

拡散レポジトリを用いた IPFS 可用性向上技術

ワン シュ[†] 上山 憲昭^{††}

[†] 立命館大学院 情報理工研究科

〒567-8570 大阪府茨木市岩倉町 2-150

^{††} 立命館大学 情報理工学部

〒567-8570 大阪府茨木市岩倉町 2-150

E-mail: [†]gr0695vx@ed.ritsumeai.ac.jp, ^{††}kamiaki@fc.ritsumeai.ac.jp

あらまし 分散型ストレージシステムとして注目される IPFS (InterPlanetary File System) は、コンテンツアドレッシングと P2P ネットワークを基盤とする革新的技術である。しかし、各ノードが保持するデータは所有者の自由意志に委ねられており、結果として同一データが地理的に集中して保存される可能性がある。このため、大規模災害や停電が発生した場合、特定データにアクセスできなくなるリスクを避けられない。既存の Filecoin や Kademia DHT では、経済的インセンティブやランダム割当によってデータの分散を試みているが、耐障害性の向上には限界がある。本稿では、各ノードの自律的な処理で、空間的に同一ノードの配置場所を拡散させることを目的に、ピア ID とデータ ID の上位ビットを照合し、一致する場合に対象データを永続的に保持する方式を提案する。さらに、地理的に分散したレポジトリに同一ビット列を割り当てるピア ID 付与アルゴリズムを導入することで、災害発生時においてもデータ可用性を維持できる IPFS 可用性向上技術を実現する。シミュレーション評価により、本提案手法が従来方式と比較して通常時の取得遅延を抑制し、災害時の復元成功率を向上させることを確認する。

キーワード IPFS, ネットワークトポロジ, 耐障害性

Availability Enhancement of IPFS Using Diffused Repository

Shu WANG[†] and Noriaki KAMIYAMA^{††}

[†] Graduate School of Information Science and Engineering, Ritsumeikan University

2-150 Iwakura-cho, Ibaraki, Osaka, 567-8570 Japan

^{††} College of Information Science and Engineering, Ritsumeikan University

2-150 Iwakura-cho, Ibaraki, Osaka, 567-8570 Japan

E-mail: [†]gr0695vx@ed.ritsumeai.ac.jp, ^{††}kamiaki@fc.ritsumeai.ac.jp

Abstract The InterPlanetary File System (IPFS), a decentralized storage system, is based on content addressing and peer-to-peer networking. However, the data stored in each node is determined by the free will of owners, which often results in the geographical concentration of identical data. Consequently, when large-scale disasters or power outages occur, there is a significant risk of losing access to specific data. Existing approaches such as Filecoin and Kademia DHT attempt to achieve data distribution through economic incentives or random allocation, but their ability to enhance fault tolerance is limited. To address this problem, we propose a novel method in which nodes permanently store data if the upper bits of their Peer ID match the upper bits of the Data ID. In addition, we introduce a Peer ID assignment algorithm that allocates identical bit strings to repositories distributed across different geographic regions. This design enables the system to maintain data availability even in the event of large-scale disasters. Through simulation-based evaluations, we demonstrate that the proposed method outperforms conventional approaches by reducing retrieval latency under normal conditions and significantly improving recovery success rates during disaster scenarios.

Key words IPFS, Network Topology, Disaster Resilience

1. はじめに

IPFS (InterPlanetary File System) は、近年分散型ウェブやブロックチェーン技術の発展とともに注目を集めている次世代型の P2P ストレージネットワークである。その基本的な特徴は、従来のインターネットにおいて広く利用されている HTTP のような位置情報ベースのアクセス方式ではなく、コンテンツアドレス方式を採用している点にある。具体的には、ファイルを固定のサーバから取得するのではなく、データ自体の内容をハッシュ関数によって生成された一意の識別子によって管理し、ネットワーク上の複数のノードから最も効率的に取得できる場所を探索する仕組みを持つ。これにより、特定のサーバへの依存が解消され、障害や検閲に対して強靱な耐性を備えることが可能となる。

さらに、IPFS ではデータがチャンク (小さなブロック) 単位で分割・保存され、複数のノードに分散配置される [2]。この仕組みによって、同一データを保持するノードが増えるほど冗長性と可用性が高まり、また最寄りのノードから効率的にデータを取得できるため、通信遅延の削減や帯域幅の有効活用が期待される。加えて、従来の CDN (Content Delivery Network) が持つキャッシュ配信の仕組みと比較しても、より自律的かつ低コストで大規模分散を実現できる点が大きな利点である。

その一方で、現行の IPFS が採用している複製戦略は静的であり、ファイルの人気度 (アクセス頻度) や時間的な需要変動を十分に反映できていないという課題が存在する。たとえば、アクセスが集中するファイルに対しては十分な複製が行われず遅延が増大し、逆にほとんど利用されないファイルが冗長に保存されることでストレージ資源が無駄に消費される可能性がある。こうした背景から、近年の研究ではファイルの需要予測やアクセスパターンの分析に基づき、複製数や配置を動的に調整する戦略の導入が検討されている。特に、機械学習を用いて将来のファイル人気度を予測し、適応的にレプリカを管理する手法は、IPFS のサービス品質を向上させる有効なアプローチとして注目を集めている。

このように、IPFS は単なる分散型ストレージにとどまらず、ブロックチェーンや Web3.0 といった新たなインターネット基盤を支える重要な技術として発展を続けている。その特性は、検閲耐性の高い情報共有、災害時のレジリエントな通信、学術データやコンテンツ配信の効率化など、多岐にわたる応用可能性を秘めており、今後も研究と実用の両面で広がりを見せることが予想される。

しかし実運用において、どのデータを自身の IPFS ノードにピンニングさせる (Pinning)、つまり永続的に保持することかはそのノードの所有者が自由意志で決める。ノードの自律的判断に依存したデータ重複保持が以下の深刻な課題を引き起こしている。

地理的集中問題: 同一データが物理的に近接したノード群に集中保持される場合あり

永続性保証欠如: 停電・災害時におけるデータアクセス不能リスク

冗長度制御不能: データ所有者が保持ノード数と配置を明示的に制御不可 [3]

そこで本稿は、Y 次ビットを用いた自律的冗長制御と空間最適化ピア ID 付与技術を用いて、より効率的な、地震等の広域災害にも耐える IPFS 可用性向上法を提案する。以下、??節で関連研究についてまとめ、3. 節で提案方式について述べる。そして 4. 節で数値評価結果について述べ、最後に 5. 節で全体をまとめる。

2. 関連研究

2.1 IPFS

IPFS (InterPlanetary File System) は、分散型の p2p ファイル保存や共有システムとして、主に以下二つの技術で実現している。

DHT ベースの配置方式: IPFS におけるコンテンツ検索は、Kademlia アルゴリズムに基づいた DHT (Distributed Hash Table) を利用して行われる [4]。各ノードにはランダムに生成された Peer ID が割り当てられ、各コンテンツはハッシュ関数によって一意に決まる CID (Content Identifier) を持つ。DHT においては、CID と Peer ID の間の XOR 距離が定義され、この距離が最も近いノードが当該コンテンツを保持する候補として選択される。検索要求が発生すると、ノードは XOR 距離の近いノードに問い合わせを転送し、段階的に目的の CID に到達する。これにより、中央サーバを必要としない分散型のルーティングが実現される。

チャンク化とデータブロック管理: IPFS ではファイル全体をそのまま保存するのではなく、通常 256KB 単位に分割されたデータブロック (チャンク) として管理する。各ブロックは独自の CID を持ち、ネットワーク上の複数ノードに複製される。ユーザが特定のファイルを要求する際には、そのファイルを構成するすべてのブロックを並列的に探索し、取得したブロックを順次結合することで完全なファイルが復元される。この仕組みによって冗長性はある程度担保されるが、災害等により特定ブロックが保持されているノードが失われた場合、そのブロックに依存するファイル全体の復元が困難になる。したがって、従来方式では可用性の保証が不十分であることが課題となっていた。

2.2 Filecoin

IPFS の実用化と性能向上を目的として、既存の Filecoin [7] や Kademlia DHT [5] では、経済的インセンティブやランダム割当によりデータ分散を試みるが、大規模災害時の耐障害性向上には限界がある。

Filecoin は IPFS 上のストレージ提供者と利用者間でインセンティブを与えるブロックチェーンベースのプロトコルであり、データ保存の契約や報酬を暗号通貨によって管理することで、ノードが長期的かつ信頼性の高いデータ保持を行う動機を与えている。これにより、IPFS 単体では課題となっていた「誰が長期間データを保持し続けるのか」という持続性の問題に一定の解決策を提示している。一方で、Filecoin は契約や証明の手続きに高い計算コストを要するため、処理遅延やエネルギー

消費の増加といった問題が存在し、また小規模ファイルや短期間利用のデータに対してはオーバーヘッドが大きい点が指摘されている。

2.3 Cluster

IPFS Cluster は複数の IPFS ノードを統合的に管理し、データ複製の配置制御や負荷分散を行うためのレイヤである [6]。従来の IPFS ではノード間での自律的なデータ保持に依存していたが、Cluster を利用することで、管理者がレプリカ数や配置ポリシーを柔軟に指定でき、サービス品質を安定的に確保できる利点がある。特に、大規模システムにおいてはデータの可用性向上やストレージ資源の効率化に効果を発揮する。しかしながら、その制御は基本的に中央的なマネージャに依存するため、完全な分散性や自律性を損なう。また、ノードの動的参加や大規模な障害環境においては、手動での調整が必要となるなど、管理者への負担が大きく、スケーラビリティやレジリエンスの面で限界が存在する。

3. レポジトリを用いた IPFS 可用性向上技術

3.1 ピア ID に基づくデータ保存方式

本稿では、複数の事業者が永続的にデータを保持するノードを「レポジトリ」と定義する。各レポジトリは通常の IPFS ノードとして動作するが、自身が永続保存対象とするデータに対してピンングを行う点が特徴である。ピンング対象の決定方法としては、各レポジトリが自身のピア ID の上位 y ビットと、データの名前のハッシュ値の上位 y ビットが一致する場合に、そのデータを永続的に保持するという規則を導入する。ここで y は制御パラメタであり、システム全体のレポジトリ数を N とすると、各データは約 $N/2^y$ 個のレポジトリで保持される。この仕組みにより、 y を小さく設定すれば保持レポジトリ数が増加し、障害やノード離脱に対する可用性が向上する。一方、 y を大きくすれば保持レポジトリ数は減少するが、インセンティブコストが低減するため、可用性とコストの間にトレードオフが成立する。この制御性こそが本方式の大きな利点である。

さらに、すべてのデータに共通の y を設定する運用方法と、データ所有者ごとに y を任意に選択できる運用方法が考えられる。いずれの場合も、Filecoin のような検証メカニズムを応用し、レポジトリ同士が定期的に相互確認を行うことで、データ保持の順守性を確保する。これにより、システムは低い制御負荷でありながら、任意のデータを任意のレポジトリに永続保持させることが可能となり、IPFS の可用性を柔軟に向上させることができる。

3.2 ピア ID 付与アルゴリズム

レポジトリに割り当てるピア ID の設計方法について述べる。本手法の核心は、地理的に分散したレポジトリに対して、上位ビットが同一となるピア ID を割り当てることである。これにより、地理的に離れた位置にあるレポジトリが同一のデータを保持する確率が高まり、大規模災害などに対する耐障害性が向上し、また各ノードからデータを取得する際の遅延を削減できる。

中央管理機関 (authority) が存在する場合、この authority が

全レポジトリを管理し、上位ビットの割当てを一括で決定する。具体的には、レポジトリ集合 S_k を定義し、その集合を第 $k+1$ ビットによって二分割する操作を繰り返す。この際、レポジトリ n から新たに生成された部分集合への最短ホップ距離の和 T_n を計算し、 $\sum T_n$ が最小化されるように 0 または 1 を割当てる。貪欲法を適用し、再帰法で割り当てる。アルゴリズムでは、ノードの連結性に基づき定義される T_n に加え、コスト関数の加重項として地理距離 d_{geo} を併用する。ここで d_{geo} は任意の二ノードの間の直線距離であり、平面座標系ではユークリッド距離によって定義する。これにより、ネットワーク構造を反映した効率的なピア ID の階層的割当てが実現される。

一方で、中央管理機関が存在しない完全分散環境では、各レポジトリは新規参加時に隣接レポジトリのピア ID と座標情報を収集する。その上で、自身の位置に近い既存レポジトリ群の上位ビット列を参照し、それらとのノルム距離 (例: ハミング距離) が最大化されるように自身の上位ビットを決定する。これにより、地理的に近いレポジトリ同士でもピア ID の識別性が確保され、システム全体で均衡の取れた分散構造が維持される。

Algorithm 1 Peer ID Assignment Algorithm

- 1: Input: Graph $G = (V, E)$ with coordinates (x_v, y_v) , max depth K , weight α
 - 2: Output: K -bit Peer ID prefixes, for all repositories
 - 3: **for** $k \leftarrow 0$ to $K - 1$ **do**
 - 4: Partition current set S_k into two subsets $S_{k+1}(0)$ and $S_{k+1}(1)$
 - 5: Compute cost $T_n = d_{hop}(n, S_{k+1}(0)) + d_{hop}(n, S_{k+1}(1))$
 - 6: Add geographic penalty:

$$C = \sum_{n \in S_k} T_n + \alpha \cdot \sum_{i, j \in S_{k+1}} d_{geo}(i, j) \quad (1)$$
 - 7: Choose partition minimizing C
 - 8: Assign $(k + 1)$ -th bit = 0 or 1 to nodes in each subset
 - 9: **end for**
 - 10: Return all nodes with K -bit prefixes
-

本提案手法を用いることで、従来の IPFS が抱えていた「保持ノード数や位置を制御できない」という課題を解消できる。Filecoin の経済的インセンティブや IPFS Cluster の複製管理では十分に対応できなかった「地理的冗長性」や「制御可能な可用性」を実現することが可能となる。また、制御パラメタ y を通じて可用性とコストのバランスを柔軟に調整できる点も、本手法の大きな特長である。さらに、中央管理型と分散管理型の双方に対応できる設計を採用することで、多様な運用環境に適応可能であり、現実的な導入を見据えた汎用性を持つ。

以上のように、本稿の提案方式「レポジトリ方式+ピア ID 付与アルゴリズム」は、レポジトリの導入とピア ID 付与アルゴリズムを組み合わせることで、IPFS におけるデータ可用性を効率的かつ柔軟に向上させる新しい枠組みを提供する。複数の事業者によって提供されるレポジトリノードが、特定条件に基づいてデータを永続的に保持し、各レポジトリは自身のピア ID の上位 y ビットと、データハッシュの上位 y ビットが一致するコンテンツを必ず保持するという規則に従う。これにより、任

意のデータについて、平均的に $N/2^y$ 個のレポジトリで永続的にキャッシュされることが保証される。

4. 性能評価

4.1 評価シナリオ

以下の二つの主要なシナリオを設定して評価を行う。

通常状態シナリオ: 全ノードがオンラインで稼働している環境を想定し、データ取得遅延を測定する。これにより、ランダムピニング方式と提案方式が正常稼働時にどの程度効率的な検索性能を示すかを評価する。

災害発生シナリオ: 地震や停電などの大規模障害を模擬するため、シミュレーション上で特定範囲内のノードを一斉に無効化した。具体的には、災害範囲を半径 r とする円形領域として定義し、その内部に含まれるノードをすべてシャットダウンする。このとき、各ファイルに対する取得要求を発行し、成功率を算出することで、両方式の耐障害性を比較する。災害範囲を段階的に拡大することで、システム全体の復元力がどのように変化するかを分析する。

4.2 比較方式

従来の IPFS において、ノードがどのデータを保持するかは各ノードの自律的な判断に委ねられており、特定のデータを長期間保持させる強制力は存在しない。そのため、計算機シミュレーションではランダムピニングを比較方式として用いる。すなわち、ファイルを構成する各データブロックについて、保持するノードがランダムに決定される。これは、実際の IPFS ネットワークにおける「自律的で不均一なデータ保持」を模擬する方法であるが、保持ノード数や配置を制御できないため、データが特定の地域に偏って保存される可能性がある。このことは、可用性および耐障害性の観点から大きなリスクを内包している。

4.3 評価指標

評価のための主要指標は以下の通りである。通常状態シナリオにおいて、両手法の各コンテンツ要求における取得時間の平均値(平均取得遅延)、要求ノードからデータ保持ノードに到達するまでの平均ホップ数(平均ホップ数)、4ホップ以内にデータを取得できる確率(4ホップ復元率)を測定する。災害発生シナリオにおいて、すべてのチャンクを取得しファイルを完全に復元できた割合である可用性(復元成功率)を評価指標とする。

4.4 公平性の確保

本稿では、両方式の比較を公正に行うため、各データブロックについて、従来方式と提案方式で同一数のノードが保持するよう調整した(人気度による y の値を選択する)。ノード数、トポロジ構造、通信帯域幅、遅延パラメタは全て同一設定とする。各シナリオについて120回の反復シミュレーションを行い、それらの平均値で評価する。災害シナリオでは、災害範囲の中心を固定し、半径のみを変化させる。これにより、ランダム性によるばらつきを排除した上で、損害規模の拡大に伴う影響を分析する。

4.5 使用データセット

評価に用いるデータセットは、既存のインターネットトラフィック研究に基づき、現実的な分布特性を反映させて生成す

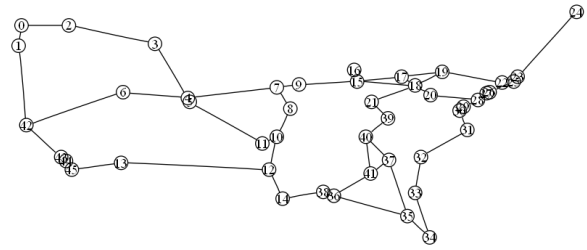


図 1: Network Topology of at home network

る。表 1 のように、実際の Web や FTP におけるファイルサイズ分布で、右に裾の長い重尾分布に従うことが知られている [8] ことからファイルサイズ分布は対数正規分布 (Lognormal) を用いる。

表 1: 1000 個ファイルの例 (単位: Mb)

	MEAN	MID	STD	MAX	MIN
File Size	4.52	2.73	7.06	139.82	0.1

ファイル数は 10, 100, 1000 の 3 パターンを評価し、それぞれの規模で両方式の性能を比較した。各ファイル人気度分布は Zipf 分布に基づき、人気度パラメタ θ を 0.6~0.9 の範囲で変化させて要求を生成した [9]。これにより、人気度が高い少数のファイルにアクセスが集中する現象を再現する。

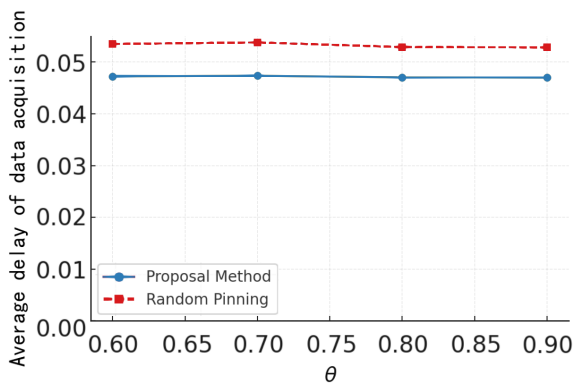
4.6 ネットワークトポロジ

図 1 に示す米国商用 ISP である at home ネットワークのネットワークトポロジを用い、提案方式と従来方式(ランダムピニング)を同一条件下で適用し、性能を比較評価する。

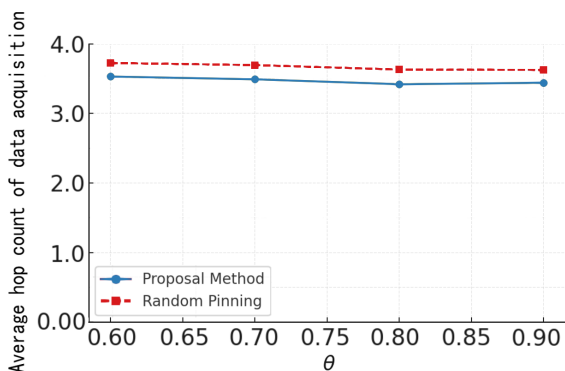
4.7 数値結果

図 2 に通常状態のシナリオで、Zipf 分布に従う 1000 ファイルを対象とし、パラメタ θ を変化させた場合の、(a) 平均データ取得遅延、(b) 平均ホップ長、(c) 4 ホップ以内でデータが取得できた割合、を各々示す。まず、平均データ取得遅延に関しては、提案方式が従来方式に比べて常に約 10%-20% 程度短縮されており、安定した低遅延を実現していることが分かる。次に、チャンクごとの平均ホップ数に着目すると、提案方式は従来方式よりも常に 10%-15% 前後少なく、効率的にデータ取得が可能であることが確認された。さらに、4 ホップ以内での復元成功率では、提案方式が従来方式に比べて常に 20% 以上高く、特に低人気ファイルを含む環境においても高い可用性を維持している。この結果から、提案方式は θ の変動、すなわち人気度分布の偏りに依存せず安定した性能を発揮することが示唆される。特に、災害時や低人気データが多数存在する状況下においても、従来方式に比べて優れた復元効率と可用性を提供できる点が、本手法の大きな強みであると結論づけられる。

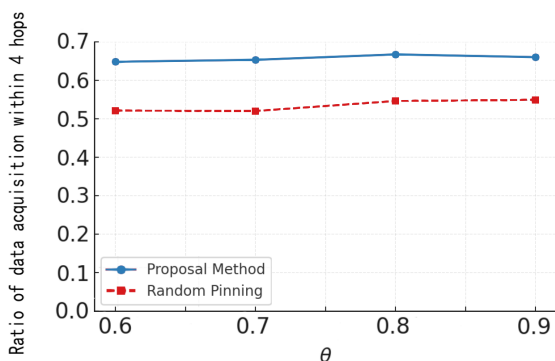
図 3 に、災害状況のシナリオで、ファイル数が 10, 100, 1000 の 3 つの各場合において、災害半径に対して復元成功率をプロットする。いずれの規模においても提案方式は従来方式を一貫して上回り、特に災害規模が大きい場合における差異は顕著となった。提案方式は従来方式に比べておよそ 20% 程度高い



(a) Average delay of data acquisition



(b) Average hop count of data acquisition

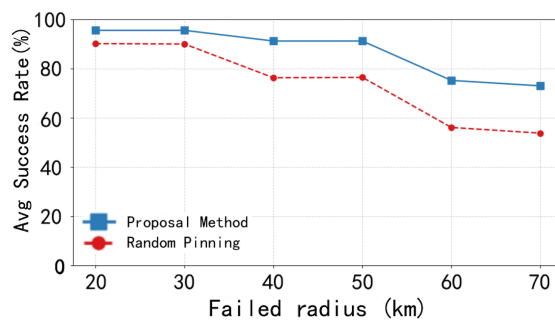


(c) Ratio of data acquisition within 4 hops

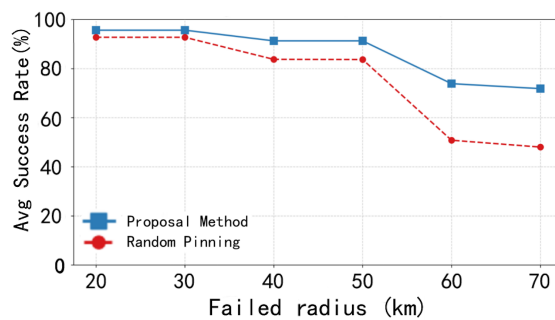
図 2: Results of normal state scenario against Zipf parameter θ

成功率を維持し、災害半径の増加に対しても緩やかな低下にとどまった。ファイル数が 100 や 1000 の場合には、従来方式が急激に成功率を低下させる局面でも、提案方式は 1.5 倍近い成功率を確保し、安定性が確認された。これらの結果は、提案方式がノード障害や広域災害に直面した際でも高い可用性を維持できることを裏付けている。特に、従来方式がファイル数の増加に伴い急速に性能を劣化させるのに対し、提案方式は規模に依存せず一貫した強靭性を発揮する点が特徴的である。

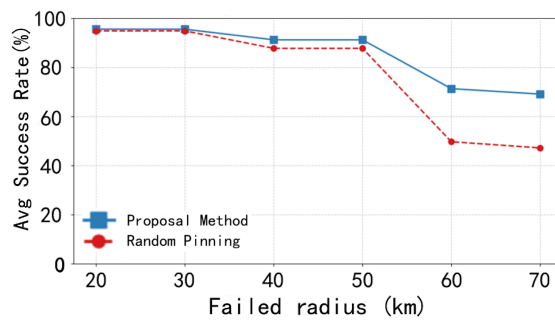
さらに実際のトレースデータに基づく評価に加えて、低人気ファイルのみを対象とした評価を行う。具体的には、すべてのファイルが低人気であり、コピーを保持するノードがネットワーク全体の 6.25% 未満となる条件下で、両方式の復元成功率を、やはり障害半径に対して図 4 (b) に示す。比較のため、す



(a) 10 files



(b) 100 files



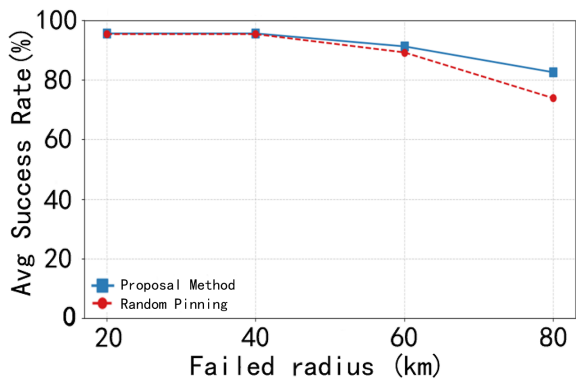
(c) 1000 files

図 3: Recover ratio against failed radius with different file count

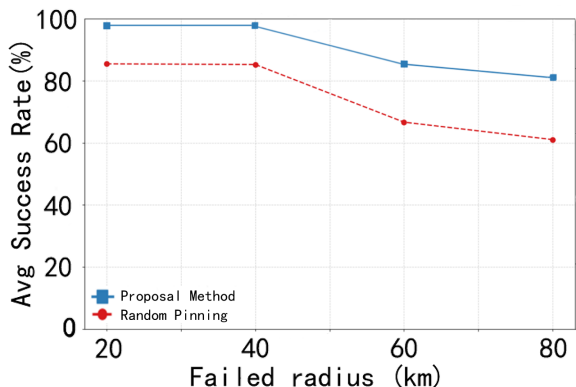
すべてのファイルが高人気（少なくとも 12.5% のノードがコピーを保持）である場合の結果を図 4 (a) に示す。提案方式は高人気ファイルに対しては従来方式と同等の性能を維持しつつ、低人気ファイルにおいては復元成功率が高人気ファイルの場合を上回ることが確認された。すなわち提案手法の復元成功率は、全体的にファイルの人気度が低いほど、従来方式に対して優位となる。

5. まとめ

本研究では、分散型ストレージシステムである IPFS が抱える「データ保持の自律性に起因する可用性低下」という課題に着目し、レポジトリ方式とピア ID 付与アルゴリズムを用いた可用性向上手法を提案した。従来の IPFS では、各ノードが自由意志でピンングを行うため、同一データが地理的に集中して



(a) Many copies of data at all nodes



(b) Few copies of data at all nodes

図 4: Recover ratio against failed radius when (a)many copies of data exist at all nodes and (b)few copies of data exist at all nodes

保持され、大規模災害や停電時にアクセス不能となるリスクが高い。この問題に対して、既存の Filecoin や IPFS Cluster は一定の改善策を示すものの、計算コストや集中管理依存、スケーラビリティの限界など、根本的な課題を解決できない。

本稿の提案方式は、まず複数事業者が提供するレポジトリを導入し、各レポジトリが自身のピア ID 上位 y ビットとデータハッシュ値上位 y ビットが一致するデータを永続保持する規則を設けた。これにより、パラメタ y を調整することで、可用性とコストのバランスを柔軟に制御できる。さらに、ピア ID 付与アルゴリズムによって地理的に分散した位置に同一ビット列を割り当てることで、災害時における同時被災のリスクを低減し、冗長性を確保できることを確認した。シミュレーション評価では、通常時においては提案方式が従来方式と比較して平均遅延やホップ数の面で同等あるいは優位な結果を示し、災害シナリオにおいては復元成功率が大幅に改善されることを確認した。特に、ファイルサイズが大きい場合や人気度の低いファイルが多い場合において、提案方式は従来方式よりも顕著な効果を発揮した。以上より、本研究の提案手法は、IPFS におけるデータ可用性を効率的かつ柔軟に向上させる有力なアプローチであると結論づけられる。今後は、より大規模なネットワーク環境や動的参加・離脱を含む現実的シナリオへの適用、ならびに実装レベルでの性能検証を進めることで、本手法の実運用可

能性をさらに高めていく予定である。

謝 辞

本研究成果は JSPS 科研費 25K03113, 23K28078 の援助を受けたものである。ここに記して謝意を表す。

文 献

- [1] N. and M. Murata, Dispersing Content Over Networks in Information-Centric Networking, IEEE Transactions on Network and Service Management 2019
- [2] B. Juan, IPFS - Content Addressed, Versioned, P2P File System, arXiv 2014
- [3] R. Shi, R. Cheng, Y. Fu, B. Han, Y. Cheng, and S. Chen, Centralization in the Decentralized Web: Challenges and Opportunities in IPFS Data Management, Proceedings of the ACM 2025
- [4] Stoica, I. Morris, R. Liben-Nowell, D. Karger, D.R. Kaashoek, M.F. Dabek, F. Balakrishnan, H., Chord: a scalable peer-to-peer lookup protocol for Internet applications, IEEE/ACM Transactions on Networking 2003
- [5] Maymounkov, P., Mazières, D. Kademlia: A Peer-to-peer Information System Based on the XOR Metric. IPTPS 2002
- [6] Protocol Labs, IPFS Cluster Documentation. <https://cluster.ipfs.io/>
- [7] Zhihao Li, Dave Levin, Neil Spring, and Bobby Bhattacharjee, Internet anycast: performance, problems, & potential, ACM SIGCOMM 2018
- [8] A. B. Downey, Evidence for Long-Tailed Distributions in the Internet, SIGCOMM Internet Measurement Workshop 2001
- [9] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, Web Caching and Zipf-like Distributions: Evidence and Implications, IEEE INFOCOM 1999
- [10] <https://texwiki.texjp.org/>