がアプリケーションログに出力される．ログ 1 に，Redmine から出力されたログの例を示す．

ログ 1: Redmine から出力されたログの例

```
I, [2025-11-03T08:23:59.198553 #245] INFO -- : [8e875b19-37d3-4e30-bdd0-a166fceef45e] Current user: tsukimori (id=6)
```

「Current user: 」の後に記述された学生のユーザアカウント名から，学生を特定する．次に Grafana から出力されたログの例を，ログ 2 に示す．

ログ 2: Grafana から出力されたログの例

```
logger=context userId=5 orgId=1 uname=tsukimori t=2025-10-29T05:04:30.673322174Z level=info msg="Request Completed" method=GET path=/api/live/ws status=-1 remote_addr=10.42.0.1 time_ms=1 duration=1.854174ms size=0 referer= handler=/api/live/ws status_source=server
```

「uname=」の後に学生のユーザアカウント名が記述される．Redmine と Grafana のログは Elasticsearch に保存されているため，curl コマンドで Elasticsearch に学生の

ユーザアカウント名を含むログを検索するクエリを作成することで，アプリケーションを使用した学生を特定する．

一方で，ユーザアカウントでログインを行わないアプリケーションが動作する Pod においては，ユーザアカウントでログインを行うアプリケーションが，あらかじめ指定した関連サーバのリストに含まれていれば，その関連サーバ内のすべてのサーバを使用したと判定する．プログラム 1 に，関連サーバのリストを示す YAML ファイルを示す．

プログラム 1: 関連サーバのリストを示す YAML ファイル

```yaml
related_servers:
  grafana-*:
    - Makaron
    - Canele
    - Lily
    - monitoring-master-ml
    - prometheus-*
  my-redmine-*:
    - Makaron
    - Canele
    - Lily
    - monitoring-master-ml
    - prometheus-*
    - alertmanager-*
    - ticket-receiver-*
    - ticket-notifier-*
```

関連サーバのリストは，Grafana や Redmine のアプリケーションを動作させるために必要なサーバをシステム管理者が判定し，提案ソフトウェアのセットアップをする前にあらかじめ作成する．例えば，grafana-*は Lily 上の monitoring-master-ml（Monitoring クラスタのマスターノード）上で動作しているため，Lily と monitoring-master-ml を使用したサーバとして判定している．grafana-*で表示されるグラフは prometheus-*が稼働しメトリクスを収集していないと表示することができないため，prometheus-*を関連サーバとして定義している．Makaron と Canele は，DNS や DHCP が稼働している．Prometheus や Grafana の Pod を稼働させるための Monitoring クラスタのマスターノードとワーカーノードは，相互にホスト名を指定して通信するため，DHCP による名前解決が必要である．そのため，DNS や DHCP が稼働している Makaron と Canele を使用したサーバとして判定している．また，my-redmine-*は Lily 上の monitoring-master-ml（Monitoring クラスタのマスターノード）上で動作している．my-redmine-*は，prometheus-*がアラートを作成し，alertmanager-*が ticket-receiver-*にアラートを通知し，Redmine にチケットを作成する一連の処理に必要である．さらに，ticket-notifier-*は作成されたチケットを担当者に通知するため，調査作業に必要である．そのため，prometheus-*，alertmanager-*，ticket-receiver-*，ticket-notifier-*を関連サーバとして定義している．学生の一度の使用でサーバを使用したと判定する理由は，一度利用したサーバやアプリケーションは，今後も使用する可能性があるからである．

チケットの並び替えと通知

テクニカルレポートの未完了である章の数を学生ごとに集計する．その後，学生が使用したサーバを特定し，そのサーバで検出されたアラートのチケットを，テクニカルレポートの未完了である章の数が多い学生から順に並び替えを行う．このとき，学生が使用したサーバで検出されたアラートのチケットが複数件ある場合は，作成日時が早いチケットから並べる．並び替えたチケットの先頭のチケットを担当者に Slack に通知する．

5. 評価実験

提案手法では，学生のメンターがテクニカルレポートの各章が未完了であるかを評価し，その結果をもとにチケットを並び替える．その他の手法として，各学生のテクニカルレポートの章ごとの記述とプロンプトを Gemma 3 に入力し，記述内容に意味の変化がある章を未完了の章として検出する手法があった．以下に，その手法である Gemma 3 を使用した場合の評価実験の結果を示す．

Gemma 3 を使用した場合のテクニカルレポートの進捗の判定

各学生のテクニカルレポートの未完了の章を特定する方法として，各学生のテクニカルレポートの章ごとの記述とプロンプトを Gemma 3 に入力し，記述内容に意味の変化がある章を未完了の章として検出することを試みた．Gemma 3 は，Google 社が提供するオープンモデルであり，パラメータサイズが 4B であるモデルを使用した．

プロンプト 1 に，Gemma 3 に入力するプロンプトを示す．

学生がアップロードしたテクニカルレポートのうち，最新とその 1 つ前のテクニカルレポートについて，章ごとの記述内容を指定のプロンプトとともに Gemma 3 に入力する．プロンプトの 1 行目は，直近 2 件の章ごとの記述内容のテキストを含め，意味に変化があるかどうかを判定させるように指示している．3 行目は，最新の 1 つ前のテクニカルレポートの章の記述内容であることを示すテキストと，その記述内容が old_section に格納される．4 行目は，最新のテクニカルレポートの章の記述内容であることを示すテキストと，その記述内容が new_section に格納される．6 行目では記述内容が含まれている章の番号が section_num に格納されており，7 行目では章のタイトルが section_title

プロンプト 1: Gemma 3 に入力するプロンプト

```
下記の２つのテキストを比較して，内容の意味に変化があるか判断してください

- 古い版のテキスト: {old_section}
- 新しい版のテキスト : {new_section}

- 章番号:{section_num}
- 章のタイトル: {section_title}

# 判定基準
## 内容の意味に変化があると判断する場合
- アルゴリズム，手法，アプローチの変更
- 入力データや前提条件の変更
- 実験結果の数値や評価指標の変更
- 提案内容の追加や削除，修正
## 内容の意味に変化がないと判断する場合
- 字数，語尾，語順，漢字表記の違い
- 単なる言い換えや言い回し，表現の変更
- フォーマットやレイアウトの変更

# 注意事項（章のタイトルに応じて以下の点を重視してください）
- 「提案」「実装」アルゴリズムや手法の内容やインプットの変更を重視:
- 「評価実験」: 実験結果の数値や評価指標の変更を重視
- 「はじめに」「議論」「まとめ」: 内容の意味の変更を重視

# 回答形式（下記のいずれかの形式で回答してください）
- YES
- NO
```

に格納される．9 行目は「判定基準」と記述されたの見出しであり，10 行目は判定基準のうち「内容の意味に変化があると判断する場合」と記述された見出しである．11 行目は 10 行目で示した条件の具体例として「アルゴリズムや，手法，アプローチの変更」，12 行目は「入力データや前提条件の変更」，13 行目は「実験結果の数値や評価指標の変更」，14 行目は「提案内容の追加や削除，修正」が記述されている．15 行目は判定基準のうち「内容の意味に変化がないと判断する場合」と記述された見出しである．16 行目は「字数，語尾，語順，漢字表記の違い」，17 行目は「単なる言い換えや言い回し，表現の変更」，18 行目は「フォーマットやレイアウトの変更」が記述されている．20 行目は「注意事項（章のタイトルに応じて以下の点を重視してください）」と記述された見出しである．21 行目は「「提案」「実装」: アルゴリズムや手法の内容やインプットの変更を重視」，22 行目には「「評価実験」: 実験結果の数値や評価指標の変更を重視」，23 行目には「「はじめに」「議論」「まとめ」: 内容の意味の変更を重視」が記述されている．26 行目には「回答形式（下記のいずれかの形式で回答してください）」と記述された見出しであり，26 行目に「YES」，27 行目には「NO」と記述されており，回答の選択肢を限定している．

表 1 に，2025 年 9 月 22 日 8 時 50 分から 2025 年 11 月 20 日 14 時 50 分までの各学生の最新と 1 つ前のテクニカルレポートから，記述内容の意味に変化があったと Gemma 3 が判定した章を示す．

表 1: 2025 年 9 月 22 日 8 時 50 分から 2025 年 11 月 20 日 14 時 50 分までの各学生の最新と 1 つ前のテクニカルレポートから，Gemma 3 が記述内容の意味に変化があったと判定した章

学生名	記述内容の意味に変化があった章
有田 海斗	2, 3, 4, 5
井出 佑	3, 4, 5
岡田 京太郎	3, 5
北川 翔也	判定不可
坂井 萌桜	1, 3, 4, 5
佐藤 健斗	1, 3, 4, 5, 6
月森 陽太	1, 3, 4, 5, 6, 7
手塚 雄星	3, 5
ムハンマド アクラム	なし
山﨑 拓海	1, 3, 4, 5
山崎 雅也	1, 3, 5

有田 海斗のテクニカルレポートの記述内容に変化があったと Gemma 3 が判定した章は，2025 年 11 月 19 日 5 時 57 分と 2025 年 11 月 20 日 2 時 9 分のテクニカルレポートを比較した結果，2 章，3 章，4 章，5 章であった．井出 佑のテクニカルレポートの記述内容に変化があったと Gemma 3 が判定した章は，2025 年 11 月 18 日 1 時 25 分と 2025 年 11 月 18 日 23 時 53 分のテクニカルレポートを比較した結果，3 章，4 章，5 章であった．岡田 京太郎のテクニカルレポートの記述内容に変化があったと Gemma 3 が判定した章は，2025 年 11 月 13 日 23 時 50 分と 2025 年 11 月 19 日 7 時 55 分のテクニカルレポートを比較した結果，3 章，5 章であった．北川 翔也のテクニカルレポートの記述内容に変化があったと Gemma 3 が判定した章は，テクニカルレポートが Google ドライブにアップロードされていなかったため，判定不可とした．坂井 萌桜のテクニカルレポートの記述内容に変化があったと Gemma 3 が判定した章は，2025 年 11 月 12 日 9 時 2 分と 2025 年 11 月 18 日 23 時 52 分のテクニカルレポートを比較した結果，1 章，3 章，4 章，5 章であった．佐藤 健斗のテクニカルレポートの記述内容に変化があったと Gemma 3 が判定した章は，2025 年 11 月 19 日 4 時 14 分と 2025 年 11 月 19 日 5 時 48 分の

テクニカルレポートを比較した結果，1 章，3 章，4 章，5 章，6 章であった．月森 陽太のテクニカルレポートの記述内容に変化があったと Gemma 3 が判定した章は，2025 年 11 月 18 日 1 時 51 分と 2025 年 11 月 19 日 0 時 7 分のテクニカルレポートを比較した結果，1 章，3 章，4 章，5 章，6 章，7 章であった．手塚 雄星のテクニカルレポートの記述内容に変化があったと Gemma 3 が判定した章は，2025 年 11 月 14 日 0 時 46 分と 2025 年 11 月 16 日 23 時 58 分のテクニカルレポートを比較した結果，3 章，5 章であった．ムハンマド アクラムのテクニカルレポートの記述内容に変化があったと Gemma 3 が判定した章は，2025 年 11 月 17 日 1 時 9 分と 2025 年 11 月 18 日 1 時 36 分のテクニカルレポートを比較した結果，記述内容の変更がなかったため，なしであった．山﨑 拓海のテクニカルレポートの記述内容に変化があったと Gemma 3 が判定した章は，2025 年 11 月 19 日 2 時 53 分と 2025 年 11 月 19 日 7 時 36 分のテクニカルレポートを比較した結果，1 章，3 章，4 章，5 章であった．山崎 雅也のテクニカルレポートの記述内容に変化があったと Gemma 3 が判定した章は，2025 年 11 月 3 日 1 時 0 分と 2025 年 11 月 4 日 23 時 58 分のテクニカルレポートを比較した結果，1 章，3 章，5 章であった．北川 翔也とムハンマド アクラムのテクニカルレポートの記述内容に変化があったと Gemma 3 が判定した章はどちらもなしと判定された．その理由は，北川 翔也の場合はテクニカルレポートが Google ドライブにアップロードされていなかったためであり，ムハンマド アクラムの場合は記述内容の変更がなかったためである．これらを区別するために，テクニカルレポートがアップロードされており記述内容に変更がない場合は「なし」と判定し，テクニカルレポートがアップロードされていない場合は「判定不可」と判定する．

図 7 に，Gemma 3 が判定した記述内容に変化があった章の数と，学生のメンターが判定した未完了の章の数を示す．

縦軸は Gemma 3 が判定した記述内容に変化があった章の数と，学生のメンターが判定した未完了の章の個数であり，横軸は CDSL の学生の名前である．青色の「※」は，テクニカルレポートがアップロードされていないため，Gemma 3 が記述内容に変化がある章を判定できなかったことを示す．赤色の「※」は，欠席しており学生のメンターの判定が存在しないため，学生のメンターによる判定できなかったことを示す．Gemma 3 による判定は，2025 年 9 月 22 日 8 時 50 分から 2025 年 11 月 20 日 14 時 50 分までの各学生の最新と 1 つ前のテクニカルレポートから，記述内容の意味に変化があったと Gemma 3 が判定した章である．学生のメンターによる判定は，2025 年 11 月 19 日 16 時 20 分時点で最新のスプレッドシートに記載されていた，学生のメンターがテクニカルレポートの各章が未完了であ

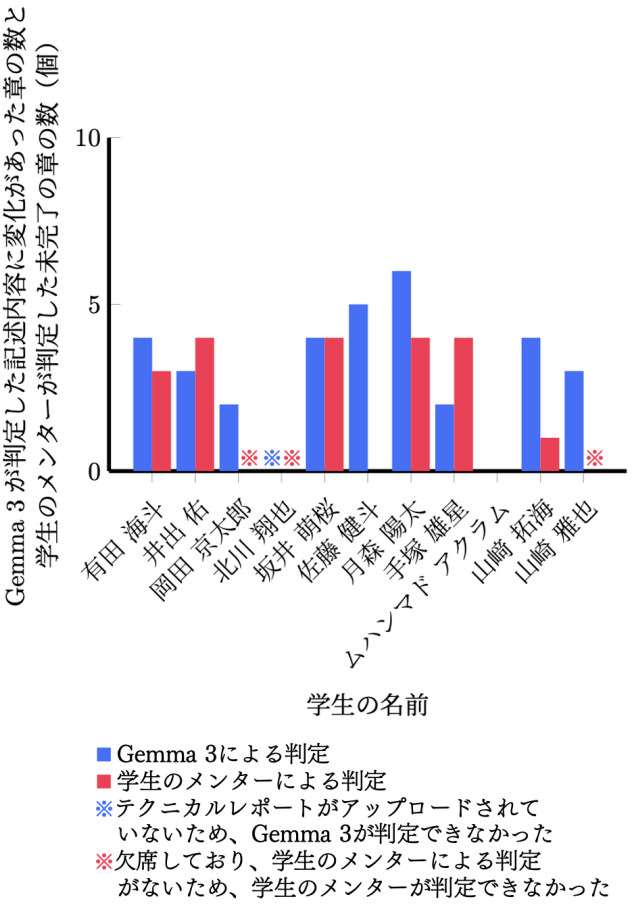

図 7: Gemma 3 が判定した記述内容に変化があった章の数と，学生のメンターが判定した未完了の章の数

ると判定した章である．図中のオレンジの囲み線がついている北川 翔也と山崎 雅也は欠席が続いており，テクニカルレポートがアップロードされていないため，未完了の章の数が 0 個と判定されている．2025 年 11 月 19 日 16 時 20 分時点で最新のスプレッドシートには，8 人の学生のテクニカルレポートの各章が未完了であると判定した章が記載されていた．有田 海斗のテクニカルレポートの未完了の章の数は，Gemma 3 による判定では 4 個，学生のメンターによる判定では 3 個である．井出 佑のテクニカルレポートの未完了の章の数は，Gemma 3 による判定では 3 個，学生のメンターによる判定では 4 個である．岡田 京太郎のテクニカルレポートの未完了の章の数は，Gemma 3 による判定では 2 個，学生のメンターによる判定では 0 個である．北川 翔也のテクニカルレポートの未完了の章の数は，テクニカルレポートが Google ドライブにアップロードされていなかったため，Gemma 3 による未完了の章を判定するプログラムが 0 個と判定した．また学生のメンターによる判定では，欠席のため学生のメンターがスプレッドシートに未完了の章の入力をしていない場合は 0 個と判定しているため，0 個である．坂井 萌桜のテクニカルレポートの未完了の章の数は，Gemma 3 による判定では 4 個，

学生のメンターによる判定では 4 個である．佐藤 健斗のテクニカルレポートの未完了の章の数は，Gemma 3 による判定では 5 個，学生のメンターによる判定では 0 個である．月森 陽太のテクニカルレポートの未完了の章の数は，Gemma 3 による判定では 6 個，学生のメンターによる判定では 4 個である．手塚 雄星のテクニカルレポートの未完了の章の数は，Gemma 3 による判定では 2 個，学生のメンターによる判定では 4 個である．ムハンマド アクラムのテクニカルレポートの未完了の章の数は，Gemma 3 による判定では 0 個，学生のメンターによる判定では 0 個である．山﨑 拓海のテクニカルレポートの未完了の章の数は，Gemma 3 による判定では 4 個，学生のメンターによる判定では 1 個である．山崎 雅也のテクニカルレポートの未完了の章の数は，Gemma 3 による判定では 3 個，また学生のメンターによる判定では，欠席のため学生のメンターがスプレッドシートに未完了の章の入力をしていない場合は 0 個と判定しているため，0 個である．Gemma 3 が判定した未完了の章の数が，メンターの判定による未完了の章の数と一致していたのは北川 翔也と，坂井 萌桜，ムハンマド アクラムである．北川 翔也のテクニカルレポートの未完了の章の数については，テクニカルレポートが Google ドライブにアップロードされていなかったため，Gemma 3 による判定では 0 個と判定されていた．学生のメンターによる判定では，欠席のため学生のメンターがスプレッドシートに未完了の章の入力をしていないため，0 個と判定されており，テクニカルレポートが完了しているわけではない．Gemma 3 が判定した未完了の章の数が，メンターの判定による未完了の章の数より多いのは有田 海斗と，岡田 京太郎，佐藤 健斗，月森 陽太，山﨑 拓海，山崎 雅也である．Gemma 3 が判定した未完了の章の数が，メンターの判定による未完了の章の数より少ないのは井出 佑と手塚 雄星である．Gemma 3 が判定した未完了の章の数がメンターの判定による未完了の章の数と一致している人数は 11 人中 3 人であり，一致率は約 27%であった．一方で，北川 翔也はテクニカルレポートをアップロードしておらず，山崎 雅也はテクニカルレポートをアップロードしているものの学生のメンターによる判定がない．そのため，学生のメンターによる判定が行われていない学生は，評価から除く必要がある．

提案を適用した後のチケットの順位と比較対象との順位相関係数

提案手法では，学生のメンターがテクニカルレポートの各章が未完了であるかを評価し，その結果をもとにチケットを並び替える．評価実験は，2 つの方法で行った．1 つ目は，2025 年 9 月 1 日 9 時 0 分から 2025 年 12 月 1 日 17 時 30 分までの間に作成された 26 件の未着手のチケットを対象に，提案を適用した後のチケットの順位と，2 種類の比較対象との順位相関係数を算出した．2 つ目は，2025 年 9 月 1 日 9 時 0 分から 2025 年 12 月 1 日 17 時 30 分までの間に調査が開始されたチケット 26 件の作成日時から調査開始までの時間と，提案を適用した後の 2025 年 12 月 11 日 12 時 30 分から 2025 年 12 月 12 日 15 時 20 分までの間に調査が開始されたチケット 2 件の作成日時から調査開始までの時間を比較した．

順位相関係数の算出方法

順位相関係数は，2 組の順位に対して，各順位の値の差の二乗和をもとに順位の一致度を算出したものであり，-1 から 1 までの値をとる．順位相関係数 ρ は，式 (1) で表される．

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{1}$$

ここで，d_i は i 番目のデータの 2 組の順位の差，n はデータ数，6 は定数である．

2 種類の比較対象の作成方法

1 つ目の比較対象は，2025 年 11 月 28 日 18 時 20 分時点でテクニカルレポートの初版が未提出の学生が使用したサーバで検出されたアラートのチケット 16 件を作成日時が早い順に並べ，その後ろに残りの 10 件を作成日時が早い順に並べた順位である．表 2 に，1 つ目の比較対象の順位を示す．

1 つ目の比較対象の順位は，以下の手順で作成した．はじめに，テクニカルレポートの初版が未提出の学生を特定した．初版を提出した学生は 2025 年 11 月 22 日に提出した坂井 萌桜，2025 年 11 月 21 日に提出した岡田 京太郎，2025 年 11 月 18 日に提出したムハンマド アクラム，2025 年 11 月 27 日に提出した井出 佑，2025 年 11 月 20 日に提出した佐藤 健斗，2025 年 11 月 26 日に提出した山﨑 拓海の合計 6 人である．初版が未提出の学生は有田 海斗，北川 翔也，月森 陽太，手塚 雄星，山崎 雅也の合計 5 人である．次に，テクニカルレポートの初版が未提出の学生が使用したサーバで検出されたアラートに対応するチケットを作成日時が早い順に並び替えた．

このとき，学生が使用したサーバが先に並べた学生が使用したと同じ場合，そのサーバ以外で検出されたアラートに対応するチケットを作成日時が早い順に並べ，先に配置したセットの後ろに配置する．初版が未提出の学生であり有田 海斗が使用したサーバは Rose, Plum, Lily, Makaron, Canele, monitoring-master-ml, grafana-*, my-redmine-*, prometheus-*, alertmanager-*, ticket-receiver-*, ticket-notifier-*であった．北川 翔也が使用したサーバは Lotus, Makaron, Canele であった．月森 陽

表 2: 1 つ目の比較対象の順位

順位	チケット名	アラートが通知されたサーバ
1	ElastAlert: Alert on {'name': 'fb-filebeat-j5c5l'}	my-redmine
2	Prometheus High Error Log Rate	prometheus
3	Internal Core Server Cluster PersistentVolumeUsageHigh	prometheus
4	Internal Core Server Cluster PersistentVolumeFullSoon	prometheus
5	Internal Monitoring Cluster Node Memory Usage High 80	monitoring-master-ml
6	Internal Monitoring Cluster Abnormal pod status	my-redmine
7	Internal Monitoring ClusterNodeDiskPressure	monitoring-master-ml
8	Internal Monitoring Cluster Node Filesystem Usage High 80	monitoring-master-ml
9	Internal ESXi ICMP Check	Plum
10	Internal ESXi ICMP Check	Lotus
11	Internal ESXi ICMP Check	Rose
12	Internal Host CPU UsageHigh-Usage100%	Mint
13	Internal Host CPU UsageHigh-Usage100%	Makaron
14	Internal Monitoring Cluster PersistentVolumeFullSoon	prometheus
15	Internal DataStoreDiskUsageHigh-Usage 80%	monitoring-master-ml
16	External_Doktor_Front_Page_HTTP_Status_Code_Err	Doktor
17	ElastAlert: Alert on {'name': 'fb-filebeat-5bjxp'}	c0117304-monitor(Violet 内のESXi上の VM)
18	External_Doktor_Front_Page_Response_Time_Late_1s	Doktor
19	External_Doktor_Front_Page_No_Response	Doktor
20	External output Lidar data check port 8002	LiDAR
21	External output Lidar data check port 8001	LiDAR
22	External_Doktor_Front_Page_No_Response	Doktor
23	Internal ESXi ICMP Check	Violet
24	ElastAlert: thanos_sidecar_no_upload_block_log - No upload block logs detected	thanos
25	External_Doktor_Front_Page_Response_Time_Late_1s	Doktor
26	External_Doktor_Front_Page_HTTP_Status_Code_Err	Doktor

太が使用したサーバは Mint, Jasmine, Lily, Makaron, Canele, monitoring-master-ml, grafana-*, my-redmine-*, prometheus-*, alertmanager-*, ticket-receiver-*, ticket-notifier-*であった．手塚 雄星が使用したサーバは Lotus, Lily, Makaron, Canele, monitoring-master-ml, grafana-*, my-redmine-*, prometheus-*, alertmanager-*, ticket-receiver-*, ticket-notifier-*であった．山崎 雅也が使用したサーバは Lotus, Makaron, Canele であった．初版が未提出の学生が使用したサーバで検出されたアラートに対応するチケットを作成日時が早い順に並べ，先頭に配置した．その後，初版を提出した学生が使用したサーバで検出されたアラートに対応するチケットを作成日時が早い順に並べ，先に配置したセットの後ろに配置した．

2 つ目の比較対象は，CDSL の「Cadence」「論文輪講会」「勉強会」の欠席数の合計が多い学生から順に，その学生が使用したサーバで検出されたアラートのチケットを，作成日時が早い順に並べた順位である．表 3 に，2 つ目の比較対象の順位を示す．

表 3: 2 つ目の比較対象の順位

順位	チケット名	アラートが通知されたサーバ
1	Internal ESXi ICMP Check	Lotus
2	Internal Host CPU UsageHigh-Usage100%	Makaron
3	ElastAlert: Alert on {'name': 'fb-filebeat-j5c5l'}	my-redmine
4	Prometheus High Error Log Rate	prometheus
5	Internal Core Server Cluster PersistentVolumeUsageHigh	prometheus
6	Internal Core Server Cluster PersistentVolumeFullSoon	prometheus
7	Internal Monitoring Cluster Node Memory Usage High 80	monitoring-master-ml
8	Internal Monitoring Cluster Abnormal pod status	my-redmine
9	Internal Monitoring ClusterNodeDiskPressure	monitoring-master-ml
10	Internal Monitoring Cluster Node Filesystem Usage High 80	monitoring-master-ml
11	Internal Host CPU UsageHigh-Usage100%	Mint
12	Internal Monitoring Cluster PersistentVolumeFullSoon	prometheus
13	Internal DataStoreDiskUsageHigh-Usage 80%	monitoring-master-ml
14	Internal ESXi ICMP Check	Violet
15	Internal ESXi ICMP Check	Plum
16	Internal ESXi ICMP Check	Rose
17	External_Doktor_Front_Page_HTTP_Status_Code_Err	Doktor
18	ElastAlert: Alert on {'name': 'fb-filebeat-5bjxp'}	c0117304-monitor(Violet 内のESXi上の VM)
19	External_Doktor_Front_Page_Response_Time_Late_1s	Doktor
20	External_Doktor_Front_Page_No_Response	Doktor
21	External output Lidar data check port 8002	LiDAR
22	External output Lidar data check port 8001	LiDAR
23	External_Doktor_Front_Page_No_Response	Doktor
24	ElastAlert: thanos_sidecar_no_upload_block_log - No upload block logs detected	thanos
25	External_Doktor_Front_Page_Response_Time_Late_1s	Doktor
26	External_Doktor_Front_Page_HTTP_Status_Code_Err	Doktor

2 つ目の比較対象の順位は，以下の手順で作成した．はじめに，2025 年 12 月 5 日 15 時 30 分時点における，各学生の「Cadence」「論文輪講会」「勉強会」の欠席数の合計を

集計し，欠席数が多い学生から順に順位をつけた．次に，各学生が使用したサーバで検出されたアラートに対応するチケットを，欠席数が多い学生から順に並び替えた．このとき，学生が使用したサーバが先に並べた学生が使用したサーバと同じ場合，そのサーバ以外で検出されたアラートに対応するチケットを作成日時の早い順に並べ，先に配置したセットの後ろに配置する．

表 4 に，各学生が使用したサーバを示す．

表 4: 各学生が使用したサーバ

学生名	使用したサーバ
有田 海斗	Rose, Plum, Lily, Makaron, Canele, monitoring-master-ml, grafana-*, my-redmine-*, prometheus-*, alertmanager-*, ticket-receiver-*, ticket-notifier-*
井出 佑	Mint, Lily, Makaron, Canele, monitoring-master-ml, my-redmine-*, prometheus-*, alertmanager-*, ticket-receiver-*, ticket-notifier-*
岡田 京太郎	Violet, Lily, Makaron, Canele, monitoring-master-ml, grafana-*, my-redmine-*, prometheus-*, alertmanager-*, ticket-receiver-*, ticket-notifier-*
北川 翔也	Lotus, Canele, Makaron
坂井 萌桜	Jasmine, Violet, Plum, Lily, Makaron, Canele, monitoring-master-ml, grafana-*, my-redmine-*, prometheus-*, alertmanager-*, ticket-receiver-*, ticket-notifier-*
佐藤 健斗	Mint, Jasmine, Violet, Lily, Makaron, Canele, monitoring-master-ml, grafana-*, my-redmine-*, prometheus-*, alertmanager-*, ticket-receiver-*, ticket-notifier-*
月森 陽太	Mint, Jasmine, Lily, Makaron, Canele, monitoring-master-ml, grafana-*, my-redmine-*, prometheus-*, alertmanager-*, ticket-receiver-*, ticket-notifier-*
手塚 雄星	Lotus, Lily, Makaron, Canele, monitoring-master-ml, grafana-*, my-redmine-*, prometheus-*, alertmanager-*, ticket-receiver-*, ticket-notifier-*
ムハンマド アクラム	Mint, Lily, Makaron, monitoring-master-ml, Canele, grafana-*, my-redmine-*, prometheus-*, alertmanager-*, ticket-receiver-*, ticket-notifier-*
山崎 雅也	Lotus, Canele, Makaron
山﨑 拓海	Violet, Lily, Makaron, Canele, monitoring-master-ml, grafana-*, my-redmine-*, prometheus-*, alertmanager-*, ticket-receiver-*, ticket-notifier-*

欠席数が 1 番目に多い学生である北川 翔也が使用したサーバで検出されたアラートに対応するチケットを作成日時が早い順に並べ，チケット順位の先頭に配置した．その後，欠席数が 2 番目に多い学生である山崎 雅也が使用したサーバは，北川 翔也が使用したサーバと同じである．そのため，山崎 雅也が使用したサーバで検出されたアラートに対応するチケットは，すでに北川 翔也が使用したサーバと同じサーバで検出されたアラートに対応するチケットは並び替えが行われているため，並び替えをスキップした．欠席数が 3 番目に多い学生である佐藤 健斗が使用したサーバのうち，Makaron と Canele 以外は並び替えが行われているため，Mint, Jasmine, Violet, Lily, monitoring-master-ml, grafana-*, my-redmine-*, prometheus-*, alertmanager-*, ticket-receiver-*, ticket-notifier-*で検出されたアラートに対応するチケットを作成日時が早い順に並べ，先に配置したセットの後ろに配置した．欠席数が 4 番目に多い学生である岡田 京太郎と，欠席数が 5 番目に多い学生であるムハンマド アクラムが使用したサーバで検出されたアラートに対応するチケットは，すでに並び替えが行われているため，並び替えをスキップした．欠席数が 6 番目に多い学生である坂井 萌桜が使用したサーバのうち，Plum 以外のサーバで検出されたアラートに対応するチケットは並び替えが行われているため，Plum で検出されたアラートに対応するチケットを先に配置したセットの後ろに配置した．欠席数が 7 番目に多い学生である月森 陽太が使用したサーバで検出されたアラートに対応するチケットは，すでに並び替えが行われているため，並び替えをスキップした．欠席数が 8 番目に多い学生である有田 海斗が使用したサーバのうち，Rose 以外のサーバで検出されたアラートに対応するチケットは並び替えが行われているため，Rose で検出されたアラートに対応するチケットを先に配置したセットの後ろに配置した．欠席数が 9 番目に多い学生である手塚 雄星と，欠席数が 10 番目に多い学生である井出 佑，欠席数が 11 番目に多い学生である山﨑 拓海が使用したサーバで検出されたアラートに対応するチケットは，すでに並び替えが行われているため，並び替えをスキップした．

実験環境

図 8 に実験環境を示す．実験環境は，CDSL で運用されている Monitoring クラスタを使用した．Monitoring クラスタは 2 台のノードで構成され，マスターノードとワーカーノードとして機能するそれぞれの仮想マシンのスペックはいずれも vCPU: 4 [Core]，RAM: 8 [GB]，SSD: 40 [GB] であり，OS として Ubuntu 24.04.2 LTS がインストールされている．

図中の Mint は，CDSL で運用している 10 台のサーバのうちの 1 台の名称である．10 台のサーバには，ハイパーバイザーとして Broadcom 社の VMware 製品である VMware

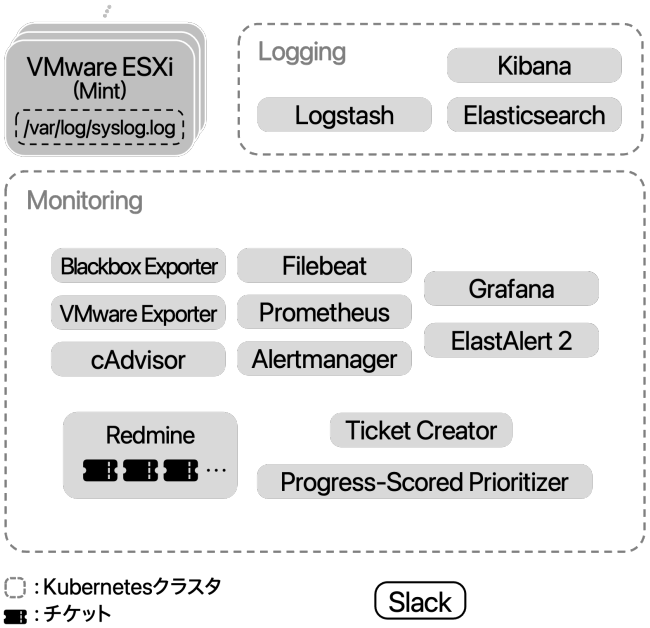

図 8: 実験環境

ESXi が導入されている．Logging クラスタ上では，サーバやアプリケーションから収集したログの構造化を行うアプリケーションや，検索や分析機能を提供するアプリケーションが Pod として稼働している．Pod とは，1 つまたは複数のコンテナのグループであり，Kubernetes の最小かつ最も基本的なデプロイ可能なオブジェクトである．Mint をはじめとした 10 台のサーバのハイパーバイザーである VMware ESXi のシステムログ (/var/log/syslog.log) は，Logging クラスタ上の Logstash に送信される．Monitoring クラスタ上で稼働している Filebeat は，Monitoring クラスタ上の Pod のログファイルを Logging クラスタ上の Logstash に送信する．Logstash は，取得した非構造化データを構造化して Elasticsearch に送信する．Elasticsearch は，送信されたデータを保存し，検索や分析機能を提供する．Kibana は，Elasticsearch 上に保存されたデータの分析や検索を行うダッシュボード機能をブラウザ上で提供する．Monitoring クラスタ上では，Prometheus は Blackbox Exporter や，VMware Exporter，cAdvisor からメトリクスを取得する．Blackbox Exporter は，VMware ESXi や仮想マシン上の OS やアプリケーションに対して HTTP や ICMP を経由したエンドポイントの外形監視を行い，その結果をメトリクスとして Prometheus に送信する．VMware Exporter は，VMware ESXi のホストや仮想マシンのメトリクスを取得し，Prometheus に送信する．cAdvisor は，実行中のコンテナのメトリクスを取得し，Prometheus に送信する．Prometheus は，取得したメトリクスがあらかじめ指定したルールを満たしたときにアラートを作成し，Alertmanager に送信する．また ElastAlert 2 は，Elasticsearch に収集されたログがあらかじめ指定したルールを満たしたときにアラートを作成する．Alertmanager と ElastAlert 2 はアラートを Ticket Creator に送信し，Ticket Creator は Redmine 上にアラートの内容を記載したチケットを作成する．Ticket Notifier は，Redmine にチケットが作成されたことをチケット担当者に Slack で通知する．チケット担当者は Redmine にアクセスする．

実験結果と分析

2025 年 9 月 1 日から 2025 年 12 月 1 日までの間に Monitoring クラスタ上の Redmine に作成されたチケットのうち，未着手として残っているチケットは 26 件であった．表 5 に，26 件の未着手のチケットを提案ソフトウェアで並び替えた結果の順位を示す．

各学生のテクニカルレポートの未完了である章の数を集計し，未完了の章の数が多い学生から順に，その学生が使用したサーバで検出されたアラートに対応するチケットを，作成日時が早い順に並び替えた後の，各チケットの「順位」と「チケット名」，「アラートが通知されたサーバ」を示している．

表 6 に，提案を適用した後のチケットの順位と 2 つの比較対象との順位相関係数の，小数第 3 位を四捨五入した値を示す．

提案を適用した後のチケットの順位と 1 つ目の比較対象との順位相関係数が約 0.85 となった．約 0.85 となった理由は，学生のメンターが 1 つでも未完了の章があると判定した場合は，テクニカルレポートの初版を提出することができないため，テクニカルレポートの初版が未提出である学生を対象に並び替えたチケットと，提案ソフトウェアをもちいた学生のメンターによる未完了の章の評価によって並び替えたチケットの順位では，各チケットの順位の差が小さくなる．

提案を適用した後のチケットの順位と 2 つ目の比較対象との順位相関係数が約 0.99 となった．これは，欠席数が多い学生の使用したサーバで検出されたアラートに対応するチケットから並び替えたときのチケットの順位と，提案ソフトウェアをもちいた学生のメンターによる未完了の章の評価によって並び替えたチケットの順位では，各チケットの順位の差が小さくなるためである．

提案手法を適用した後のチケットの順位と，2 つの比較対象との順位相関係数を比較した結果，提案手法を適用したチケットの順位は，1 つ目の比較対象で優先されている「初版が未提出の学生が使用したサーバで検出されたアラートに対応するチケット」よりも，2 つ目の比較対象で優先されている「欠席数が多い学生が使用したサーバで検出されたアラートに対応するチケット」を優先して並び替えていると言える．

表 5: 26 件の未着手のチケットを提案ソフトウェアで並び替えた結果の順位

順位	チケット名	アラートが通知されたサーバ
1	Internal ESXi ICMP Check	Lotus
2	Internal Host CPU UsageHigh-Usage100%	Makaron
3	ElastAlert: Alert on {'name': 'fb-filebeat-j5c5l'}	my-redmine
4	Prometheus High Error Log Rate	prometheus
5	Internal Core Server Cluster PersistentVolumeUsageHigh	prometheus
6	Internal Core Server Cluster PersistentVolumeFullSoon	prometheus
7	Internal Monitoring Cluster Node Memory Usage High 80	monitoring-master-ml
8	Internal Monitoring Cluster Abnormal pod status	my-redmine
9	Internal Monitoring ClusterNodeDiskPressure	monitoring-master-ml
10	Internal Monitoring Cluster Node Filesystem Usage High 80	monitoring-master-ml
11	Internal ESXi ICMP Check	Violet
12	Internal Monitoring Cluster PersistentVolumeFullSoon	prometheus
13	Internal DataStoreDiskUsageHigh-Usage 80%	monitoring-master-ml
14	Internal ESXi ICMP Check	Plum
15	Internal ESXi ICMP Check	Rose
16	Internal Host CPU UsageHigh-Usage100%	Mint
17	External_Doktor_Front_Page_HTTP_Status_Code_Err	Doktor
18	ElastAlert: Alert on {'name': 'fb-filebeat-5bjxp'}	c0117304-monitor(Violet内のESXi上のVM)
19	External_Doktor_Front_Page_Response_Time_Late_1s	Doktor
20	External_Doktor_Front_Page_No_Response	Doktor
21	External output Lidar data check port 8002	LiDAR
22	External output Lidar data check port 8001	LiDAR
23	External_Doktor_Front_Page_No_Response	Doktor
24	ElastAlert: thanos_sidecar_no_upload_block_log - No upload block logs detected	thanos
25	External_Doktor_Front_Page_Response_Time_Late_1s	Doktor
26	External_Doktor_Front_Page_HTTP_Status_Code_Err	Doktor

表 6: 提案を適用した後のチケットの順位と 2 つの比較対象との順位相関係数の，小数第 3 位を四捨五入した値

比較対象	順位相関係数
1 つ目の比較対象	約 0.85
2 つ目の比較対象	約 0.99

そのため提案の順位は，欠席数の多い学生が使用したサーバの異常のアラートに対応するチケットの調査を優先的に行うことができるようになっている．

提案を適用した後のチケットの順位と，1 つ目の比較対象との順位相関係数は，式 (1) を用いて計算した．はじめに，26 件のチケット（$n = 26$）について，各チケット i（$i = 1, 2, \ldots, 26$）の提案を適用した後のチケットの順位と 1 つ目の比較対象の順位の差を d_i とし，d_i の二乗和 $\sum_{i=1}^{26} d_i^2$ を計算した．各チケットの順位と d_i の値は以下の通りである．

- チケット「Internal ESXi ICMP Check」（Lotus）
  提案を適用した後の順位は 1 位，1 つ目の比較対象の順位は 10 位である．
  したがって，$d_1 = 1 - 10 = -9$
- チケット「Internal Host CPU UsageHigh-Usage100%」（makaron）
  提案を適用した後の順位は 2 位，1 つ目の比較対象の順位は 13 位である．
  したがって，$d_2 = 2 - 13 = -11$
- チケット「ElastAlert: Alert on {'name': 'fb-filebeat-j5c5l'}」（my-redmine）
  提案を適用した後の順位は 3 位，1 つ目の比較対象の順位は 1 位である．
  したがって，$d_3 = 3 - 1 = 2$
- チケット「Prometheus High Error Log Rate」（prometheus）
  提案を適用した後の順位は 4 位，1 つ目の比較対象の順位は 2 位である．
  したがって，$d_4 = 4 - 2 = 2$
- チケット「Internal Core Server Cluster PersistentVolumeUsageHigh」（prometheus）
  提案を適用した後の順位は 5 位，1 つ目の比較対象の順位は 3 位である．
  したがって，$d_5 = 5 - 3 = 2$
- チケット「Internal Core Server Cluster PersistentVolumeFullSoon」（prometheus）
  提案を適用した後の順位は 6 位，1 つ目の比較対象の順位は 4 位である．
  したがって，$d_6 = 6 - 4 = 2$
- チケット「Internal Monitoring Cluster Node Memory Usage High 80」（monitoring-master-ml）
  提案を適用した後の順位は 7 位，1 つ目の比較対象の順位は 5 位である．
  したがって，$d_7 = 7 - 5 = 2$
- チケット「Internal Monitoring Cluster Abnormal pod status」（my-redmine）
  提案を適用した後の順位は 8 位，1 つ目の比較対象の順位は 6 位である．
  したがって，$d_8 = 8 - 6 = 2$
- チケット「Internal Monitoring ClusterNodeDiskPressure」（monitoring-master-ml）
  提案を適用した後の順位は 9 位，1 つ目の比較対象の順位は 7 位である．

したがって，$d_9 = 9 - 7 = 2$

- チケット「Internal Monitoring Cluster Node Filesystem Usage High 80」（monitoring-master-ml）
  提案を適用した後の順位は 10 位，1 つ目の比較対象の順位は 8 位である．
  したがって，$d_{10} = 10 - 8 = 2$
- チケット「Internal ESXi ICMP Check」（Violet）
  提案を適用した後の順位は 11 位，1 つ目の比較対象の順位は 23 位である．
  したがって，$d_{11} = 11 - 23 = -12$
- チケット「Internal Monitoring Cluster PersistentVolumeFullSoon」（prometheus）
  提案を適用した後の順位は 12 位，1 つ目の比較対象の順位は 14 位である．
  したがって，$d_{12} = 12 - 14 = -2$
- チケット「Internal DataStoreDiskUsageHigh-Usage 80%」（monitoring-master-ml）
  提案を適用した後の順位は 13 位，1 つ目の比較対象の順位は 15 位である．
  したがって，$d_{13} = 13 - 15 = -2$
- チケット「Internal ESXi ICMP Check」（Plum）
  提案を適用した後の順位は 14 位，1 つ目の比較対象の順位は 9 位である．
  したがって，$d_{14} = 14 - 9 = 5$
- チケット「Internal ESXi ICMP Check」（Rose）
  提案を適用した後の順位は 15 位，1 つ目の比較対象の順位は 11 位である．
  したがって，$d_{15} = 15 - 11 = 4$
- チケット「Internal Host CPU UsageHigh-Usage100%」（Mint）
  提案を適用した後の順位は 16 位，1 つ目の比較対象の順位は 12 位である．
  したがって，$d_{16} = 16 - 12 = 4$
- チケット「[Alert] External_Doktor_Front_Page_HTTP_Status_Code_Err」（Doktor）
  提案を適用した後の順位は 17 位，1 つ目の比較対象の順位は 16 位である．
  したがって，$d_{17} = 17 - 16 = 1$
- チケット「ElastAlert: Alert on {'name': 'fb-filebeat-5bjxp'}」（c0117304-monitor）
  提案を適用した後の順位は 18 位，1 つ目の比較対象の順位は 17 位である．
  したがって，$d_{18} = 18 - 17 = 1$
- チケット「[Alert] External_Doktor_Front_Page_Response_Time_Late_1s」（Doktor）
  提案を適用した後の順位は 19 位，1 つ目の比較対象の順位は 18 位である．
  したがって，$d_{19} = 19 - 18 = 1$
- チケット「External_Doktor_Front_Page_No_Response」（Doktor）
  提案を適用した後の順位は 20 位，1 つ目の比較対象の順位は 19 位である．
  したがって，$d_{20} = 20 - 19 = 1$
- チケット「External output Lidar data check port 8002」（LiDAR）
  提案を適用した後の順位は 21 位，1 つ目の比較対象の順位は 20 位である．
  したがって，$d_{21} = 21 - 20 = 1$
- チケット「External output Lidar data check port 8001」（LiDAR）
  提案を適用した後の順位は 22 位，1 つ目の比較対象の順位は 21 位である．
  したがって，$d_{22} = 22 - 21 = 1$
- チケット「External_Doktor_Front_Page_No_Response」（Doktor）
  提案を適用した後の順位は 23 位，1 つ目の比較対象の順位は 22 位である．
  したがって，$d_{23} = 23 - 22 = 1$
- チケット「ElastAlert: thanos_sidecar_no_upload_block_log - No upload block logs detected」（thanos）
  提案を適用した後の順位は 24 位，1 つ目の比較対象の順位は 24 位である．
  したがって，$d_{24} = 24 - 24 = 0$
- チケット「External_Doktor_Front_Page_Response_Time_Late_1s」（Doktor）
  提案を適用した後の順位は 25 位，1 つ目の比較対象の順位は 25 位である．
  したがって，$d_{25} = 25 - 25 = 0$
- チケット「External_Doktor_Front_Page_HTTP_Status_Code_Err」（Doktor）
  提案を適用した後の順位は 26 位，1 つ目の比較対象の順位は 26 位である．
  したがって，$d_{26} = 26 - 26 = 0$

式 (2) に，26 件のチケットに対する提案を適用した後の順位と 1 つ目の比較対象の順位の差 d_i の二乗和 $\sum_{i=1}^{26} d_i^2$ の算出過程を示す．

$$\begin{aligned}\sum_{i=1}^{26} d_i^2 &= (-9)^2 + (-11)^2 + 2^2 + 2^2 + 2^2 + 2^2 \\ &\quad + 2^2 + 2^2 + 2^2 + 2^2 + (-12)^2 + (-2)^2 \\ &\quad + (-2)^2 + 5^2 + 4^2 + 4^2 + 1^2 + 1^2 \\ &\quad + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 0^2 \\ &\quad + 0^2 + 0^2 \\ &= 450\end{aligned} \tag{2}$$

順位相関係数 ρ の計算は式 (1) を用いる．式 (3) に，順位相関係数の算出過程を示す．

$$\begin{aligned}\rho &= 1 - \frac{6\sum_{i=1}^{26} d_i^2}{26(26^2-1)} \\ &= 1 - \frac{6 \times 450}{26(26^2-1)} \\ &= 1 - \frac{2700}{17550} \\ &= \frac{14850}{17550} \\ &= 0.8462\ldots\end{aligned} \tag{3}$$

提案を適用した後のチケットの順位と 1 つ目の比較対象との順位相関係数 ρ は，小数第 3 位を四捨五入し，約 0.85 である．

提案を適用した後のチケットの順位と，2 つ目の比較対象との順位相関係数は，式 (1) を用いて計算した．はじめに，26 件のチケット（$n = 26$）について，各チケット i（$i = 1, 2, \ldots, 26$）の提案を適用した後のチケットの順位と 2 つ目の比較対象の順位の差を d_i とし，d_i の二乗和 $\sum_{i=1}^{26} d_i^2$ を計算した．各チケットの順位と d_i の値は以下の通りである．

- チケット「Internal ESXi ICMP Check」（Lotus）
  提案を適用した後の順位は 1 位，2 つ目の比較対象の順位は 1 位である．
  したがって，$d_1 = 1 - 1 = 0$
- チケット「Internal Host CPU UsageHigh-Usage100%」（makaron）
  提案を適用した後の順位は 2 位，2 つ目の比較対象の順位は 2 位である．
  したがって，$d_2 = 2 - 2 = 0$
- チケット「ElastAlert: Alert on {’name’: ’fb-filebeat-j5c5l’}」（my-redmine）
  提案を適用した後の順位は 3 位，2 つ目の比較対象の順位は 3 位である．
  したがって，$d_3 = 3 - 3 = 0$
- チケット「Prometheus High Error Log Rate」（prometheus）
  提案を適用した後の順位は 4 位，2 つ目の比較対象の順位は 4 位である．
  したがって，$d_4 = 4 - 4 = 0$
- チケット「Internal Core Server Cluster PersistentVolumeUsageHigh」（prometheus）
  提案を適用した後の順位は 5 位，2 つ目の比較対象の順位は 5 位である．
  したがって，$d_5 = 5 - 5 = 0$
- チケット「Internal Core Server Cluster PersistentVolumeFullSoon」（prometheus）
  提案を適用した後の順位は 6 位，2 つ目の比較対象の順位は 6 位である．
  したがって，$d_6 = 6 - 6 = 0$
- チケット「Internal Monitoring Cluster Node Memory Usage High 80」（monitoring-master-ml）
  提案を適用した後の順位は 7 位，2 つ目の比較対象の順位は 7 位である．
  したがって，$d_7 = 7 - 7 = 0$
- チケット「Internal Monitoring Cluster Abnormal pod status」（my-redmine）
  提案を適用した後の順位は 8 位，2 つ目の比較対象の順位は 8 位である．
  したがって，$d_8 = 8 - 8 = 0$
- チケット「Internal Monitoring ClusterNodeDiskPressure」（monitoring-master-ml）
  提案を適用した後の順位は 9 位，2 つ目の比較対象の順位は 9 位である．
  したがって，$d_9 = 9 - 9 = 0$
- チケット「Internal Monitoring Cluster Node Filesystem Usage High 80」（monitoring-master-ml）
  提案を適用した後の順位は 10 位，2 つ目の比較対象の順位は 10 位である．
  したがって，$d_{10} = 10 - 10 = 0$
- チケット「Internal ESXi ICMP Check」（Violet）
  提案を適用した後の順位は 11 位，2 つ目の比較対象の順位は 14 位である．
  したがって，$d_{11} = 11 - 14 = -3$
- チケット「Internal Monitoring Cluster PersistentVolumeFullSoon」（prometheus）
  提案を適用した後の順位は 12 位，2 つ目の比較対象の順位は 12 位である．
  したがって，$d_{12} = 12 - 12 = 0$
- チケット「Internal DataStoreDiskUsageHigh-Usage 80%」（monitoring-master-ml）
  提案を適用した後の順位は 13 位，2 つ目の比較対象の順位は 13 位である．
  したがって，$d_{13} = 13 - 13 = 0$
- チケット「Internal ESXi ICMP Check」（Plum）
  提案を適用した後の順位は 14 位，2 つ目の比較対象の順位は 15 位である．
  したがって，$d_{14} = 14 - 15 = -1$
- チケット「Internal ESXi ICMP Check」（Rose）
  提案を適用した後の順位は 15 位，2 つ目の比較対象の順位は 16 位である．
  したがって，$d_{15} = 15 - 16 = -1$
- チケット「Internal Host CPU UsageHigh-Usage100%」（Mint）
  提案を適用した後の順位は 16 位，2 つ目の比較対象の

順位は 11 位である．
したがって，$d_{16} = 16 - 11 = 5$

- チケット「[Alert] External_Doktor_Front_Page_HTTP_Status_Code_Err」（Doktor）
  提案を適用した後の順位は 17 位，2 つ目の比較対象の順位は 17 位である．
  したがって，$d_{17} = 17 - 17 = 0$
- チケット「ElastAlert: Alert on {'name': 'fb-filebeat-5bjxp'}」（c0117304-monitor）
  提案を適用した後の順位は 18 位，2 つ目の比較対象の順位は 18 位である．
  したがって，$d_{18} = 18 - 18 = 0$
- チケット「[Alert] External_Doktor_Front_Page_Response_Time_Late_1s」（Doktor）
  提案を適用した後の順位は 19 位，2 つ目の比較対象の順位は 19 位である．
  したがって，$d_{19} = 19 - 19 = 0$
- チケット「External_Doktor_Front_Page_No_Response」（Doktor）
  提案を適用した後の順位は 20 位，2 つ目の比較対象の順位は 20 位である．
  したがって，$d_{20} = 20 - 20 = 0$
- チケット「External output Lidar data check port 8002」（LiDAR）
  提案を適用した後の順位は 21 位，2 つ目の比較対象の順位は 21 位である．
  したがって，$d_{21} = 21 - 21 = 0$
- チケット「External output Lidar data check port 8001」（LiDAR）
  提案を適用した後の順位は 22 位，2 つ目の比較対象の順位は 22 位である．
  したがって，$d_{22} = 22 - 22 = 0$
- チケット「External_Doktor_Front_Page_No_Response」（Doktor）
  提案を適用した後の順位は 23 位，2 つ目の比較対象の順位は 23 位である．
  したがって，$d_{23} = 23 - 23 = 0$
- チケット「ElastAlert: thanos_sidecar_no_upload_block_log - No upload block logs detected」（thanos）
  提案を適用した後の順位は 24 位，2 つ目の比較対象の順位は 24 位である．
  したがって，$d_{24} = 24 - 24 = 0$
- チケット「External_Doktor_Front_Page_Response_Time_Late_1s」（Doktor）
  提案を適用した後の順位は 25 位，2 つ目の比較対象の順位は 25 位である．
  したがって，$d_{25} = 25 - 25 = 0$
- チケット「External_Doktor_Front_Page_HTTP_Status_Code_Err」（Doktor）
  提案を適用した後の順位は 26 位，2 つ目の比較対象の順位は 26 位である．
  したがって，$d_{26} = 26 - 26 = 0$

式 (4) に，26 件のチケットに対する提案を適用した後の順位と 2 つ目の比較対象の順位の差 d_i の二乗和 $\sum_{i=1}^{26} d_i^2$ の算出過程を示す．

$$\begin{aligned}\sum_{i=1}^{26} d_i^2 &= 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 \\ &\quad + 0^2 + 0^2 + 0^2 + 0^2 + (-3)^2 + 0^2 \\ &\quad + 0^2 + (-1)^2 + (-1)^2 + 5^2 + 0^2 + 0^2 \\ &\quad + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 \\ &\quad + 0^2 + 0^2 \\ &= 36\end{aligned} \tag{4}$$

順位相関係数 ρ の計算は式 (1) を用いる．式 (5) を用いて順位相関係数の算出過程を示す．

$$\begin{aligned}\rho &= 1 - \frac{6\sum_{i=1}^{26} d_i^2}{26(26^2 - 1)} \\ &= 1 - \frac{6 \times 36}{26(26^2 - 1)} \\ &= 1 - \frac{216}{17550} \\ &= \frac{17334}{17550} \\ &= 0.9877\ldots\end{aligned} \tag{5}$$

提案を適用した後のチケットの順位と 2 つ目の比較対象との順位相関係数 ρ は，小数第 3 位を四捨五入し，約 0.99 である．

提案を適用した前と適用した後のチケットの調査開始までの時間

提案を適用した前と適用した後のチケットの調査開始までの時間を比較する．提案を適用した前のチケットは 2025 年 9 月 1 日から 2025 年 12 月 1 日までの期間で作成されたチケットのうち，調査が開始されている 23 件のチケットを対象とする．提案を適用した後のチケットは 2025 年 12 月 11 日から 2025 年 12 月 12 日までの期間で作成されたチケットのうち，未着手のチケット 2 件を対象とする．

図 9 に，2025 年 9 月 1 日から 2025 年 12 月 1 日までの間に作成された，調査が開始されているチケット 23 件の，提案を適用する前のチケット作成から調査開始までの時間の分布を示す．

横軸の間隔を 1000 分とした．チケットの作成から調査の開始までにかかった時間が 0 分以上 1000 分未満のチケッ

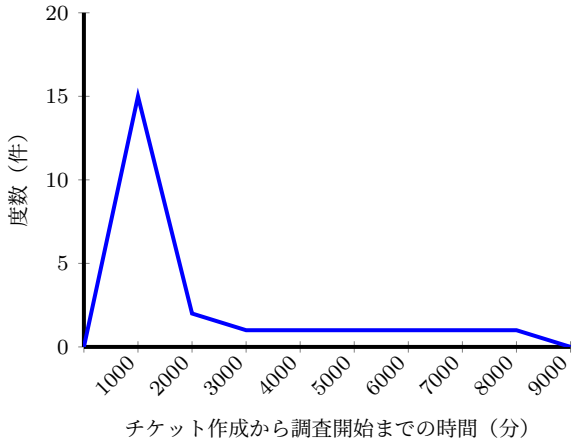

図 9: 2025 年 9 月 1 日から 2025 年 12 月 1 日までの間に作成された，調査が開始されているチケット 23 件の，提案を適用する前のチケット作成から調査開始までの時間の分布

トは 15 件，1000 分以上 2000 分未満のチケットは 2 件，2000 分以上 3000 分未満のチケットは 1 件，3000 分以上 4000 分未満のチケットは 1 件，4000 分以上 5000 分未満のチケットは 1 件，5000 分以上 6000 分未満のチケットは 1 件，6000 分以上 7000 分未満のチケットは 1 件，7000 分以上 8000 分未満のチケットは 1 件，8000 分以上 9000 分未満のチケットは 0 件である．

23 件の調査が完了または途中のチケットのうち，2 件のチケットでチケットの作成から調査の開始までに時間がかかった理由を聞くことができた．チケットの作成から調査の開始までにかかった時間が 4740 分である「Internal Monitoring Cluster Pod Memory Usage High」は，2025 年 9 月 30 日 6 時 24 分に発行され，2025 年 10 月 3 日 13 時 24 分に有田 海斗によって調査が開始された．チケットの作成から調査の開始までにかかった時間が 4740 分かかった理由は，2025 年 9 月 30 日 6 時 24 分はチケットの担当者が割り当てられる時間ではないため，チケット担当者が指定されずにチケットが作成された．チケットが作成される際に平日の 9 時 00 分から 18 時 00 分の間に監視作業のシフトに入っている学生が担当者として割り振られる．それ以外の時間に登録されたチケットは，担当者が未割り当ての状態で登録される．未割り当てのチケットは，手動でチケットの割り振りを行う．割り振りを行った後，Slack でチケットの割り振りが変わったことを監視作業者に報告する．2025 年 9 月 30 日 6 時 24 分に登録されたチケットは担当者の割り振りを忘れており，担当者が未割り当ての状態が 3 日以上放置されていた．その後，このチケットの調査を大学院生である平尾 真斗が有田 海斗に Slack で調査を依頼したため，チケットの作成から調査の開始までに 4740 分が経過した．チケットの作成から調査の開始までにかかった時間が 1539 分である「Internal Monitoring Cluster Node OOM Kill」は，Ubuntu 内のプロセスで OOM Killer が発生したことを示すアラートのチケットである．このチケットは 2025 年 10 月 22 日 12 時 37 分に発行され，2025 年 10 月 23 日 14 時 16 分に有田 海斗によって調査が開始された．チケットの作成から調査の開始までにかかった時間が 1539 分かかった理由は 2025 年 10 月 22 日 12 時 37 分時点のチケット担当者である岡田 京太郎が欠席しチケットが調査されておらず，2025 年 10 月 23 日 14 時 10 分に大学院生である平尾 真斗が有田 海斗に Slack で調査を依頼したためである．

図 10 に，2025 年 12 月 11 日から 2025 年 12 月 12 日までの間に作成された，調査が開始されているチケット 2 件の，提案を適用した後のチケット作成から調査開始までの時間の分布を示す．

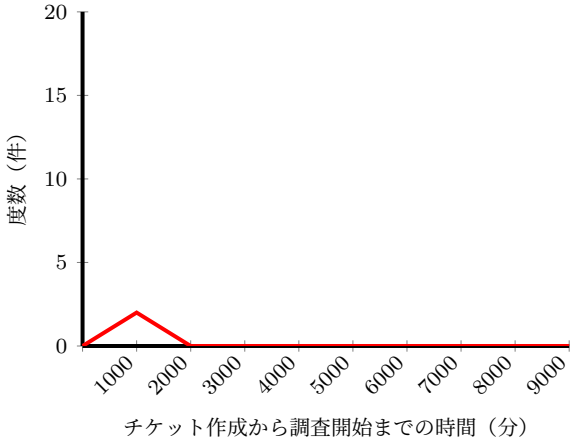

図 10: 2025 年 12 月 11 日から 2025 年 12 月 12 日までの間に作成された，調査が開始されているチケット 2 件の提案を適用した後のチケット作成から調査開始までの時間の分布

提案を適用した後に作成されたチケットは 2 件あった．担当者が調査を行ったときの，チケットの作成から調査の開始までの時間を計測した．チケット作成から調査開始までの時間が 0 分以上 1000 分未満のチケットが 2 件であった．2025 年 12 月 11 日 17 時 09 分に「Internal Host CPU UsageHigh-Usage100%」のチケットが作成された．アラートが通知されたサーバは Makaron である．過去に Makaron を使用した学生は井出 佑，有田 海斗，岡田 京太郎，北川 翔也，坂井 萌桜，佐藤 健斗，手塚 雄星，月森 陽太，ムハンマド アクラム，山﨑 拓海，山崎 雅也であり，今後 Makaron を使用する可能性がある．ただし，それぞれの学生によって Makaron を使用する可能性や頻度は異なる．チケットの担当者は有田 海斗であり，調査を開始した日時は 2025 年 12 月 11 日 17 時 17 分であり，調査を終了した日時は，2025 年 12 月 11 日 18 時 02 分である．チケットが作成されてから調査を開始するまでの時間は約 8 分であり，調査を開始してから終了するまでの時間は約

45 分である．2025 年 12 月 12 日 11 時 27 分に「Internal Host CPU UsageHigh-Usage100%」のチケットが作成された．アラートが通知されたサーバは Makaron である．過去に Makaron を使用した学生は，2025 年 12 月 11 日 17 時 09 分に「Internal Host CPU UsageHigh-Usage100%」のチケットが作成されたときに，今後 Makaron を使用する可能性があると判定された学生と同じである．チケットの担当者は山﨑 拓海であり，調査を開始した日時は 2025 年 12 月 12 日 12 時 05 分であり，調査を終了した日時は，2025 年 12 月 12 日 12 時 35 分である．チケットが作成されてから調査を開始するまでの時間は約 38 分であり，調査を開始してから終了するまでの時間は約 30 分である．

提案を適用する前は，チケット作成から調査開始までの時間が最短で約 1 分，最長で約 4 日 18 時間 27 分であった．提案適用後に調査が開始されたチケット 2 件は，チケット作成から調査開始までの時間が約 8 分と約 38 分，調査開始から調査終了までの時間が約 45 分と約 30 分であった．「卒業課題 II」の進捗が遅れている学生について，その学生が使用したサーバで検出されたアラートに対応するチケット作成から調査開始までの最短の時間は，提案適用前で約 1 分，適用後は約 8 分であり，提案手法なしと比較して約 7 分増加した．また最長の時間は，提案適用前で約 4 日 18 時間 27 分，適用後は約 38 分であり，提案手法なしと比較して約 4 日 17 時間 49 分減少した．

6. 議論

提案では，仮想マシンに Ubuntu をインストールしたサーバの所有者を，仮想マシンに登録されたサーバ名から特定した．仮想マシンのサーバ名の先頭に学籍番号が記載されているが，必ずしもその学籍番号の学生が使用したとは限らない．例えば，Monitoring クラスタのマスターノードのサーバ名は「G2124034-monitoring-master-ml」である．このサーバは，Monitoring クラスタで稼働している Redmine や Grafana を使用するその他の学生も使用した．そのため，学籍番号が G2124034 である学生だけが使用したわけではない．解決策として，仮想マシンにインストールされた Ubuntu を Kubernetes のノードとしている場合，Kubernetes クラスタ上の Pod 内で動作するアプリケーションに学生のユーザアカウントがアクセスしたログがあるとき，そのユーザアカウントの学生は仮想マシンにインストールされた Ubuntu を使用した学生と判定する．

提案では，学生のメンターがテクニカルレポートの各章が未完了であるかを評価し，その結果をもとにチケットを並び替えていたが，この方法では，テクニカルレポートが提出できたかどうかを判定できない．提案を適用した後の順位と 1 つ目の比較対象との順位相関係数が約 0.85 であったのは，学生のメンターによるテクニカルレポートの各章の完了についての判定が必ずしもテクニカルレポートが提出できる状態であると示しているわけではないためである．したがって，学生のメンターがすべての章を完了と判定した学生が使用したサーバで検出されたアラートのチケットは順位が低く設定される．解決策として，テクニカルレポートに，指導教員や学生のメンター，学生が記入したフィードバックを含むコメントがある章を検出した場合は，その章を未完了と判定し，そのテクニカルレポートを執筆している学生が使用したサーバで検出されたアラートに対応するチケットを優先するようにする．

提案では，後期が開始してから一度でもアクセスが行われたサーバは使用したと判定しているが，実際には後期開始以降に一度だけアクセスした後，全く使用していない学生も存在する．また，学生がサーバを作成したが，使用しない場合がある．解決策として，テクニカルレポートの執筆を開始した日時から，チケットが登録され提案ソフトウェアが動作するまでの期間で，サーバを使用したかどうかを判定する．理由は，テクニカルレポートには実験の結果の数値を記述する必要があり，そのためにはサーバやアプリケーションにアクセスして実験結果の数値を取得する必要がある．そのため，テクニカルレポートの執筆期間に使用されていないサーバはテクニカルレポートの執筆とは関係がないと判定できるためである．テクニカルレポートの執筆を開始した日時の取得方法は，テクニカルレポートをアップロードする指定の Google ドライブ上のフォルダ内を検索し，各学生が最初にアップロードしたテクニカルレポートの作成日時を，執筆を開始した日時として取得する．

提案では，学生が使用したサーバで検出されたアラートに対応するチケットが複数件ある場合，チケットが作成された日時が古いチケットから順に対応するが，チケットが作成された日時が古いチケットが必ずしも緊急度が高いとは限らない．解決策として，サーバの配置関係にもとづく優先順位付けを行う．具体的には，Node と Pod で同時にアラートが発生した場合は，Node を優先する．これは，Node は Pod を実行するために必要であり，Node に問題があるとその Node 上で動作するすべての Pod に影響が及ぶためである．

提案では，欠席数が多い学生のサーバで検出されたアラートに対応するチケットを優先して並び替えるようになっている．欠席数が多くテクニカルレポートをアップロードしていない学生は未完了の章の数が 0 個であると判定されるため，サーバを使用していない場合でも優先して並び替える．解決策として，欠席により学生のメンターによる判定がない学生は，テクニカルレポートの未完了の章の数の判定を行う対象から除外する．ただし，1 回だけ欠席しテクニカルレポートをアップロードしていない学生

は，病気や用事があり欠席している可能性があるため，テクニカルレポートの未完了の章の数の判定を行う対象に含める．これにより，実際に進捗が遅れている学生が使用したサーバで検出されたアラートに対応するチケットを優先して調査できるようになる．

提案では，後期の始めにサーバを使用していたが，その後欠席が続き，使用されなくなったサーバを優先して並び替えを行う．使用されなくなったサーバを優先して調査すると，実際に使用されてるサーバの調査が遅れる．解決策として，学生の最新のテクニカルレポートのアップロード日時からチケットが作成された日時までの期間で，学生が使用したと判定したサーバやアプリケーションを優先することで，欠席の状況を反映してチケットを並び替えられるようになる．

7. おわりに

課題は，作成日時が早いチケットから順に調査を開始すると，学生の「卒業課題 II」の進捗が考慮されず，進捗が遅い学生が使用したサーバの調査が遅れる．提案では，学生のメンターが記録した，学生のテクニカルレポートの未完了の章の数を学生ごとに集計する．その後，各学生が使用したサーバで検出されたアラートに対応するチケットを，テクニカルレポートの未完了である章の数が多い学生から順に並び替え，先頭のチケットを担当者に通知する．1 人の学生が使用したサーバで検出されたアラートに対応するチケットが複数ある場合は，作成日時が古いチケットから並べる．評価実験では，2025 年 9 月 1 日から 2025 年 12 月 1 日までの間に作成された 26 件の未着手のチケットを対象に，提案を適用した後のチケットの順位と 2 つの比較対象との順位相関係数を比較した．1 つ目の比較対象は，テクニカルレポートの初版が未提出の学生が使用したサーバで検出されたアラートに対応するチケット 16 件を作成日時が早い順に並べ，その後ろに残りの 10 件を作成日時が早い順に並べた順位である．2 つ目の比較対象は，CDSL の「Cadence」「論文輪講会」「勉強会」の欠席数の合計が多い学生から順に，その学生が使用したサーバで検出されたアラートに対応するチケットを，作成日時が早い順に並べた順位である．提案を適用した後のチケットの順位と 1 つ目の比較対象との順位相関係数は約 0.85 となった．これは，1 つ目の比較対象は初版が未提出の学生が使用したサーバで検出されたアラートに対応するチケットを基準に順位を決定しているのに対し，提案は学生のメンターによる未完了の章の数にもとづいて順位を決定しているためである．提案を適用した後のチケットの順位と 2 つ目の比較対象との順位相関係数は約 0.99 となった．これは，2 つ目の比較対象は欠席数が多い学生が使用したサーバで検出されたアラートに対応するチケットを基準に順位を決定しているのに対し，提案は学生のメンターによる未完了の章の数にもとづいて順位を決定しているためである．提案を適用した後の順位と 2 つの比較対象との順位相関係数から，提案によるチケットの並び替えは，テクニカルレポートの初版が未提出の学生のサーバで検出されたアラートに対応するチケットよりも，欠席が多い学生のサーバで検出されたアラートに対応するチケットを優先する傾向にある．欠席数が多い学生は，研究室でサーバを使用しておらず，今後も使用しない可能性が高い．また提案を適用する前と適用した後で，チケット作成から調査開始までの時間を計測した．提案を適用する前である 2025 年 9 月 1 日から 2025 年 12 月 1 日に調査が開始されたチケット 23 件では，チケット作成から調査開始までの時間は最短で約 1 分，最長で約 4 日 18 時間 27 分であった．提案を適用した後である 2025 年 12 月 11 日から 2025 年 12 月 12 日に調査が開始されたチケット 2 件のうち，いずれのチケットもテクニカルレポートの第 2 版が未提出の学生が使用したサーバで通知されたアラートのチケットであり，チケット作成から調査開始までの時間は約 8 分と約 38 分であった．「卒業課題 II」の進捗が遅れている学生が使用したサーバで検出されたアラートに対応するチケット作成から調査開始までの最短の時間は，提案適用前で約 1 分，適用後は約 8 分であり，提案手法なしと比較して約 7 分増加した．また最長の時間は，提案適用前で約 4 日 18 時間 27 分，適用後は約 38 分であり，提案手法なしと比較して約 4 日 17 時間 49 分減少した．

参考文献

[1] Amanullah Bashir, T. R. S.: Comparative Study on Incident Management, *International Journal of Applied Information Systems*, Vol. 3, No. 2, pp. 21–23 (online), DOI: http:/ijais12-450467 (2012).

[2] Liu, J., He, S., Chen, Z., Li, L., Kang, Y., Zhang, X., He, P., Zhang, H., Lin, Q., Xu, Z., Rajmohan, S., Zhang, D. and Lyu, M. R.: Incident-aware Duplicate Ticket Aggregation for Cloud Systems, *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pp. 2299–2311 (online), DOI: 10.1109/ICSE48619.2023.00193 (2023).

[3] Marcu, P., Grabarnik, G., Luan, L., Rosu, D., Shwartz, L. and Ward, C.: Towards an optimized model of incident ticket correlation, *2009 IFIP/IEEE International Symposium on Integrated Network Management*, pp. 569–576 (online), DOI: 10.1109/INM.2009.5188863 (2009).

[4] Xue, J., Birke, R., Chen, L. Y. and Smirni, E.: Spatial–Temporal Prediction Models for Active Ticket Managing in Data Centers, *IEEE Transactions on Network and Service Management*, Vol. 15, No. 1, pp. 39–52 (online), DOI: 10.1109/TNSM.2018.2794409 (2018).

[5] Vinnakota, K. and Kolla, M.: Creating Effective Alerts for Monitoring Distributed Systems, *International Journal of Computer Trends and Technology*, Vol. 73, pp. 172–178 (online), DOI: 10.14445/22312803/IJCTT-V73I5P122 (2025).

[6] Huang, H., Jennings III, R. B., Ruan, Y., Sahoo, R. K.,

Sahu, S. and Shaikh, A.: PDA: A Tool for Automated Problem Determination., *LISA*, Vol. 7, pp. 1–14 (2007).

[7] SPADACCINI, A. and GULIANI, K.: Being an On-Call Engineer, *Operating Systems and Sysadmin*, p. 43.

[8] Panaite, D. C.: Evaluating the performance of eBPF-based security software in a virtualized 5G cluster, PhD Thesis, Politecnico di Torino (2024).

[9] Sukhija, N. and Bautista, E.: Towards a Framework for Monitoring and Analyzing High Performance Computing Environments Using Kubernetes and Prometheus, *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pp. 257–262 (online), DOI: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00087 (2019).

[10] Pai, K. and Srinivas, B.: Enhanced Visibility for Real-time Monitoring and Alerting in Kubernetes by Integrating Prometheus, Grafana, Loki, and Alerta, *International Journal of Scientific Research in Engineering and Management* (2024).

[11] BRIGUGLIO PELLEGRINO, R. A., Xompero, M., Riva, M., Selmi, C., Urban, C., AZZAROLI, N., OGGIONI, L., SCALERA, M. A., RICCARDI, A., BALESTRA, A. et al.: Be social, be agile: team engagement with Redmine, SPIE, The International Society for Optical Engineering (2022).

[12] Yamaoka, H., Yamamoto, K., Nagai, T. and Masuda, H.: Case Study of Implementing an IT Service Desk Ticketing System at Small Computer Center, *Proceedings of the 2019 ACM SIGUCCS Annual Conference*, SIGUCCS '19, New York, NY, USA, Association for Computing Machinery, p. 140–144 (online), DOI: 10.1145/3347709.3347820 (2019).

[13] German, K. and Ponomareva, O.: An Overview of Container Security in a Kubernetes Cluster, *2023 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT)*, pp. 283–285 (online), DOI: 10.1109/USBEREIT58508.2023.10158865 (2023).

[14] Doan, D. N. and Iuhasz, G.: Tuning Logstash Garbage Collection for High Throughput in a Monitoring Platform, *2016 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pp. 359–365 (online), DOI: 10.1109/SYNASC.2016.063 (2016).

[15] Gupta, S. and Rani, R.: A comparative study of elasticsearch and CouchDB document oriented databases, *2016 International Conference on Inventive Computation Technologies (ICICT)*, Vol. 1, pp. 1–4 (online), DOI: 10.1109/INVENTIVE.2016.7823252 (2016).

[16] Ahmed, F., Jahangir, U., Rahim, H., Ali, K. and Agha, D.-e.-S.: Centralized Log Management Using Elasticsearch, Logstash and Kibana, *2020 International Conference on Information Science and Communication Technology (ICISCT)*, pp. 1–7 (online), DOI: 10.1109/ICISCT49550.2020.9080053 (2020).

[17] Sharma, B. and Nadig, D.: eBPF-Enhanced Complete Observability Solution for Cloud-native Microservices, *ICC 2024 - IEEE International Conference on Communications*, pp. 1980–1985 (online), DOI: 10.1109/ICC51166.2024.10622329 (2024).

[18] Trikusuma Dewo, K., Yasin, V., Budiman, T., Zulkarnain Sianipar, A. and Budi Yulianto, A.: IT Infrastructure Dashboard Monitoring Application Development Using Grafana And Promotheus, a Case Study at Astra Polytechnic School, *2023 International Conference of Computer Science and Information Technology (ICOSNIKOM)*, pp. 1–5 (online), DOI: 10.1109/ICoSNIKOM60230.2023.10364485 (2023).

[19] Roy, S., Muni, D. P., Tack Yan, J.-J. Y., Budhiraja, N. and Ceiler, F.: Clustering and labeling IT maintenance tickets, *International Conference on Service-Oriented Computing*, Springer, pp. 829–845 (2016).

[20] Subbarao, M. V., Venkatarao, K. and Suresh, C.: Automation of incident response and IT ticket management by ML and NLP mechanisms, *Journal of Theoretical and Applied Information Technology*, Vol. 100, No. 12, pp. 3945–3955 (2022).

[21] Miao, G., Moser, L. E., Yan, X., Tao, S., Chen, Y. and Anerousis, N.: Generative models for ticket resolution in expert networks, *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 733–742 (2010).

