

ECサイトのバックアップ期限とディスク帯域幅使用率にもとづく読み取り量の制御によるレスポンス数の減少の抑制

井田 尚樹¹ 高橋 風太² 串田 高幸¹

概要：ユーザが商品を購入した場合、データベースへの書き込みが行われ、ディスクの帯域幅を使用する。また、ECサイトのデータのバックアップを行うことでもディスクの帯域幅を使用する。課題はECサイトへのリクエストとバックアップが同時に発生した時、ディスク帯域幅がボトルネックとなり、ユーザからのリクエストへの単位時間当たりのレスポンス数が、バックアップを行っていない時を比較して減少することである。提案方式ではリクエストとバックアップによる使用ディスク帯域幅が100%に達していないときはバックアップによるディスクへの読み取り量を増加させる。それ以外の場合はバックアップの期限を超過しないように、バックアップ開始前にバックアップするデータ容量とバックアップ期限から算出した値だけディスクへの読み取りを行う。また、サーバが処理できる最大秒間リクエストで増加するディスクの帯域幅使用率を事前に計測しておき、このディスクの帯域幅を常に使わずに残しておく。基礎実験ではECサイトへのリクエストを送信する実験、ECサイトへのリクエスト送信とバックアップを同時に発生させる実験をそれぞれ5回行い、その平均の単位時間当たりのレスポンス数で比較した。結果は前者の実験が25.0req/s、後者の実験が23.1req/sで7.6%減少していた。

1. はじめに

背景

ECサイトとはElectronic Commerce(以下eコマース)を行うサイトである。eコマースとは企業や個人がインターネット上で取引を行うビジネスモデルである [1]。

株式会社シャープは2020年4月21日にこの時期に品薄が続いていたマスクの販売を行い、オンラインショップや会員サイトにアクセスが集中した^{*1}。ECサイトではこのようにアクセスが急増することがある [2]。そのため、顧客の満足度を満たすには、eコマース用のサーバを常に利用できるようにする必要があり、システムが効率的に動作し続け、災害が発生した場合にシステムを回復させる必要がある [3]。

そこでバックアップを行う。機器の故障やユーザエラーによるデータの破損、自然災害により消失が発生した際に復旧できるように別の媒体にデータを複製することで、元の情報を復元できるようにしておく [4]。

大規模な高性能コンピュータシステムのI/Oパフォーマンスは、コンピューティング能力の成長に追いついていないため、I/Oパフォーマンスが、高性能コンピュータシステム全体のボトルネックになることがある [5]。カールスルーエ工科大学のForHLR IIのコンピューティングノードでは、ネットワーク転送速度が7GB/sと12.5GB/sなのに対し、600MB/sの読み取りパフォーマンスと400MB/sの書き込みパフォーマンスのディスク搭載されている。このような環境では、ディスクのI/Oパフォーマンスがボトルネックとなり、高性能コンピュータシステム全体の速度が低下させ、同じI/Oサブシステムを使用する他のアプリケーションを中断させる原因となる。

課題

図1に課題の概要を示す。

課題はバックアップとECサイトへのリクエスト送信が同時に発生するとバックアップを行っていない時と比較して単位時間当たりのレスポンス数が減少することである。サーバ上にはECサイトがあり、ユーザからのリクエスト処理やバックアップサーバへのバックアップを行う。ユーザからのリクエストではサーバのディスクへの書き込み、バックアップサーバへのバックアップではディスクへの読み込みが行われる。これらが同時に発生した時、サーバの

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・情報メディア研究科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} ECサイトのアクセス集中対策をシャープの事例から考える
[https://caina.co.jp/service/ec-sharp/\(2023/07/11\)](https://caina.co.jp/service/ec-sharp/(2023/07/11))

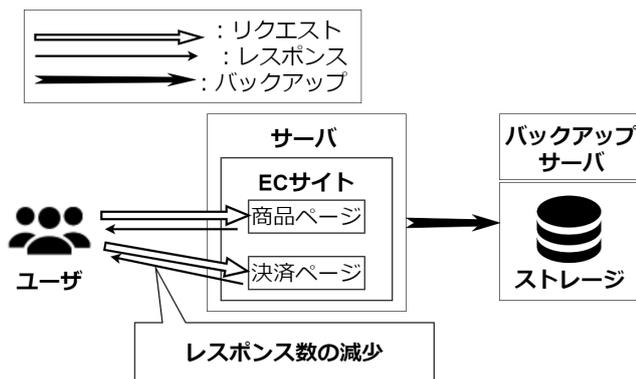


図 1 課題の概要

ディスクへの読み書きが行われ、ディスクの帯域幅使用率は 100 % ほどになる。この時にユーザへのレスポンスに遅延が発生し、単位時間当たりのユーザへのレスポンス数が減少する。ユーザへのレスポンスの遅延は売り上げの減少につながる*2。

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述べる。第 4 章では実装を述べる。第 5 章では基礎実験の実装や実験結果を述べる。第 6 章では本稿の提案方式についての議論を述べる。第 7 章では本稿のまとめと成果を述べる。

2. 関連研究

空きディスク領域での動的なデータ複製によるディスクのパフォーマンスとエネルギー消費の改善を提案している研究がある [6]。実行時に観察されるディスクアクセスパターンに従ってファイルシステムの空きブロックにデータのコピーを動的に配置することで、元のデータブロックに加えて 1 つ以上のレプリカにアクセスできるようになり、最速のアクセスを提供する最も近いレプリカを選択することで、ディスク I/O 操作のパフォーマンスが大幅に向上した。しかし、Ext2 ファイルシステムをベースとした設計が前提であるため、本稿の課題は解決できない。

アクセス要求に対するディスク応答時間を最小限に抑え、ディスクシステムの I/O パフォーマンスを向上させるために、最適に近いストライピングユニットを選択する新しい方法について提案している研究がある [7]。ストライピングユニットは単一のディスクに保存される論理的に連続したデータの最大量である。この公式により、さまざまなリクエストサイズとさまざまなリクエスト到着率の場合に、最適に近いストライピングユニットを迅速かつ簡単に取得できる。しかし、ハードウェアが違う環境では、すべ

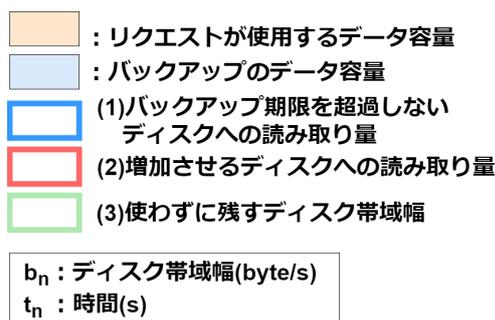
てのリクエストサイズとリクエスト到着率、ストライピングユニットの関係を調べなければならない。

関数呼び出しインターセプトベースの方法を使用して I/O を集約し、データがメインメモリに書き込まれる回数を最小限に抑え、ほとんどセットアップを必要とせずにパフォーマンスを向上させる方法を提案している研究がある [8]。ダブルバッファ設計により、OS によって実行される最適化に関係なく、それぞれにオーバーヘッドが必要となる多数の小規模な読み取りまたは書き込みを防止する中間ストレージおよび集約システムが可能になった。しかし、この提案は C 言語のソースコードからコンパイルされたプログラムによる書き込み処理にのみ機能するため、本稿の課題は解決できない。

3. 提案

提案方式

本稿では、バックアップ期限とディスク帯域幅使用率にもとづくディスクへの読み取り量の制御を提案する。バックアップ期限は企業の業務開始時刻とする。図 2 に提案方式の概要を示す。



ディスク帯域幅(byte/s)

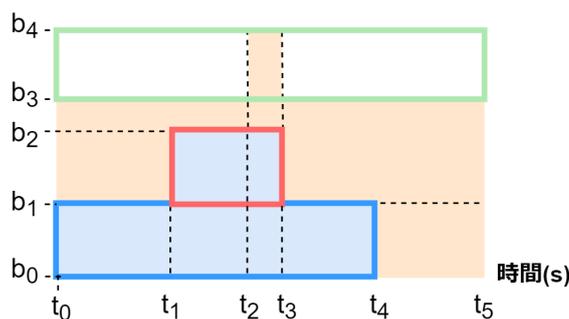


図 2 提案方式の概要

本稿の提案方式を図 2 に示す (1), (2), (3) に分けて説明する。

(1) バックアップの期限を超過しないように、バックアップ開始前にバックアップするデータ容量とバックアップ期限から算出した値だけディスクへの読み取りを行う。制限したディスクへの読み取り量の計算方法を以下に示す。制限したディスクへの読み取り量を

*2 <http://robotics.stanford.edu/~ronnyk/2007IEEEComputerOnlineExperiments.pdf>
[https://onl.sc/isy2J5m\(2023/07/17\)](https://onl.sc/isy2J5m(2023/07/17))

L [MB/s], バックアップするデータ容量を S [MB], バックアップ期限を D [s] とする.

$$L = \frac{S}{D} \quad (1)$$

t_5 をバックアップ期限とすると, バックアップするデータ容量を t_0 から t_5 の時間で割ることによって b_0 から b_1 のディスク帯域幅が算出される. t_0 から t_5 の時間まで b_0 から b_1 のディスク帯域幅をバックアップに使用することで, バックアップを期限までに終了できる. t_4 から t_5 の時間に b_0 から b_1 のディスク帯域幅をリクエストに使用している理由は (2) で述べる.

- (2) リクエストとバックアップによる使用ディスク帯域幅使用率が 100%に達していないときはバックアップによるディスクへの読み取り量を増加させる. リクエストが t_1 から t_3 の時間に b_2 から b_3 のディスク帯域幅しか使用しない場合, バックアップに使用する t_1 から t_3 の時間の b_1 から b_2 のディスク帯域幅を増加させる. こうすることでバックアップを期限までに終了でき, バックアップが終了してからバックアップ期限までの時間はディスクの帯域幅をすべてリクエストに使用できる. t_4 から t_5 の時間に b_0 から b_1 のディスク帯域幅をリクエストに使用している理由は, t_1 から t_3 の時間にバックアップに使用するディスク帯域幅を b_1 から b_2 の分だけ増加させることができ, t_0 から t_4 の時間でバックアップが終了したためである.

- (3) ユーザからのリクエストの増加検知後にディスクへの読み取り量の制限を行うまでにディスク帯域幅使用率が 100%に達し, 単位時間当たりのレスポンス数が減少する. これを解決するために, サーバが処理できる最大秒間リクエストで増加するディスクの帯域幅使用率を事前に計測しておき, このディスクの帯域幅を常に使わずに残しておく. b_4 をディスク帯域幅の 100% とすると, t_2 からリクエストが増加した場合, t_2 から t_3 の時間はリクエストの増加検知とディスクへの読み取り制限を行うため, ディスク帯域幅が b_4 に達してしまい, 単位時間当たりのレスポンス数が減少する. そこで b_3 から b_4 までのディスク帯域幅を常に使用しないで残しておく, リクエストとバックアップによる使用ディスク帯域幅が 100%になることを防ぐ.

ユースケース・シナリオ

ユースケースは動画の販売, 配信を行う EC サイトを想定する. ユースケースシナリオを図 3 に示す.

EC サイトを構築したサーバでユーザからのリクエストの処理を行い, バックアップサーバに EC サイトのバックアップを行う. そのため, ディスクへの読み書きが同時に行われ, 単位時間当たりのレスポンス数が減少する. そこで, 本稿の提案方式を実装したソフトウェアを使用するこ

深夜

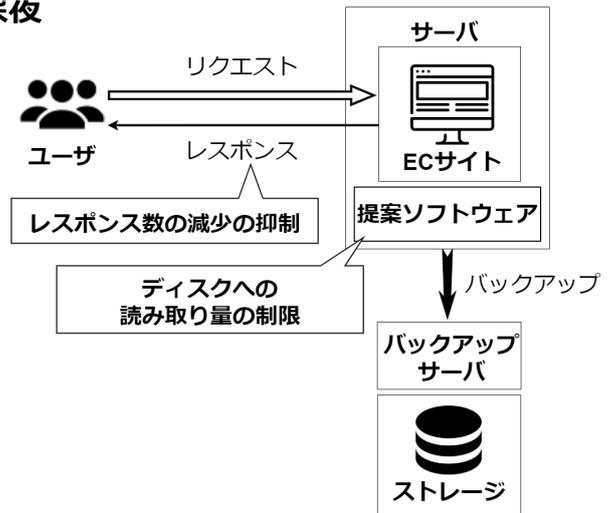


図 3 ユースケースシナリオ

とで, バックアップを期限までに終わらせつつ, ユーザからのリクエストに使用できるディスクの帯域幅を増加させることができる.

4. 実装

図 4 に実装の全体構成を示す. 本稿では提案方式を実装

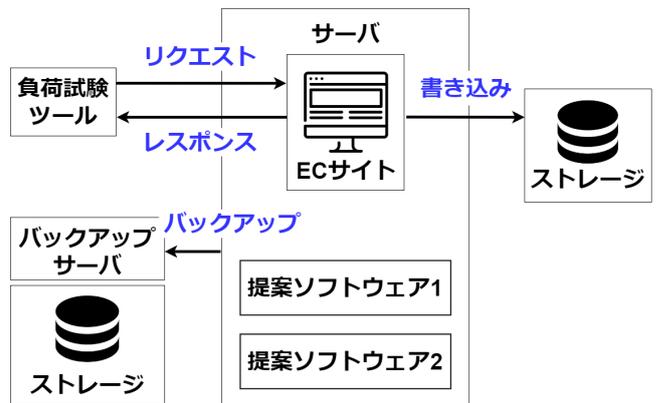


図 4 実装の全体構成

したソフトウェアを 2 つ開発する. 提案ソフトウェア 1 は EC サイトを構築したサーバが処理できる最大秒間リクエストで増加するディスクの帯域幅をバックアップ開始前に計測する. 提案ソフトウェア 2 はリクエストとバックアップによる使用ディスク帯域幅が 100%に達していないときはバックアップによるディスクへの読み取り量を増加させ, それ以外の場合はバックアップの期限を超過しないように, バックアップ開始前に式 (1) より算出した値だけディスクへの読み取りを行う. また, 提案ソフトウェア 1 で計測したディスクの帯域幅を常に使わずに残しておく. 提案ソフトウェアはどちらも EC サイトを構築したサーバに配置する.

図 5 の提案ソフトウェア 1 の処理の流れを以下に示す.

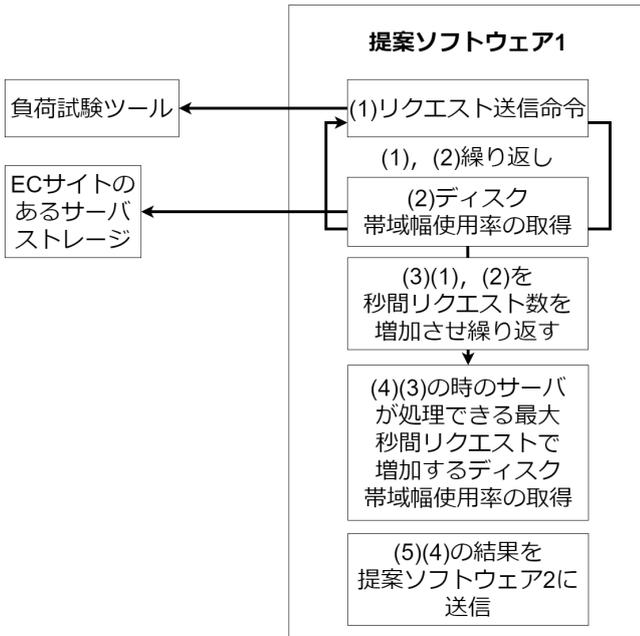


図 5 提案ソフトウェア 1 の処理の流れ

- 図 5 の提案ソフトウェア 1 の処理の流れを以下に示す。
- (1) 負荷試験ツールに EC サイトへリクエストを送信するよう命令する。
 - (2) 負荷試験ツールから EC サイトへリクエストを送信する。この時の EC サイトのサーバのディスクの帯域幅使用率を取得する。
 - (3) 負荷試験ツールから EC サイトへの秒間リクエスト数を増加させ (1), (2) 繰り返し行う。
 - (4) (3) の処理でサーバが処理できる最大秒間リクエストで増加するディスクの帯域幅使用率を取得する。
 - (5) (4) で取得したディスクの帯域幅使用率を提案ソフトウェア 2 へ送信する。

図 6 に提案ソフトウェア 2 の処理の流れを示す。

図 6 の提案ソフトウェア 2 の処理の流れを以下に示す。

- (1) 提案ソフトウェア 1 からディスクの帯域幅使用率を受け取る。
- (2) EC サイトを構築したサーバのストレージからバックアップのデータ容量を取得する。
- (3) 式 (1) より制限したディスクへの読み取り量を算出する。
- (4) rsync コマンドによるバックアップを開始する。
- (5) EC サイトを構築したサーバのストレージからディスク帯域幅を取得する。
- (6) リクエストとバックアップによる使用ディスク帯域幅使用率が (5) のディスク帯域幅使用率から (1) のディスク帯域幅使用率を引いた値に達しているか確認する。
- (7) 達していた場合は式 (1) より算出した値だけディスクへの読み取りを行う。達していなかった場合は 100% から提案ソフトウェア 1 から取得したディスクの帯域

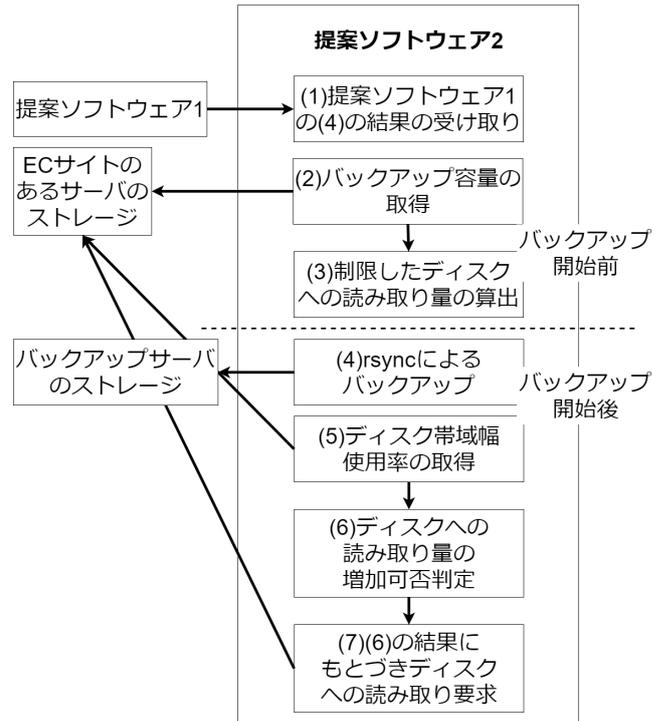


図 6 提案ソフトウェア 2 の処理の流れ

幅使用率までディスクへの読み取り量を増加させる。

5. 実験

基礎実験では EC サイトへのリクエストを送信する実験 (以下, 実験 1), EC サイトへのリクエスト送信とバックアップを同時に発生させる実験 (以下, 実験 2) を行う。リクエストの送信には Locust を用いた。Locust とは Python ベースのソフトウェアで, Web サイトやサービスのテストに役立つオープンソースの負荷テストツールである [9]。Locust から送信するリクエストのアクセスシナリオはサイト閲覧率は 52%, カート追加率は 44%, 購入率は 4% とした^{*3*4}。バックアップのデータ容量は 100GB とする。ビットレートを 800kbps とした場合, 3 時間の動画が 1GB である^{*5}。本稿では 3 時間の動画が 100 本ある EC サイトを想定する。実験 1, 実験 2 はどちらもバックアップにかかる時間だけ行う。Locust は実行から 1 分は挙動が安定しないため, 実行から 1 分経過後から計測を行う。これらの条件でそれぞれの実験を 5 回を行い, その平均の単位時間当たりのレスポンス数を比較し, 減少率を確認する。また, CPU 使用率, メモリ使用率, ネットワーク帯域幅使用率, ディスク帯域幅使用率を確認し, ボトルネックの特定を行う。ボトルネックは CPU 使用率, メモリ使用率, ネットワーク帯域幅使用率, ディスク帯域幅使用率の最も値の変

*3 Average ecommerce conversion rate benchmarks
[https://onl.sc/isy2JSm\(2023/06/29\)](https://onl.sc/isy2JSm(2023/06/29))

*4 What is Cart Abandonment?
[https://onl.sc/SGWPnyw\(2023/06/29\)](https://onl.sc/SGWPnyw(2023/06/29))

*5 動画の容量が 1GB だと何時間位の映像になりますか?
[https://faq.classstream.jp/42\(2023/07/19\)](https://faq.classstream.jp/42(2023/07/19))

化量が大きいのとする。

実験環境

図 7 に実験環境を示す。

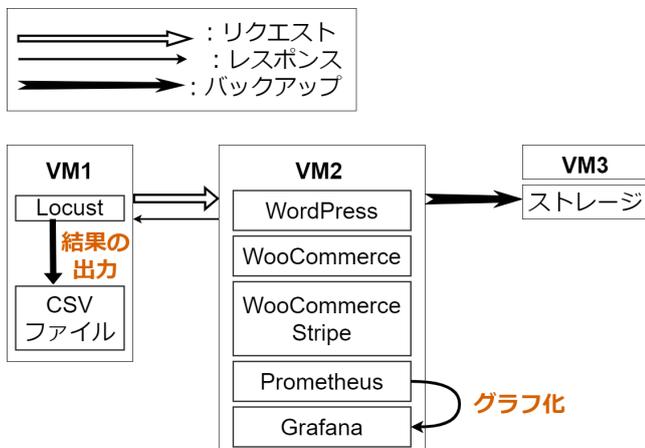


図 7 実験環境

実験 1 は VM1 と VM2, 実験 2 は VM1 と VM2, VM3 を用いてそれぞれ行う。VM1 には Locust を導入し, VM2 に対してリクエストを送信する。その実験結果のリクエスト数やレスポンス数を CSV ファイルに出力する。VM2 には WordPress で EC サイトを構築し, VM1 に対してレスポンスを返す。WordPress とは PHP プログラミング言語と MySQL または MariaDB リレーショナルデータベース管理システムをベースにした人気のある Web サイトのコンテンツ管理システムである [10]。EC サイトを構築する上で WooCommerce, WooCommerce Stripe を使用した。WooCommerce とは WordPress 上に構築された, 柔軟なオープンソースの電子商取引プラットフォームであり, 小規模から大規模のオンラインショッピング Web サイトの開発に適用される [11]。WooCommerce Stripe ゲートウェイとはクレジットカード支払いを実現するプラグインである [12]。また, Prometheus, Grafana をインストールし, CPU, メモリ, ネットワーク, ディスクの監視を行う。Prometheus とはオープンソースのシステム監視, 警告ソフトウェアであり, 時系列データによる多次元データを収集し, これらの次元を利用するための柔軟なクエリ言語を持つ。Grafana とはオープンソースの可視化および分析ソフトウェアであり, Grafana を使用すると, 保存されたメトリクスを視覚化し, リマインドし, 探索することができる [13]。VM3 は VM2 のバックアップ用のストレージである。基礎実験の環境では研究室にある ESXi 上に VM を作成した。以下に VM の構成要素を示す。

- VM1
OS Ubuntu 20.04.3
vCPU 4 コア

RAM 8GB

HDD 20GB

- VM2
OS Ubuntu 20.04.3
vCPU 6 コア

RAM 8GB

HDD 30GB

- VM3
OS Ubuntu 22.04
vCPU 2 コア

RAM 4GB

HDD 25GB

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
Locust-2.12.1
- VM2
WordPress v6.1.1
- VM2
Prometheus-2.37.1
Grafana v9.2.2

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

実験結果と分析

図 8 に実験 1, 実験 2, それぞれの単位時間当たりのレスポンス数を示す。

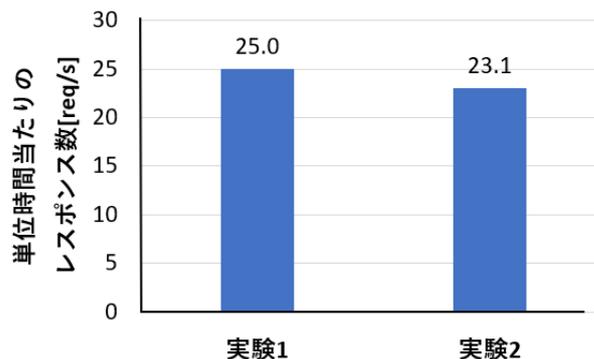


図 8 レスポンス数

縦軸はレスポンス数, 横軸は実験番号を示す。比較すると実験 1 が 25.0req/s, 実験 2 が 23.1req/s で実験 2 のほうが 7.6%減少していた。このことから単位時間当たりのレスポンス数はバックアップが原因で減少することがわかる。

図 9 に実験 1, 実験 2, それぞれの CPU 使用率を示す。縦軸は CPU 使用率, 横軸は時間を示す。実験 1, 実験 2

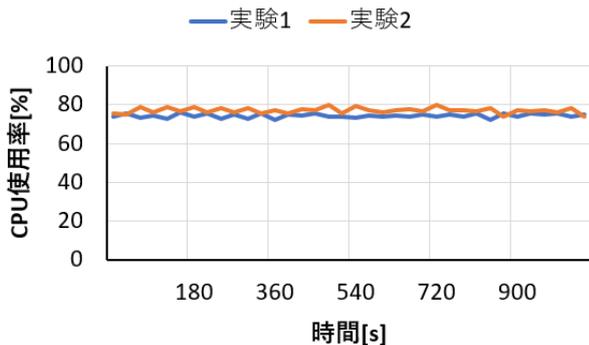


図 9 CPU 使用率

ともに CPU 使用率は 80% ほどを推移している。

図 10 に実験 1, 実験 2, それぞれのメモリ使用率を示す。

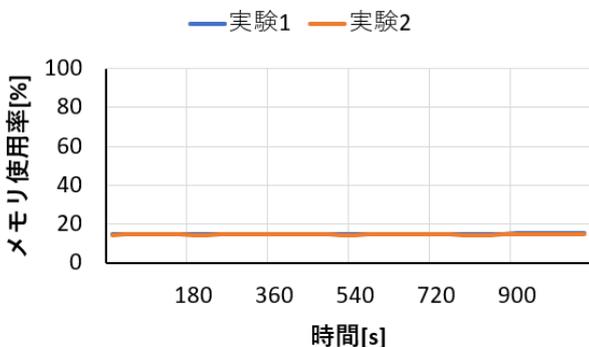


図 10 メモリ使用率

縦軸はメモリ使用率, 横軸は時間を示す。実験 1, 実験 2 ともにメモリ使用率は 15% ほどを推移している。

図 11 に実験 1, 実験 2, それぞれのネットワーク帯域幅使用率を示す。

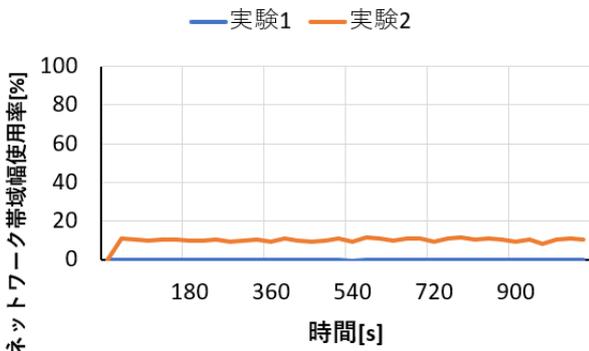


図 11 ネットワーク帯域幅使用率

縦軸はネットワーク帯域幅使用率, 横軸は時間を示す。VM は 1GB の LAN ケーブルにより接続されている。実験 1 では 1%, 実験 2 では 10% ほどを推移している。

図 12 に実験 1, 実験 2, それぞれのディスク帯域幅使用率を示す。

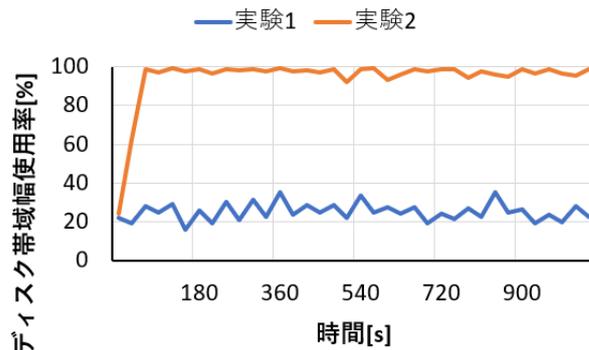


図 12 ディスク帯域幅使用率

縦軸はディスクの帯域幅使用率, 横軸は時間を示す。実験 1 のディスクの帯域幅使用率は 20% から 40% ほどを推移している。実験 2 では開始から 60 秒で 100% ほどになり, そのまま終了まで 100% ほどで推移している。

これらの結果から, CPU 使用率, メモリ使用率, ネットワーク帯域幅使用率, ディスクの帯域幅使用率を比較すると, 値の変化量が最も大きいディスクの帯域幅使用率がボトルネックと言える。

6. 議論

本稿ではリクエストとバックアップによる使用ディスク帯域幅使用率が 100% に達していないときはバックアップによるディスクへの読み取り量を増加させ, それ以外の場合はバックアップの期限を超過しないように, 式 (1) より算出した値だけディスクへの読み取りを行う提案を行った。しかし, バックアップ開始前に式 (1) より算出した値だけディスクへの読み取りを行う場合でも, 実際にはディスクへの読み取り量にばらつきが生じることにより, バックアップ期限通りにバックアップが終了しないことが問題となる。バックアップデータ容量を 100GB でバックアップをした結果, ディスクからの読み取り量の最大値が 96MB/s であった。そこでバックアップのデータ容量は 100GB, ディスクからの読み取り量の上限を 96MB/s として実験した結果, 3MB/s の誤差が出るのが分かった。バックアップ期限を超過する場合は, バックアップが正常に終了しないことにより, データの整合性が取れなくなる。バックアップ期限より早くバックアップが終了する場合は, ユーザの使用できるディスクの帯域幅が減少する。そこで, 制限したディスクへの読み取り量の計算をバックアップを行う前だけでなく, バックアップ中にも行うことによってディスクへの読み取り量のばらつきによる蓄積誤差を抑制し, バックアップ終了時刻をバックアップ期限ちょうどに近づけられる。計算頻度は制限したディスクへ

の読み取り量の誤差が蓄積する直前に行く。これは制限したディスクへの読み取り量の誤差の蓄積が、バックアップが1秒遅れることになるからである。

7. おわりに

課題はECサイトへのリクエスト送信とバックアップが同時に発生するとバックアップを行っていない時と比較して単位時間当たりのレスポンス数が減少することである。提案方式はECサイトのバックアップ期限とディスク帯域幅使用率にもとづくバックアップ制御によるレスポンス数の減少の抑制である。基礎実験ではECサイトへのリクエストを送信する実験、ECサイトへのリクエスト送信とバックアップを同時に発生させる実験をそれぞれ5回行い、その平均の単位時間当たりのレスポンス数で比較した。結果は前者の実験が25.0req/s、後者の実験が23.1req/sで7.6%減少していた。この単位時間当たりのレスポンス数はバックアップが原因で減少することがわかった。

参考文献

- [1] Pan, C.-L., Bai, X., Li, F., Zhang, D., Chen, H. and Lai, Q.: How Business Intelligence Enables E-commerce: Breaking the Traditional E-commerce Mode and Driving the Transformation of Digital Economy, *2021 2nd International Conference on E-Commerce and Internet Technology (ECIT)*, pp. 26–30 (online), DOI: 10.1109/ECIT52743.2021.00013 (2021).
- [2] Zheng, K. and Wang, X.: Dynamic Control of Flow Completion Time for Power Efficiency of Data Center Networks, *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 340–350 (online), DOI: 10.1109/ICDCS.2017.138 (2017).
- [3] Kim, S.-K.: Enhanced Maintenance Architecture for E-commerce Applications, *2009 International Conference on Computer and Automation Engineering*, pp. 3–7 (online), DOI: 10.1109/ICCAE.2009.10 (2009).
- [4] 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).
- [5] Soysal, M., Berghoff, M., Zirwes, T., Vef, M.-A., Oeste, S., Brinkmann, A., Nagel, W. E. and Streit, A.: Using On-Demand File Systems in HPC Environments, *2019 International Conference on High Performance Computing Simulation (HPCS)*, pp. 390–398 (online), DOI: 10.1109/HPCS48598.2019.9188216 (2019).
- [6] Huang, H., Hung, W. and Shin, K. G.: FS2: dynamic data replication in free disk space for improving disk performance and energy consumption, *ACM SIGOPS Operating Systems Review*, Vol. 39, No. 5, pp. 263–276 (2005).
- [7] Ni, Y. and Li, Z.: Minimizing the Response Time of a Striped Disk Array, *2010 Second International Conference on Intelligent Human-Machine Systems and Cybernetics*, Vol. 2, pp. 200–204 (online), DOI: 10.1109/IHMSC.2010.150 (2010).
- [8] Jensen, Q., Jagodzinski, F. and Islam, T.: FILCIO: Application Agnostic I/O Aggregation to Scale Scientific Workflows, *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 1587–1592 (online), DOI: 10.1109/COMPSAC51774.2021.00236 (2021).
- [9] Challapalli, S. S. N., Kaushik, P., Suman, S., Shiva-hare, B. D., Bibhu, V. and Gupta, A. D.: Web Development and performance comparison of Web Development Technologies in Node.js and Python, *2021 International Conference on Technological Advancements and Innovations (ICTAI)*, pp. 303–307 (online), DOI: 10.1109/ICTAI53825.2021.9673464 (2021).
- [10] Tomiša, M., Milković, M. and Čačić, M.: Performance Evaluation of Dynamic and Static WordPress-based Websites, *2019 23rd International Computer Science and Engineering Conference (ICSEC)*, pp. 321–324 (online), DOI: 10.1109/ICSEC47112.2019.8974709 (2019).
- [11] Dehghani, M.-H. and Kolahdouz-Rahimi, S.: An Automatic Generation of Android Application for WooCommerce, *2019 9th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 194–200 (online), DOI: 10.1109/ICCKE48569.2019.8964732 (2019).
- [12] Ravensbergen, R.: *Building E-commerce solutions with WooCommerce*, Packt Publishing Ltd (2015).
- [13] Rahman, D., Amnur, H. and Rahmayuni, I.: Monitoring server dengan prometheus dan grafana serta notifikasi telegram, *JITSI: Jurnal Ilmiah Teknologi Sistem Informatika*, Vol. 1, No. 4, pp. 133–138 (2020).