

# Zabbixによるブロックチェーン型 NTMobileサーバ統合管理システムの評価

鈴木 洸太<sup>1</sup> 松岡 穂<sup>1</sup> 鈴木 秀和<sup>1</sup> 内藤 克浩<sup>2</sup>

**概要:** IPv4/IPv6 混在環境下で通信接続性と移動透過性を同時に実現する技術として, ブロックチェーン型 NTMobile (Network Traversal with Mobility) が提案されている. NTMobile を実環境下で運用する場合, 運用規模の拡大により監視対象のサーバ台数が増加するため, サーバ管理者の監視負担が増加することが懸念されている. また, NTMobile に基づく通信では, 通信開始時にエンド端末および NTMobile 独自サーバである Direction Coordinator 及び Relay Server などとトンネル構築処理を行う必要があるため, NTMobile 特有の処理にエラーが発生した場合, 構築処理に関わる全端末のログを収集して確認する必要があり, エラー発生の原因究明に多大な時間を要する場合があった. 上記の課題を解決するために, 筆者らはネットワーク機器の統合管理を行う Zabbix を用いた NTMobile サーバ管理システムを提案してきた. 本稿では, 提案システムのプロトタイプを実装し, 実ネットワーク環境において評価実験を行う. その結果, 提案システムを用いることにより, 監視時間を従来手法より約 75%削減できることを確認した.

## An Evaluation of Server Integration Management System Using Zabbix for Blockchain-Based NTMobile

KOTA SUZUKI<sup>1</sup> MINORU MATSUOKA<sup>1</sup> HIDEKAZU SUZUKI<sup>1</sup> KATSUHIRO NAITO<sup>2</sup>

### 1. はじめに

スマートフォンやタブレットなどのモバイル端末や LTE などの無線通信技術の急速な普及により, 移動しながら通信を行う機会が増加している. しかし, 移動により接続先ネットワークが変化した場合, 端末の IP アドレスも共に変化してしまう. そのため, 移動前に開始していた通信フローを識別することができなくなってしまうため, 通信が断絶するという課題が発生する. 移動に伴う通信断絶を回避するためには, 移動透過性を実現する必要がある [1], [2].

また, IPv4 ネットワークにはグローバルアドレス空間とプライベートアドレス空間の間に存在する NAT (Network Address Translation) により, 自由に双方向通信を開始することができない NAT 越え問題が存在する. さらに, IPv4 と IPv6 の間で互換性がないため, 直接通信を行うことも

できない. そのため, モバイル端末の接続先ネットワークの種類や通信相手端末のネットワーク環境に影響されことなく, 通信接続性を確保しつつ, 移動透過性を実現することが要求されている. このような背景から, 筆者らは IPv4/IPv6 混在環境において通信接続性と移動透過性を同時に実現する NTMobile (Network Traversal with Mobility) [3], [4], [5] や, その改良型であるブロックチェーン型 NTMobile[6] を提案してきた.

ブロックチェーン型 NTMobile は, 通信経路指示や端末管理を行う DC (Direction Coordinator) と, 端末同士が直接通信が行えない場合に通信の中継を行う RS (Relay Server) と呼ぶ専用のサーバ群が定義されている. NTMobile を利用する端末の規模に応じて, DC 及び RS は複数台設置することが可能である. つまり, NTMobile を実環境下での運用を想定した場合, 運用規模の拡大に応じて DC 及び RS 稼働数が増加するため, ネットワーク管理者の運用管理負担が増加する懸念がある. また, NTMobile は通信開始時にエンド端末間で暗号化トンネルを動的に

<sup>1</sup> 名城大学大学院理工学研究科  
Graduate School of Science and Technology, Meijo University

<sup>2</sup> 愛知工業大学情報科学部  
Faculty of Information Science, Aichi Institute of Technology

形成し、トンネルのエンドポイントの情報や RS 中継の有無、暗号鍵など、NTMobile における暗号化トンネル通信に関する情報はメモリ上に生成されるトンネルテーブルのみに存在する情報もある。したがって、サーバ管理者がその情報を参照する為にメモリダンプを行って解析する必要があった。このような状況下において、NTMobile 特有の処理を行っている最中に異常が発生し、トンネル構築処理が失敗したり、認証処理が失敗したりすると、エンド端末だけでなく、トンネル構築処理に関わった DC や RS のログやメモリダンプなどの情報を俯瞰して解析する必要があり、エラーの原因究明に時間を要する場合があった。そのため、ブロックチェーン型 NTMobile の運用において低負担かつ高効率なサーバ管理手法は非常に重要である。

これまでに、NTMobile サーバ群を統合監視する手法として、SNMP (Simple Network Management Protocol) を用いた方法が検討されていた [7]。NTMobile 独自の拡張 MIB (Management Information Base) を定義し、各サーバの死活監視や CPU 使用率等の負荷情報を取得する。しかし、この手法は拡張 MIB に基づいた規定的な状態情報しか取得できなかったため、NTMobile の一連の処理が正常に動作していない場合の状況分析や原因の特定に必要な情報を網羅的に収集することができなかった。また、NTMobile 機能を有するエンド端末はホームネットワークや社内 LAN、セルラーネットワークに接続している状況がほとんどであり、エンド端末と DC および RS 間には NAT やファイアウォールが存在するため、環境によって SNMP による情報取得が不可能な場合も考えられる。

このような背景から、筆者らはサーバやネットワーク機器の統合監視ソフトウェアである Zabbix[8] を用いることにより、先行研究の課題を解決するとともに、ブロックチェーン型 NTMobile サーバ群を低負担で状況把握が可能となる統合管理システムを提案している [9]。本稿では、Zabbix に NTMobile 対応させるための拡張機能を追加し、提案システムを Linux 環境上に実現したプロトタイプを用いて、評価実験を行う。障害検知時の対応について、提案システムを利用することにより管理者の対応時間がどの程度改善するかを明らかにする。

以後、2 章でブロックチェーン型 NTMobile の概要と、現状の異常発生時における管理者の対応状況について整理する。3 章で提案システムの概要、4 章で Zabbix で NTMobile 特有の情報を統合監視するための実装とプロトタイプシステムの評価実験と結果について示し、最後に 5 章でまとめる。

## 2. ブロックチェーン型 NTMobile

### 2.1 概要

NTMobile では、ネットワークに依存しない Identifier としての役割を担う仮想 IP アドレスと、接続先ネットワー

クから割り当てられる Locator としての役割を担う実 IP アドレスの両方を利用する。エンド端末のアプリケーション間で交換されるメッセージは仮想 IP アドレスに基づく IP パケットで交換され、これをネットワークでルーティングさせるために実 IP アドレスに基づく UDP/IP ヘッダでカプセル化される。

NTMobile を実装したエンド端末 (以後、NTM 端末) は、起動時に DC から仮想 IP アドレスを割り当てられ、通信開始時に通信相手 NTM 端末との間で暗号鍵の交換や UDP トンネルを動的に構築し、仮想 IP アドレスに基づく IP パケットを UDP トンネルを通じて通信相手 NTM 端末へ伝送する。これにより、ネットワークの切り替えや通信経路上にある NAT の影響を受けない通信が可能である。また、ハンドオーバーにより NTM 端末の接続先ネットワークが変化した場合でも、仮想 IP アドレスは変化しないため通信が断絶しない。ブロックチェーン型 NTMobile は、従来の NTMobile にブロックチェーンの技術を導入し、耐障害性の向上及び認証処理の簡略化を図った改良型である。

### 2.2 構成

図 1 にブロックチェーン型 NTMobile の概要を示す。ブロックチェーン型 NTMobile は NTM 端末の他に、DC 及び RS の独自サーバから構成され、IPv4/IPv6 のデュアルスタックネットワーク上に分散配置される。

DC は NTM 端末の認証及び管理を行う。各 DC の管理下にある NTM 端末実際の IP アドレスと仮想 IP アドレスを対応関係を管理し、NTM 端末の通信開始時にトンネル構築の指示を出す。また、各 DC はブロックチェーンネットワークに接続され、ログインしている NTM 端末の情報を分散管理する。

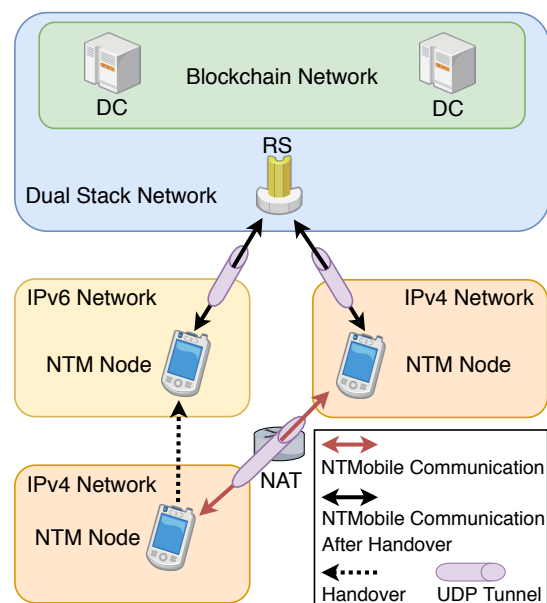


図 1 ブロックチェーン型 NTMobile の概要

RS は NTM 端末同士が異なる NAT 配下にある場合や、IPv4/IPv6 間の互換性がないネットワークで通信する場合のようにエンドツーエンドで通信を行うことができない場合に通信の中継を行う。

### 2.3 通信シーケンス

NTM 端末は通信開始時に、自身の実 IP アドレスや公開鍵情報を DC に登録するログイン処理を行う。ログイン処理後に NTM 端末は、仮想 IP アドレスを DC より割り当てられる。DC はログインした NTM 端末の仮想 IP アドレス、NTM 端末の公開鍵情報、ログイン先 DC の情報をブロックチェーンに追加する。通信開始時において NTM 端末は、通信相手と通信を開始する要求である Direction Request を登録先 DC へ送信する。その後、DC は Direction Request の内容を基にブロックチェーンから通信相手のログイン先 DC を特定し、トンネル構築を行うシグナリング処理を実施する。ただし、NTM 端末同士が直接相互通信できないと DC が判断した場合は、RS を経由する経路でトンネル構築を行う。RS は、メモリ上にリレーテーブルを保持しており、構築したトンネルの PathID や実 IP アドレスを対応付けた情報を登録する。RS は、リレーテーブルに基づいて通信の中継を行う。

### 2.4 異常発生時における原因究明作業とその課題

ブロックチェーン型 NTMobile において異常が発生した場合、原因となるサーバにリモートアクセスを行いログファイルを調査する必要がある。特にトンネル構築処理は、DC、RS が関与するため、トンネル構築処理時に異常が発生した場合、DC、RS が出力するログファイルを付き合わせて調査する必要がある。

また、通信経路の追跡を行う際においてリレーテーブルは有用な情報である。しかし、リレーテーブルはメモリ上にのみ存在するため、サーバ管理者が参照するためにメモリダンプの情報を解析する必要がある。

更に、ログイン中の NTM 端末増加や RS が構築するトンネル数の増加により処理遅延の発生やトンネルの新規構築が行えない状況の発生が考えられる。これらの状況を検知する手法は検討されておらず、サーバ管理者が問題の発生に気づかない場合も考えられる。

以上より、ブロックチェーン型 NTMobile に関する一連の処理において異常が発生した場合の状況分析、原因の特定に時間を要する懸念がある。

## 3. 提案システム

NTMobile の内部情報可視化やサーバ群の状態を統合管理するために、Zabbix を導入する。Zabbix はネットワーク機器の統合管理を行うオープンソースで提供されているソフトウェアであり、企業や団体が組織内システムの監視

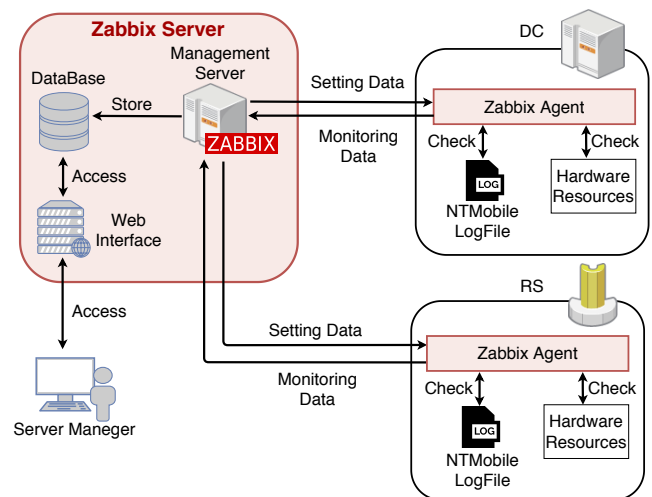


図 2 Zabbix を用いた統合管理システムの概要

や監視情報の収集システムとして利用されている [10], [11]. Zabbix が通常取得可能な情報である CPU やメモリの使用率等に加え、RS が持つリレーテーブル、DC の管理下にある NTM 端末の台数取得機能を追加実装する。

### 3.1 システム構成

図 2 にブロックチェーン型 NTMobile サーバ群を一元管理するシステムの構成を示す。監視対象である DC 及び RS に監視情報を取得するデーモンプロセスの Zabbix Agent を導入する。また、Zabbix Agent が取得した監視情報を回収し、それらを統括する Zabbix サーバを新規に設置する。

Zabbix サーバは監視情報を回収する管理サーバ、監視情報を保存するデータベースの他に、Web インターフェイスが組み込まれており、サーバ状況をグラフィカルに表示することが可能である。NTMobile サーバ群の管理者は Web ブラウザを利用して Zabbix Server にアクセスすることにより、Zabbix Agent 経由で取得した DC および RS の様々な情報をワンストップで閲覧することができる。

### 3.2 障害検知

NTMobile サーバ群で発生した障害の検知には、NTMobile が出力するログの監視及びプロセス死活監視により行う。ログファイルの内容検知には Zabbix トリガーを利用する。図 3 に監視情報の流れと Zabbix トリガーの動作を示す。Zabbix トリガーは、事前に条件式及び動作アクションを定義し、条件式が真となる監視情報を Zabbix が取得した場合、指定した動作アクションを実行することができる。本管理システムでは、NTMobile が出力したログファイルを Zabbix サーバが回収し、その内容に基づいた障害情報を Zabbix 上に表示するよう Zabbix トリガーを定義する。また、ログ取得によりネットワーク帯域を圧迫しないように、NTMobile が出力するログレベルごとに監視周期

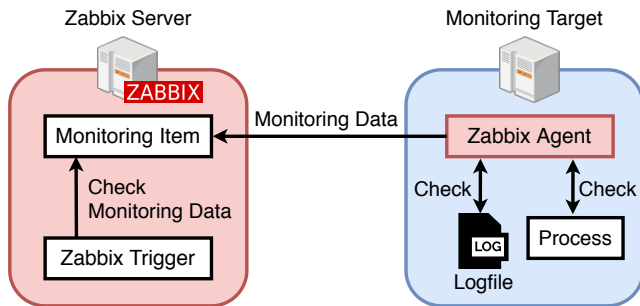


図 3 監視情報の流れと Zabbix トリガーの動作概要

を変化させ、システムエラー等重要なログは監視周期を短く、NTMobile 処理の動作結果等重要ではないものは監視周期を長く設定する。

更に、監視対象上で DC と RS で稼働する NTMobile プロセスを死活監視することにより、NTMobile のシステムダウンを検知する。

以上により、サーバ管理者は監視対象における NTMobile の稼働状況を統合監視することが可能である。

### 3.3 リレーテーブル可視化

本管理システムでは、Zabbix が提供する機能であるスクリーンを用いてリレーテーブルの可視化を行う。

図 4 にリレーテーブル可視化におけるシーケンスを示す。Zabbix が提供するスクリーン機能は、Zabbix が取得した監視情報を基に監視対象の状況を一覧表示する機能である。スクリーンは、各種監視情報の他に HTTP を用いて取得した情報の表示が可能である。RS にプロセス間通信を行う機能、プロセス間通信で受信したリレーテーブルの情報を HTML ページで表示するリレーテーブル可視化プロセスを追加実装しスクリーンを用いて Zabbix 上で可視化を行う。リレーテーブル可視化プロセスは、RS プロセスとプロセス間通信を行いリレーテーブルの情報を取得する。可視化プロセスは受信した情報を元に HTML ページを作成し、Zabbix のスクリーンからアクセスできるよう展開する。

これにより、RS 群が持つリレーテーブルを容易に監視することが可能となる。

### 3.4 負荷検知

DC は NTM 端末の増加に伴う処理遅延の発生が予測される。また、RS はトンネル作成時に UDP ポート番号をエフェメラルポートの範囲 [12] で割り当てるため、作成できるトンネル数は制限される。Zabbix では、上記の内容は監視することができないため、NTMobile 特有の負荷状態を見落とす懸念がある。本管理システムでは Zabbix に標準で用意されている CPU やメモリ使用率等の監視項目に加え、割り当可能なエフェメラルポートの残数、DC が管理している NTM 端末の台数を取得する NTMobile に関する

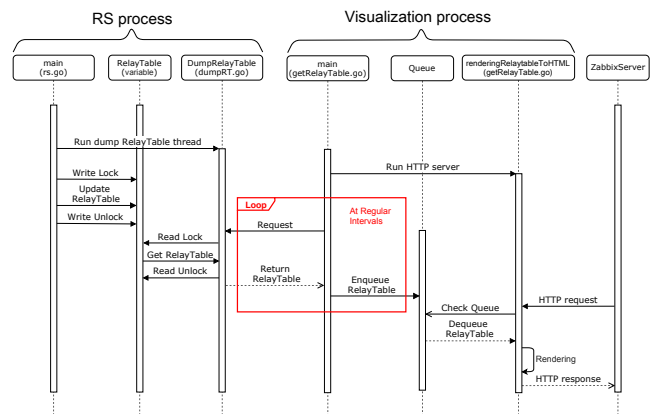


図 4 リレーテーブル可視化シーケンス

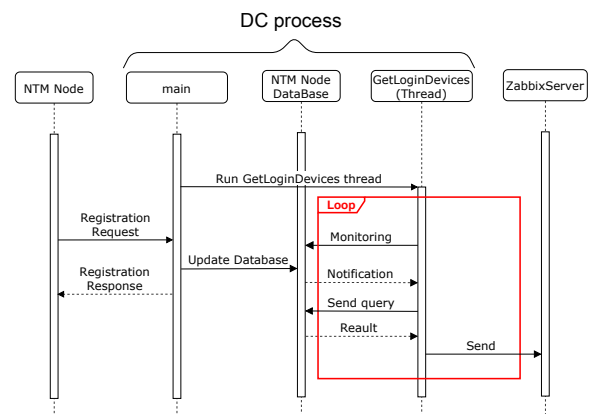


図 5 ログイン中 NTM 端末台数取得シーケンス

監視項目を Zabbix の UserParameter を用いて実装する。UserParameter は、標準で提供されていない監視や特定のアプリケーションが出力する情報を監視項目として定義する機能である。

図 5 にリレーテーブル可視化におけるシーケンスを示す。DC はシングナリング処理を行うためにログインしている NTM 端末を管理するデータベースを持つ。このデータベースにクエリを送信し、ログイン中の NTM 端末台数を取得する機能を追加実装する。

また、RS で割り当可能な UDP ポート番号残数はサーバのリソース情報から取得する。一般的な Linux システムの場合、`/proc/net/udp`、エフェメラルポート範囲の設定情報 (`/proc/sys/net/ipv4/ip_local_port_range`) が格納されている。この 2 つのファイルを読み込み、割り当可能な UDP ポート番号残数を計算する機能を RS に追加実装する。

追加実装により取得した NTM 端末ログイン台数、割り当可能な UDP ポート番号残数は、Zabbix で監視できるよう UserParameter で監視項目をそれぞれ定義する。以上により、NTMobile に関する高負荷状態の検知も可能になる。

表 1 SNMP を用いた管理手法と提案手法の比較

	SNMP を用いた手法	提案手法
統合監視	○	○
一般的な障害検知	○	○
監視情報取得の柔軟性	△	○
NTMobile の障害検知	×	○
内部情報可視化の容易さ	×	○
NTMobile の高負荷状態検知	×	○

## 4. 実装・評価

### 4.1 実装

本管理システムにおいて監視に用いる追加機能は Go 言語を用いて実装した。プロセス間通信は、Go 言語の net パッケージを用いて作成した Unix Domain Socket を用いて行っている。また、リレーテーブル可視化時に利用する HTML ページは、Gin パッケージ [13] が提供する Web フレームワークを用いて実装した。

VirtualBox 上で Ubuntu 18.04 を用意し、Zabbix 4.2 を実行した。また、DC 及び RS は、それぞれ VirtualBox を用いて起動した Ubuntu 18.04 上で実行し、それぞれに Zabbix Agent を導入しローカルネットワーク環境において動作検証および評価実験を行った。

図 6 に Zabbix 上における NTMobile サーバ群の監視画面を示す。その結果、今回実装した監視項目の取得及び障害検知を行えることを確認した。また、検知した障害情報に色をつけて表示することによる強調表示が可能であり、訴求効果の向上が可能である。

### 4.2 定性評価

本稿で提案する管理手法と SNMP を用いた管理手法について定性評価を行った。表 1 に提案手法と SNMP を用いた管理手法の比較表を示す。

両手法ともに、NTMobile サーバ群の CPU やメモリ使用率等の一般的な監視情報を統合的に監視することが可能であり、サーバ群の死活監視を行うことができる。SNMP を用いた手法では規定的な情報しか取得できないという課題に対して、提案手法では Zabbix を用いることによりサーバ管理者は必要に応じて監視項目の変更が可能であるため、監視項目指定の柔軟性が向上している。障害検知に関して、SNMP を用いた手法では NTMobile の処理に関する障害は検知を行うことができず、サーバ管理者が対象サーバに直接アクセスしログ調査を行う必要があった。一方、提案手法では NTMobile が出力するログファイルの取得及び内容検知を自動で行うため、サーバ管理者は低負担で障害状況を把握することができる。内部情報可視化に関して、SNMP を用いた手法ではリレーテーブル可視化手法は検討されておらず、サーバ管理者はメモリダンプを行う必要があった。提案手法では、可視化機能の追加実装により、Zabbix 上でリレーテーブルの表示が可能であり、サーバ管理者がメモリダンプを行う必要がない。負荷状態の検知に関しては、NTMobile の性質に則して負荷状況の検知が行えるため、従来手法よりも過負荷防止の信頼性向上が期待できる。

### 4.3 定量評価

サーバ管理者の負担を評価するために、従来手法及び提案手法で障害調査時間の計測及び比較を行う。ブロックチェーン型 NTMobile の基礎的な知識を持つ被験者 3 名に対して、NTMobile の障害調査を行って頂き、各処理ごとの障害調査時間を計測した。なお、サーバの障害により NTMobile の一連の処理が行えない状況を再現し、被験者にはリレーテーブルの調査、DC 及び RS のプロセス稼働状況の確認、稼働していない場合に原因となるエラーログの調査を行わせた。

図 7 に従来手法における各処理ごとの調査時間の積立グラフを示す。SSH ログイン時間、リレーテーブル調査時間、プロセス確認時間、ログ調査の平均時間はそれぞれ 27.1 秒、175.7 秒、20.3 秒、77.1 秒であった。特に SSH ログイン、ログ調査について、前者は 10.75 秒～57.4 秒、後者は 39.7 秒～101.1 秒のように時間にばらつきが生じた。これは、調査を行うためにサーバ管理者がコマンドを入力する必要があるため、タイピングの速度やコマンドライン操作の熟練度の違いにより差が生じたと考えている。また、リレーテーブル調査に関して、メモリダンプの取得に平均 111.8 秒の時間を費やしている。これは、全調査時間のうち 37.2% をメモリダンプの取得に費やしていたことになる。

次に図 8 に提案手法における各処理ごとの調査時間の積立グラフを示す。Zabbix ログイン時間、リレーテーブル調査時間、プロセス確認時間、ログ調査の平均時間はそれぞれ 23.7 秒、21.2 秒、10.3 秒、16.0 秒であった。提案手法では、主にマウス操作による調査になるため、タイピング速度の違いに起因するような調査時間のばらつきは生じなかった。

提案手法を用いることにより調査時間の総計は従来手法よりも約 75 % の時間削減を確認した。主な要因として、ログなどの監視情報取得や検知は Zabbix が自動で行うため、サーバ管理者は Zabbix にアクセスのみ行うことで NTMobile サーバ群の状況を把握することができるためと考える。特に、リレーテーブル及びログの調査において負担軽減の効果が大きいことを確認した。

提案手法ではメモリダンプを取得する必要がないため、調査時間の大幅な削減が可能である。



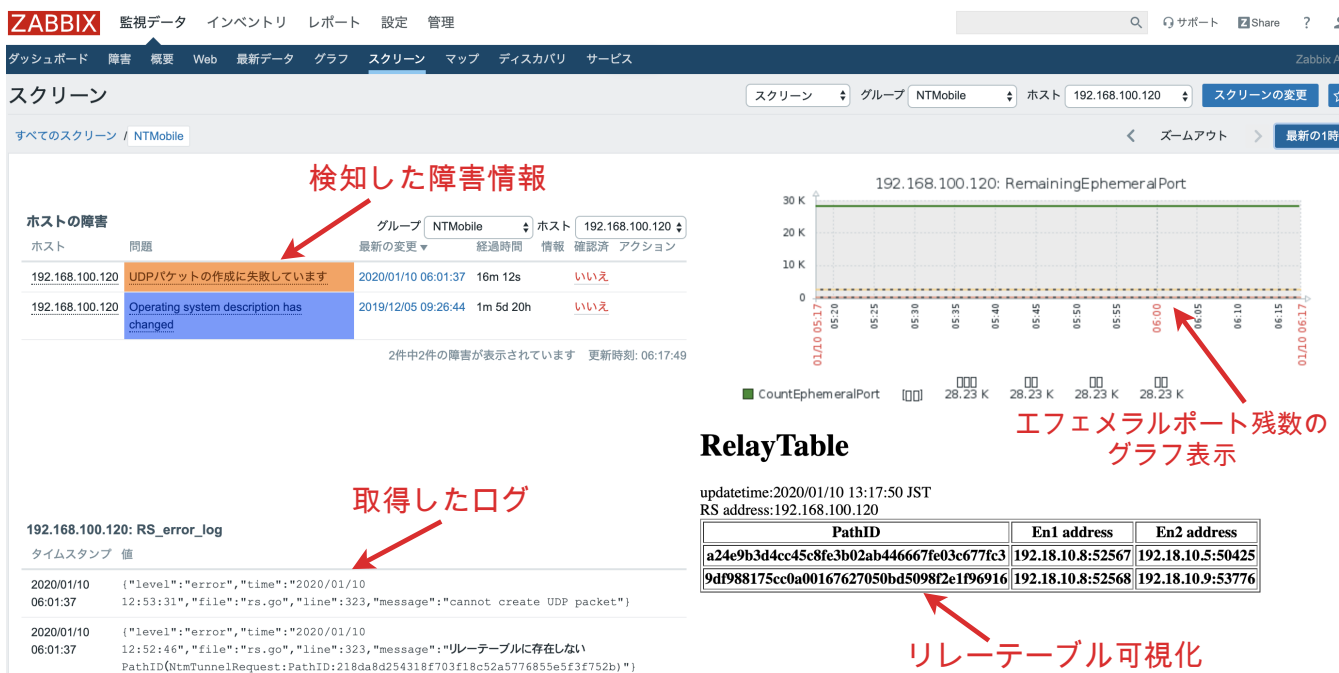


図 6 Zabbix による NTMobile サーバ群統合監視画面

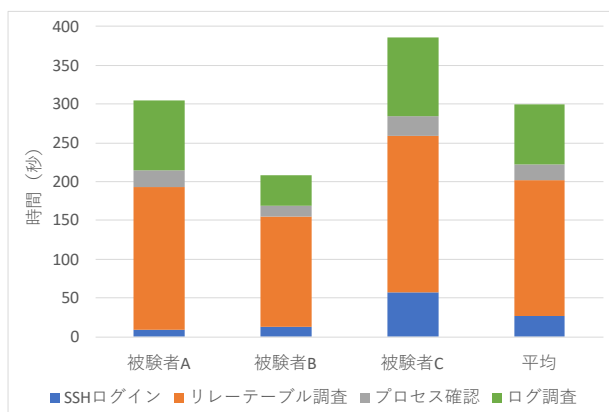


図 7 従来手法における各処理ごとの調査時間

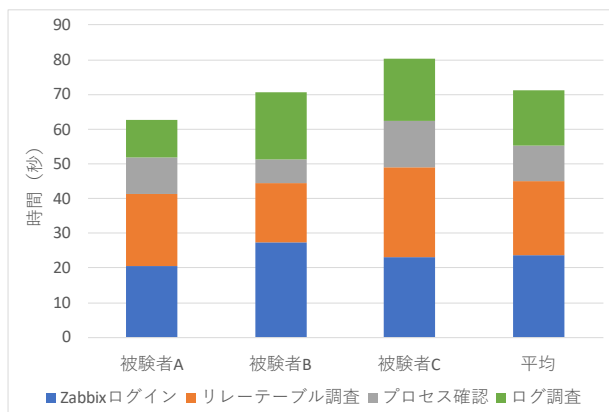


図 8 提案手法における各処理ごとの調査時間

## 5. まとめ

本稿ではブロックチェーン型 NTMobile サーバ群を Zab-

bix を用いて効率的に統合管理するシステムについて評価を行った。本管理システムでは、Zabbix に追加実装した機能を組み合わせることにより、従来では困難であったリレーテーブルの可視化や、NTMobile 特有の高負荷状態検知を実現した。更にログファイルの内容検知によりサーバ管理者は直接ログファイルを調査することなく障害状況の把握が可能である。評価実験により、本管理システムを用いることで障害調査に費やす時間を約 75%低減することが可能であり、提案システムが有用であることを確認した。

**謝辞** 本研究の一部は名城大学学術研究奨励助成制度（基礎的研究促進事業費）の助成を受けたものである。

## 参考文献

- [1] Le, D., Fu, X. and Hogrefe, D.: A review of mobility support paradigms for the internet, *IEEE Communications Surveys Tutorials*, Vol. 8, No. 1, pp. 38–51 (2006).
- [2] C.Perkins: IP Mobility Support for IPv4, Revised, RFC 5944, IETF (2010).
- [3] 鈴木秀和, 上醉尾一真, 水谷智大, 西尾拓也, 内藤克浩, 渡邊 晃: NTMobile における通信接続性の確立手法と実装, 情報処理学会論文誌, Vol. 54, No. 1, pp. 367–379 (2013).
- [4] 内藤克浩, 上醉尾一真, 西尾拓也, 水谷智大, 鈴木秀和, 渡邊 晃, 森香津夫, 小林英雄: NTMobile における移動透過性の実現と実装, 情報処理学会論文誌, Vol. 54, No. 1, pp. 380–393 (2013).
- [5] 上醉尾一真, 鈴木秀和, 内藤克浩, 渡邊 晃: IPv4/IPv6 混在環境で移動透過性を実現する NTMobile の実装と評価, 情報処理学会論文誌, Vol. 54, No. 10, pp. 2288–2299 (2013).
- [6] 木村信裕, 柳瀬知広, 田中久順, 鈴木秀和, 内藤克浩, 渡邊 晃: ブロックチェーン上のスマートコントラクトを利用した NTMobile システムの基礎的検証, 第 81 回情報処理学会全国大会講演論文集, Vol. 2019, No. 1, pp. 65–66

- (2019).
- [7] 田内千裕, 上醉尾一真, 鈴木秀和, 内藤克浩, 渡邊晃: NTMobile におけるサーバ群の統合的管理システムの提案, 第 76 回全国大会講演論文集, Vol. 2014, No. 1, pp. 263–264 (2014).
  - [8] Zabbix LLC.: Zabbix :: The Enterprise-Class Open Source Network Monitoring Solution, <https://www.zabbix.com>. (2019/05/18 参照).
  - [9] 鈴木洸太, 松岡穂, 柳瀬知広, 鈴木秀和, 内藤克浩, 渡邊晃: Zabbix によるブロックチェーン型 NTMobile サーバ統合管理システムの提案, Vol. 2020, No. 1, pp. 93–94 (2020).
  - [10] 竹内純人: OSS を活用したグラフィカルなセンサデータ監視システムの構築, 電気通信大学紀要, Vol. 31, No. 1, pp. 104–113 (2019).
  - [11] Gustafsson, J., Fredriksson, S., Nilsson-Mäki, M., Olsson, D., Sarkinen, J., Niska, H., Seyvet, N., Minde, T. B. and Summers, J.: A Demonstration of Monitoring and Measuring Data Centers for Energy Efficiency Using Opensource Tools, p. 506512 (online), DOI: 10.1145/3208903.3213522 (2018).
  - [12] NcFTP: The Ephemeral Port Range, [https://www.ncftp.com/ncftpd/doc/misc/ephemeral\\_ports.html](https://www.ncftp.com/ncftpd/doc/misc/ephemeral_ports.html). (2019/11/07 参照).
  - [13] Gin-Gonic: Gin Web Framework, <https://gin-gonic.com/>. (2019/11/17 参照).