

RPSの低下検知に基づくファイルのコピーの中断と再開によるHTTPレスポンス数の減少の軽減

井田 尚樹¹ 飯島 貴政² 串田 高幸¹

概要: スナップショットを用いたバックアップによるファイルのコピーはCPUやメモリ、ネットワーク、ディスクというリソースを使用する。そのため、ファイルのコピーがそのECサイトサービスの妨げになる。また、ストレージは増設できるがディスクI/O速度やスループットは上がらないため、CPUやメモリを同じく増設した場合、ディスクがボトルネックとなる。よって、ファイルのコピーとECサイトサービスのディスクへの要求が重なった時にディスク帯域幅の使用率が高くなり、ECサイトのサービスの妨げとなる。本稿では仮想マシン(以下、VM)上のWordPressをECサイトとし、そのECサイトが1秒間に処理を行うHTTPリクエストの数であるRequest Per Second(以下、RPS)の監視を行い、閾値と比較することでファイルのコピーの中断と再開を行う。閾値は1秒間に100アクセスを5分間発生させる実験を5回行った時のRPSの最低値とした。評価方法として、ECサイトへのHTTPリクエストとファイルのコピーを同時に行い、本提案とファイルのコピーをやめなかった場合のHTTPレスポンス数を比較する。結果として、HTTPレスポンス数の減少を62.3%抑制できた。

1. はじめに

背景

バックアップとは機器の故障やデータの破損、消失が発生した際に復旧できるように別の媒体に業務データ、情報セキュリティ統合管理システム、管理対象システムの業務データを複製することである [1]。

ファイルシステムのサイズはペタバイトに達し、さらに増大すると予測していく。その結果、すべてのデータにアクセスする必要があるバックアップのようなオペレーションは、完了までに長い時間がかかる [2]。バックアップ時間が長いということは、システムのダウンタイムが長いということの意味する [3]。

そこでスナップショットという仕組みが存在する。スナップショットとは作成時点でのファイルの状態を保存しておくものである。作成したスナップショットは削除しない限り、ファイルへの変更を別領域に保存しておくことができる [4]。そのため、バックアップの欠点であるシステムのダウンタイムが長いことを克服できる。Amazon EBS スナップショットというサービスでは、取得したスナップ

ショットをAmazon S3というクラウドストレージにバックアップする。^{*1}しかし、スナップショットのメカニズムでは、占有されたCPU、メモリ、およびディスクI/Oによる読み取り、書き込みパフォーマンスの低下といった他の課題に直面する [5]。

課題

課題はファイルのコピーとECサイトへのHTTP GETリクエストやHTTP POSTリクエスト(以下、HTTPリクエスト)が同時に発生するとHTTPリクエストが発生しただけの時と比較してHTTPレスポンス数が減少することである。図1に課題の概要を示す。ユーザはVM上にあるWordPressで作られたECサイトに対してHTTPリクエストを送信している。VM上にはWordpressで作成されたECサイトがある。NASはVM上のファイルのバックアップを保存しておくストレージである。リクエストとファイルのコピーが同時に発生するとユーザへのレスポンスに遅延が発生し、単位時間当たりのレスポンス数が減少する。

基礎実験環境の実装

ファイルのコピーとHTTPリクエストが同時に発生し

¹ 東京工科大学コンピュータサイエンス学部
〒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・情報メディアコンピュータサイエンス専攻
〒192-0982 東京都八王子市片倉町1404-1

^{*1} Amazon EBS スナップショット https://docs.aws.amazon.com/ja_jp/AWSEC2/latest/UserGuide/EBSSnapshots.html(2022/11/28)

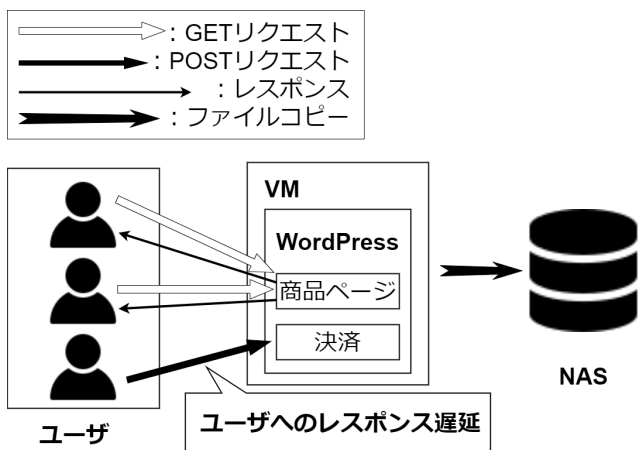


図 1 課題の概要

た時に HTTP レスポンス数が減少することを示すための基礎実験を行った。また、ディスクの帯域幅使用率がボトルネックであることを示すために CPU やメモリ使用量、ネットワーク帯域幅の監視を行う基礎実験を行った。

以下に基礎実験環境の VM 構成情報を示す。

- VM1
OS Ubuntu 20.04.3
vCPU 6 コア
RAM 8GB
HDD 30GB
- VM2
OS Ubuntu 20.04.3
vCPU 4 コア
RAM 8GB
HDD 20GB

また、各 VM にインストールしたソフトウェアを以下に示す。

- VM1
WordPress v6.1.1
Prometheus-2.37.1
Grafana v9.2.2

- VM2
Locust-2.12.1

WordPress は多くのブログ、ホームページ、およびソーシャルコミュニティを強化するために使用される人気のあるオンラインコンテンツ管理プラットフォームである [6]。Prometheus はオープンソースの監視およびアラートを行うプラットフォームである [7]。Grafana はオープンソースの時系列データやクエリの視覚化、およびアラートを行うツールである [8]。Locust は複数の同時ユーザーによる Web アプリケーションの動作を評価するための Python

ベースのツールである [9]。Prometheus では VM1 の CPU やメモリ、ネットワークのメトリクスの取得、Grafana ではそのメトリクスのグラフ化を行った。また、Locust は負荷試験を行うために使用した。

さらに、WordPress にインストールしたプラグインを以下に示す。

- WooCommerce v7.1.0
- WooCommerce Stripe ゲートウェイ v7.0.1

WooCommerce は WordPress 上に構築されたオープンソースの E コマースプラットフォームである。これは、小規模から大規模なオンラインショッピング Web サイトの開発に適用される [10]。WooCommerce Stripe ゲートウェイは、従来の決済ゲートウェイとマーチャントアカウントを 1 つのサービスに統合したプラグインである [11]。

基礎実験の環境

基礎実験は 2 つ行った。1 つ目は EC サイトへの HTTP リクエストを 4 分間送信した時の HTTP レスポンス数を確認する実験、2 つ目は HTTP リクエストとファイルのコピーを同時に行った時の HTTP レスポンス数を確認する実験である。この 2 つの基礎実験の結果を比較し、HTTP レスポンス数の減少を示す。また、2 つ目の基礎実験の時の CPU やメモリ使用量、ネットワーク帯域幅を計測し CPU やメモリ、ネットワークがボトルネックとなっていないことを示す。CPU やメモリ使用量、ネットワーク帯域幅は 2 つ目の基礎実験の時のものを使用する。1 つ目と 2 つ目の基礎実験ではファイルのコピーを同時に行っている 2 つ目の基礎実験のほうが CPU やメモリ、ネットワークに負荷がかかっているからである。図 2 に 1 つ目の基礎実験の環境を示す。VM1 上には EC サイトを作成する Wordpress が

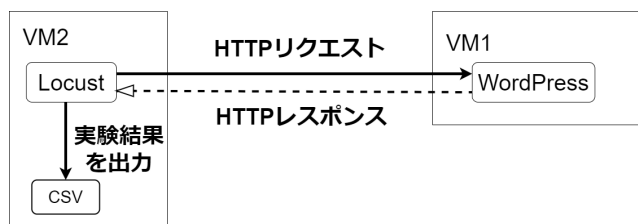


図 2 基礎実験 1

ある。VM2 上には Locust があり、EC サイトへの HTTP リクエストを発生させ、実験結果の CSV ファイルへの出力を行っている。VM1 上の WordPress と VM2 上の Locust で HTTP リクエストの送信とレスポンスを行っている。それが終わった後に CSV ファイルに結果を出力している。

図 3 に 2 つ目の基礎実験の環境を示す。1 つ目の基礎実験と同じ実験を行っている時にファイルのコピーと Prometheus による VM1 のメトリクスを取得、その結果の Grafana を使ったグラフ化を同時に実行している。

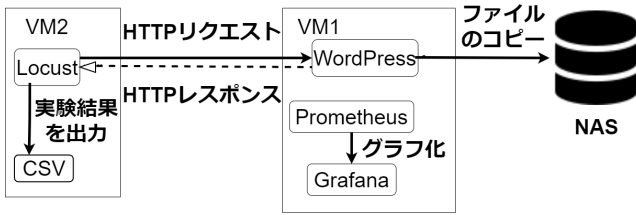


図 3 基礎実験 2

基礎実験の結果

図 4 に 1 つ目の基礎実験, 2 つ目の基礎実験を行った時のそれぞれの HTTP レスポンス数を示す. 実験は 4 分間行った. 縦軸はそれぞれの基礎実験で得られた HTTP レス

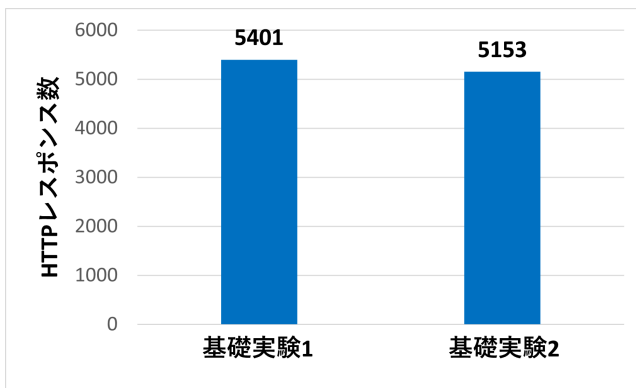


図 4 HTTP レスポンス数の差

ポンス数である. 基礎実験 2 は基礎実験 1 と比べて HTTP レスポンス数が 248 少ない.

図 5 に CPU 使用率を示す. 縦軸は CPU 使用率を表す.

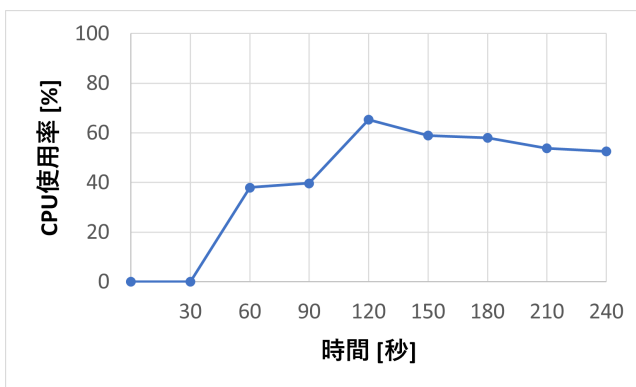


図 5 CPU 使用率

横軸は基礎実験の時間を示す. CPU 使用率は最大でも 120 秒時点の 65.3%であり, ボトルネックとは言えない.

図 6 にメモリ使用率を示す. 縦軸はメモリ使用率を表す. 横軸は基礎実験の時間を示す. メモリ使用率は最大でも 20%に達しておらず, ボトルネックとは言えない.

図 7 にネットワーク使用量を示す. 縦軸はネットワーク使用率を表す. 横軸は基礎実験の時間を示す. VM 同士は最大通信速度 1Gbps の LAN ケーブルで接続されてい

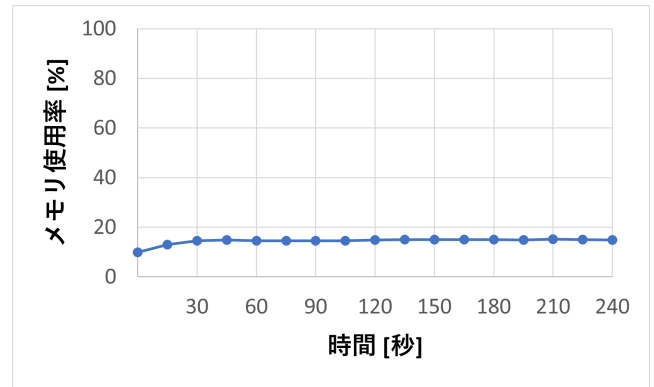


図 6 メモリ使用率

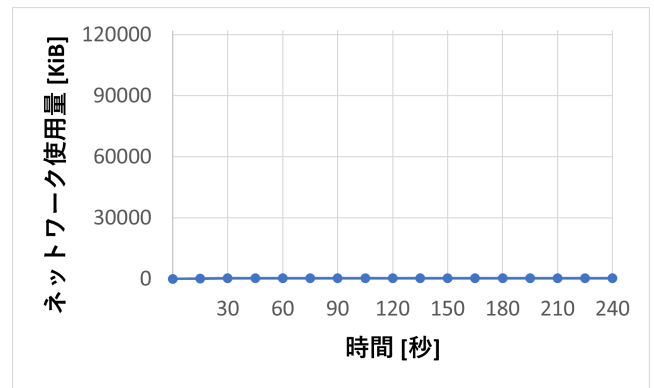


図 7 ネットワーク使用量

る. ネットワーク使用量は最大でも 365KiB であり, ボトルネックとは言えない.

これらの結果から, HTTP リクエストがファイルのコピーが原因で減少したこと, ディスクの帯域幅使用率がボトルネックだと言える.

各章の概要

第 2 章では関連研究を述べる. 第 3 章では本稿の提案方式を説明する. 第 4 章では実装及び実験方法を述べる. 第 5 章では提案方式の評価と分析を行う. 第 6 章では本稿の提案方式についての議論を行う. 第 7 章では本稿のまとめと成果を述べる.

2. 関連研究

Ajith Padyana らはハイパフォーマンスコンピューティング (以下, HPC) システムにおける GPUs と CellBE プロセッサを用いた浮動小数点圧縮に基づくディスク I/O 帯域幅ボトルネックの軽減を提案している [12]. HPC システムにおける計算集約型のタスクでは, 大量のデータ, 特に浮動小数点データをネットワーク経由で転送する. 提案手法では, GPU と CellBE プロセッサの計算能力を活用することで, データの圧縮や解凍を高速化でき, ディスク I/O 帯域幅ボトルネックを改善し, データ転送量を削減している. しかし, 圧縮や解凍によるディスク使用量がディスク

I/O 帯域幅を上回る場合、この手法では効果がない。

Matthew Malensek らは長時間バッチ処理の平均性能を維持しながら、保守的に I/O 資源を使用するタスクに対して高いスループットを確保するために、ディスクスケジューリングを独断的に決定するフレームワークを提案している [13]。そのフレームワークは構成コンポーネントが VM の外部に存在し、カーネルやハイパーバイザーの変更を必要としないため、物理および仮想マシンの設定で広く展開可能である。また、実行されている I/O 操作の数と種類に基づいてプロセスの効果的な優先順位付けを行うことで、全体的なスループットを損なうことなく公平性を確保している。しかし、優先すべきタスクが実行毎に変わってしまうため、本研究の課題を解決できない。

3. 提案

提案方式

図 8 に提案の概要を示す。本稿では HTTP レスポンス数の減少の軽減を行うため、RPS 監視に基づくファイルのコピーの中断と再開を行う。ユーザは WordPress で作成

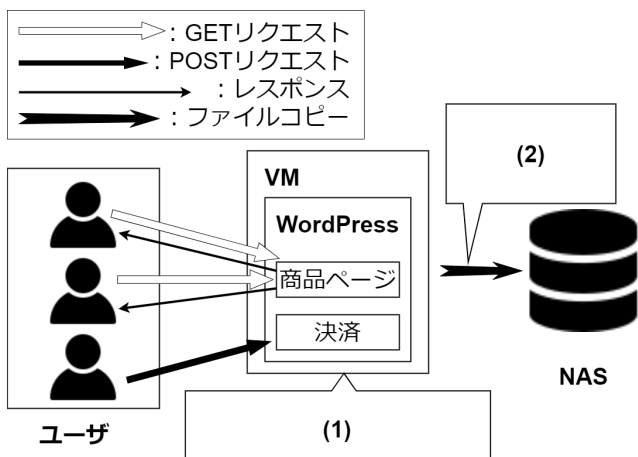


図 8 提案の概要

された EC サイトへアクセスする利用者を表している。図 8 の提案のアルゴリズムについて説明する。

- (1) ユーザの EC サイトへのアクセス時に RPS を計測しておき、閾値と比較し上回っているか、下回っているか判断する。
- (2) RPS が閾値を下回っている間はファイルのコピーを中断し、上回っている間はファイルのコピーを実行する。

閾値決め

本稿では提案の中で RPS と閾値を比較することによる、ファイルのコピーの中断と再開を行う。そのための閾値を定める。

閾値は基礎実験環境を使い、1 秒間に 100 アクセスを 5 分間発生させる実験を 5 回行った時の RPS の最低値とし

た。大量のアクセスを処理しきれない状況を再現すること、1 つ目の基礎実験での RPS の最大値が 22.4 であったことより 1 秒間に 100 アクセスとした。ユーザからのアクセスが集中した時の RPS の最低値を用いることでファイルのコピーによる RPS の低下かどうかを見分けることができる。

図 9 に 1 秒間に 100 アクセスを 5 分間発生させる実験を 5 回行った時のそれぞれの RPS の最低値を示す。縦軸は

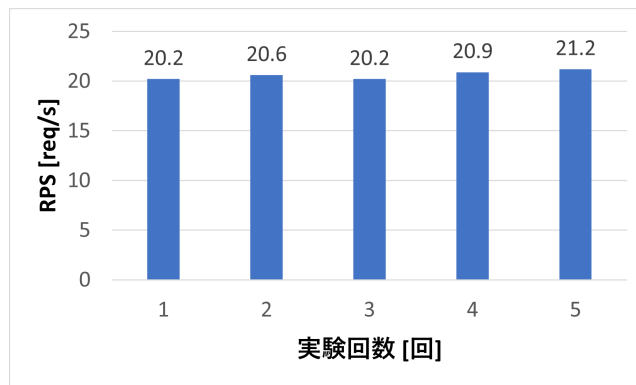


図 9 閾値決めの実験

RPS を表している。横軸は実験回数を表している。RPS の最低値は 1 回目と 3 回目の実験の 20.2 であったため、本稿では 20.2 をファイルのコピーの中断と再開を決める閾値とする。

ユースケース・シナリオ

図 10 にユースケースはお菓子の販売を行っている EC サイトを想定している。ユースケースシナリオを示す。Youtube や Twitich といった動画配信サイトには配信者と

深夜

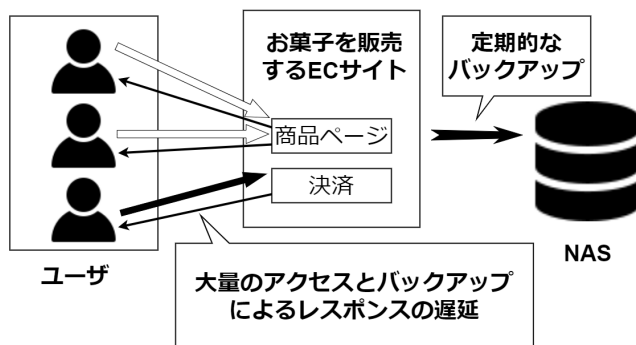


図 10 ユースケースシナリオ

呼ばれるリアルタイムのライブストリーミングを行う人たちがいる。Amazon ではその配信者による宣伝で、その日の売り上げが 1 位となったお菓子がある。また、ライブストリーミングの配信時間は不定期であり深夜に行われることもある。そのため、深夜に設定していた定期バックアッ

プの時間とライブストリーミングの時間が重なることがある。その時、配信者が商品の宣伝をした場合、その商品を販売する EC サイトでは大量のアクセスによる応答時間の遅延が発生し、売り上げの低下につながる。

4. 実装

本稿では4つのソフトウェアを開発した。図11に実装の概要を示す。研究室の2つのESXiにそれぞれVMをたて

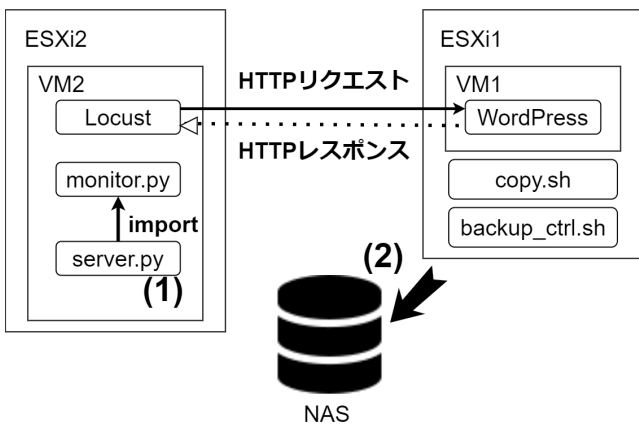


図 11 実装の概要

た。ESXi1のVM1にはWordPressをESXi2のVM2にはLocustを構築した。提案ソフトウェアはESXi1内、ESXi2上のVM2内にそれぞれ作成し、Python、Shellscriptで実装した。ESXi2上のVM2内にはLocustによるWordPressへのHTTPリクエストを行っている時のRPSの取得、取得したRPSが閾値を上回っているか下回っているかどうかを判別する関数を定義したmonitor.py、Flaskサーバを構築し、monitor.pyの関数を使い判断したRPSの結果をbool型の変数としてFlaskサーバにアップロードするserver.pyを作成した。ESXi1にはファイルのコピーとファイルのコピーにかかった時間の計測を行うcopy.sh、Flaskサーバの変数をもとにファイルのコピーの中断と再開を行うbackup_ctrl.shを作成した。図11の(1)ではserver.pyによるFlaskサーバへのbool型の変数のアップロードを行っている。(2)ではbackup_ctrl.shによるFlaskサーバの変数をもとにファイルのコピーの中断と再開を行っている。

5. 実験と分析

評価実験では3種類の実験を行った。HTTPリクエストの送信だけを行った時のHTTPレスポンスとファイルのコピーにかかった時間を計測する実験、HTTPリクエストとファイルのコピーを同時に行った時のHTTPレスポンスとファイルのコピーにかかった時間を計測する実験、HTTPリクエストとファイルのコピー、提案プログラムの実行を同時に実行した時のHTTPレスポンスとファイルのコ

ピーにかかった時間を計測する実験である。ここでは便宜上HTTPリクエストの送信だけを行った時のHTTPレスポンスとファイルのコピーにかかった時間を計測する実験を実験1、HTTPリクエストとファイルのコピーを同時に行った時のHTTPレスポンスとファイルのコピーにかかった時間を計測する実験を実験2、HTTPリクエストとファイルのコピー、提案プログラムの実行を同時に実行した時のHTTPレスポンスとファイルのコピーにかかった時間を計測する実験を実験3とする。

実験環境

図12に実験1の環境を示す。VM2上のLocustでVM1

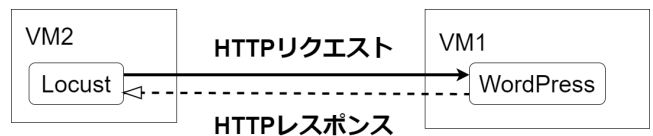


図 12 実験1の構成

上のWordPressに対してHTTPリクエストを送信し、そのHTTPレスポンス数を計測する。実験は4分間行った。

図13に実験2の環境を示す。VM2上のLocustでVM1

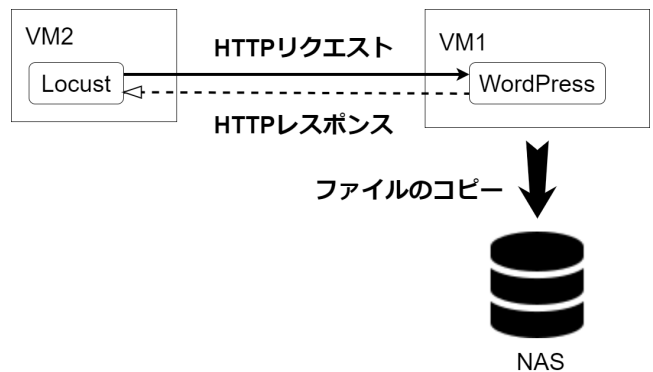


図 13 実験2の構成

上のWordPressに対してHTTPリクエストを送信する時にファイルのコピーも同時に行う。その時のHTTPレスポンス数を計測する。実験は4分間行った。

図14に実験3の環境を示す。VM2上のLocustでVM1上のWordPressに対してHTTPリクエストを送信する時にファイルのコピーも同時に行い、提案ソフトウェアも実行する。その時のHTTPレスポンス数を計測する。実験は4分間行った。

実験結果と分析

図15に3種類の評価実験のHTTPレスポンス数を示す。縦軸はHTTPレスポンス数を表している。HTTPリクエストだけの実験1では5065だったHTTPレスポンス数が、ファイルのコピーも同時に行った実験2では4874

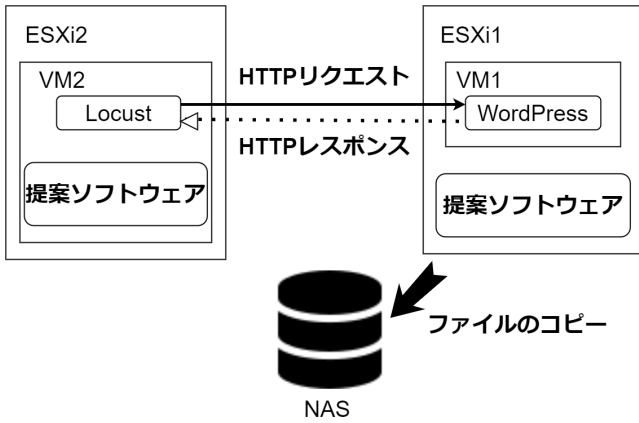


図 14 実験 3 の構成

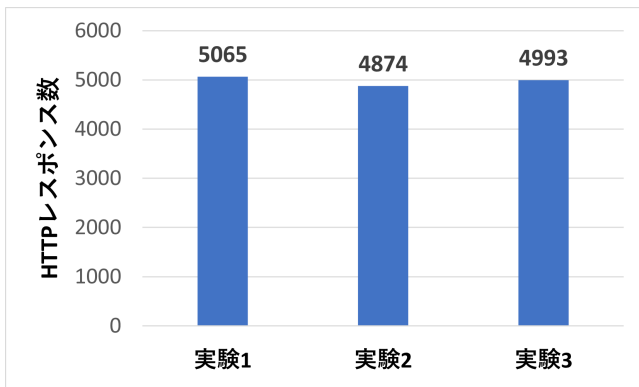


図 15 各実験の HTTP レスポンス数の比較

と 191 減少している。しかし、提案ソフトウェアも同時に実行した実験 3 では 4993 と 72 の減少に抑えられている。これはファイルのコピーの中断により、ディスク帯域幅使用量が減少し、HTTP レスポンスの遅延が軽減されたためである。

図 16 に 3 種類の評価実験のファイルのコピーにかかった時間を示す。縦軸はファイルのコピーにかかった時間を

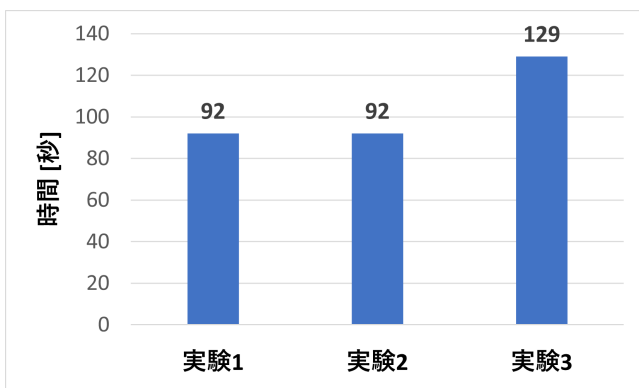


図 16 各実験のファイルのコピーにかかった時間の比較

表している。ファイルのコピーにかかった時間は実験 1, 実験 2 とともに 92 秒だったが実験 3 では 129 秒まで 37 秒増加している。これはファイルのコピーの中断によってファイルのコピー完了までにかかる時間が増加したためである。

6. 議論

本提案では HTTP レスポンス数の減少を 62.3%抑制できた。しかし、ファイルのコピーにかかる時間が 1 分 32 秒から 2 分 9 秒までの 37 秒増加した。コピーするファイルのサイズが大きくなり、アクセスの集中が続けばそれだけファイルのコピーにかかる時間が増加してしまう。今後の展望として、アクセスに対してレスポンスを行う処理を分散させるソフトウェアを提案する。分散処理を行うことでファイルのコピーと競合してディスクの帯域幅使用率が 100 % になってしまうことを回避できる。

図 17 に実験 3 の開始から 2 分間の RPS の推移を示す。縦軸は RPS、横軸は時間を表す。本稿ではファイルのコ



図 17 実験 3 の RPS の推移

ピーの中断を RPS の低下に基づいて行った。RPS が閾値より低下した時間からファイルのコピーを中断することで、ファイルのコピーの中断を行わない時より、RPS が閾値より低下している時間を減らした。しかし、図 17 より 52 秒から 1 分 6 秒までの 14 秒間は RPS が閾値より低下してしまっている。そこで計測している RPS の値より以前の期間での減少率を用いることで、RPS の低下を予測することができ、RPS が閾値より低下している時間をより減らすことができる。

7. おわりに

課題はファイルのコピーと EC サイトへの HTTP GET リクエストや HTTP POST リクエストが同時に発生すると HTTP リクエストが発生しただけの時と比較して HTTP レスポンス数が減少することである。提案では HTTP レスポンス数の低下の軽減を行うため、RPS 監視に基づくファイルのコピーの中断と再開を行う。評価では EC サイトへの HTTP リクエストとファイルのコピーを同時に行い、本提案とファイルのコピーをやめなかった時の HTTP レスポンス数で比較する。結果として、ファイルのコピー時間は 37 秒増加したが、HTTP レスポンス数の減少を 62.3%抑制できた。

謝辞 本テクニカルレポートの執筆にあたり実装にご協力いただいた東京工科大学大学院バイオ・情報メディアコンピュータサイエンス専攻の小山 智之さん, 東京工科大学コンピュータサイエンス学部の富田 啓太さん, 坂本 一俊にお礼申し上げます。

national Symposium on Big Data Computing (BDC), IEEE, pp. 1–10 (2015).

参考文献

- [1] Zhao, Y. and Lu, N.: Research and implementation of data storage backup, *2018 IEEE International Conference on Energy Internet (ICEI)*, IEEE, pp. 181–184 (2018).
- [2] Henson, V., van de Ven, A., Gud, A. and Brown, Z.: Chunkfs: Using Divide-and-Conquer to Improve File System Reliability and Repair., *HotDep* (2006).
- [3] Deka, L. and Barua, G.: Consistent Online Backup in Transactional File Systems, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 26, No. 11, pp. 2676–2688 (online), DOI: 10.1109/TKDE.2014.2302297 (2014).
- [4] Hao, Z., Wang, W., Cui, L., Yun, X. and Ding, Z.: iConSnap: An Incremental Continuous Snapshots System for Virtual Machines, *IEEE Transactions on Services Computing*, Vol. 15, No. 1, pp. 539–550 (online), DOI: 10.1109/TSC.2019.2955700 (2022).
- [5] Chuang, P.-J. and Wong, W.-C.: Generating Snapshot Backups in Cloud Virtual Disks, *2014 IEEE 17th International Conference on Computational Science and Engineering*, pp. 1860–1863 (online), DOI: 10.1109/CSE.2014.341 (2014).
- [6] Koskinen, T., Ihantola, P. and Karavirta, V.: Quality of WordPress plug-ins: an overview of security and user ratings, *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing*, IEEE, pp. 834–837 (2012).
- [7] 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/CBDCCom/IOP/SCI)*, IEEE, pp. 257–262 (2019).
- [8] Chakraborty, M. and Kundan, A. P.: Grafana, *Monitoring Cloud-Native Applications*, Springer, pp. 187–240 (2021).
- [9] Pradeep, S. and Sharma, Y. K.: A pragmatic evaluation of stress and performance testing technologies for web based applications, *2019 Amity International Conference on Artificial Intelligence (AICAI)*, IEEE, pp. 399–403 (2019).
- [10] Ravensbergen, R.: *Building E-Commerce Solutions with WooCommerce*, Packt Publishing Ltd (2015).
- [11] Rauland, P.: *WooCommerce Cookbook*, Packt Publishing Ltd (2015).
- [12] Padyana, A., Sudheer, D., Baruah, P. K. and Srinivasan, A.: Reducing the disk io bandwidth bottleneck through fast floating point compression using accelerators, *International Journal of Advanced Computer Research*, Vol. 4, No. 1, pp. 134–144 (2014).
- [13] Malensek, M., Pallickara, S. L. and Pallickara, S.: Alleviation of disk I/O contention in virtualized settings for data-intensive computing, *2015 IEEE/ACM 2nd Inter-*