DEIM Forum 2023 4b-8

ユーザ透過なP2P型Webアーカイブの導入検討

伊藤　璃音†　　松原　克弥††

† 公立はこだて未来大学　〒041–8655 北海道函館市亀田中野町 116 番地 2
†† 公立はこだて未来大学　〒041–8655 北海道函館市亀田中野町 116 番地 2
E-mail: †{b1019252,matsu}@fun.ac.jp

あらまし　今日の高度情報化社会において必要不可欠となった World Wide Web において，日々更新・消失するリスクがある Web ページを収集し蓄積する「Web アーカイブ」という取り組みがある．単一のシステムで世界中の Web ページを収集する，現状の集中型 Web アーカイブでは，運用コスト増加や非効率な Web ページ収集処理が課題となっている．本研究では，前述の課題に対処できる P2P 型 Web アーカイブ実装のひとつである IPWB を導入することを目的として，その導入障壁を軽減するために，ネットサーフィン中のユーザ透過な Web アーカイブ実行を可能にするブラウザ・プラグイン型の Web アーカイブ機構を実現した．

キーワード　InterPlanetary File System，InterPlanetary WayBack，WARC，分散ストレージ

1　はじめに

1.1　背　　景

インターネット上で最も普及し，利用されている情報システムである World Wide Web(以降 Web) では日々多様な情報が含まれたコンテンツが公開されている．膨大な量の Web コンテンツは現在も作成，更新され続けている．一方，更新に伴い消失するコンテンツも存在している．池内らの調査 [1] によると 2001 年に収集された 1000 万件の Web ページを対象として 2003 年に生存調査を行ったところ，12 年間で 90 %以上の Web ページにアクセスできなくなっている．そこで，削除・変更されて情報が失われるコンテンツを保存・蓄積する，Web アーカイブという取り組みがある．デジタル化が進み，Web への情報集中度が高まった現代社会では，重要な情報が含まれている可能性のあるコンテンツを削除される前に収集・蓄積し，後世に残すという取り組みは，現代のデジタル社会における重要な使命の一つである．世界中で Web アーカイブの取り組みは行われており，日本では国立国会図書館がインターネット資料収集保存事業 (WARP) [2] という事業として国内の Web のアーカイブを行っている．WARP は Web アーカイブを作成する際にユーザーが自分で url を指定して Web アーカイブを作成しなければならないが，ユーザーが url を指定しなくても網羅的に Web を収集しアーカイブを作成するサービスもある．例として，Internet Archive が運営する Web アーカイブサービスの，WayBack Machine がある．世界中に公開されているすべての Web コンテンツを対象として収集を行い，時系列順に並べて公開している．

1.2　Web アーカイブにおける課題

アーカイブ保存容量の増大

収集した Web ページを蓄積するストレージ容量の急激な増加によるシステム運用コストが大きな課題の一つとなっている．実態として WayBack Machine のストレージ容量は 70 ペタバイトにまで達しており [4]，そのコストのほとんどを寄付に頼っている．よって，単一の団体や組織で Web ページを収集，蓄積することは大きなコストがかかる．

効率的なアーカイブ収集

現在最も大規模である Web アーカイブサービスである WayBack Machine は 5850 億もの Web ページを蓄積しているが，それでも世界中の全ての Web ページを収集できているわけではない．これには，コンテンツの動的化や，収集プログラムのアクセスを拒否するページの存在，著作権など法的な課題なども影響している．それらが原因で既存の Web アーカイブサービスは Web ページの増加に対して収集，保管が追いついていない．

サーバクライアント型通信による保守管理コスト

サーバに全てのアクセスが集中する，サーバクライアント型システムではサーバのネットワーク通信費用がコストの大きな割合を占めており，非営利的な目的で運営される Web アーカイブシステムにとって大きな課題であると考えた．また，サーバを保守管理する人手が必要になるため，金銭面だけでなく人材を用意しなければいけない人的コストも存在する．

2　関連技術

本研究に関連する技術について述べる．

2.1　Peer-To-Peer(P2P)

Peer To Peer(P2P) 通信とはコンピュータが通信を行う際の方式の一つ．対等の通信相手と相互に通信を行い，データのやり取りを行う．対になる通信方式であるサーバクライアント型通信はデータやサービスを提供するサーバに利用者であるクライアントが要求を行い，それに対応した返答を行う通信方式である．サーバクライアント型通信は別のクライアントに通信を行う際に，サーバが中継して通信を行うのに対して，P2P 通信

はサーバやクライアントといった立場の違いが存在しないため，直接通信を行うことができる．
メリット：

- 特定の箇所に通信や負荷が集中しにくく，膨大な数のコンピュータによるネットワークでもパフォーマンスを保ったシステムの構築が可能 (高スケーラビリティ)
- サーバを利用しないため負荷が集中せず，保守管理や回線の用意が必要がないため低コストでシステムの運用が可能 (低コスト)
- 通信相手の端末全てが運用システム内で等価値であるため，特定の機材の故障によってシステム全体に影響を及ぼすような単一障害点が存在しない (耐障害性)

デメリット：

- 通信を集約するサーバが存在しないため，システム全容の把握が困難
- 一度共有されたデータは完全に削除することが困難
- ウィルスに感染した端末が P2P システムに自身の個人情報を共有してしまう危険があるなど，セキュリティ対策が困難
- ネットワーク全体に掛かる負荷が高くなるため，接続できる端末の上限がある

2.2 分散ハッシュテーブル

分散ハッシュテーブル (Distributed Hash Table：DHT) は P2P 通信を用いてハッシュテーブルを複数の端末で共有して管理する技術．保存する要素 (Value) をハッシュ関数に掛けて導かれた値 (Key) と紐づけて管理することで探索・追加の時間をテーブル内の要素数によらず定数時間 O(1) で行うことができる．機能は通常のハッシュテーブルと同等であるが，探索方法の特徴として，端末の離脱によるテーブルの消失や従来の P2P システムの欠点であったネットワークに掛かる負荷を低減できるため，DHT を利用しない P2P システムよりも使用帯域の削減や探索速度の向上といった利点がある．

2.3 kademlia

kademlia は DHT におけるノード (分散ハッシュテーブルに参加する端末) 間のルーティングや，データの探索を行うためのアルゴリズムである．ノード間の探索や他ノードとの接続を二分木構造を用いて行い，効率的に DHT を動作させ，ネットワークの負荷を低減している．

a） ルーティングテーブルの作成

ノードに対してハッシュ値を割り当てることで，[ハッシュ値・ノードの接続情報]：[nodeKey・nodeValue] で二分木構造のハッシュテーブルを生成することができる．このテーブルは他ノードと通信を行う際の接続情報として利用され，ノードに与えられたハッシュ値の XOR 距離が近い近隣のノードの情報を任意の数，経路表に記憶させる．

b） データの保存と参照

保存したいデータに対してハッシュ関数 (SHA-3) を掛けることで入力に対して一意で偏りが小さい Hash 値を生成する．ノードに保存を行う際には，ノードに与えられたハッシュ値と dataValue のハッシュ値の XOR 距離が最も近いノードにデータを保持させる．この保存方式を実現するために，特定の端末へのデータの集中を防ぐための偏りのないハッシュ値を生成する必要がある．

c） データの探索方法

保存されたデータを検索するためには，探索を始めたノードに登録されている他ノードの情報を参照しリクエストと探索対象の dataKey を送信する．リクエストを受け取ったノードはキーに対応する dataValue を自身が持っているかを確認し，持っていた場合は dataValue を返却する．持っていない場合は経路表の中で最も dataKey に XOR 距離が近い NodeKey を返却する．nodeKey を返却された場合には，再度返却された nodeKey に対して再度リクエストを送信するというプロセスをデータが見つけるか，一定回数繰り返すまで続ける．

2.4 IPFS

IPFS(InterPlanetary Filesystem) [5] は，Kademlia を利用した P2P 通信を採用した分散ファイルシステムサービスの 1 つであり，現在のインターネットの主要な通信プロトコルである HTTP に変わる通信方式として開発されている．分散ハッシュテーブルに基づいて一つのファイルを様々なネットワークに参加している様々な端末上に配置する．ファイルやディレクトリを参照する際はハッシュ値によって決定されたコンテンツ ID を IPFS ネットワークに問い合わせることで，通信者のネットワーク的な位置に応じた最適な場所からファイルの取得を行う．

メリット

- 耐障害性: 接続ノード 1 つ 1 つが等価値で，特定の 1 台の通信が止まっても障害にならない
- 負荷分散：通信が分散しているためネットワークへの負荷が小さい
- 耐検閲性：データの保存場所が分散されているため，削除されにくい
- 耐改ざん性：ハッシュ値によってデータを管理しているため，改ざんが困難

2.5 IPWB

IPWB(InterPlanetary WayBack) [6] は IPFS を使った Web ページデータの分散管理と，アーカイブ用のファイル形式である WARC ファイルフォーマット [7] を利用したアーカイブ登録を行うシステムである．アーカイブした Web ページを復元し閲覧する機能も備えている．IPWB は WARC 形式のファイルを読み取り，HTTP ヘッダーの情報とコンテンツブロックの情報を抜き出した後，コンテンツブロックを IPFS へ追加する．その後 IPFS へ追加した際に返るハッシュ値や，元の URL などメタ情報を返す．これを CDXJ と呼ばれるアーカイブ形式で保存する．

Index ：渡された WARC ファイルを IPFS へ登録する．登録したページの要素は CDXJ ファイルとして返却される

Replay ：アーカイブとして保存した Web ページを復元し，URL

で検索を行うことができる
IPWB は Docker コンテナ上で動作するイメージが公開されており，これを利用することで，アーカイブ作成・復元環境の複製が容易になる．

2.6 WARC File

WARC File は Web コンテンツのアーカイブ収集用途に特化したフォーマットで作られるファイル形式．同じ Web コンテンツでも異なる時間毎にデータを保存することができるため，時系列ごとのアーカイブを行うことが可能になる．作成された WARC ファイルは IPWB だけでなく，現在運用されている様々なアーカイブシステムで利用されている，汎用的なアーカイブファイルである．

2.7 WARC レコード

WARC ファイルには，保存するデータと，その補足情報が 2 組 1 セットで WARC レコードとして書き込まれる．WARC レコードには種類があり，種類によって補足情報の内容が異なってくる．WARC ファイルに書き込まれる内容（WARC record）は主に 3 つに分類される．

- warcinfo レコード
- request レコード
- response レコード

a） warcinfo レコード

warcinfo には，ファイルの作成日時，端末情報，使用しているフォーマットの指定など WARC ファイルの情報が記述される．warcinfo レコードは 1 つの WARC ファイルに 1 つのみ記述される．

b） request レコード

request レコードには，送信した HTTP リクエストの情報を記述する．

c） responce レコード

responce レコードでは，リクエストによって返却されたデータとその付加情報を記述する．

3 先行研究：伊藤らの研究

本研究に関連する研究について述べる．

subsection 先行研究における提案伊藤らの研究 [8] では前述したような課題の解決のために，各 Web ページの権利を保持するコミュニティ毎に Web アーカイブを行い，コンテンツ収集やデータ保存のコストを分散する非集中型の Web アーカイブシステムを提案した．

3.1 先行研究における課題

この提案では課題を 3 つに分け，それぞれについて解決を行った．下記に課題とその解決手法を述べる．

a） Web アーカイブにおけるコスト

既存の Web アーカイブサービスでは，コンテンツの収集にサーバを設置し，保存に大量のストレージを使用しているため，それらの用意や保守管理に多くのコストが掛かっている．伊藤らはコミュニティ内に存在する端末の余剰ストレージを利用した分散ファイルシステムを構築することで，保存用のストレージを不要とする手法を提案した．また，端末のブラウジングに連動してアーカイブも同時に行うことで，コンテンツ収集サーバの設置も不要とした．特定のサーバやストレージを保守管理する必要がなくなったことから，人件費も削減が可能となる．

b） 効率的な Web コンテンツの収集手法

多くの Web アーカイブシステムでは，コンテンツの収集のために収集プログラムであるクローラを用いている．しかし，収集対象であるコンテンツが膨大であるため，収集が追い付いていないという課題がある．伊藤らは，コミュニティに参加している端末のブラウジングを利用してコンテンツを収集する手法を提案した．これによって，価値のある情報が含まれている可能性の高いコンテンツの収集を効率的に行うことができる．

c） コンテンツの著作権問題

他人が公開した Web コンテンツを無断で複製し，別の場所で再公開する Web Archive は著作権の問題を抱えている．伊藤らの提案では，コンテンツの著作権を保有するコミュニティ内でアーカイブを行うことを推進しているため，著作権の問題を回避することができる．

3.2 提 案 手 法

ブラウザで開いた Web ページを自動的にアーカイブし IPFS に公開する手法を提案した．ブラウザ上に設置されたボタンを押すことで，ブラウザが開いている Web ページ GoogleChrome の拡張機能である，WARCreate [9] を利用して使用者がアクセスした Web ページのアーカイブファイルを作成する．ファイルの作成を検知すると IPWB に作成されたファイルを渡すという実装であった．この提案の概要を図で示す．

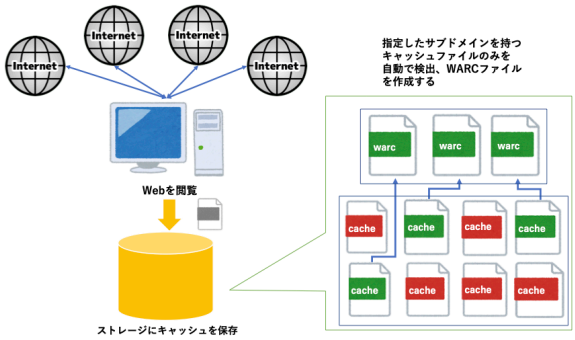


図 1 キャッシュをアーカイブの形式に変換

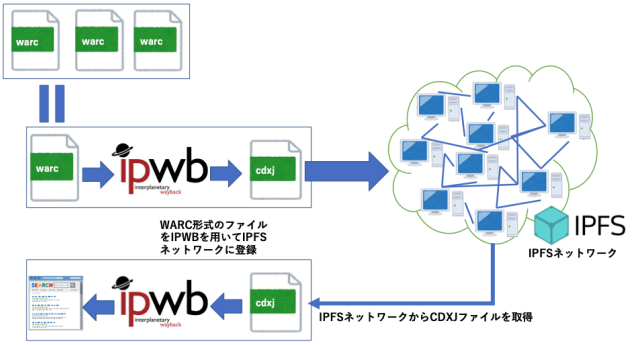


図 2　IPFS ネットワークでアーカイブを管理

3.3　先行研究における残課題

伊藤らの実験で用いられたアーカイブ生成システムでは，アーカイブ生成毎にユーザがボタン入力を行わなければならず，自動化が行われていないため個々のユーザーに負担が掛かってしまうことや，最新のブラウザアップデートに対応していないため，1 年以内にシステムが利用できなくなってしまうといった問題から，効率的な Web コンテンツの収集の手法が確立されていなかった．

4　提　　案

本研究では，新たなコミュニティの Web アーカイブ参入障壁を下げることを目的とする．目的を達成するための提案として，ブラウザ操作と連動するユーザ透過な P2P 型 Web アーカイブ機構を提案する．この提案では，ユーザのブラウジング情報から自動的にアーカイブファイルを作成し，ファイルの共有には分散型ファイルシステムである IPFS を利用してアーカイブファイルを保存することで，ファイルサーバやクローラ用サーバを使用せずに Web アーカイブ機構を実現することができる．P2P 通信でデータのやり取りをすることで，通信費用，電気代，保守管理等の人件費の削減になる．低コストであり，かつアーカイブ収集やシステムの保守管理に人的資源を消費しない Web アーカイブを実現することができる．

4.1　提案における技術的課題

本提案における技術的課題として，ブラウザ操作と連動したアーカイブファイルの自動収集機構を実現することがあった．アーカイブファイル作成の自動化は本提案における人的資源を削減するための必要機能であり，この課題を解決することで，提案の実現が可能になる．

4.2　技術的課題の解決方法

本提案の実現手法として，2 つの手法を検討した．

a）　ブラウザのキャッシュ情報を利用したアーカイブファイル作成

ユーザ端末に保存されているキャッシュファイルを利用して，アーカイブを作成する手法を調査した．調査内容として，OSS ブラウザである FireFox の実装を元にしキャッシュファイルの仕組みを解析し，アーカイブファイル作成に活用できるかを検討した．ブラウザのキャッシュファイルは，キャッシュした全てのコンテンツを保存しておくフォルダと，それらの情報を管理する Index ファイルから構成されていた．Index ファイルはバイナリで記述がなされており，内容を読み解くことで，コンテンツデータと Web ページが関連付けされていた．

b）　ブラウザ拡張機能を利用したアーカイブファイル作成

ブラウザ拡張機能を利用して，Web コンテンツをアーカイブする手法を調査した．調査内容として，GoogleChrome の拡張機能で利用可能なライブラリや API 機能，類似実装を確認した．結果として，開いている Web ページに含まれている情報をプログラム的に取得できる Document Object Model(DOM) と，URI に対して HTTP リクエストを送信することができる Fetch API を利用することで課題を解決できることが判明した．

4.3　実装手法の決定

実現可能な手法として，ブラウザのキャッシュファイルを解析し作成する手法，ブラウザの Web ページ上でページの操作や情報の取得を行うスクリプトを実行することができるブラウザ・プラグインとして実装する手法の二つを検討した．本研究では，普及率が高いブラウザに着目し，開発者サポートや開発情報が多く開発環境が充実していることや，システム作成後のインストールが手軽であるので利用しやすいという点から GoogleChrome のプラグインである Chrome 拡張機能として実装することとする．

5　実　　装

P2P 型 Web アーカイブシステムを実現するために必要な，効率的なコンテンツ保存の手法を提案する．

5.1　実 装 内 容

本研究では，現状の Web アーカイブサービスの課題である，「コンテンツ収集効率の向上」を解決し、非集中型 Web アーカイブを実現するために，ブラウザ・プラグイン型のアーカイブファイル自動生成システムを提案する．この手法で実現されたシステムを用いることで，課題であったアーカイブの収集効率の向上や使用者の操作負担を軽減することができる．伊藤らの研究で利用していた WARC ファイル生成システムであるブラウザ拡張機能の WARCreate では，ページを保存するために操作者がボタン操作を行う必要があるため，前述の課題を解決することができない．そのため自動でファイルを生成する新たな手法の提案が必要である．

5.2　提 案 手 法

- アクセスした Web ページに含まれる要素の URI を取得する機能
- URI を元に，Web ページのデータを取得する機能
- WARC ファイルに必要なメタデータを生成する機能
- 取得・生成したデータを WRAC ファイルに整形し出力する機能

上記の機能について実装を行う．2023 年に行われる Chrome

拡張機能のマニュフェスト更新によって利用できなくなる機能として，アクセスしたページのデータを取得できる機能を持つ XMLHttpRequest がある．類似の実装である WARCreate ではページのソース (HTML ファイル) や画像データを取得する際に利用していたが，代替機能として fetchAPI [11] が推奨されているため，本実装では fetchAPI を利用する．また，アクセスしているサイトの URL やアクセス日時・使用している端末の IP の情報が WARC ファイルに必要なため，ページの情報をプログラムで管理するための仕組みである Document Object Model(DOM) を利用することで取得する．これらの情報を WARC ファイルフォーマットに沿って整形することでファイルの作成を行う．

a） アクセスした Web ページに含まれる要素の URI を取得する機能

ページに含まれる要素を取得する手法として，Web ページ内に含まれる要素を参照し操作する DOM を利用した．本実装では，下記の要素を取得する．

- document.documentURI：ページ本体の URI（HTML）
- document.images：画像データのリスト（JPEG，PNG，SVG，etc.）
- document.styleSheets：スタイルシートのリスト（CSS）

取得した要素のネットワーク的な場所を示す URI のみを抜き出し，リストに格納する．

コンテンツ URI リストの例

```
pageURI: 'https://www.fun.ac.jp/president-message', imageURIs: Array(6), styleURIs: Array(8)}
imageURIs:Array(6)
0: "https://www.fun.ac.jp/wp-content/themes/fun_2204/img/title-ja.svg"
1: "https://www.fun.ac.jp/wp-content/themes/fun_2204/img/icon-nav-search.svg"
2: "https://www.fun.ac.jp/wp-content/themes/fun_2204/img/icon-nav-search.svg"
3: "https://www.fun.ac.jp/wp-content/uploads/2020/02/MG_6259-e1458735442869-624x693-480x533.jpg"
4: "https://www.fun.ac.jp/wp-content/themes/fun_2204/img/title-ja.svg"
5: "https://www.fun.ac.jp/wp-content/themes/fun_2204/img/title-ja.svg"
length: 6
[[Prototype]]: Array(0)
pageURI: "https://www.fun.ac.jp/president-message"
styleURIs: Array(8)
0: "https://www.fun.ac.jp/wp-includes/css/dist/block-library/style.min.css"
1: "https://www.fun.ac.jp/wp-includes/css/classic-themes.min.css"
2: null
3: "https://www.fun.ac.jp/wp-content/themes/fun_2204/style.css?fver=20220418010104"
4: "https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css"
5: "https://fonts.googleapis.com/icon?family=Material+Icons"
6: "https://fonts.googleapis.com/css?family=Roboto:900&display=swap"
7: "https://fonts.googleapis.com/css?family=Red+Hat+Display:900&display=swap"
```

5.2.1 取得した URI を元に，ページのコンテンツを取得する機能

ページ要素の URI リストを元に，URI に対して HTTP リクエストを作成する機能を持つ Fetch API を使用しリクエストを送信する．URI リストに重複がある場合があり，無駄な通信を行わないよう，リクエスト送信前に重複データの削除と NULL チェックを行う．返却されたレスポンスは格納され，WARC ファイルフォーマットに沿って整形する．

a） 取得した URI を元に，ページのコンテンツを取得する機能

ページ要素の URI リストを元に，URI に対して HTTP リクエストを作成する機能を持つ Fetch API を使用しリクエストを送信する．URI リストに重複がある場合があり，無駄な通信を行わないよう，リクエスト送信前に重複データの削除と NULL チェックを行う．返却されたレスポンスは格納され，WARC ファイルフォーマットに沿って整形する．

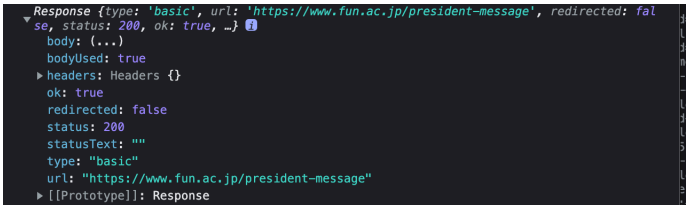


図 3　HTTP リクエストのレスポンス

b） WARC ファイルに必要なメタデータを生成する機能

WARC ファイルに書き込まれる内容（WARC record）は主に 3 つに分類される．

- warcinfo レコード
- request レコード
- response レコード

i） warcinfo レコード

warcinfo には，ファイルの作成日時，端末情報，使用しているフォーマットの指定など WARC ファイルの情報が記述される．warcinfo レコードは 1 つの WARC ファイルに 1 つのみ記述される．

warcinfo レコードの例

```
WARC/1.0
WARC-Type: warcinfo
Content-Type: application/warc-fields
WARC-Date: 2023-01-11T05:28:46Z
WARC-Filename: 2023111.warc
WARC-Record-ID: <urn:uuid:14dea44f-5253-4816-8c8c-b73ebb5b8ed6>
Content-Length: 279

software: getPageData
format: WARC File Format 1.0
conformsTo: http://bibnum.bnf.fr/WARC/WARC_ISO_28500_version1_latestdraft.pdf
http-header-user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36
```

ii） request レコード

request レコードには，送信した HTTP リクエストの情報を記述する．Fetch API では，Request() コンストラクタによって，リクエストをプログラム内で処理できるオブジェクトとして扱うことができるため，request レコードには Request オブジェクトの内容が記載される．Request オブジェクト内にはリクエスト先の URI やリクエストの形式，リクエスト送信元の端末情報が含まれており，それらを request レコードに格納する．

request レコードの例

```
WARC/1.0
WARC-Type: request
WARC-Target-URI: https://www.fun.ac.jp/president-message
WARC-Date: 2023-01-11T05:28:46Z
```

```
WARC-Filename: 2023111.warc
WARC-Record-ID: <urn:uuid:bba9a594-d8bb-41f6-aa7c-14085597fa57
    >
Content-Length: 258

Content-Type: application/http;msgtype=request
GET https://www.fun.ac.jp/president-message HTTP/1.1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0
     Safari/537.36
Connection: close
```

iii） response レコード

response レコードには，HTTP リクエストを送信した後に返却されたレスポンスを記述する．返却されたレスポンスを必要な情報の粒度に分解し，記述する．また，レスポンスがベクターデータ以外の画像データの場合は，Base64 形式に変換を行った後に記述する．

response レコードの例

```
WARC-Type: response
WARC-Target-URI: https://www.fun.ac.jp/president-message
WARC-Date: 2023-01-11T05:28:46Z
WARC-Filename: 2023111.warc
WARC-Record-ID: <urn:uuid:f31c8f03-46c0-499d-a918-b95e1dbd16a2
    >
Content-Length: 40560

HTTP/1.1 200 basic
Content-Type: application/http;msgtype=request
Content-Type: text/html
Content-Length: 40438

text data here
```

c） 取得したデータを WRAC ファイルに整形し出力する機能

生成した WARC record を，ファイルに記述する．初めに warcinfo レコードを記述し，その後に request レコード，対応する response レコードの順で並ぶように整形を行う．生成された WARC ファイルは DOM によってページに追加され，端末へのダウンロードが開始される．全ての WARC record に含まれている WARC ヘッダの記述も，同時に行う．

6 評価実験

本研究で実装したアーカイブファイル自動生成システムを用いて，提案システムの評価実験を行う予定である．

6.1 実験の概要

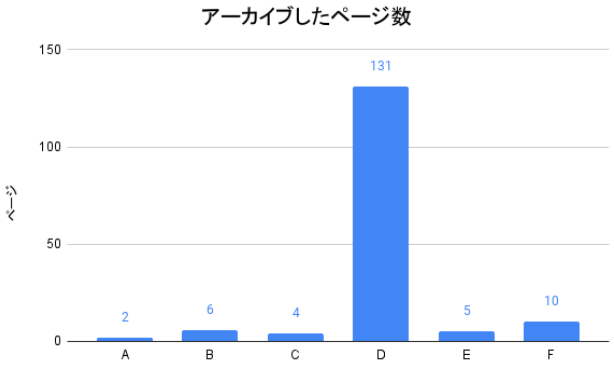
本研究で提案したブラウザ操作と連動するユーザ透過な P2P 型 Web アーカイブ機構の評価実験を 5 章で示した実装を元に行う．評価項目として，アーカイブの生成頻度や消費したディレクトリの容量などを検討している．

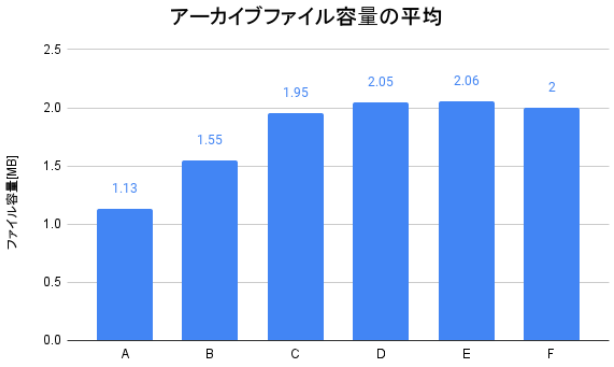
- 対象者：Mac と Linux を利用している人物 6 名
- 期間：1 週間
- アーカイブ対象サイト：気象協会 Web サイト（tenki.jp）

この実験では，Web アーカイブシステムがが正常に動作しており，目的であった Web アーカイブの低コスト化が実現してるかどうかを評価する．

6.2 結　　果

結果として，期間内で合計 158 ページのアーカイブを行い，合計容量は 195MB だった．1 ページあたりの平均容量は 1.79MB でアーカイブのページ数が特に多かった被験者に対しても大きく平均から逸脱することはなかった．





6.3 考　　察

今回のアーカイブでは，ページの平均サイズが比較的小さく，大量のページを保存しても使用した保存容量のサイズはあまり大きくなかったが，画像が大量に含まれているページではさらにアーカイブの容量が増大することが考えられる．今回アーカイブの対象とした Web ページは天気の情報をもとに動的に生成される Web ページであり，天気のアイコンなど重複したデータが保存されていることが考えられる．このことから画像サイズなどが変更された類似データを複数保存するのではなく，1 つのデータからパラメータを用いて画像を生成するような仕組みがあれば，保存容量の削減が望めると考えた．

7 おわりに

まとめ

本研究はブラウザキャッシュから WARC ファイルを自動的に生成し分散ファイルシステムによって管理することで，従来の Web アーカイブの抱える問題を解決することを目標とした．この提案が実現することで，アーカイブした Web ページが分散して保存されるようになり，一つの組織が莫大な費用を負担す

る，Web ページの増加に対してアーカイブの速度が追いついていないという課題を解決できる．そして，実用的・重要な情報を損なうことなく後世に伝えることが可能になる．

今後の課題

アーカイブの自動作成がユーザのブラウジングに与える心理的影響や，アーカイブが透過的に行えているかの調査を行うことが必要であると考える．

文　献

[1] 池内 淳，安形 輝，野末 道子: ウェブの動的変化に関する調査，三田図書館・情報学会研究大会発表論文集, 19-22, 2003.

[2] 国立国会図書館インターネット資料収集保存事業: https://warp.da.ndl.go.jp/

[3] InternetArchive: https://archive.org/

[4] Donate to the Internet Archive: https://blog.adafruit.com/2020/12/01/donate-to-the-internet-archive-digital-library-of-free-borrowable-books-movies-music-wayback-machine-internetarchive/

[5] IPFS(InterPlanetary FileSystem): https://ipfs.tech

[6] IPWB(InterPlanetary WayBack): https://github.com/oduwsdl/ipwb

[7] WARCFileFormat: https://archive-access.sourceforge.net/warc/

[8] 伊藤晋梧，松原克弥: ブラウザキャッシュを活用した P2P 型 Web アーカイブ機構の検討 https://www.ipsj.or.jp/event/taikai/84/ipsj_web2022/data/pdf/2N-08.html

[9] WARCreate: https://warcreate.com/

[10] XMLHttpRequest: https://developer.mozilla.org/ja/docs/Web/API/XMLHttpRequest

[11] fetchAPI: https://developer.mozilla.org/ja/docs/Web/API/Fetch_API

[12] DOM(DocumentObjectModel): https://developer.mozilla.org/ja/docs/Web/API/Document_Object_Model.