TXT レコードを用いた仮想マシンが登録されているハイパーバイザー特定時間の短縮

鈴木 友也[1]　高橋 風太[2]　串田 高幸[1]

概要：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では，学生が自由に使用できるハイパーバイザーが 6 台用意されている．仮想マシンの作成は自由であるため，仮想マシンの情報が作成者本人以外に共有されていない．課題は，対象の仮想マシンがどのハイパーバイザーに登録されているかわからないことである．提案方式として，すべてのハイパーバイザーから仮想マシンの情報を取得し，データベースに登録し，DNS の TXT レコードに仮想マシンのデータベースから取得したレコードの ID と仮想マシンのホスト名を登録する．基礎実験として 9 人がハイパーバイザーのダッシュボードにアクセスし，対象の仮想マシンを見つけるまでの時間とアクセスしたハイパーバイザーの数を計測した．対象の仮想マシンを見つけるまでのハイパーバイザーへのアクセス回数は最も少なくて 2 回，最も多くて 9 回だった．また，対象の仮想マシンを見つけるまでの所要時間は，最も短くて約 62 秒，最も長くて約 322 秒だった．評価方法として，ハイパーバイザーのダッシュボードにアクセスし，提案方式を用いる場合と用いない場合の対象の仮想マシンを見つけるまでにかかる時間を比較する．

1. はじめに

背景

ハイパーバイザーを利用することで，単一の物理コンピュータ上で複数の独立した OS を同時に実行することができる [1]．ハイパーバイザーによって作成された仮想化したコンピューターを仮想マシンという．ハイパーバイザーには，2 つのタイプがある [2]．タイプ 1 のハイパーバイザーはその下に OS を持たず，代わりに仮想マシン間でシステムリソースのスケジューリングと割り当てを行う [3, 4]．タイプ 2 のハイパーバイザーは，ホスト OS が I/O デバイスのサポートやメモリ管理のサービスを提供する [4, 5]．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（Cloud and Distributed Systems Laboratory，以後 CDSL とする）には，学生が自由に使用できるハイパーバイザーが 6 台用意されている．使われているのは，VMware ESXi であり，タイプ 1 のハイパーバイザーである [6, 7]．仮想マシンの作成は自由であるため，仮想マシンの情報が作成者本人以外に共有されていない．

Domain Name System（以後 DNS とする）は，ドメイン名を対応するマシンの IP アドレスにマッピングすることを主な機能としている [8–10]．DNS は，クエリを受け取るとデータベースを参照し，レコード情報を返す．一般的なレコードタイプに，A レコード，CNAME レコード，MX レコード，NS レコード，PTR レコードがある [11]．また，ドメイン名と文字列をマッピング TXT レコードもある [12]．TXT レコードの使用用途に，Sender Policy Framework（以後 SPF とする）がある．SPF は，新規登録されたドメインの DNS トラフィックと TXT レコードの内容，特にドメイン名のなりすまし防止プロトコルである [13–15]．

DNS では，再帰バグ，ゼロ応答バグ，サーバー障害検出バグ，再送信タイマーの不具合が確認されている [16]．仮想マシンは，新しく作成したとき，再起動したときに IP アドレスが Dynamic Host Configuration Protocol（以後 DHCP とする）によって割り当てられる．DHCP によって IP アドレスが割り当てられると，Dynamic DNS（以後 DDNS とする）によって DNS に登録される．仮想マシンに IP アドレスが割り当てられない，名前解決ができないという不具合が起きる．図 1 に不具合が疑われる際の対応を示す．ユーザーが仮想マシンに不具合を確認したら，CDSL の DNS の管理者は，その不具合がどこにあるか，本当に不具合が起きているか，確認する．図 1 では，ユー

[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

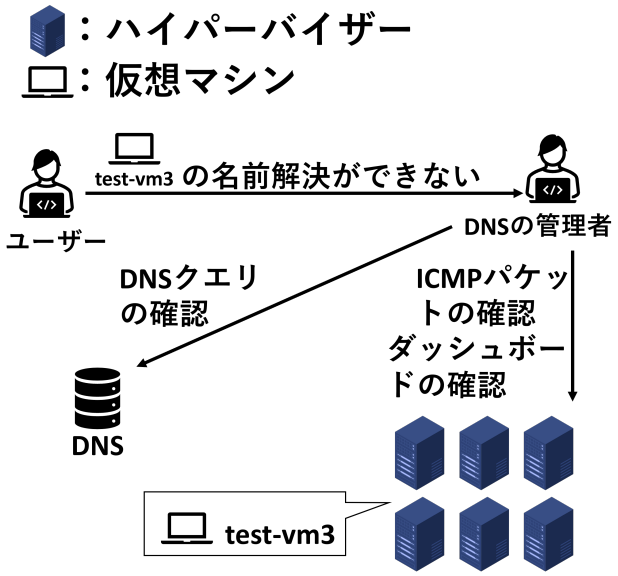

図 1　不具合が疑われる際の対応

ザーが test-vm3 という名前の仮想マシンの名前解決ができないことを確認したため，DNS の管理者に不具合の確認を依頼している．DNS の管理者は不具合の確認として，DNS に対する DNS クエリの送信，対象の仮想マシンに対する ICMP パケットの送信，ハイパーバイザーのダッシュボードへのアクセスを行う．

課題

課題は，対象の仮想マシンがどのハイパーバイザーに登録されているかわからないことである．図 2 に課題の概要を示す．不具合が疑われる際に，DNS の管理者は，ハイパーバイザーのダッシュボードにアクセスし，仮想マシンの状態を確認する．CDSL では，ハイパーバイザーを 6 台使用している．仮想マシンの作成は自由であるため，仮想マシンの情報が作成者本人以外に共有されていない．DNS の管理者は，不具合が疑われている仮想マシンが作成されているハイパーバイザーのダッシュボードにアクセスするために，アクセスする必要のないハイパーバイザーにアクセスすることになる．

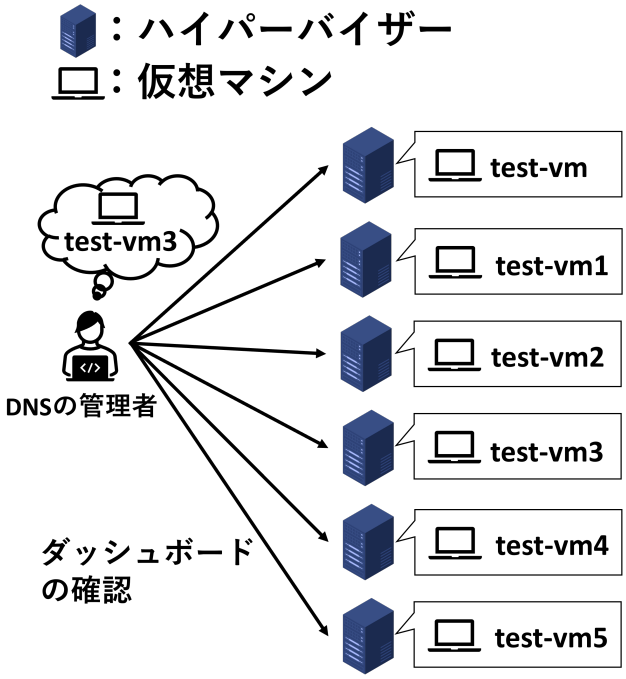

図 2　課題の概要

各章の概要

第 2 章の関連研究では，関連する既存研究を述べる．第 3 章の提案では，課題を解決する提案方式，ユースケース・シナリオの説明をする．第 4 章の実装では，実装方法の説明する．第 5 章の実験では，実験環境，実験結果の分析の説明をする．第 6 章の議論では，提案方式の議論をする．第 7 章の結論では，全体をまとめる．

2.　関連研究

TXT レコードのクエリを分析し，ボットネット通信を検出する方法について述べた研究がある [17]．この研究では，約 550 万の TXT レコードのクエリを 3 か月以上にわたって取得し，分析している．無料のサードパーティセキュリティチェック Web サイトである「Virustotal.com」を使用し，TXT レコードの目的が未確認のクエリを分析した．その結果，悪意のある通信に関与している可能性のある一意の宛先 IP アドレスを 330 件検出した．ボットネット通信を検出するには，人間のオペレーターが「未確認」の TXT レコードのクエリをチェックする必要がある．この研究の結果では，1 日に約 13.2 個の IP アドレスをチェックする必要があるが，これは多くの組織にとって許容できる量である．この研究は，TXT レコードのセキュリティに関して述べている．TXT レコードの新しい使い方や，仮想マシンの管理については述べていない．

セキュリティへの影響に焦点を当てた，TXT レコードの構造について述べた研究がある [18]．この研究では，Open-INTEL を使用し，データセットを分析している．TXT レコードの構造化されていない部分を調査し，セキュリティに影響を与える可能性のある TXT レコードの使用法を明らかにしている．1 文字の TXT レコードが「CONFLUENCE-NETWORK-INC」(AS 40034) のネットワークから発信されていることを示されいる．このネットワークはマルウェアを拡散することで悪名高く，その IP アドレスの多くは多数のブラックリストに掲載されている．また，セキュリティ上問題のあるミスをしている TXT レコードも確認されている．秘密鍵が登録されている TXT レコードが存在した．悪意のあるコマンドを含んでいる TXT レコードも確認された．この研究では，悪意のある TXT レコードの存在を分析しているが，TXT レコードの新しい使い方や，

仮想マシンの管理については述べていない．

ネットワーク運用サービスの中断を引き起こす可能性のある問題である，人為的な設定ミスによる DNS レコードと IP アドレスの競合の両方を回避するためのプロビジョニングメカニズムを提案している研究がある [19]．仮想マシンはサーバーとして運用され，静的に IP アドレスが与得られることを想定している．一意の名前レコードと一意のネットワークアドレスを用いて，一意の識別子を仮想マシンに与える．一意の識別子をもとに，仮想マシンの作成時に DNS サーバーに接続し，DNS レコードを自動的に更新する．データルームまたはクラウドアーキテクチャに，数千の管理対象のサーバーとなる仮想マシンがあることをユースケースとしている．このような量の仮想マシンを人間が介入して管理すると，エラーや大幅な遅延が発生する．この研究の提案では，各仮想マシンの DNS レコードを手動で管理する必要がなくなり，DNS レコードの重複やエラーがなくなる．しかし，DNS レコードのみで管理しているため，仮想マシンに関する他の情報を登録することができない．

統合された情報通信技術環境向けの統合サーバーネットワークリソース管理を提案している研究がある [20]．一時的なネットワーク情報を活用して仮想マシンを移行し，結果として生じるトラフィックダイナミクスのネットワーク全体の通信コストを最小限に抑える，仮想マシン管理用のソフトウェア定義ネットワークベースのオーケストレーションフレームワークを紹介している．特にコストが高く輻輳が発生しやすいデータセンターの集約層とコア層で，ネットワーク全体の通信コストを削減することが示されている．その結果，実験では輻輳が大幅に軽減され，全体のスループットが 6 倍以上増加し，仮想マシンの 50 %未満を移行することで 70 %以上のコスト削減が実現している．仮想マシンを移行するため，移行できない環境では使うことができない．

3. 提案

提案方式

提案の概要を図 3 に示す．ハイパーバイザーから仮想マシンの名前，ホスト名，IP アドレスを取得する．データベースに，仮想マシンの名前，ホスト名，IP アドレス，UUID，作成者，ハイパーバイザーの名前を登録する．UUID は，ハイパーバイザーの IP アドレスと Vmid を組み合わせて決定する．仮想マシンの名前には，作成者の学籍番号必ず入っている．そのため作成者は，仮想マシンの名前から判断する．DNS には，TXT レコードとして UUID とハイパーバイザーの名前を登録する．DNS の管理者は，DNS クエリを送り，A レコードと TXT レコードを受け取る．

図 4 に UUID の取得方法を示す．UUID は，ハイパーバイザーの IP アドレスの第 4 オクテットと Vmid を組み合わせて決定する．図 4 では，ハイパーバイザーの IP アドレスは 192.168.100.22，Vmid は 15 となっているため，UUID は 2215 となる．

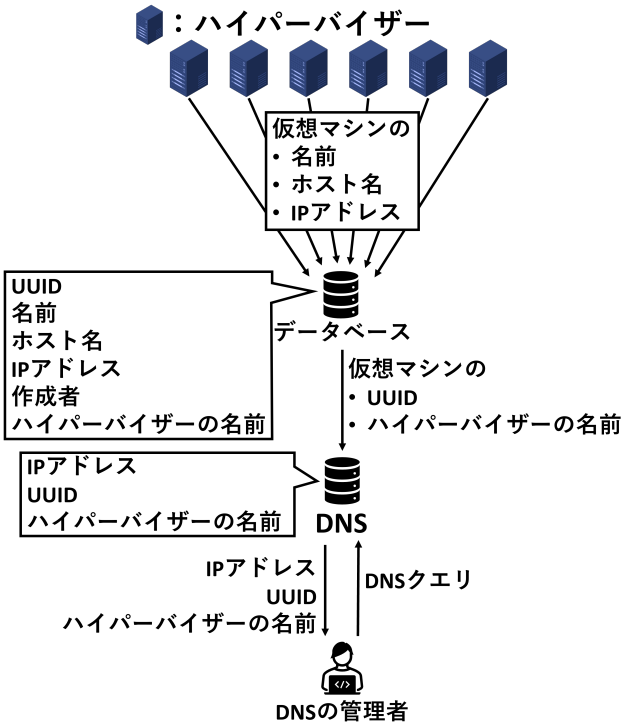

図 3　提案の概要

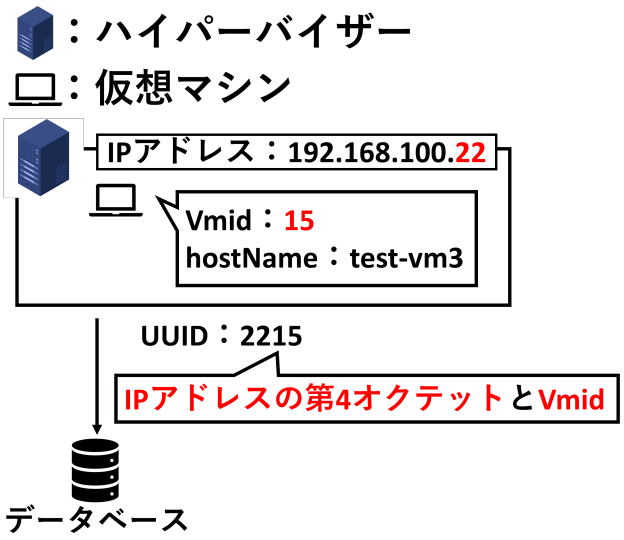

図 4　UUID の取得方法

ユースケース・シナリオ

CDSL で DNS の管理者が仮想マシンに不具合の確認を依頼される場面をユースケースとする．不具合の確認の依頼を図 5 に示す．CDSL の学生をユーザーとしている．ユーザーが仮想マシンに不具合を感じ，DNS の管理者に不具合の確認を依頼する．

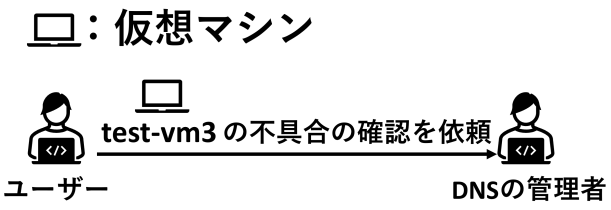

図 5　不具合の確認の依頼

DNS への問い合わせを図 6 に示す．不具合の確認を行う

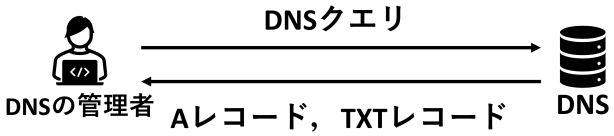

図 6　DNS への問い合わせ

ために，DNS の管理者は対象の仮想マシンに関して，DNS に問い合わせを行う．DNS は，A レコードと TXT レコードを返答する．A レコードでは，IP アドレスを確認する．TXT レコードでは，仮想マシンの UUID とハイパーバイザーの名前を確認する．

図 7 に対象のハイパーバイザーのダッシュボードにアクセスする図を示す．DNS の管理者は，TXT レコードから

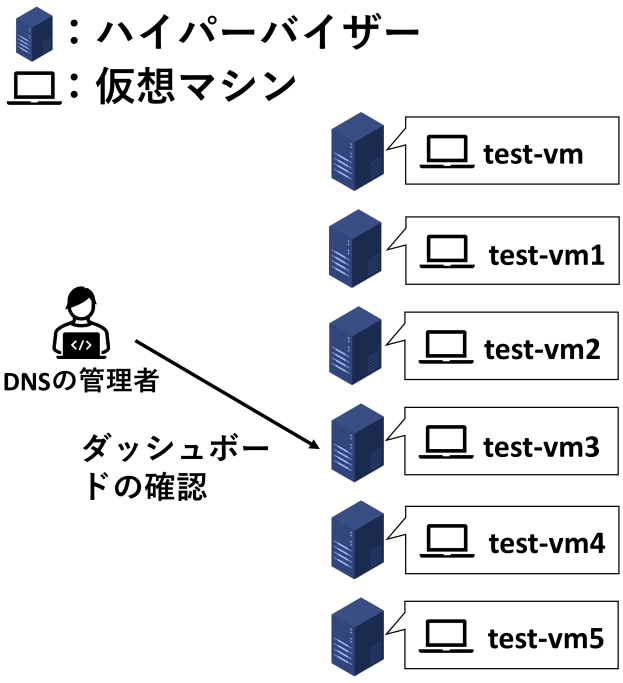

図 7　対象のハイパーバイザーのダッシュボードにアクセス

取得したハイパーバイザーの名前を参考にダッシュボードにアクセスする．

4.　実装

提案方式をもとに Python3.12.3 を使用したソフトウェアを作成した．提案ソフトウェアとデータベースを実装するための Ubuntu 24.04 をインストールした仮想マシンを作成した．データベースは，MariaDB 10.11.8 を使用した．テーブルのスキーマをコード 1 に示す．レコードには，id，

コード 1　テーブルのスキーマ

```
1  CREATE TABLE tbl_nm (
2    id   INT AUTO_INCREMENT,
3    uuid INT NOT NULL,
4    vmname   VARCHAR(255) NOT NULL,
5    hostname   VARCHAR(255) DEFAULT NULL,
6    ipaddress VARCHAR(255) DEFAULT NULL,
7    created_by VARCHAR(255) NOT NULL,
8    esxi VARCHAR(255) NOT NULL,
9    PRIMARY KEY (id, uuid)
10 );
```

uuid，vmname，hostname，ipaddress，created_by，esxi の項目がある．PRIMARY KEY として，id と uuid を設定した．id はレコードごとの番号，uuid は仮想マシンに与える固有の番号，vmname は仮想マシンの名前，hostname は仮想マシンのホスト名，ipaddress は仮想マシンの IPv4 アドレス，created_by は仮想マシンの作成者，esxi はハイパーバイザーの名前を示す．

図 8 に実装の概要を示す．提案ソフトウェアの処理は，(1) から (3) の三段階に分けられる．以下に (1) から (3) の詳細を示す．

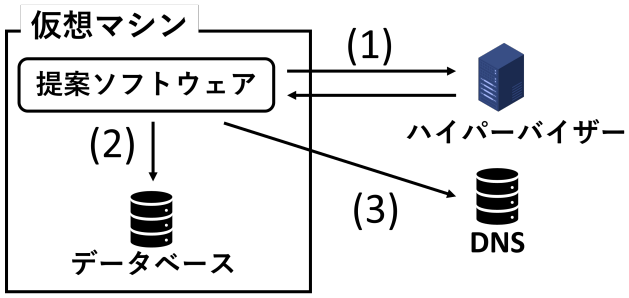

図 8　実装の概要

(1) 仮想マシンから SSH し，vim-cmd コマンドを実行する．まず，‘vim-cmd vmsvc/getallvms‘を実行しハイパーバイザーに登録されているすべての仮想マシンのリストを取得する．このコマンドで，Vmid を取得し，‘vim-cmd vmsvc/get.guest {Vmid}‘を実行する．仮想マシンの情報を取得し，必要な情報を取り出す．

(2) データベースに仮想マシンの情報を登録する．情報に

は, uuid, vmname, hostname, ipaddress, created_by, esxi が含まれる.

(3) DNS にハイパーバイザーの名前と uuid を TXT レコードとして登録する.

5. 実験

実験環境

ハイパーバイザーは, VMware ESXi 8.0U2 を 6 台使用した. 各ハイパーバイザーには, plum, jasmine, rose, lotus, violet, mint という名前が付けられている. plum には 14 台, jasmine には 44 台, rose には 20 台, lotus には 25 台, violet には 23 台, mint には 16 台の仮想マシンが登録されている. 9 人がハイパーバイザーのダッシュボードにアクセスし, 対象の仮想マシンを見つけるまでの時間とアクセスしたハイパーバイザーの数を計測した.

実験結果と分析

基礎実験の結果を図 9 に示す. 左軸に対象の仮想マシンを見つけるまでの所要時間, 右軸に対象の仮想マシンを見つけるまでのハイパーバイザーへのアクセス回数, 下軸に実験の対象者を示す. 対象の仮想マシンを見つけるまでのハイパーバイザーへのアクセス回数は最も少なくて 2 回, 最も多くて 9 回だった. また, 対象の仮想マシンを見つけるまでの所要時間は, 最も短くて約 62 秒, 最も長くて約 322 秒だった.

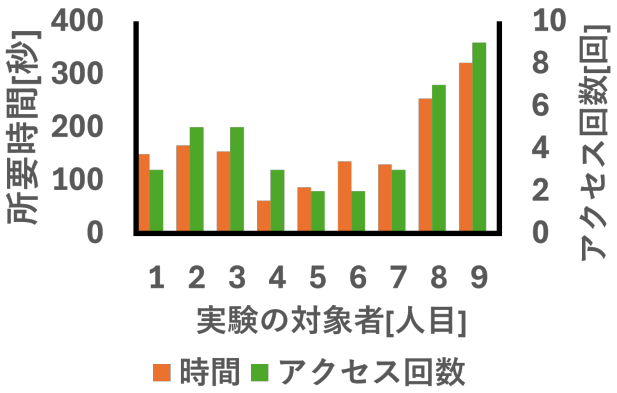

図 9 基礎実験の結果

評価実験計画

ハイパーバイザーのダッシュボードにアクセスし, 提案方式を用いる場合と用いない場合の対象の仮想マシンを見つけるまでにかかる時間を比較する.

6. 議論

本提案方式では, 仮想マシンの情報をデータベースに登録する. 仮想マシンの情報を得るためには, データベースを参照する. データベースそのものに不具合が起きた場合, 仮想マシンの情報を得ることができない. データベースの冗長化やバックアップの作成により, 不具合に対応することができる.

DNS に DNS クエリを送信し, TXT レコードの情報を取得し, 仮想マシン情報のデータベースを参照する. DNS に DNS クエリを受けつけない不具合が発生した場合, 仮想マシン情報のデータベースのレコードの ID を取得できない. DNS を冗長化することで, 不具合に対応し TXT レコードを取得できる.

DNS のキャッシュが残っている場合, レコードの更新が即座に行われない. キャッシュのクリアや Time to Live を短く設定することで対応できる.

提案ソフトウェアを実行した際に登録されている仮想マシンしかデータベースに登録されない. 新しく仮想マシンが作成された際に提案ソフトウェアが自動で実行されるようにすることで, データベースを更新することができる.

7. おわりに

CDSL では, 学生が自由に使用できるハイパーバイザーが 6 台用意されている. 仮想マシンの作成は自由であるため, 仮想マシンの情報が作成者本人以外に共有されていない. 課題は, 対象の仮想マシンがどのハイパーバイザーに登録されているかわからないことである. 提案方式として, すべてのハイパーバイザーから仮想マシンの情報を取得し, データベースに登録し, DNS の TXT レコードに仮想マシンのデータベースから取得したレコードの ID と仮想マシンのホスト名を登録する. 基礎実験として 9 人がハイパーバイザーのダッシュボードにアクセスし, 対象の仮想マシンを見つけるまでの時間とアクセスしたハイパーバイザーの数を計測した. 対象の仮想マシンを見つけるまでのハイパーバイザーへのアクセス回数は最も少なくて 2 回, 最も多くて 9 回だった. また, 対象の仮想マシンを見つけるまでの所要時間は, 最も短くて約 62 秒, 最も長くて約 322 秒だった. 評価方法として, ハイパーバイザーのダッシュボードにアクセスし, 提案方式を用いる場合と用いない場合の対象の仮想マシンを見つけるまでにかかる時間を比較する.

参考文献

[1] Bauman, E., Ayoade, G. and Lin, Z.: A Survey on Hypervisor-Based Monitoring: Approaches, Applications, and Evolutions, *ACM Comput. Surv.*, Vol. 48, No. 1 (online), DOI: 10.1145/2775111 (2015).

[2] Awasthi, A. and Gupta, R.: Multiple hypervisor based Open Stack cloud and VM migration, *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*, pp. 130–134 (online), DOI: 10.1109/CONFLUENCE.2016.7508101 (2016).

[3] Ally, S.: Comparative analysis of proxmox VE and

xenserver as type 1 open source based hypervisors, *International Journal of Scientific & Technology Research*, Vol. 7, No. 3 (2018).

[4] Aalam, Z., Kumar, V. and Gour, S.: A review paper on hypervisor and virtual machine security, *Journal of Physics: Conference Series*, Vol. 1950, No. 1, IOP Publishing, p. 012027 (2021).

[5] Pandey, R.: Comparing vmware fusion, oracle virtualbox, parallels desktop implemented as type-2 hypervisors, *National College of Ireland* (2020).

[6] orević, B., Timčenko, V., Sakić, D. and Davidović, N.: File system performance for type-1 hypervisors on the Xen and VMware ESXi, *2022 21st International Symposium INFOTEH-JAHORINA (INFOTEH)*, pp. 1–6 (online), DOI: 10.1109/INFOTEH53737.2022.9751288 (2022).

[7] Djordjevic, B., Timcenko, V., Kraljevic, N. and Macek, N.: File System Performance Comparison in Full Hardware Virtualization with ESXi, KVM, Hyper-V and Xen Hypervisors., *Advances in Electrical & Computer Engineering*, Vol. 21, No. 1 (2021).

[8] Shaikh, A., Tewari, R. and Agrawal, M.: On the effectiveness of DNS-based server selection, *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, Vol. 3, pp. 1801–1810 vol.3 (online), DOI: 10.1109/INFCOM.2001.916678 (2001).

[9] Jung, J., Sit, E., Balakrishnan, H. and Morris, R.: DNS performance and the effectiveness of caching, *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, IMW '01, New York, NY, USA, Association for Computing Machinery, p. 153–167 (online), DOI: 10.1145/505202.505223 (2001).

[10] Zhauniarovich, Y., Khalil, I., Yu, T. and Dacier, M.: A Survey on Malicious Domains Detection through DNS Data Analysis, *ACM Comput. Surv.*, Vol. 51, No. 4 (online), DOI: 10.1145/3191329 (2018).

[11] Weimer, F.: Passive DNS replication, *FIRST conference on computer security incident*, Vol. 98, pp. 1–14 (2005).

[12] Singh, S. P.: The Use of DNS Resource Records, *International Journal of Advances in Electrical and Electronics Engineering (IJAEEE, ISSN: 2319-1112)*, Vol. 1, No. 02, pp. 230–236 (2012).

[13] Fernandez, S., Korczyński, M. and Duda, A.: Early detection of spam domains with passive DNS and SPF, *International Conference on Passive and Active Network Measurement*, Springer, pp. 30–49 (2022).

[14] Wong, M. and Schlitt, W.: RFC 4408: Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1 (2006).

[15] Ashiq, M. I., Li, W., Fiebig, T. and Chung, T.: SPF Beyond the Standard: Management and Operational Challenges in Practice and Practical Recommendations.

[16] Brownlee, N., Claffy, K. and Nemeth, E.: DNS measurements at a root server, *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No.01CH37270)*, Vol. 3, pp. 1672–1676 vol.3 (online), DOI: 10.1109/GLOCOM.2001.965864 (2001).

[17] Ichise, H., Jin, Y. and Iida, K.: Analysis of via-resolver DNS TXT queries and detection possibility of botnet communications, *2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pp. 216–221 (online), DOI: 10.1109/PACRIM.2015.7334837 (2015).

[18] der Toorn, O. v., van Rijswijk-Deij, R., Fiebig, T., Lindorfer, M. and Sperotto, A.: TXTing 101: Finding Security Issues in the Long Tail of DNS TXT Records, *2020 IEEE European Symposium on Security and Privacy Workshops (EuroSPW)*, pp. 544–549 (online), DOI: 10.1109/EuroSPW51379.2020.00080 (2020).

[19] Marian, C. V.: DNS Records Secure Provisioning Mechanism for Virtual Machines automatic management in high density data centers, *2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pp. 1–5 (online), DOI: 10.1109/BlackSeaCom52164.2021.9527811 (2021).

[20] Cziva, R., Jouët, S., Stapleton, D., Tso, F. P. and Pezaros, D. P.: SDN-Based Virtual Machine Management for Cloud Data Centers, *IEEE Transactions on Network and Service Management*, Vol. 13, No. 2, pp. 212–225 (online), DOI: 10.1109/TNSM.2016.2528220 (2016).

