

# バックアップのディスク読み取り速度の制限による 許容時間内のリストア完了

石川裕人<sup>1</sup> 高橋風太<sup>2</sup> 串田 高幸<sup>1</sup>

**概要：**SaaS アプリケーションの開発者は過去 1 年以内に約 53 %の割合でデータ紛失や破損をしたことがあり、深夜までの長時間労働が人為的ミスの原因となっている。データを紛失した際はバックアップされていたデータをリストアする必要がある。企業は業務の妨げにならないようするため、業務時間外の深夜にバックアップを行う。したがって、深夜に紛失したデータをリストアする場合、バックアップと時間帯が重複する。課題は、バックアップが期限を超過しない条件でバックアップ中のリストア完了時間を短縮する必要があることである。提案方式では、問題解決や作業回復に費やした時間がシステムの安定時に想定される平均的な完了時間と比較して 38 %増加するとフラストレーションを抱えることから、データサイズごとにリストアを単体で行った場合の完了時間を算出する。算出した時間に 38 %を加えた時間を許容時間とし、リストアが許容時間内に完了するようにバックアップにおけるディスク読み取り速度を制限する。そのため、データサイズにもとづいてリストアに必要なディスク読み取り速度を算出する。この速度を確保するために、バックアップの制限速度は、最大のディスク読み取り速度とリストアに必要なディスク読み取り速度の差とする。評価実験では、提案方式をもちいてバックアップ中にリストアを行った際、リストア完了時間が許容時間内(約 190 秒)に収まり、リストア完了時間を短縮できた。提案方式無しのリストア完了時間は約 380 秒、提案方式有りのリストア完了時間は約 187 秒で完了した。一方、バックアップ完了時間は提案方式無しの約 415 秒から提案方式有りは約 553 秒であり、約 33 %増加したが、バックアップの期限内に完了した。さらに、本稿のユースケースで提案方式をもちいた場合、データサイズが約 556GB 以内であれば追加のバックアップ要求をされてもバックアップの期限を超過しない見込みとなる。

## 1. はじめに

### 背景

バックアップとは、紛失や破損したデータを復元する目的で作成および保持される、データのコピーである [1-3]。また、バックアップからデータやシステムを復元するためのリストアは、欠かせないプロセスである [4]。リストアとは、データのバックアップを保存しているストレージから、データを取得することである [5]。

東京工科大学コンピュータサイエンス学部にある研究室の Cloud and Distributed Systems Laboratory(以下 CDSL)では、Broadcom 社の VMware 製品であるハイパーバイザーの ESXi を使用して仮想マシン (以下 VM) を作成している。CDSL 所属の学生は、ESXi がインストールされた 7 台の物理マシン (以下 PM) 上で、自身の VM を利用するこ

とができる。CDSL では平日の午前 7 時から午後 7 時の間に研究室で作業を行うことができる。そして、バックアップは翌日の午前 3 時から行い、研究室が開く午前 7 時までに完了する必要がある。CDSL で行っているバックアップとして、毎日各 PM に入っている VM のスナップショットを作成し、保存している。スナップショットとは、ある時点で凍結されたデータセットの読み取り専用コピーを指す技術であり、アプリケーションが引き続きデータへの書き込みを行えるようにするものである [6]。このコピーは、VM の状態を記録するものであり、具体的には、メモリの内容、ディスクの状態、および CPU の実行状態を含む。これにより、データを破損した際に、スナップショットを作成時の状態に戻すことができる手段として利用される [7-9]。

バックアップにより、保存されたデータをいつでも復元できる [10]。そして、サイバー攻撃やデータの盗難、システム障害の発生件数が過去数十年と比較して増加しており、バックアップの重要性が高まっている [11-13]。COVID-19 によるリモートワークの増加に伴い、オフィスや拠点から

<sup>1</sup> 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

<sup>2</sup> 東京工科大学院バイオ・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

離れた場所で働く人たちのデータを守る方法が必要になっている [14]. 特に気を付けなければならないのが、人為的ミスでデータが損失、破損することである。バックアップは、企業規模に関わらずストレージ管理の一環として行われており、データの損失を防ぐために欠かせない取り組みである。企業にとってデータは顧客情報や財務記録、業務の基盤となるものであり、その損失は事業の停止、信頼の喪失を招くため、許容できない [15].

調査によると、SaaS アプリケーションを開発しているエンジニアは過去 1 年間で 53 % の割合でデータの紛失や破損をしたことがある [16]. また、データの紛失や破損をしたことがある人の約 74 % は、その理由がエラーや誤用、人為的要因にあると報告されている\*1. 例として、深夜までの長時間労働による睡眠不足が人為的ミスの原因となっている [17]. また、アニメ業界は月間 225 時間以上の労働が常態化している\*2. そして、突然のスケジュール変更や深夜作業が避けられない状況が続いている\*3. これにより、人為的ミスが増加し、データの紛失や破損のリスクが高まっている。データが紛失、破損した場合、リストアが必要となり、深夜にバックアップデータをもちいたリストアが行われる。しかし、企業は、一般的にバックアップを業務時間外の深夜に行う [18]. そのため、深夜に紛失したデータをリストアする場合、バックアップと時間帯が重複する。

バックアップとリストアを同時実行することでリストアによる書き込み速度とバックアップによる読み取り速度が競合することでリストア完了時間とバックアップ完了時間が増加する [19].

## 課題

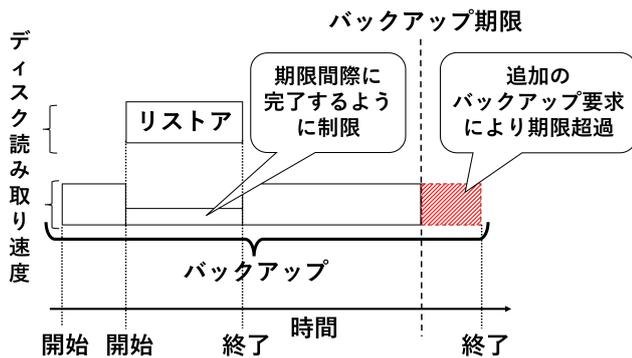


図 1: 課題の概要

本稿の課題を図 1 に示す。課題は、バックアップが期限を超過しない条件でバックアップ中のリストア完了時間を短縮する必要があることである。リストア完了時間の増加はユーザのフラストレーションにつながる。バックアップ

\*1 <https://00m.in/ZnjpJ>

\*2 <https://00m.in/Ov sob>

\*3 <https://www.kartworks.com/guide/sche.html>

の中断や制限によりバックアップ中のリストア完了時間を短縮できるが、バックアップの期限は厳守する必要がある。また、リストア完了時間を最大限短縮するためにバックアップを期限間際に完了するよう制限した場合、バックアップ中にユーザが転送したファイルを追加でバックアップする際、バックアップが期限を超過する。したがって、ユーザによりファイルが転送されるタイミングやファイルサイズが予測できない状況で、バックアップの期限を超過させずにリストア完了時間を短縮する必要がある。

## 各章の概要

本稿は以下のように構成される。第 2 章では、本稿の関連研究について述べる。第 3 章では、本稿での課題を解決するための提案方式について述べる。第 4 章では、提案方式の実装について述べる。第 5 章では、評価実験の内容とその分析について述べる。第 6 章では、提案方式の議論を述べる。第 7 章では、本稿のまとめを述べる。

## 2. 関連研究

既存の VM リストア技術では、チェックポイントイメージのサイズ縮小に成功しているが、リストア完了時間が増加する課題が残っている。この問題に対処するため、新たなアプローチが提案されている [20]. この研究では、既存の重複メモリページの除去手法に代わり、lazyfetch(遅延フェッチ)とアクセスインターセプト技術を活用し、チェックポイントイメージのサイズを小さくするために、リストア時にアクセスされるメモリページのみを保存し、使用されないメモリページを除外している。また、リストアに必要なデータとその時点で取得することで、リストアにかかる時間を短縮している。しかし、この手法は主に VDC 環境での仮想マシンをリストアすることを対象としている。VDC 環境とは Virtual Data Center(仮想データセンター)の略である。主にクラウドサービスプロバイダーや大規模な IT インフラを運用する企業で利用されている。また、他の環境や異なる種類のシステム障害への適用は限られているため、本稿の課題を解決することはできない。

既存のクラッシュ防止に比べて短時間でリストアを優先する設計が、低コストで高い可用性を提供する上で有効であることが示されている。この課題に対応するため、UNIX に類似した分散型 OS である Sprite では、リカバリーボックスを使用したリストア手法が提案されている [21]. リカバリーボックスは、システムの重要部分を保持する安定したメモリ領域であり、チェックサムを利用してエラーを検出し、リストア時には本来の再起動プロセスに戻す。Sprite ファイルサーバの実装では、システムは約 26 秒、データベースマネージャは約 6 秒でリストアが完了し、ユーザやアプリケーションがシステムダウンを意識

しないほどのリストアが可能である。しかし、この設計は Sprite システム専用であり、他のシステムには適用できない欠点がある。

データ重複排除はバックアップシステムで広く使用されており、冗長データの量を削減する手法として有用であることが示されている。元のバックアップデータを再構築するために、インライン重複排除の一般的な手順でリストアが行われ、重複チャンクが異なるデータストリーム間で共有されている場合には、読み取り増幅が生じる。この研究では、バックアップシステムにおけるリストアパフォーマンスを改善し、高い重複排除率を保証するリージョンパーティショニングにもとづく書き換え手法 (ERP) を提案している [22]。ERP の主要なアイデアは、コンテナ間の冗長性情報を調査し、リストアにおいて必要な範囲を絞り込むことで、適切なコンテナ数を選択することである。しかし、この手法はデータ重複排除とリストアにおいて有用であるが、バックアップを行っている最中のリストア完了時間の増加を抑制する課題には対応していない。したがって、本稿の課題を解決することはできない。

CABdedupe と呼ばれる因果関係にもとづく重複排除パフォーマンスブースターを提案している研究がある [23]。これは、複数のバックアップ、リストアで行われるデータセットの時系列バージョン間の因果関係を捕捉し、バックアップ操作だけでなくリストア操作中にも送信から変更されていないデータを削除して、バックアップとリストアの両方のパフォーマンスを向上させる。結果として、削減率は最大約 99.03 % であり、バックアップ完了時間とリストア完了時間の両方が短縮されたことを示されている。しかし、CABdedupe は、因果関係ベースのシステムのため、データバージョン間の因果関係に関するメタデータを保存する追加のメモリが必要になる。その結果、既存の重複排除手法と比較して、メモリオーバーヘッドが増加し、メモリが限られている環境ではバックアップやリストアが完了するまでの時間が増加する欠点がある。

### 3. 提案

本稿の目的は、バックアップの期限を超過しない条件でバックアップ中にリストア完了時間を短縮することである。提案方式は、単体で行ったリストア完了時間から予測完了時間を算出し、その時間に 38 % を加えた時間を許容時間として設定する。その許容時間の上限にリストアを完了させるためにバックアップにおけるディスク読み取り速度を制限し、リストアを行う。

#### 提案方式

本稿の提案方式として、バックアップにおけるディスク読み取り速度をリストアが完了するまで制限し、リストアを単体で行った時間とその時間に 38 % を加えた時間を許

容時間とし、その許容時間の上限でリストアを完了させる。38 % の理由は、問題解決や作業回復に費やした時間がシステムの安定時に想定される平均的な完了時間と比較して 38 % 増加すると、ユーザは「思ったよりも時間がかかる」と感じ、フラストレーションを抱えるためである [24]。また、制限の理由は、バックアップを中断すると、リストアするデータサイズによってバックアップが期限までに完了できないためである。

許容時間の上限でリストアを完了させる理由は、バックアップとリストアを同時に行う際にリストアを優先するためにディスク読み取り速度を最大まで使用してしまうと、追加でバックアップ要求が発生した場合、バックアップが期限までに完了できないためである。そのため、リストアを許容時間の上限で完了させ、バックアップに余裕を持たせることで、追加でバックアップ要求が発生した場合でもバックアップを期限までに完了させる。

提案方式の流れを図 2 に示す。

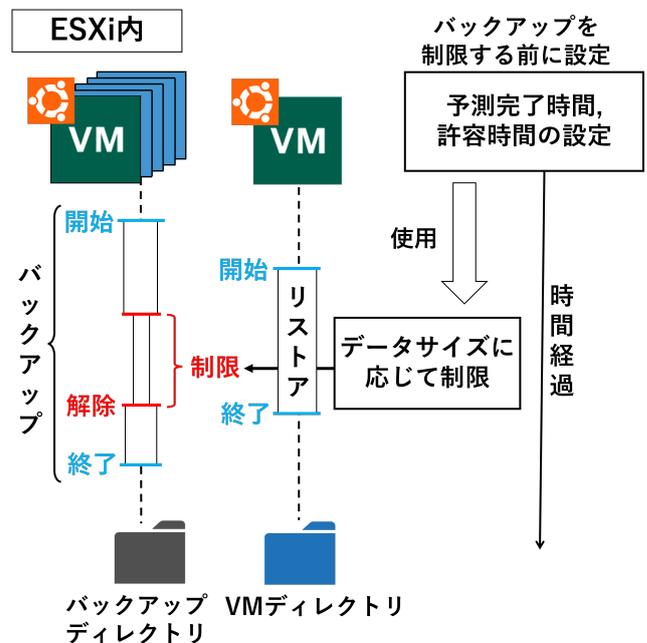


図 2: 提案方式の流れ

提案方式の流れとして、バックアップにおけるディスク読み取り速度を制限する前に、リストアにおける予測完了時間と許容時間を設定する。予測完了時間とは、リストアが完了すると予測される時間のことである。許容時間とは、リストアにおける予測完了時間に 38 % 分を加えた時間のことである。リストアにおける予測完了時間と許容時間を設定した後、これらの時間をもちいて、データサイズに応じてバックアップにおけるディスク読み取り速度を制限する。また、リストア完了後、バックアップにおけるディスク読み取り速度の制限を解除する。

リストアにおける予測完了時間を設定するために実験を

行う。実験では、ファイル転送時間をデータサイズごとに算出する。この実験では、ファイル転送を行うデータサイズを 10GB から 100GB まで 10GB 間隔で変化させ、それぞれ 10 回ずつ測定する。この実験により、データサイズとファイル転送時間の関係から 1GB あたりのファイル転送時間の増加率を求める。この増加率をもちいて、データサイズからリストアを単体で行った場合の完了時間を算出する。データサイズを 10GB ごとの間隔にした理由は、網羅的にデータサイズを変化させてファイル転送時間の測定を行うことが望ましいが、5GB 間隔では測定回数が増加し、実験時間が長くなり、15GB 間隔で行うと 10GB 間隔より測定間隔が広くなってしまい、ファイル転送時間とデータサイズの関係を近似する際の精度が低下し、ファイル転送時間の変動を十分に把握できないためである。また、ファイル転送を行うデータサイズの上限を 100GB とした理由は、映像制作会社をユースケースとする CDSL の研究で 100GB のファイルが使われているためである [25]。

実験結果として、ファイル転送時間とデータサイズの関係が得られた。ファイル転送時間とデータサイズの間を線形近似で図 3 に示す。

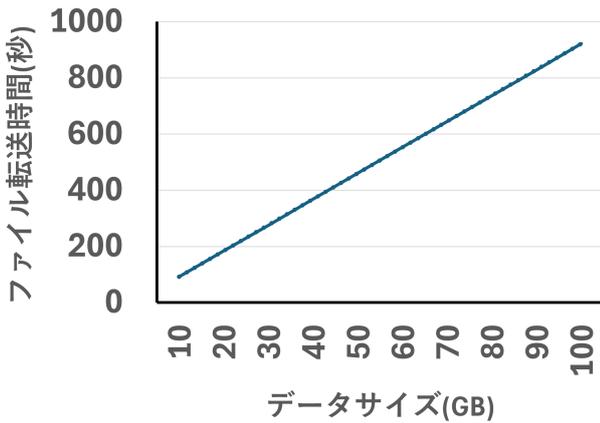


図 3: ファイル転送時間とデータサイズの関係

図 3 のデータをもとに、リストア完了時間がデータサイズに対して 1GB あたりどの程度増加するかを求める。増加率の求め方は、式 (1) の最小二乗法を使用する。

$$a = \frac{n \sum_{i=1}^n (S_i T_i) - \sum_{i=1}^n S_i \sum_{i=1}^n T_i}{n \sum_{i=1}^n S_i^2 - (\sum_{i=1}^n S_i)^2} \quad (1)$$

式 (1) の  $S_i$  はデータサイズ (GB)、 $T_i$  は転送時間 (秒)、 $n$  はデータの数 (この場合は 10) となる。実験結果の値を式 (1) に代入すると、1GB あたりのリストア完了時間の増加率は約 9.22 秒となる。この増加率を参照し、データサイズにおけるリストア完了時間の予測をする。リストア完了時間の求め方は、式 (2) に示す。

$$\text{予測完了時間 [秒]} = \text{データサイズ [GB]} \times \text{増加率 [秒/GB]} \quad (2)$$

例として、リストアを行うデータサイズを 50GB で求める。式 (2) にデータサイズと増加率を代入すると、予測されたリストア完了時間は約 461 秒となる。この算出できた時間を予測完了時間とする。次に、リストア完了時間が予測完了時間に 38% を加えた時間までに完了するための許容時間を求める。先程と同様に例として、リストアするデータサイズが 50GB の場合は予測完了時間が約 461 秒であるため、予測完了時間に 38% を加えた許容時間は約 635.7 秒となる。次に、データサイズと許容時間をもとに、リストアを許容時間の上限に完了するための必要なディスク読み取り速度を求める。この速度の求め方は式 (3) に示す。

$$\text{必要な読み取り速度 [MB/s]} = \frac{\text{データサイズ [GB]} \times 1024}{\text{許容時間 [秒]}} \quad (3)$$

例として、リストアを行うデータサイズが 50GB、許容時間が約 635.7 秒の場合、式 (3) に代入すると、リストアに必要なディスク読み取り速度は約 80.3[MB/s] となる。この速度でリストアを行えば許容時間内に完了することができる。次に、制限されるディスク読み取り速度の求め方を式 (4) に示す。

$$V_{\text{制限}} = V_{\text{最大}} - \text{必要読み取り速度 [MB/s]} \quad (4)$$

$V_{\text{制限}}$  は、制限されるディスク読み取り速度 [MB/s] のことで、 $V_{\text{最大}}$  は、最大のディスク読み取り速度 [MB/s] のことである。制限されるディスク読み取り速度  $V_{\text{制限}}$  は、最大のディスクの読み取り速度  $V_{\text{最大}}$  とリストアに必要なディスク読み取り速度 [MB/s] の差とする。リストアにおける予測完了時間を設定するためにリストアを単体で合計 100 回行った結果、ディスク読み取り速度の最大が約 110[MB/s] であったため、 $V_{\text{最大}}$  を約 110[MB/s] とする。リストアに必要な読み取り速度 [MB/s] は式 (3) で求めた約 80.3[MB/s] を使用し、式 (4) に代入すると、 $V_{\text{制限}}$  は約 29.7[MB/s] となる。したがって、リストアを行うデータサイズが 50GB の場合、バックアップにおけるディスク読み取り速度を約 29.7[MB/s] に制限することで、リストアを許容時間内に完了させる。

## ユースケース・シナリオ

ユースケースシナリオを図 4 に示す。本稿では、CDSL における学生の VM 使用時に VM 内にある Ubuntu Server のデータが破損してリストアする状況をユースケースとして想定する。以下に Ubuntu Server のデータをリストアする流れを①から③で示す。①は、深夜に学生が研究を行おうとしたところ、VM 内にインストールされた Ubuntu

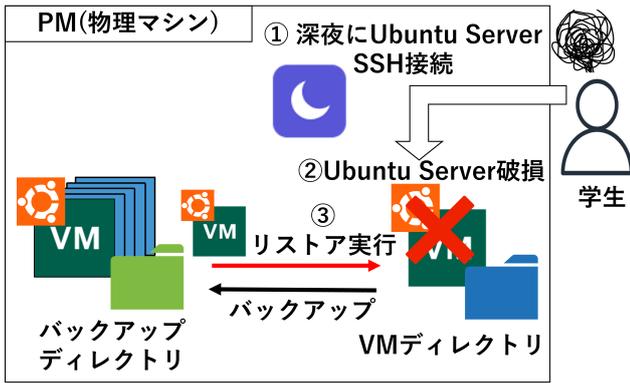


図 4: ユースケースシナリオ

Server へ SSH 接続をした。しかし、②は、Ubuntu Server のデータが破損していたため、Ubuntu Server に入れない状態となった。そのため③は、破損した Ubuntu Server のデータをリストアする必要がある。しかし、CDSL では深夜に VM のバックアップを行っており、PM1 台に保存されている VM40 台のバックアップ完了時間は約 7020 秒を要する。そのため、バックアップとリストアを同時実行することでリストアによる書き込み速度とバックアップによる読み取り速度が競合することでリストア完了時間が増加してしまい学生は研究する時間が少なくなる。また、バックアップのデータサイズによってはバックアップの完了時間が増加し、期限を超過する。本稿の提案方式をもちいてバックアップを行っている最中にリストアを行った場合、許容時間の上限にリストアを完了させ、バックアップ完了時間に余裕を持たせることで、追加のバックアップ要求が発生してもバックアップを期限までに完了させる。

#### 4. 実装

提案ソフトウェアの処理の流れを図 5 に示す。

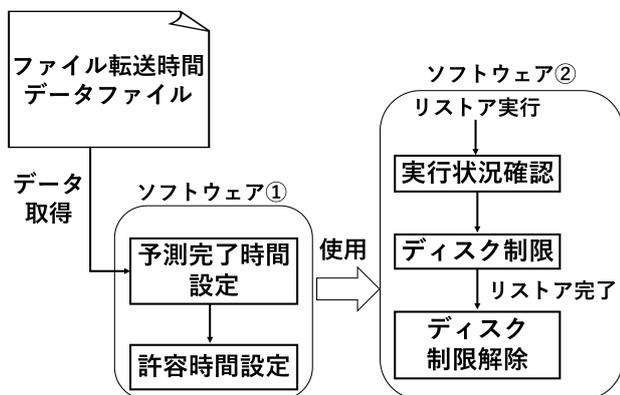


図 5: 提案ソフトウェアの処理の流れ

提案ソフトウェアは 2 つのソフトウェアで構成されている。ソフトウェア①は、データサイズごとに算出したファイル転送時間をデータファイルから取得し、予測完了時間

を設定する。このデータファイルは、予測完了時間を求めるために、データサイズを 10GB から 100GB まで 10GB 間隔で変化させた各データサイズのファイル転送時間が記録されている。次に、予測完了時間に 38% を加えた時間までにリストアを完了できるように許容時間を設定する。ソフトウェア②は、バックアップを行っている最中にリストアの実行状況を確認する。リストアが実行していると判断された場合、ソフトウェア①から許容時間をもとにリストアに必要なディスク読み取り速度を求める。その速度を確保するために提案方式の式 (4) から最大のディスク読み取り速度とリストアに必要なディスク読み取り速度の差で制限されるディスク読み取り速度を求め、バックアップにおけるディスク読み取り速度を制限する。リストア完了後、バックアップにおけるディスク読み取り速度の制限を解除する。

#### 5. 評価実験

提案方式が実際にバックアップを行っている最中にリストアを行った場合、リストア完了時間を許容時間の上限で完了し、バックアップ中のリストア完了時間が短縮できるかどうかを確認する。また、追加でバックアップ要求をされてもバックアップの期限を超過せずに完了するかどうかを確認する。

##### 実験環境

実験環境を図 6 に示す。

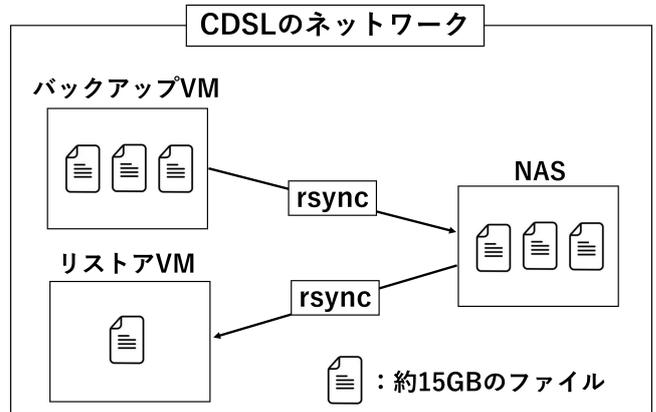


図 6: 実験環境

実験環境として、バックアップを行う VM とリストアを行う VM を使用する。バックアップは、約 15GB のファイル 3 つを使用し、CDSL に構築されている Network Attached Storage(以下 NAS) に rsync コマンドをもちいて行う。また、リストアは、NAS に保存されている約 15GB のファイル 1 つを対象として、バックアップが実行している最中に rsync コマンドをもちいて行う。なお、約 15GB のファイルを選んだ理由は、CDSL の PM1 台を対象に保存されて

いる VM40 台の平均サイズを基準としたためである。

### 基礎実験

基礎実験では、単体でファイル転送を行った場合と、2つのファイルを同時に転送した場合を比較してディスク読み取り速度が競合し、ファイル転送時間および平均待ち時間が増加するかどうかを 20GB のデータサイズで確認した。2つのファイルを同時に転送する場合では、1つのファイル転送を対象にファイル転送時間および平均待ち時間を測定した。

### 基礎実験の結果

基礎実験の結果を図 7 に示す。

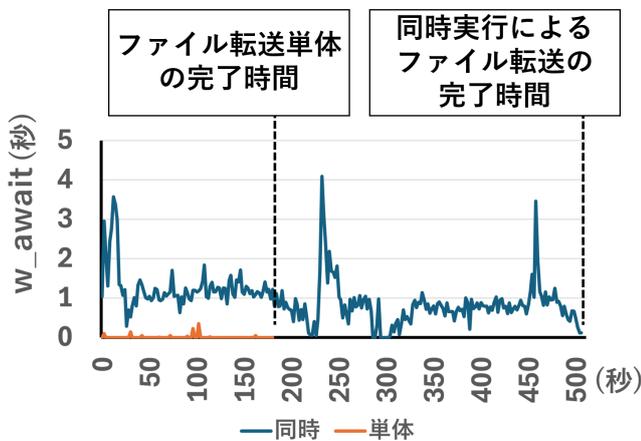


図 7: ファイル転送時間

縦軸は  $w\_await$  (平均待ち時間 [秒]) である。横軸はファイル転送が完了するまでの時間である。 $w\_await$  は、Linux の性能監視ツールである `iostat` の出力項目の 1 つであり、ストレージデバイスへの書き込みリクエストが完了するまでにかかる平均待ち時間を示す。結果として、オレンジ色の折れ線はファイル転送を単体で行った場合である。平均待ち時間は約 0.01 秒、ファイル転送時間は約 184 秒であった。そして、青色の折れ線は 2つのファイル転送を同時にを行った場合である。平均待ち時間は約 0.90 秒、ファイル転送時間は約 506 秒であった。この結果から、ファイル転送を同時に行うとファイル転送時間が増加することが示されている。

### 評価結果と分析

リストア完了時間の結果を図 8 に示す。縦軸はリストアが完了した時間、横軸は提案方式無しと提案方式有りを表している。赤色の点線は、リストアを単体で行った場合のリストア完了時間を表す。リストア単体の完了時間は約 137 秒であり、予測完了時間に 38% を加えた許容時間は約 190 秒である。バックアップとリストアを同時に行った場

合、提案方式無しのリストア完了時間は約 380 秒であった。一方、提案方式有りのリストア完了時間は約 187 秒であり、許容時間までに完了することができた。この結果は、提案方式の式 (4) で述べた通り、バックアップにおけるディスク読み取り速度を最大のディスク読み取り速度とリストアに必要なディスク読み取り速度の差で制限することで、許容時間の上限に近い値でリストアを完了することができた。また、バックアップが期限を超過しない条件でバックアップ中のリストア完了時間を短縮することができた。

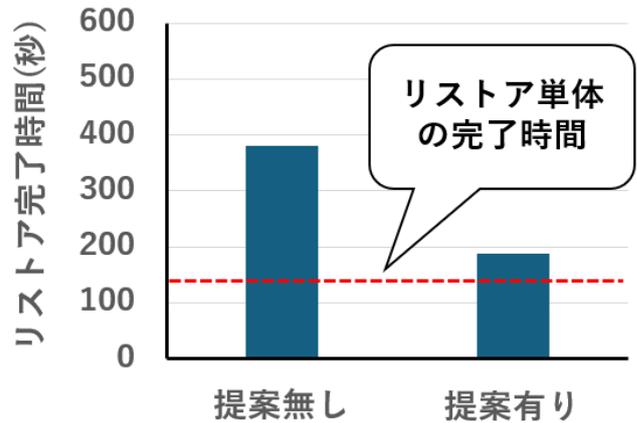


図 8: リストア完了時間

バックアップ完了時間の結果を図 9 に示す。

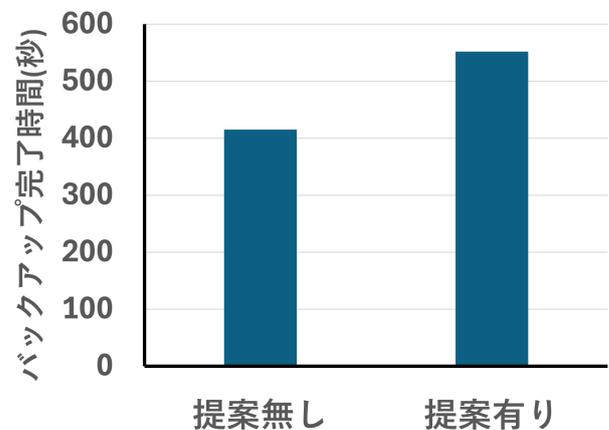


図 9: バックアップ完了時間

縦軸はバックアップが完了した時間、横軸は提案方式無しと提案方式有りを表している。バックアップを単体で行った提案方式無しのバックアップ完了時間は約 415 秒であった。一方、バックアップにおけるディスク読み取り速度を制限した提案方式有りのバックアップ完了時間は約 553 秒であり、約 33% 増加した。これは、リストアを優先するためにバックアップにおけるディスク読み取り速度を制限した結果、バックアップ完了時間が増加したといえる。また、バックアップ完了時間は増加したが、本稿のユースケース

で提案方式を使用するとバックアップの期限までに完了することができ、データサイズが約 556GB 以内であれば追加でバックアップ要求をされてもバックアップの期限を超過しない見込みとなる。

## 6. 議論

提案方式では、リストアの許容時間やディスク読み取り速度を求める際に、データサイズをもとにした近似式を使用している。しかし、リストア完了時間は、ファイル数の増加に伴って増加する。これは、ファイル数が増えるほど、ディスクの読み書き回数が増加し、ストレージデバイスの読み出し要求が遅れるためである。特にファイル数が多い場合、ディスク内のアクセスが広範囲に分散し、各要求に対する対応時間が増加することで遅延が発生する [26]。また、ファイル数が増加するほど、ディスクのスループットよりも Input/Output Per Second(以下 IOPS) が性能のボトルネックになりやすい。IOPS とは、ハードディスクや SSD を例とし、ストレージデバイスが 1 秒間にデータの読み込みまたは書き込みができる回数を示す性能指標である。データサイズだけをもとにした提案方式では、ディスクの読み書き回数の増加や I/O 性能の制限を取り入れていないため、リストア完了時間を正確に予測できない。また、評価実験では、許容時間の上限(約 190 秒)付近で完了することができたが、リストア完了時間は約 187 秒で約 3 秒分は正確にリストアを完了することができなかった。これらの課題は、本稿の提案方式にファイル数やストレージデバイスの性能指標を追加することで改善可能である。具体的には、IOPS とリストア完了時間の関係を、提案方式のファイル転送時間とデータサイズの間を基礎実験と同様に実験から示し、この結果を提案方式の式 (2) に変数として取り入れることで、本稿の提案方式より正確にリストア完了時間を見積もることができる。また、バックアップ時にデータを圧縮することで、リストア時のディスクの書き込み回数を削減する。

## 7. おわりに

評価実験では、提案方式が実際にバックアップを行っている最中にリストアを行った場合、リストア完了時間を許容時間の上限で完了し、バックアップ中のリストア完了時間が短縮できるかどうかを提案有無で比較した。また、追加でバックアップ要求をされてもバックアップの期限を超過せずに完了するかどうかを提案方式の有無で比較した。結果として、提案方式無しのリストア完了時間は約 380 秒、提案方式有りのリストア完了時間は約 187 秒で予測完了時間に 38% を加えた許容時間(約 190 秒)の上限付近で完了することができ、バックアップが期限を超過しない条件でバックアップ中のリストア完了時間を短縮できた。一方、バックアップを単体で行った提案方式無しのバックアップ

完了時間は約 415 秒であった。バックアップにおけるディスク読み取り速度を制限した提案方式有りのバックアップ完了時間は約 553 秒であり、約 33% 増加した。これは、リストアを優先するためにバックアップにおけるディスク読み取り速度を制限した結果、バックアップ完了時間が増加した。また、バックアップ完了時間は増加したが、本稿のユースケースで提案方式をもちいることでバックアップの期限までに完了することができ、データサイズが約 556GB 以内であれば追加でバックアップ要求をされてもバックアップの期限を超過しない見込みとなる。

## 参考文献

- [1] Akbar, R., Husain, M. S. and Suaib, M.: Comparative study of various backup and monitoring techniques, *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, pp. 1530–1537 (online), DOI: 10.1109/ICGCIoT.2015.7380710 (2015).
- [2] Mukherjee, P. and Salapura, V.: Challenges of DB2 Restore in a Distributed Systems Environment and Engineered Solutions, *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pp. 38–42 (online), DOI: 10.1109/DSN-W.2018.00021 (2018).
- [3] Xia, R., Yin, X., Alonso Lopez, J., Machida, F. and Trivedi, K. S.: Performance and Availability Modeling of IT Systems with Data Backup and Restore, *IEEE Transactions on Dependable and Secure Computing*, Vol. 11, No. 4, pp. 375–389 (online), DOI: 10.1109/TDSC.2013.50 (2014).
- [4] Shriwas, M. S., Gupta, N. and Sinhal, A.: Efficient method for backup and restore data in Android, *2013 International Conference on Communication Systems and Network Technologies*, IEEE, pp. 693–697 (2013).
- [5] Chang, V.: Towards a big data system disaster recovery in a private cloud, *Ad hoc networks*, Vol. 35, pp. 65–82 (2015).
- [6] Chen, L., Liu, Z. and Jia, W.: A snapshot system based on cloud storage Log-Structured Block System, *Proceedings of the 11th IEEE International Conference on Networking, Sensing and Control*, pp. 392–398 (online), DOI: 10.1109/ICNSC.2014.6819658 (2014).
- [7] Xie, G., Qi, L., Wang, F., Wang, G., Liu, X. and Liu, J.: ESnapII: A Writable Dependent Snapshot System with Shared Cache, *2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, pp. 677–682 (online), DOI: 10.1109/SNP.D.2008.94 (2008).
- [8] Li, J., Liu, H., Cui, L., Li, B. and Wo, T.: iROW: An Efficient Live Snapshot System for Virtual Machine Disk, *2012 IEEE 18th International Conference on Parallel and Distributed Systems*, pp. 376–383 (online), DOI: 10.1109/ICPADS.2012.59 (2012).
- [9] Xiao, W., Yang, Q., Ren, J., Xie, C. and Li, H.: Design and analysis of block-level snapshots for data protection and recovery, *IEEE Transactions on Computers*, Vol. 58, No. 12, pp. 1615–1625 (2009).
- [10] Levina, A. and Semenov, A.: Jump-Based Backup: An Efficient Data Backup, *2024 13th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1–4 (online), DOI: 10.1109/MECO62516.2024.10577875 (2024).
- [11] Paul, S., K.A., S. and Abraham, J. P.: Block Level In-

- cremental Backup With Brotli Compression, *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, pp. 1–5 (online), DOI: 10.1109/ICCSDET.2018.8821114 (2018).
- [12] Levitin, G., Xing, L., Zhai, Q. and Dai, Y.: Optimization of Full versus Incremental Periodic Backup Policy, *IEEE Transactions on Dependable and Secure Computing*, Vol. 13, No. 6, pp. 644–656 (online), DOI: 10.1109/TDSC.2015.2413404 (2016).
- [13] Song, Y., Liu, K., Gu, Y., Jiang, Y., Sun, W. and Fang, L.: Data Resilience Protection for Power Data Center Based on Automatic Identification of Key Data and Incremental Backup, *2023 International Conference on Intelligent Management and Software Engineering (IMSE)*, pp. 56–59 (online), DOI: 10.1109/IMSE61332.2023.00018 (2023).
- [14] Sokolov, S., Vlaev, S. and Iliev, T. B.: Technique for Improvement of Backup and Restore Strategy Based on Blockchain, *2022 International Conference on Communications, Information, Electronic and Energy Systems (CIEES)*, pp. 1–6 (online), DOI: 10.1109/CIEES55704.2022.9990781 (2022).
- [15] Javaraiah, V.: Backup for cloud and disaster recovery for consumers and SMBs, *2011 Fifth IEEE International Conference on Advanced Telecommunication Systems and Networks (ANTS)*, pp. 1–3 (online), DOI: 10.1109/ANTS.2011.6163671 (2011).
- [16] Enterprise Strategy Group (ESG): Data Protection for SaaS, Technical report, Enterprise Strategy Group (2023).
- [17] 駒田陽子, 井上雄一: 睡眠障害の社会生活に及ぼす影響 (シンポジウム: 心身機能と睡眠障害, 2006年, 第47回日本心身医学会総会(東京)), *心身医学*, Vol. 47, No. 9, pp. 785–791 (2007).
- [18] Hasan, M. Z., Sarwar, N., Alam, I., Hussain, M. Z., Siddiqui, A. A. and Irshad, A.: Data Recovery and Backup management: A Cloud Computing Impact, *2023 IEEE International Conference on Emerging Trends in Engineering, Sciences and Technology (ICEST)*, pp. 1–6 (online), DOI: 10.1109/ICEST56843.2023.10138852 (2023).
- [19] 石川裕人, 高橋風太, 串田高幸: リストアにおける進行状況にもとづくバックアップのCPU使用量の制限による時間の増加の抑制, 技術報告 CDSL-TR-200, Tokyo University of Technology CDSL Technical Report (2024). (online), Available at: <https://ja.tak-cslab.org/tech-report>.
- [20] Egger, B., Gustafsson, E., Jo, C. and Son, J.: Efficiently restoring virtual machines, *International Journal of Parallel Programming*, Vol. 43, pp. 421–439 (2015).
- [21] Baker, M.: The recovery box: Using fast recovery to provide high availability in the UNIX environment, *In Proceedings USENIX Summer Conference*, Citeseer (1992).
- [22] Liu, W., Lu, Y., Wu, C., Li, J. and Guo, M.: ERP: An Efficient Rewrite Scheme to Improve the Inline Deduplication Restore Performance in Backup Systems, *2022 IEEE 28th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 371–378 (online), DOI: 10.1109/ICPADS56603.2022.00055 (2023).
- [23] Tan, Y., Jiang, H., Feng, D., Tian, L. and Yan, Z.: CABdedupe: A Causality-Based Deduplication Performance Booster for Cloud Backup Services, *2011 IEEE International Parallel Distributed Processing Symposium*, pp. 1266–1277 (online), DOI: 10.1109/IPDPS.2011.76 (2011).
- [24] Ceaparu, I., Lazar, J., Bessiere, K., Robinson, J. and Shneiderman, B.: Determining causes and severity of end-user frustration, *International journal of human-computer interaction*, Vol. 17, No. 3, pp. 333–356 (2004).
- [25] 高橋風太, 串田高幸: 業務開始時刻に完了させるバックアップに対する読み取り制御によるユーザファイルの転送時間の最短化, 技術報告 CDSL-TR-209, Tokyo University of Technology CDSL Technical Report (2024). (online), Available at: <https://ja.tak-cslab.org/tech-report>.
- [26] Bose, R., Roy, S. and Sarddar, D.: On Demand IOPS Calculation in Cloud Environment to Ease Linux-Based Application Delivery, *Proceedings of the First International Conference on Intelligent Computing and Communication*, Springer, pp. 71–77 (2017).