

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: 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実行された時に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。提案方式ではディスク帯域幅使用率に空きがある場合はディスクへの読み取り量を増加させ、ない場合は読み取り量を抑制させる。評価実験では 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行う実験 (以下、実験 2)、EC サイトへのリクエスト送信と提案ソフトウェアの実行を同時に行う実験 (以下、実験 3) を行う。バックアップするデータ容量は 600[GB]、バックアップ期限を 8 時間に設定し、各実験を 3 回ずつ行い、その平均の単位時間当たりのレスポンス数とバックアップにかかる時間で評価した。結果は実験 1 が約 22.6[req/s]、実験 2 が約 20.4[req/s]、実験 3 が約 22.4[req/s] であり、実験 1 を 100%とすると実験 2 は 90.3%、実験 3 は 99.1%となった。また、バックアップにかかった時間は実験 2 が約 110.7 分、実験 3 が約 192.6 分であり、実験 2 を 100%とすると実験 3 は 174.0%となったがバックアップを期限までに終了できた。

1. はじめに

背景

ネットワーク経済の急速な発展に伴い、B2C の EC サイトが出現し続けている [1]。オンラインビジネスウェブサイトが多数出現し、徐々にオンラインショッピングの主流の傾向になった。EC サイトではトラフィックは時間的に非常に複雑であり、長距離依存性と非定常性の両方を示すことがわかっている [2]。LINE LIVE は 2020 年 9 月 26 日に Official 髭男 dism のオンラインライブを配信し、アクセスが集中した*¹。そのため、顧客の満足度を満たすには、EC サイト用のサーバを常に利用できるようにする必要がある。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。構造情報および保存されたデータを含む定期的なバックアップにより、ハードウェア障害、バグによるデータ破損、システムクラッシュで失われたデータを復元できるようにしておく [4-8]。

高い計算速度と大規模なストレージ容量を備えたハイパフォーマンスコンピューティング (HPC) は科学シミュレーションやビッグデータ分析の科学技術コンピューティングに使用される [9]。その一方で計算量とデータ量が急速に増加しており、処理能力とストレージのパフォーマンスの歴史的なギャップが拡大している。アルゴンスの Intrepid システム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 4.3TB/s なのに対し、0.5TB/s の実行速度のパフォーマンスのディスクが搭載されている [10]。このような環境で I/O 操作にかかる時間を分析したところ、輻輳によりアプリケーションの I/O 効率が最大 70%低下することが示された。

課題

図 1 に課題となる状況を示す。ユーザは EC サイトへアクセスを行う。また、EC サイトがあるサーバはバックアップサーバにバックアップを行う。この時、ユーザから EC サイトへのリクエストによる書き込み処理とサーバからバックアップサーバへのバックアップによる読み取り処理が競合し、ユーザに対しての単位時間当たりのレスポンス数が減少する。ユーザに対してのレスポンス数の減少は売り上げの減少につながる*²。

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 「Official 髭男 dism ONLINE LIVE」のサーバダウンから学ぶ教訓
[https://onl.tw/WxJQUK3\(2023/10/29\)](https://onl.tw/WxJQUK3(2023/10/29))

*² Online Experiments: Lessons Learned
<http://robotics.stanford.edu/~ronnyk/2007IEEEComputerOnlineExperiments.pdf> (2023/07/17)

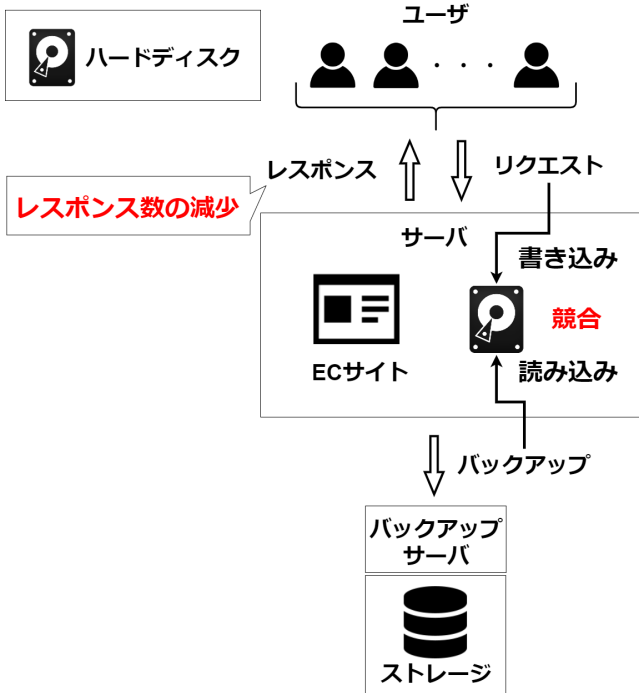


図 1 課題の概要

基礎実験では EC サイトへ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行う実験を行った [11]. これらの実験では単位時間当たりのレスポンス数を計測し、前者の実験から後者の実験への減少率を確認した. 結果では、単位時間当たりのレスポンス数は実験 1 が約 25.0[req/s], 実験 2 が約 23.1[req/s] となり、実験 1 を 100% とすると実験 2 は 92.4% となった. 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.

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 章では評価実験の実験結果と分析を述べる. 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

Linux オペレーティングシステムのデフォルトのディスクスケジューラである Complete Fair Queuing (CFQ) に変更を加えた新しいディスクスケジューラの Prioritized Complete Fair Queuing (PCFQ) による、優先度の高いプログラムの実行時間の削減を提案している研究がある [12]. プログラムの優先順位とタイムスタンプによりサービスツリーの構築を行うことですべてのディスク要求がホスティングプログラムのディスク I/O 優先順位に従って処理されることを保証している. しかし、本稿ではランダムに実行される複数のプログラムを用いないため、本稿の課題は解

決できない.

オンボードディスクキャッシュを圧縮して有効なキャッシュサイズを増やし、ディスクのパフォーマンスを向上させることを提案している研究がある [13]. オンボードディスクキャッシュのプリフェッチはディスクのパフォーマンスに大きな影響を与えるため、複雑さと圧縮の副作用を最小限に抑えるために、プリフェッチされたデータのみを圧縮している. しかし、ユーザに転送するデータは圧縮しないため、本稿の課題は解決できない.

ファイルサーバを調整して一度に 1 つのアプリケーションにサービスを提供することで完了時間を短縮し、その間のサーバの使用率と公平性の維持を提案している研究がある [14]. ファイルサーバ上で I/O リクエストを再配置し、関連するすべてのノード上のアプリケーションに関して同じ順序でリクエストが処理されるようにすることで、すべての I/O リクエストの平均完了時間を短縮している. しかし、並列ファイルシステムにおける課題、提案であるため本稿の課題は解決できない.

待ち行列の長さ依存するサービス時間を持つ M/G/1 待ち行列の応答時間分布に対する新しい近似を提示し、I/O 要求の並び替えを実装したゾーン化ディスクドライブの具体的なケースに適用した研究がある [15]. 想定される最大待ち行列長を大きくした場合の到着率やリクエストサイズに依存しない平均応答時間について、モデルと実測は一致している. しかし、読み込みと書き込みが混合しているリクエスト、様々なサイズのリクエストにはモデルが対応できないため、本稿の課題は解決できない.

3. 提案

提案方式

図 2 に提案方式の概要を示す. 図 2 はユーザからの EC サイトへのリクエスト送信とサーバからバックアップサーバに対してのバックアップが同時に実行された状況を示した図である. 提案手法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行う. ディスクへの書き込み量、ディスク帯域幅使用率はユーザからのリクエストの量や内容によって変化する. それに伴い、提案手法ではバックアップによるディスクへの読み取り量が増減させることで、ディスク帯域幅使用率が 100% になるのを防ぐ. その結果、課題の状況と比較してユーザへの単位時間当たりのレスポンス数を増加させる. バックアップ期限は企業の業務開始時刻とする.

次に本稿の提案方式について (1) 最低限読み取るディスクへのデータ量の計算, (2) ディスク帯域幅使用率の確保, (3) バックアップによる読み取り量の制御の 3 つに分けて説明する.

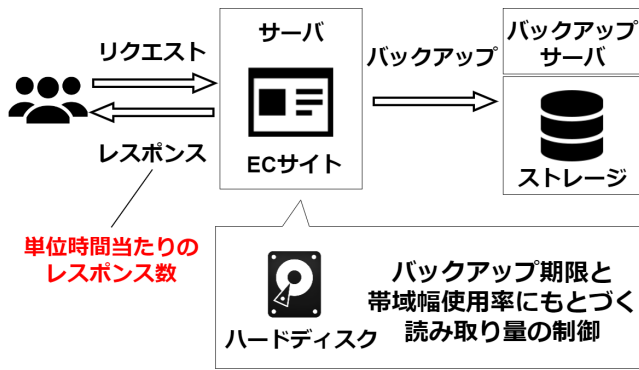


図 2 提案方式の概要

(1) 最低限読み取るディスクへのデータ量の計算

本提案ではバックアップを期限までに終わらせることが目的の1つである。そこで、バックアップが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を用い、最低限読み取るディスクへのデータ量を求める。最低限読み取るディスクへのデータ量の計算方法を以下に示す。最低限読み取るディスクへのデータ量を M [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。

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

(2) ディスク帯域幅使用率の確保

本提案では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。そのため、その間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。これを防ぐために、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の時間で、サーバが処理できる最大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く、このディスク帯域幅使用率を確保して残しておく。

図3に確保しておくディスク帯域幅使用率の決定の仕方の例を示す。リクエスト増加検知後にディスクへの読み取

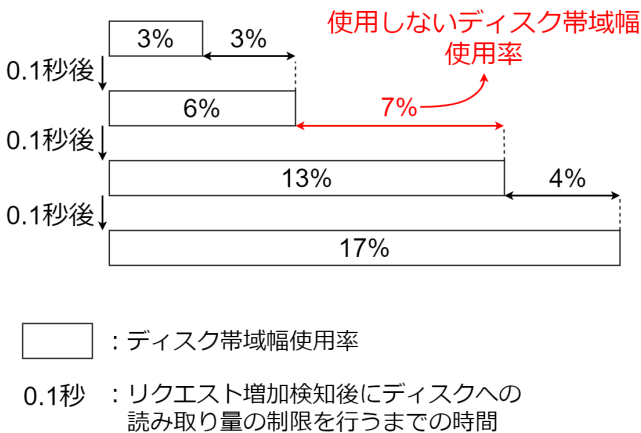


図 3 確保しておくディスク帯域幅使用率の決定の仕方の例

り量の制限を行うまでの時間でサーバが処理できる最大リクエストを送信した時のディスク帯域幅使用率が図3のようになっている時、0.1秒後のディスク帯域幅使用率との差を求める。その中で最大のディスク帯域幅使用率の差である7%を確保しておくディスク帯域幅使用率とする。

(3) バックアップによる読み取り量の制御

ユーザからのリクエストとバックアップによるディスク帯域幅使用率が100%から(2)で説明した値を減算したディスクの帯域幅使用率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。その結果、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帯域幅使用率をすべてリクエストによるディスクへの書き込みに確保できる。

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。図4にユースケースシナリオを示す。

深夜

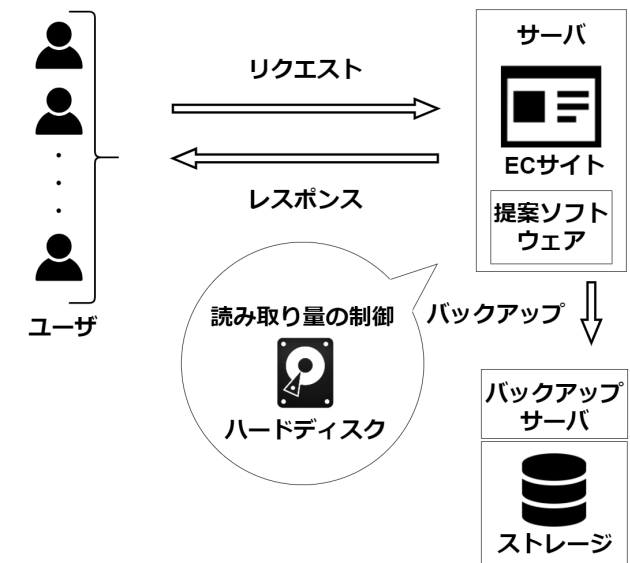


図 4 ユースケースシナリオ

深夜にユーザはECサイトで商品の閲覧や購入というリクエストを行い、ECサイトを構築したサーバはユーザからのリクエストの処理を行う。また、サーバはバックアップサーバにECサイトのデータベースのバックアップを行う。そのため、ECサイトを構築したサーバのディスクへの読み書きが同時に行われ、ディスク帯域幅使用率が100%になることで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とで、ディスクへの読み取り量を制御し、バックアップを期限までに終わらせることができる。また、ユーザか

らのリクエストが確保できるディスク帯域幅使用率を増加させることができ、ユーザへの単位時間当たりのレスポンス数の減少を抑制できる。

4. 実装

本稿では提案方式を実装したソフトウェアである `caord.sh` と `dadlt.py` の2つを開発した。どちらも EC サイトを構築したサーバに実装した。図5にソフトウェアの概要を示す。`dadlt.py` は EC サイトを構築したサーバが処理できる最大

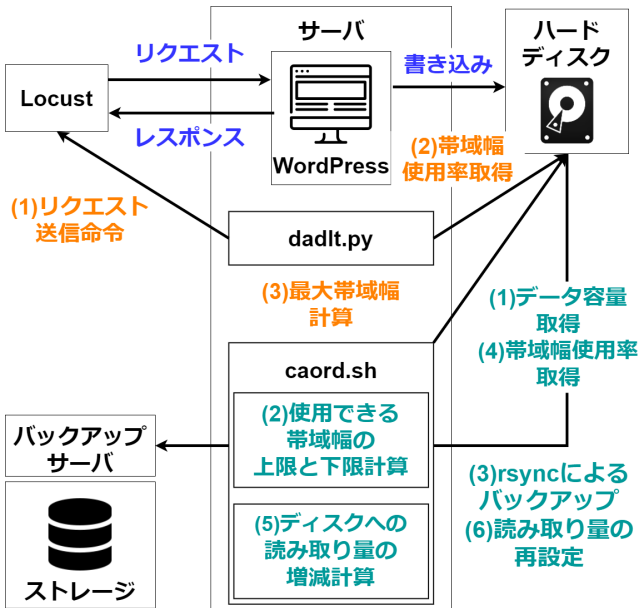


図5 ソフトウェアの概要

秒間リクエストで増加するディスク帯域幅使用率をバックアップ開始前に計測する。EC サイトへのリクエスト送信には 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 [16]。dadlt.py の処理を (1), (2), (3) に分けて説明する。

- (1) Locust によって EC サイトへリクエストを送信する。
- (2) その時のディスク帯域幅使用率を計測しておく。
- (3) リクエストの増加検知後にディスクへの読み取り量を制限を行うまでの時間での最大ディスク帯域幅使用率を計測し、その最大ディスク帯域幅使用率をファイルに書き出す。

`caord.sh` はリクエストとバックアップによるディスク帯域幅使用率が 100% から `dadlt.py` で計測した値を減算した値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。`caord.sh` の処理を (1), (2), (3), (4), (5), (6) に分けて説明する。

- (1) バックアップするデータ容量を取得する。

- (2) (1) の値をバックアップ開始から期限までの時間で割ることによりディスク帯域幅使用率の下限を算出する。また、100% から送信された値を減算し上限を算出する。
- (3) `rsync` コマンドによってバックアップを開始する。
- (4) ディスク帯域幅使用率を取得する。
- (5) ディスク帯域幅使用率の上限から (4) で取得したディスク帯域幅使用率を減算する。
- (6) (5) の値にディスクへの読み取り量を再設定する。

5. 評価実験

評価実験では 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実行させる実験 (以下、実験 2)、EC サイトへのリクエスト送信と提案ソフトウェアを同時に実行させる実験 (以下、実験 3) を行う。リクエストの送信には Locust を用いた。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*}。バックアップのデータ容量は 600[GB] とした^{*5}。バックアップ期限は 8 時間とした^{*6}。実験 1、実験 2 は実験 3 でバックアップにかかる時間だけ行う。確保しておくディスク帯域幅使用率は約 6.67% とした。これらの条件でそれぞれの実験を 3 回行い、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、実験 2、実験 3 ではバックアップにかかる時間を計測する。

実験環境

図6に実験環境を示す。実験1は VM1 と VM2、実験

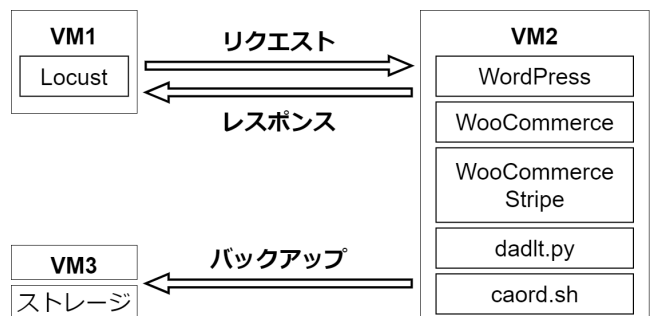


図6 実験環境

2、実験3は VM1 と VM2、VM3 を用いてそれぞれ行う。VM1 には Locust を導入し、VM2 の WordPress で作成

*3 Average ecommerce conversion rate benchmarks
[https://onl.sc/isy2J5m\(2023/06/29\)](https://onl.sc/isy2J5m(2023/06/29))

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

*5 SHISEIDO Taiwan
[https://x.gd/eqzyq\(2023/11/12\)](https://x.gd/eqzyq(2023/11/12))

*6 バックアップの使用 - Amazon Relational Database Service
[https://x.gd/5d1wt\(2023/10/21\)](https://x.gd/5d1wt(2023/10/21))

したサイトに対してリクエストを送信する。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 [17]。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 [18]。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 [19]。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。表1にVMの構成要素を示す。

表 1 VMの構成情報

VM	OS	vCPU(コア)	RAM(GB)	HDD(GB)
VM1	Ubuntu-20.04	4	8	20
VM2	Ubuntu-20.04	8	32	800
VM3	Ubuntu-22.04	2	4	700

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

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7に各実験の評価実験の単位時間当たりのレスポンス数を示す。縦軸は単位時間当たりのレスポンス数を表

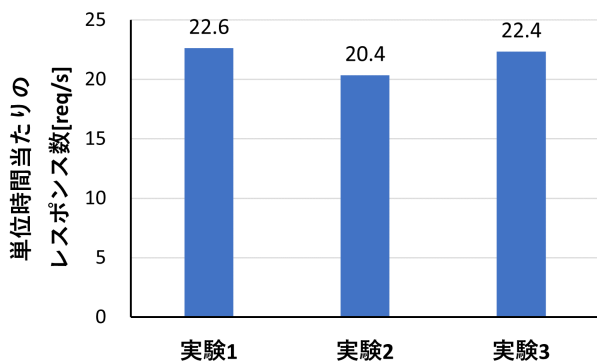


図 7 各実験の単位時間当たりのレスポンス数

している。横軸は実験の種類を表している。単位時間当たりのレスポンス数は実験1では約22.6[req/s]あったが、実験2では約20.4[req/s]であった。実験1と比較すると2.2[req/s]減少しており、実験1を100%とすると実験2は90.3%となった。しかし、実験3では約22.4[req/s]であった。実験1と比較すると減少は0.2[req/s]であり、実験1を100%とすると実験3は99.1%となった。これは、バックアップでのディスクへの読み取り量の制御により、ディスクへの書き込みが確保できるディスク帯域幅使用率が増加したためである。

図8に実験2と実験3のバックアップにかかった時間を示す。縦軸はバックアップにかかった時間を表している。

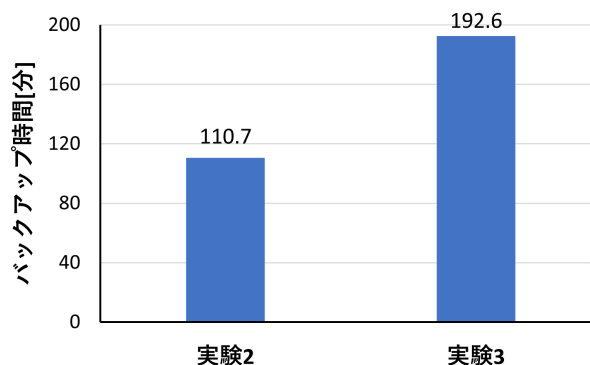


図 8 実験2、実験3のバックアップにかかった時間

横軸は実験の種類を表している。バックアップにかかった時間は実験2は約110.7分で実験3は約192.6分であった。実験2と実験3を比較するとバックアップにかかった時間は実験3のほうが81.9分延長されており、実験2を100%とすると実験3は174.0%となったが、バックアップ期限を超過しなかった。しかし、バックアップ期限までには時間が約287.4分残っている。これは、提案の(1)で求めた値を常にディスクに対して読み取り、リクエストが確保できるディスク帯域幅使用率が実験1の場合より減少したためである。この時間はバックアップにかかる時間をさらに延長することで、単位時間当たりのレスポンス数の減少をさらに抑制できる。また、バックアップを期限までに終わらせることにより、バックアップ期限までの時間はリクエストによる書き込みがディスク帯域幅使用率を100%確保できる。

6. 議論

本稿では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の時間で、サーバが処理できる最大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、このディスク帯域幅使用率を確保し、残しておくという提案を行った。これは、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とを避けるためである。しかし、本番環境で事前に計測した値以上のディスク帯域幅使用率にならないとは言えず、事前に計測した値以上のディスク帯域幅使用率に達した時にディスク帯域幅使用率が100%を超え単位時間当たりのレスポンス数が減少する。そこで、本稿で提案するディスク帯域幅使用率は初期値に設定し、本番環境でも1秒ごとにディスク帯域幅使用率を監視し、初期値を上回った時に更新するようにする。監視はiostatを用いるため、設定できる最小監視間隔の1秒とする。その結果、ディスク帯域幅使用率が設定された値を超過する値に達しても2回目以降は単位時間当たりのレスポンス数が減少することはない。

評価実験の結果、バックアップを期限内に終了できたが、単位時間当たりのレスポンス数は実験1が約22.6[req/s]、実験3は約22.4[req/s]であり、2.2[req/s]減少していた。原因は提案の(1)で求めた値を常にディスクに対して読み取り、ディスク帯域幅使用率が増加することにより、リクエストによるディスク帯域幅使用率が実験1の場合より減少したことである。1秒ごとに残りファイルサイズをバックアップ期限までの時間で割ることにより、ディスク帯域幅使用率に空きがあり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られた場合に最低限読み取るディスクへのデータ量を減少させることができる。よって、リクエストによるディスク帯域幅使用率を増加させることができ、単位時間当たりのレスポンス数の減少を抑制できる。

本稿ではディスクへの書き込み量、ディスク帯域幅使用率はユーザからのリクエストの量や内容によって変化するため、それに伴い提案手法ではバックアップによるディスクへの読み取り量が増減させることを提案した。しかし、評価実験の結果、単位時間当たりのレスポンス数の減少をすべて抑制できなかった。原因はリクエストが確保しないディスク帯域幅使用率をバックアップが余分に確保していることである。そこで、提案(1)で求めた値を常にディスクに対して読み取りを行うことでディスク帯域幅使用率を最低限に抑え、単位時間当たりのレスポンス数の減少を抑制できる。

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単位時間当たりのレスポンス数の減少の抑制である。評価実験では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行う実験、ECサイトへのリクエスト送信と提案ソフトウェアの実行

を同時に行う実験を行い、平均の単位時間当たりのレスポンス数とバックアップにかかる時間で比較した。結果は実験1が約22.6[req/s]、実験2が約20.4[req/s]、実験3が約22.4[req/s]で、実験1を100%とすると実験2は90.3%、実験3は99.1%となった。バックアップにかかった時間は実験2が約110.7分、実験3が約192.6分であり、実験2を100%とすると実験3は174.0%となったがバックアップを期限までに終了できた。

参考文献

- [1] Zhao, Y.: Evaluation model of B2C e-commerce site based on consumer Perspective, *2011 International Conference on Computer Science and Service System (CSSS)*, pp. 2230–2232 (online), DOI: 10.1109/CSSS.2011.5974611 (2011).
- [2] Kant, K. and Venkatachalam, M.: Transactional characterization of front-end e-commerce traffic, *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, Vol. 3, pp. 2523–2527 vol.3 (online), DOI: 10.1109/GLOCOM.2002.1189085 (2002).
- [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] Kim, H., Yeom, H. Y. and Son, Y.: An Efficient Database Backup and Recovery Scheme using Write-Ahead Logging, *2020 IEEE 13th International Conference on Cloud Computing (CLOUD)*, pp. 405–413 (online), DOI: 10.1109/CLOUD49709.2020.00062 (2020).
- [5] Bhattacharya, S., Mohan, C., Brannon, K. W., Narang, I., Hsiao, H.-I. and Subramanian, M.: Coordinating backup/recovery and data consistency between database and file systems, *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pp. 500–511 (2002).
- [6] Amvrosiadis, G. and Bhadkamkar, M.: Identifying trends in enterprise data protection systems, *2015 USENIX Annual Technical Conference (USENIX ATC 15)*, pp. 151–164 (2015).
- [7] Amvrosiadis, G. and Bhadkamkar, M.: Getting back up: Understanding how enterprise data backups fail, *2016 USENIX Annual Technical Conference (USENIX ATC 16)*, pp. 479–492 (2016).
- [8] Kaiser, J., Süß, T., Nagel, L. and Brinkmann, A.: Sorted deduplication: How to process thousands of backup streams, *2016 32nd Symposium on Mass Storage Systems and Technologies (MSST)*, IEEE, pp. 1–14 (2016).
- [9] Zha, B. and Shen, H.: Memory Hierarchy Aware I/O Scheduling under Contention for Hybrid Storage Based HPC, *2018 9th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*, pp. 69–73 (online), DOI: 10.1109/PAAP.2018.00019 (2018).
- [10] Gainaru, A., Aupy, G., Benoit, A., Cappello, F., Robert, Y. and Snir, M.: Scheduling the I/O of HPC Applications Under Congestion, *2015 IEEE International Parallel and Distributed Processing Symposium*, pp. 1013–1022 (online), DOI: 10.1109/IPDPS.2015.116 (2015).
- [11] 井田尚樹, 高橋風太, 串田高幸: 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, Tokyo University of Technology CDSL Technical Report, CDSL-TR-

- 149, Jul.21 2023. (online: <https://ja.tak-cslab.org/tech-report>).
- [12] Yeh, T. and Yang, S.: Improving the program performance through prioritized disk operation, *2012 International Conference on High Performance Computing Simulation (HPCS)*, pp. 574–580 (online), DOI: 10.1109/HPCSim.2012.6266976 (2012).
 - [13] Jiang, W., Deng, Y., Meng, X., Hu, C. and Zhou, Y.: Boosting Disk Performance by Compressing On-board Disk Cache, *2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 98–105 (online), DOI: 10.1109/HPCC-SmartCity-DSS.2017.13 (2017).
 - [14] Song, H., Yin, Y., Sun, X.-H., Thakur, R. and Lang, S.: Server-side I/O coordination for parallel file systems, *SC '11: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–11 (2011).
 - [15] Lebrecht, A. S., Dingle, N. J. and Knottenbelt, W. J.: A Performance Model of Zoned Disk Drives with I/O Request Reordering, *2009 Sixth International Conference on the Quantitative Evaluation of Systems*, pp. 97–106 (online), DOI: 10.1109/QEST.2009.31 (2009).
 - [16] 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).
 - [17] 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).
 - [18] 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).
 - [19] Ravensbergen, R.: *Building E-commerce solutions with WooCommerce*, Packt Publishing Ltd (2015).