

ページビューの増加をもちいた記事数の限定による リストアのテスト時間の短縮

西村 克己¹ 平尾 真斗² 串田 高幸¹

概要: システム運用において、システムデータは重要であり必要不可欠な要素である。データを損失した際に、システムを回復できるようにバックアップは行われる。課題は、バックアップデータに含まれる WordPress の記事や固定ページのコンテンツを全てテストすることは、コンテンツを限定してテストを行った際と比べて実行時間がかかることである。提案方式で、VM 上に構築したアプリケーションを復元した際に、テストの対象を絞り込むことで確認時間の短縮を行うことを目的としている。本稿では、バックアップデータが作成し終えたことをバックアップ完了のログの追加から判定し、Web ページの閲覧数の差分をバックアップデータに含まれているデータベースの Dump ファイルの比較から取得した。Web ページの閲覧数の差分の値が 1 件以上ある記事を取得し、パレートの法則から閲覧数が 1 件以上増加した記事の上位 20% をテスト対象として含めることとした。評価方法は、テスト項目を適用させる対象を絞り込むことで、提案方式を使用する前と後で、どれだけの時間が削減できたかを調査する。課題を証明する基礎実験として、CDSL のブログサイトを対象に、公開されている固定ページと投稿の記事の内容すべてを確認する自動化したプログラムを 2 回実行して、実行時間を Python の time モジュールを利用して、計測を行った。実験結果として、プログラム 2 回実行の時間はそれぞれ約 56 秒と約 62 秒であった。このことから、テスト範囲の対象を広げると時間がかかることがわかった。

1. はじめに

背景

企業では、システムを運用する際のシステムデータを保存しておくためにバックアップを作成する [1, 2]。バックアップデータとは、障害が発生した際に復元できるように、データのコピーを作成し、別のストレージデバイス上に保存しておくことで、データの可用性を確保するものである [3]。

データを損失する原因として、人為的ミス、自然災害、機器の故障があげられる [4]。データを損失した際に、迅速にシステムを回復するために、データバックアップ技術を実現することが重要になっている [5, 6]。バックアップデータを復元時に利用できなければ、全く意味をなさないデータとなり、データの安全性と可用性を確保できなくなる [7]。そのため、バックアップとリストアにおいて、テストを実行してバックアップが機能することを確認することの重要性が注目されている [8]。

東京工科大学 コンピュータサイエンス学部にある、Cloud and Distributed Systems Laboratory (以後 CDSL と呼ぶ)

では、普段の様子やテクニカルレポートをインターネット上に公開するために、VM 上に WordPress を構築し、CDSL のブログサイトを運用している*¹。

CDSL のブログサイトのシステム構成を図 1 に示す。CDSL のブログサイトの VM は Google Cloud Platform (以後 GCP と呼ぶ) 上にある。図 1 の、インターネット上にいる AccessUsers は、CDSL のブログサイトにアクセスする際、cdsl-vm2 上にある Nginx のリバースプロキシにアクセスする。その後、WordPress が動作している cdsl-vm1 に AccessUser はアクセスが転送され、CDSL のブログサイトにアクセスできるようになる。VM 上で動作している WordPress のデータは、CDSL の Local Network 上に存在する Synology 社の Network Attached Storage (以後 NAS と呼ぶ) に WordPress 本体のデータとデータベースの内容を dump したファイルを圧縮し、1 ヶ月間保存している。リストアを行う際には、NAS 上にあるバックアップデータを復元してデータの復旧を試みる [9]。

2024 年 07 月 01 日時点の CDSL のブログサイトの総公開項目数 (固定ページ、投稿件数) を表 1 に示す。

CDSL のブログサイトでは、固定ページが 12 件、投稿件数が 418 件あった。2024 年 07 月 01 日 0:00 と 2024 年 07

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

〒192-0982 東京都八王子市片倉町 1404-1

*¹ <https://ja.tak-cslab.org/>

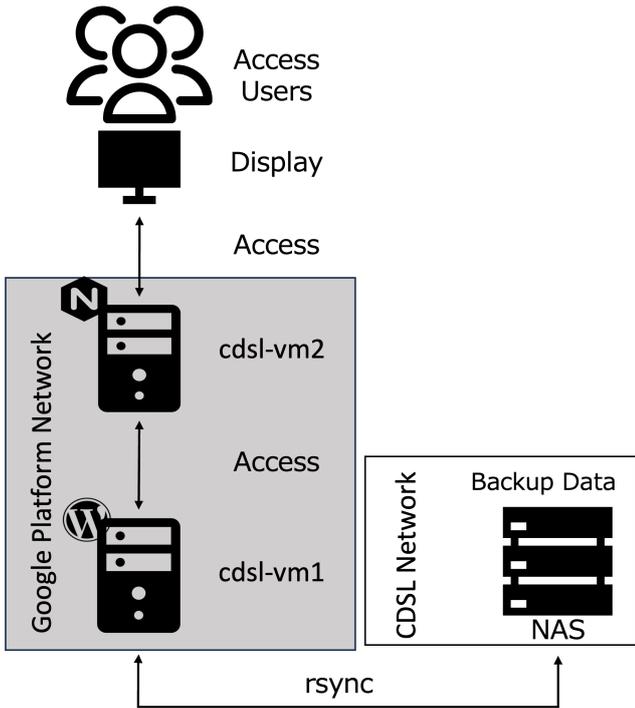


図 1: CDSL のブログサイトのシステム構成図

表 1: CDSL のブログサイト総公開項目数

項目	項目数
固定ページ	12
投稿件数	418

月 02 日 0:00 までにそれぞれ取られたバックアップデータから閲覧数の差分順に並べ替えた上位 10 件の記事の結果のグラフを図 2 に示す。

図 2 では、記事 ID として tech-report が 1 番閲覧数の差分が大きい結果となった。

2024 年 07 月 02 日 0:00 と 2024 年 07 月 03 日 0:00 までにそれぞれ取られたバックアップデータから閲覧数の差分順に並べ替えた上位 10 件の記事の結果のグラフを図 3 に示す。

図 3 では、記事 ID として 2898 が 1 番閲覧数の差分が大きい結果となった。

図 2 と図 3 から、日々のバックアップデータから記事の閲覧数の変化があることがわかる。

また、WordPress のテスト項目として確認すべき項目を以下に示す。

(1) ブログ記事のコンテンツ

- ブログ記事のホームページが閲覧できる状態であるか。
- 投稿内容の確認ができていないか。
- ブログ記事内にある URL のリンクが正しく動作しているか。
- 画像が表示されているかどうか。

(2) ブログダッシュボードの確認

- 新しい記事が作成できるか。
- 投稿済みの記事が編集できるか。
- 投稿済みの記事が削除できるか。

上記のように、テスト項目の設定をする [10]。

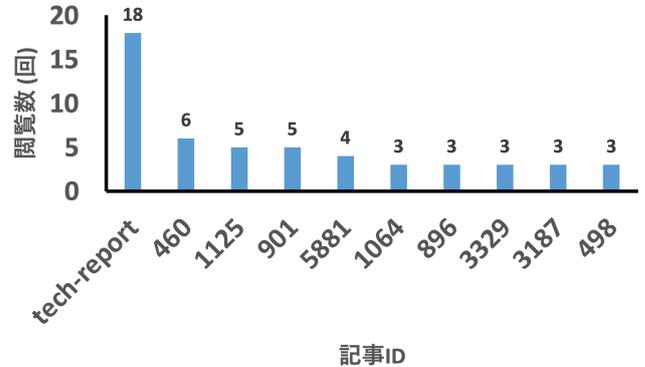


図 2: 07 月 01 日と 07 月 02 日の閲覧数の差分

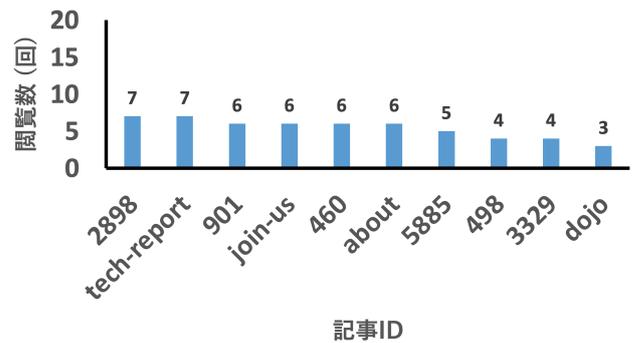


図 3: 07 月 02 日と 07 月 03 日の閲覧数の差分

課題

課題は、バックアップデータに含まれる WordPress の記事や固定ページのコンテンツを全てテストすることは、コンテンツを限定してテストを行ったサイトと比べて実行時間がかかることである。バックアップデータには、過去に投稿された WordPress のブログ記事のデータが保存されている。課題の概要を図 4 に示す。

バックアップデータの作成後、テスト実行者は PC 内にあるテストの自動化プログラムを経由して、WordPress のブログ記事のデータすべてをテストの実行範囲として設定し、設定されてあるテスト項目を確認する。ブログ記事のデータ全てをテストの実行範囲として設定し、確認することは時間がかかる。

各章の概要

本テクニカルレポートでは以下の内容で構成している。第 1 章では、本稿の背景と課題について述べる。第 2 章で

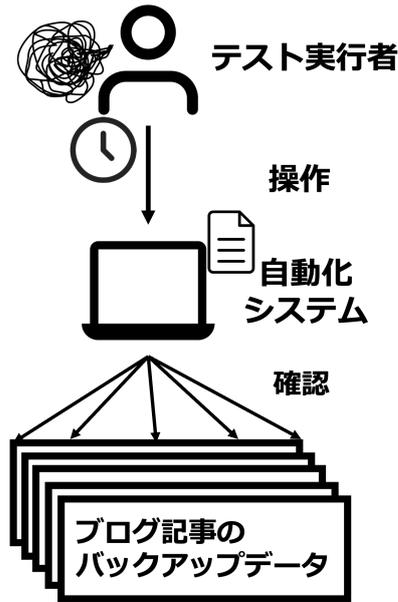


図 4: 課題の概要

は、本稿の関連研究について述べる。第 3 章では、本稿の課題について解決するための提案方式について述べる。第 4 章では、提案した手法の実装について述べる。第 5 章では、評価実験として実験内容と実験結果と分析について述べる。第 6 章では、提案方式についての議論を述べる。最後に、第 7 章にて結論を述べる。

2. 関連研究

スナップショットのロールバックにおける、データの損傷の問題を解決するための新しいバックアップの方法の提案を行っている研究がある [11]。リストアする際のもとなるスナップショットを利用したバックアップデータ作成の場面で提案を適用しようとしているが、バックアップ作成後のリストア時の確認までは行っていない。そのため、リストア後バックアップデータの利用可能かどうかの判断ができないため改善の余地がある。

オンデマンドリストアを実現する新しいバックアップ方法の提案、特定のディレクトリやファイルのみをリストアする提案を行っている研究がある [12]。特定のディレクトリのみをリストアする方法は示されているが、リストア時の確認のテストは行っていない。そのため、リストア時の確認を行わないことで、データを復元できない原因に繋がると、改善の余地がある。

リレーショナルデータベースにおける、バックアップとリカバリを行う際のデータの一貫性に対する提案を行っている研究がある [13]。メンテナンスの問題性を解決しようとしているが、リストア後の確認のテストは行っていない。そのため、リストアの動作自体に潜む問題を発見できず、リストアが失敗する原因に繋がると、改善の余地がある。

プログラムやアプリケーション内のエラーやバグを検出

して、ソフトウェアの品質を保つ提案を行っている研究がある [14]。リストア前のチェックにおける観点では確認していない。そのため、プログラムやアプリケーションの信頼性を損なうため改善の余地がある。

ソフトウェアのリリース前に Ranorex と呼ばれる自動テストツールを提案し、テストツールの機能を評価した研究がある [15]。リリース前に適切なテストが行われていないソフトウェア製品が存在している。ソフトウェアの内容をチェックしているが、個人が読み書きするデータの話までは踏み込んでいない。そのため、ユーザーがシステムを使用する際に問題が発生する原因に繋がると、改善の余地がある。

自動化されたバックアップデータのテストは、測定異常を検出および修正する上で重要であり、測定データの誤りを特定および修正するためのウェーブレット閾値分析を使用した方法を提案している研究がある [16]。この手法は測定データの誤り検出および修正に重点を置いており、アプリケーション全体の動作確認や動作条件の自動生成については考慮していない。そのため、予期しない不具合が発生する原因に繋がると、改善の余地がある。

3. 提案方式

3.1 提案方式の目的

提案手法にて、バックアップデータのテスト対象を絞ることでテストの実行時間を削減することを目的とする。

3.2 提案の概要

本稿では、Web ページの閲覧数に差がある記事をバックアップデータにあるデータベースの Dump ファイルをリストアして比較を行い、テスト対象を定めることを提案する。

図 5 に、提案の概要を示す。図 5 にある提案ソフトウェア

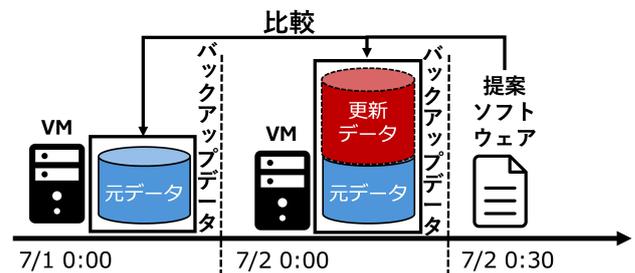


図 5: 提案の概要

アは、7/1 0:00 時点での VM 上で作成されたバックアップデータ内の Web ページと、7/2 0:00 時点での VM 上で作成されたバックアップデータ内の Web ページを比較する。バックアップデータに保存されている記事の閲覧数が 1 件以上増加しているページを取得する。その後パレートの法則から、閲覧数が 1 件以上増加した記事の件数上位 20%を

テスト対象として含めた [17]. なお本稿では、バックアップが行われた際に提案アルゴリズムは適用し、実施する.

3.3 閲覧数の差分の取得方法

バックアップが終了した旨のログがプログラム 1 のように、ログファイルに追加される.

プログラム 1: バックアップ完了のログ

```
1 Backup completed on Wed Jul 10 23:30:04 JST
  2024
```

該当のログファイルに追記があったかどうかで、バックアップデータの作成が終了したことの判断を行う. バックアップデータの比較を図 6 に示す. バックアップデータが作成

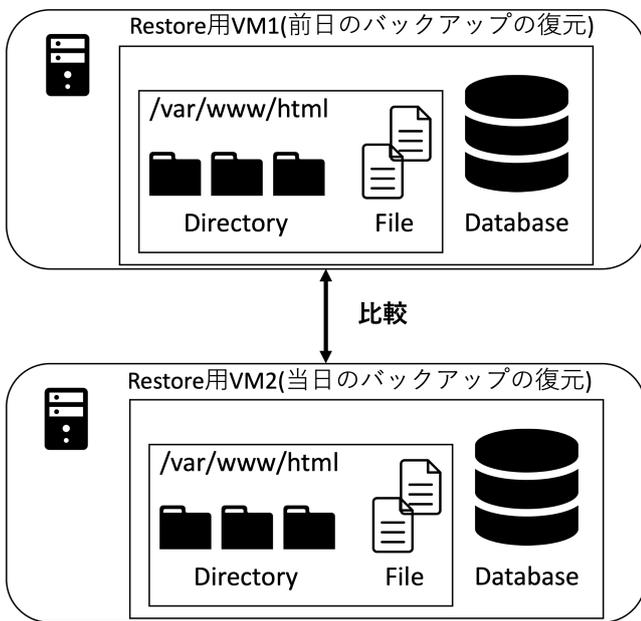


図 6: バックアップデータの比較

されたことを確認できたら、Restore 用の VM1 にテスト実行日前日のバックアップデータを復元する。また、Restore 用の VM2 にテスト実行日当日のバックアップデータを復元する。復元が完了したら、Restore 用の VM1, VM2 上にある WordPress のデータを保存しているデータベースの table の比較を行う。その際にデータベース内に保存されている記事の閲覧数の要素からテスト対象をテスト実行日前日のバックアップとの差分で絞り込みを行う。

3.4 パレートの法則を利用して対象の絞り込みを行う方法

7/1 0:00 時点でのバックアップデータに記録されている閲覧数の例を表 2 に示す。表 2 にある PageID はそれぞれの記事の URL を表している。閲覧数は PageID に該当する記事の累積の閲覧数を表している。また、7/2 0:00 時点でのバックアップデータに記録されている閲覧数の例を表 3 に示す。表 3 は、表 2 と同じ項目が含まれている。本稿では、

表 2: 7/1 0:00 時点のブログサイト閲覧数

PageID	閲覧数
/tech-report	18
/archives/460	6
/archives/1125	5
/archives/901	5
/archives/5881	4
/archives/1064	3
/archives/896	3
/archives/3329	3
/archives/3187	3
/archives/498	3
/archives/5885	3
/archives/4860	3
/archives/2238	2
...	...

表 3: 7/2 0:00 時点のブログサイト閲覧数

PageID	閲覧数
/archives/2898	7
/tech-report	7
/archives/901	6
/join-us	6
/archives/460	6
/about	6
/archives/5885	5
/archives/498	4
/archives/3329	4
/dojo	3
...	...

表 2 と表 3 のように、バックアップデータに保存されているそれぞれの閲覧数の差分を出し、テスト項目を実施する範囲を決定する。

例えば、表 2 と表 3 上の /tech-report と示されている PageID の記事に注目する。表 2 での /tech-report の記事の閲覧数が 18 であった。7/1 0:00 時点での閲覧数が 1 件以上増加があった記事は 62 件あった。パレートの法則を利用すると、閲覧数の増加があった記事の上位 2 割を対象とするため、表 2 の PageID にある /tech-report から /archives/4860 の計 12 件をテスト対象とした。また、表 3 での /tech-report の記事の閲覧数が 7 であった。7/2 0:00 時点での閲覧数が 1 件以上増加があった記事は 41 件あった。パレートの法則を利用すると、表 3 の PageID にある /archives/2898 から /archives/3329 の計 8 件をテスト対象とした。

ユースケース・シナリオ

本稿では、WordPress が動作している仮想マシン上にあるブログサイトのバックアップデータをテストすること

を想定する。バックアップデータをチェックする動作を図7に示す。

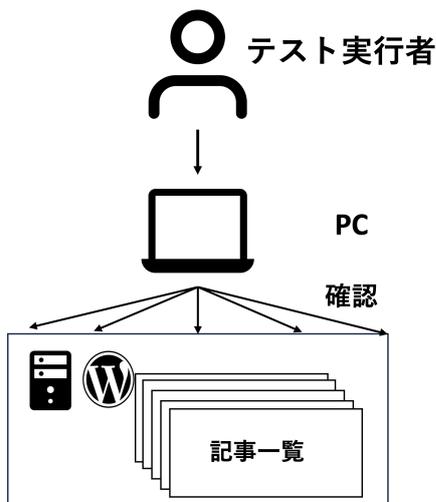


図 7: チェックする動作を示した図

現状テスト実行者はバックアップデータから復元した際に、自動化ツールを導入していない場合、手動でテスト項目を確認しているが、この手法は多くの労力と時間を要し、確認漏れが発生ししやすい [18]。

提案では、毎日定期的に作成されるバックアップデータを元に、バックアップデータのテスト項目を実行する対象を絞ることを行っている。これにより、アプリケーション全体を対象にテスト項目を実施するより時間の削減を行おうとしている。

例えば、WordPress の記事の更新があった場合でも、テスト実施前日に作成されたバックアップデータをもとに比較を行い、テスト対象を絞り込む。これにより、運用中の Web サイトが常に正常に動作することを確保し、ユーザー体験の向上にもつながる [19]。

4. 実装

実装では、VM 上にある `/var/www/html` の内容と、MySQL の Dump ファイルである `.sql` ファイルを圧縮したデータを WordPress のブログサイトのバックアップデータとして用意し、NAS 上に保管をする。

バックアップとして保存したデータを検証のため、Restore 用 VM として用意する Ubuntu 上の `/tmp` ディレクトリに展開を行う。その際に Python プログラムでアプリケーションの判別を行い、バックアップデータのテストを実行する。

実装プログラムの処理の流れを図8に示す。

バックアップデータを作成し終えたことを、ログファイルから判断する。ログファイル上にプログラム1の記載があった場合、バックアップデータが作成し終えたことを判断する。プログラム1のように、バックアップ完了のログが追

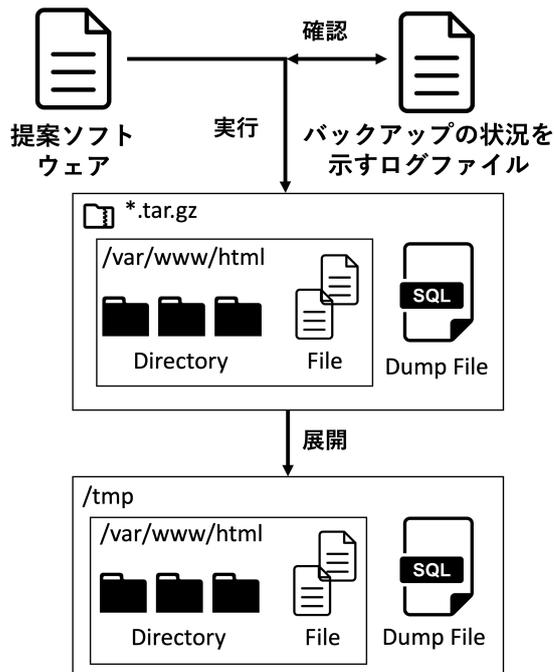


図 8: 実装プログラムの処理の流れ

加されたら、Linux の `inotify-tools` に含まれる `inotifywait` でログの追加の読み取りを行う*2。

WordPress には閲覧数を統計情報を取得するプラグインとして WP Statistics がある*3。記事ごとの閲覧数の情報をプラグインから取得する。テスト実行前日のバックアップデータとテスト当日のバックアップデータにあるデータベースの Dump ファイルを復元して閲覧数の比較を行い、対象を定める。記事ごとの閲覧数の差分を出した例をプログラム2に示す。

プログラム 2: 記事ごとの閲覧数の差分の例

```
1 Main: {'cleaned_uri': '/archives/460', 'total_count': 3910, 'post_title': 'ビザンチン将軍問題 ~合意するって難しい~'}
2 Sub: {'cleaned_uri': '/archives/460', 'total_count': 3890, 'post_title': 'ビザンチン将軍問題 ~合意するって難しい~'}
3 ---
4 Main: {'cleaned_uri': '/archives/5881', 'total_count': 26, 'post_title': '1年前の自分と振り返って'}
5 Sub: {'cleaned_uri': '/archives/5881', 'total_count': 22, 'post_title': '1年前の自分と振り返って'}
6 ---
7 Main: {'cleaned_uri': '/about/it-facility', 'total_count': 302, 'post_title': 'IT ファシリティ'}
8 Sub: {'cleaned_uri': '/about/it-facility', 'total_count': 300, 'post_title': 'IT ファシリティ'}
```

*2 <https://github.com/inotify-tools/inotify-tools>

*3 <https://wordpress.org/plugins/wp-statistics/>

```
IT ファシリティ'}
9 ---
10 Main: {'cleaned_uri': '/archives/3329', '
total_count': 725, 'post_title': 'インド
の大学と共同で論文を執筆し、公開'}
11 Sub: {'cleaned_uri': '/archives/3329', '
total_count': 722, 'post_title': 'インド
の大学と共同で論文を執筆し、公開'}
12 ---
13 Main: {'cleaned_uri': '/archives/3187', '
total_count': 471, 'post_title': 'Slay
the Spire、上手になりたい'}
14 Sub: {'cleaned_uri': '/archives/3187', '
total_count': 467, 'post_title': 'Slay
the Spire、上手になりたい'}
15 ---
16 Main: {'cleaned_uri': '/archives/498', '
total_count': 659, 'post_title': '分散シ
ステムにおけるCAP 定理とは'}
17 Sub: {'cleaned_uri': '/archives/498', '
total_count': 653, 'post_title': '分散シ
ステムにおけるCAP 定理とは'}
18 ---
19 Main: {'cleaned_uri': '/tech-report', '
total_count': 3098, 'post_title': 'テク
ニカルレポート'}
20 Sub: {'cleaned_uri': '/tech-report', '
total_count': 3074, 'post_title': 'テク
ニカルレポート'}
```

Main は 7/2 0:00 現在のデータ, Sub は 7/1 0:00 現在のデータで保存されている閲覧数をそれぞれ示している. プログラム 2 のように, 閲覧数の差分を出力し, 記事の閲覧数に差がある対象をテストの実行範囲として含める. なお, プログラム 2 に載っていない記事はテストを実行する対象として含めない.

5. 評価実験

評価方法

評価方法では, Application バックアップデータのテスト項目を作成し, 対象を絞り込むことで提案方式を使用する前と後でどれだけの時間が削減できたかを調査する [20]. 2024 年 7 月 01 日, 2024 年 7 月 02 日時点での CDSL ブログサイトの総公開項目数はそれぞれ 430 件であった. 提案方式を利用することで, テストの実施対象を 62 件に限定した. 総公開項目数である 430 件と, 提案方式利用後の 62 件で, 削減した時間から評価を行う.

実験環境

今回の提案を実験した環境を図 9 に示す.

実験環境は Ubuntu 24.04 LTS をインストールした仮想マシン上に, Web サーバーである Nginx をインストールし

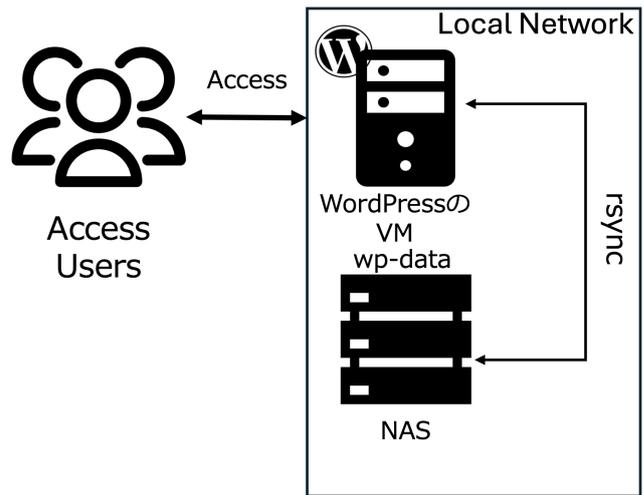


図 9: 実験環境

た. 以下に, 実験時の仮想マシンの構成要素を示す.

- 仮想マシン構成要素
 - OS: Ubuntu 24.04 LTS**
 - vCPU: 1Core**
 - RAM: 1GB**
 - HDD: 25GB**

その後動作に必要な PHP とデータベースサーバーである, mariadb-server をインストールして, WordPress を動作させた. またバックアップ先として, 現在 CDSL で運用している NAS 上にディレクトリ 'wp-data' を作成し, Network File System(以後 NFS と呼ぶ) にて共有した状態で用意した.

CDSL のブログサイトでは, GCP 上にあるデータを 1 日 1 回 24 時に WordPress のデータである, /var/www/html とデータベースのデータである mysql-dump コマンドを使って取得した dump ファイルを圧縮している. 実験時, CDSL のブログサイトで取られているバックアップ方式と同様に, 仮想マシン上にある, /var/www/html 内にあるディレクトリ 'wp-admin', 'wp-content', 'wp-includes' の 3 つとディレクトリに付随する WordPress の構成ファイルをコピーする. また, データベースである, mariadb-server の内容を mysql-dump コマンドを使って .sql ファイルに書き出しを行った. /var/www/html 内のディレクトリと, データベースの .sql ファイルを tar ファイルで圧縮して, バックアップデータとして保存する. その後, NAS にあるタスクスケジューラーの機能を使い, バックアップ用に作成したあるデータのディレクトリを rsync で同期させた*4. これにより, 仮想マシン上にあるバックアップデータを NAS 上に置き, バックアップのテストを実行することが可能になる.

*4 https://kb.synology.com/ja-jp/DSM/help/DSM/AdminCenter/system_taskscheduler?version=7

基礎実験

基礎実験として、自動化したプログラムでテストの範囲をブログのコンテンツ全体で設定した際に、テストのチェック時間がどのくらいかかったかを計測した。基礎実験の実行結果を図 10 に示す。

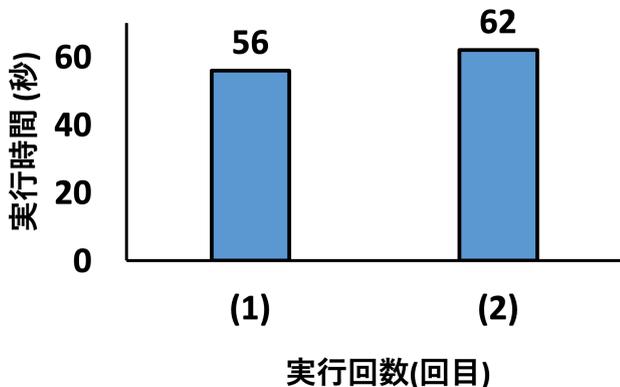


図 10: 基礎実験の実行結果

図 10 では、テストのチェック時間がどのくらいかかったかを示している。

ブログ全体のチェックを行う Python プログラムは、WordPress の Rest API を使い、全体のうちインターネット上に公開されている記事全体を抜き出して検索を行っている*5。(1) が 1 回目にプログラムを動作させたもの、(2) が 2 回目にプログラムを動作させたものとなっている。

Python プログラムの time モジュールで計測した結果は、(1) では約 56 秒、(2) では約 62 秒となった。

6. 議論

提案アルゴリズムでは、ログファイルに追記されたメッセージを元にバックアップの完了を確認している。ログファイルの操作や追記に失敗するとともとなるバックアップの作成終了を確実に判断する手段として、バックアップファイルの存在の確認やファイルのタイムスタンプを要素として使用することにより、バックアップの作成終了の判断を確実にすることができる。

自動テストでは Web ページにおける、レイアウトの崩れや環境、状況に依存する動作を実行することは難しい。すべてのテスト項目やテスト項目を適用する範囲を自動で生成することは難しく、特定のケースやシナリオに対応する範囲を決定することは難しい。Web ページのテストの際、環境依存で自動テストの対応が難しい場合、複数のテスト環境を用意しておくこととする。テスト環境で表示されている UI 同士を比較することで対応でき、テストの網羅性の向上につながる [21]。

提案方式で使用したテストの実行範囲を、バックアップ

データ同士を比較した閲覧数の差分を元に、記事や固定ページの指定をした。この提案方式では、投稿日時が古い重要なページや、一時的にアクセスが集中したページがテスト項目の適用範囲として漏れる可能性がある。適用範囲に漏れることで、テスト本来の目的であるバックアップデータが使えるかどうかの判断ができなくなる。そのため、バックアップデータのみでなく、リアルタイムでのアクセスログやユーザーの行動パターンを活用し、特定期間に急激にアクセスが増加したページを記録し、ユーザーがどのページを経由して目的ページに到達しているかを分析して、ページの重み付けをつけることで、テスト対象を更新でき、テスト実行時間の短縮に繋げられる。

7. おわりに

課題は、バックアップデータに含まれる WordPress の記事や固定ページのコンテンツを全てテストすることは、テストの実行時間がかかることである。提案では、バックアップデータが作成し終えたことを判定し、前日とのバックアップを比較して Web ページの閲覧数の差分の値が 1 件以上ある記事を取得し、パレートの法則から閲覧数の 1 件以上増加した記事の上位 20% をテスト対象として含めた。その後、事前に作成してあるテスト項目を提案手法にて決定した対象を元にテストの実行をした。実験として、CDSL のブログサイトを元に再現した環境のもと、アプリケーション全体をバックアップのテスト対象としておいたものと、テストを実施する前日と当日のバックアップデータを比較し、時間の削減を目標とした。課題を証明する基礎実験として、CDSL のブログサイトを対象に、公開されている固定ページと投稿の記事の内容すべてを確認する自動化したプログラムを 2 回実行して、実行時間を Python の time モジュールを利用して、計測を行った。実験結果として、プログラム 2 回実行した際の時間はそれぞれ約 56 秒と約 62 秒だった。このことから、テスト範囲の対象を広げると時間がかかる。

謝辞 本稿の執筆に当たりご助言を賜りました、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の高橋 風太さんに御礼申し上げます。

参考文献

- [1] Bruderer, H. and Vakulov, A.: The Dangers of Digitization, and the Importance of Data Backup, *Commun. ACM*, Vol. 67, No. 5, p. 23–25 (online), DOI: 10.1145/3643990 (2024).
- [2] Ivanov, M. and Ravanliyski, R.: Copy Services: A Client's Shield Against Data Loss, *2023 Eight Junior Conference on Lighting (Lighting)*, pp. 1–4 (online), DOI: 10.1109/Lighting59819.2023.10299430 (2023).
- [3] Olaoye, G. and Luz, A.: Data backup and disaster recovery in the cloud (2024).
- [4] Suguna, S. and Suhasini, A.: Overview of data backup and disaster recovery in cloud, *International Conference on Information Communication and Embedded Sys-*

*5 <https://developer.wordpress.org/rest-api/>

- tems (ICICES2014)*, pp. 1–7 (online), DOI: 10.1109/ICICES.2014.7033804 (2014).
- [5] Xu, C. and Xu, C.: Design of Computer data Remote Backup System Based on Artificial Intelligence, *2024 International Conference on Distributed Computing and Optimization Techniques (ICDCOT)*, pp. 1–5 (online), DOI: 10.1109/ICDCOT61034.2024.10515712 (2024).
- [6] Eppen, G. D. and Iyer, A. V.: Backup agreements in fashion buying—the value of upstream flexibility, *Management science*, Vol. 43, No. 11, pp. 1469–1484 (1997).
- [7] Olaoye, G. and Luz, A.: Data backup and disaster recovery in the cloud (2024).
- [8] Moisescu, R.-C., Olteanu, C.-D., Deac-Suteu, D. V. and Țițu, A. M.: Deduplication Data Technologies Impact on the Backup Systems in Intellectual Property Organizations, *2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1–6 (online), DOI: 10.1109/ECAI54874.2022.9847495 (2022).
- [9] Chang, V.: Towards a Big Data system disaster recovery in a Private Cloud, *Ad Hoc Networks*, Vol. 35, pp. 65–82 (online), DOI: <https://doi.org/10.1016/j.adhoc.2015.07.012> (2015). Special Issue on Big Data Inspired Data Sensing, Processing and Networking Technologies.
- [10] Gupta, K. and Goel, A.: Requirement checklist for blog in web application, *International Journal of System Assurance Engineering and Management*, Vol. 3, No. 2, pp. 100–110 (online), DOI: 10.1007/s13198-012-0116-7 (2012).
- [11] 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).
- [12] Nemoto, J., Sutoh, A. and Iwasaki, M.: File System Backup to Object Storage for On-Demand Restore, *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, pp. 946–952 (online), DOI: 10.1109/IIAI-AAI.2016.116 (2016).
- [13] 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*, SIGMOD '02, New York, NY, USA, Association for Computing Machinery, p. 500–511 (online), DOI: 10.1145/564691.564749 (2002).
- [14] Sneha, K. and Malle, G. M.: Research on software testing techniques and software automation testing tools, *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, pp. 77–81 (online), DOI: 10.1109/ICECDS.2017.8389562 (2017).
- [15] Mallick, S. R., Lenka, R. K., Sudershana, S., Sahoo, A., Palei, S. and Barik, R. K.: An Investigation Into the Efficacy of RANOREX Software Test Automation Tool, *2023 3rd International Conference on Innovative Sustainable Computational Technologies (CISCT)*, pp. 1–5 (online), DOI: 10.1109/CISCT57197.2023.10351231 (2023).
- [16] Zhu, S., Fang, R., Wang, Y., Qiu, Y. and Shao, Z.: The determination and correction of measurement anomaly based on wavelet-threshold method, *2011 IEEE Power Engineering and Automation Conference*, Vol. 1, pp. 402–405 (online), DOI: 10.1109/PEAM.2011.6134906 (2011).
- [17] Yamashita, K., McIntosh, S., Kamei, Y., Hassan, A. E. and Ubayashi, N.: Revisiting the applicability of the pareto principle to core development teams in open source software projects, *Proceedings of the 14th International Workshop on Principles of Software Evolution, IWPSE 2015*, New York, NY, USA, Association for Computing Machinery, p. 46–55 (online), DOI: 10.1145/2804360.2804366 (2015).
- [18] Spool, J. and Schroeder, W.: Testing web sites: five users is nowhere near enough, *CHI '01 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '01, New York, NY, USA, Association for Computing Machinery, p. 285–286 (online), DOI: 10.1145/634067.634236 (2001).
- [19] Mahmoud, T., Girgis, M., Abdullatif, B. and Zaki, A.: An Automated Web Application Testing System, *International Journal of Computer Applications*, Vol. 99, pp. 37–44 (online), DOI: 10.5120/17387-7926 (2014).
- [20] Swathi, B.: Automated Test Case Prioritization and Evaluation using Genetic Algorithm, *2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS)*, pp. 1–5 (online), DOI: 10.1109/IC3SIS4991.2022.9885535 (2022).
- [21] Zheng, Y., Liu, Y., Xie, X., Liu, Y., Ma, L., Hao, J. and Liu, Y.: Automatic Web Testing Using Curiosity-Driven Reinforcement Learning, *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pp. 423–435 (online), DOI: 10.1109/ICSE43902.2021.00048 (2021).