レポジトリを用いたIPFS可用性向上技術

立命館大学
ワン　シュ，上山憲昭

研究背景

- IPFS
  - ファイル保存や共有システム
  - すべてのノードが同一のデータを重複して保持しない，各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で決まる
  - 常に個数以上のノードに保持させることを保証できない
- 従来手法：
  - ピニング（Pinning）：特定のデータをピニングし，自身がそのデータを永続的に保存する
  - Filecoin：インセンティブを与え，ノードに長期間、ノードにデータを保持させることが可能
  - クラスタ（Cluster）： 複数のIPFSノードを管理・調整する方法．データの分散保存や複製の作成を行い，耐障害性や可用性の向上

研究目的

- 障害や地震等の大規模災害に対する耐障害性の向上することが重要
- authority やデータ所有者が明示的に、データを永続的に保持するノードの数や、それらの位置を制御
- 効率的な IPFS のデータ配布方法
  - レポジトリを用いたIPFS可用性向上技術
  - NID付与アルゴリズム

提案手法(レポジトリを用いたIPFS可用性向上技術)

- レポジトリは特定のデータを永続的にピニング

- y :各レポジトリは自身のバイナリピアIDビット数
- k :上位kビットとデータのハッシュ上位kビットが一致するデータを保持(人気度)
- S_k : 上位kビットのノード集合に定義

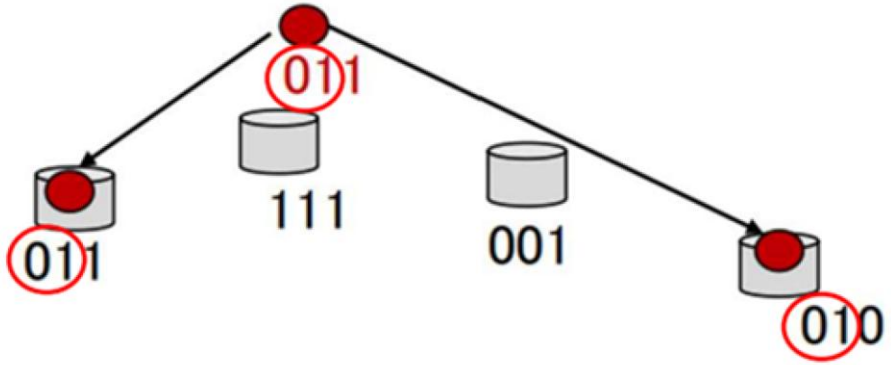


図1. y = 3, k = 2, S_k = (0,1)

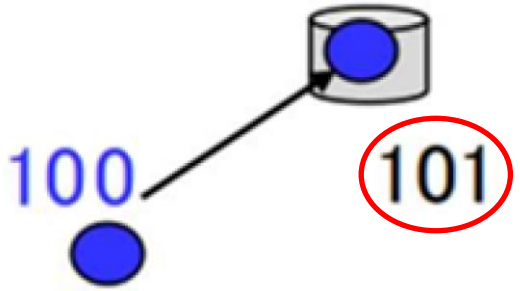


図2. y = 3, k = 3 S_k = (1,0,1)

提案手法(レポジトリを用いたIPFS可用性向上技術)

- 制御パラメータ k により、同一データを保持するレポジトリ数を調整可能
  - 小さい k → 高可用性(多重保持)
  - 大きい k → 低コスト(少数保持)

- 小さな制御負荷で柔軟な可用性制御が可能

提案手法(NID付与アルゴリズム)

- T_n がノードnから$S_{k+1}(S_k, 0)$と$S_{k+1}(S_k, 1)$へのホップ距離の総合と定義
- 貪欲法でT_nが最小化の第 k+1 ビットを割当てるアルゴリズムを確立

- Authorityありの場合
  - 集中的にID割当 → 最短ホップ距離を最小化するアルゴリズム
  - 空間的に離れたレポジトリに同一ビットを配置
- Authorityなしの場合
  - 近隣レポジトリの座標を取得
  - 空間的に最も離れたビット値を自律的に設定

- データの冗長化と空間的分散を両立

- 大規模障害への耐性を強化

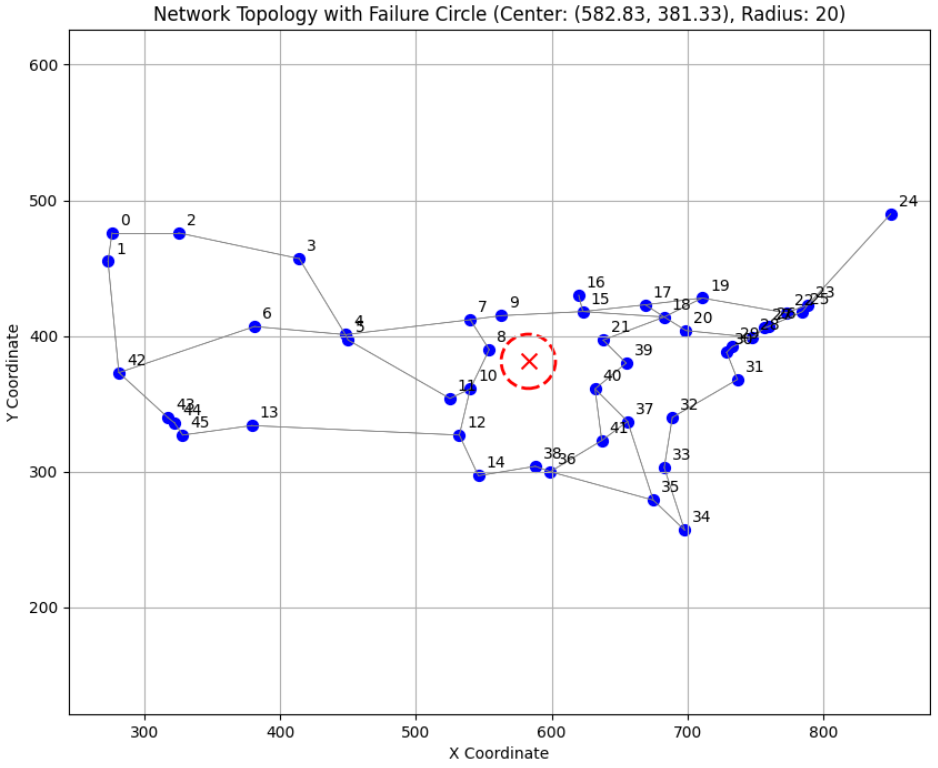
性能評価条件

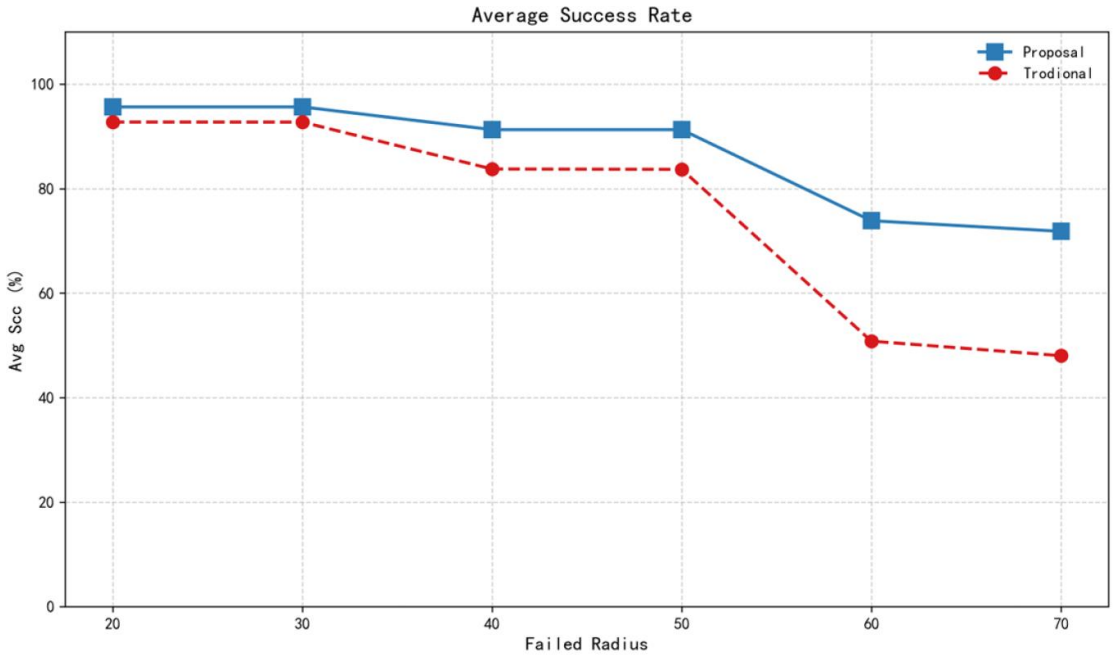
- 災害シミュレーション
  - 固定点を中心とした半径 R の損害範囲を設定
  - 範囲内のノードは停止 ⇒ データ取得不能
  - 半径を段階的に拡大し，復元成功率を測定

- 一般シミュレーション
  - 障害なしの状態で両方式を比較
  - 10,000回のデータ取得要求を実行
  - ファイルサイズ：lognormal 分布
  - 人気度：Zipf 分布で，異なる θ 値（0.6~0.9）でデータセットを生成
  - 評価指標
    - 平均復元遅延
    - 平均ホップ数
    - 4ホップ以内の復元成功率

性能評価

- 米国商用ISP@homeネットワークトポロジー
- 災害シミュレーション結果
  - y = 5
  - k ≥ 3 の条件下で提案方式は高い成功率を維持

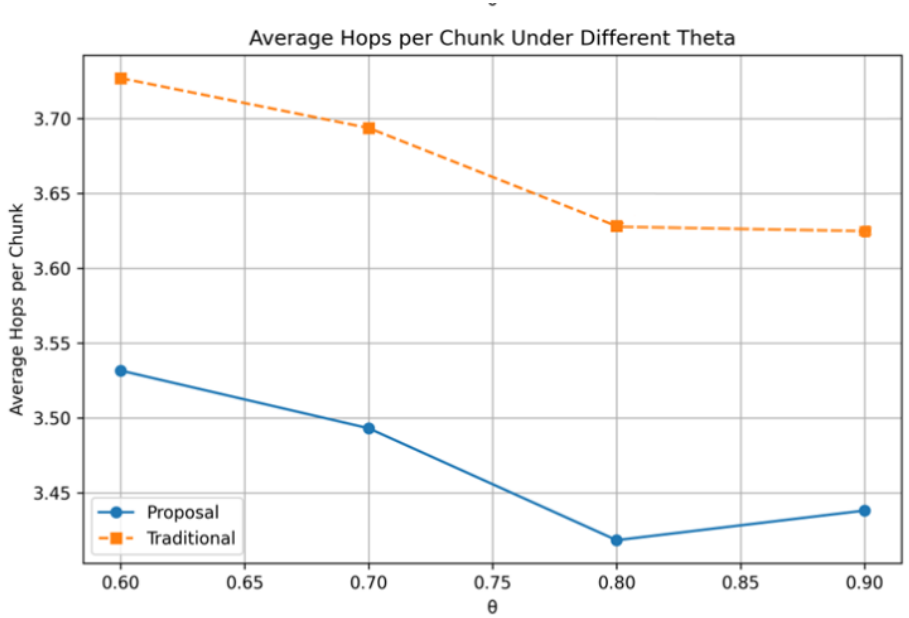


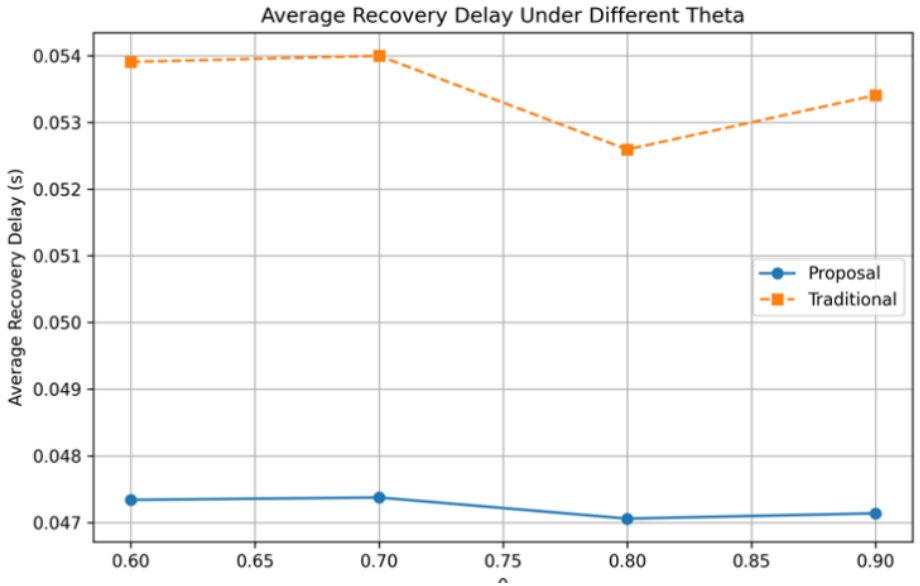


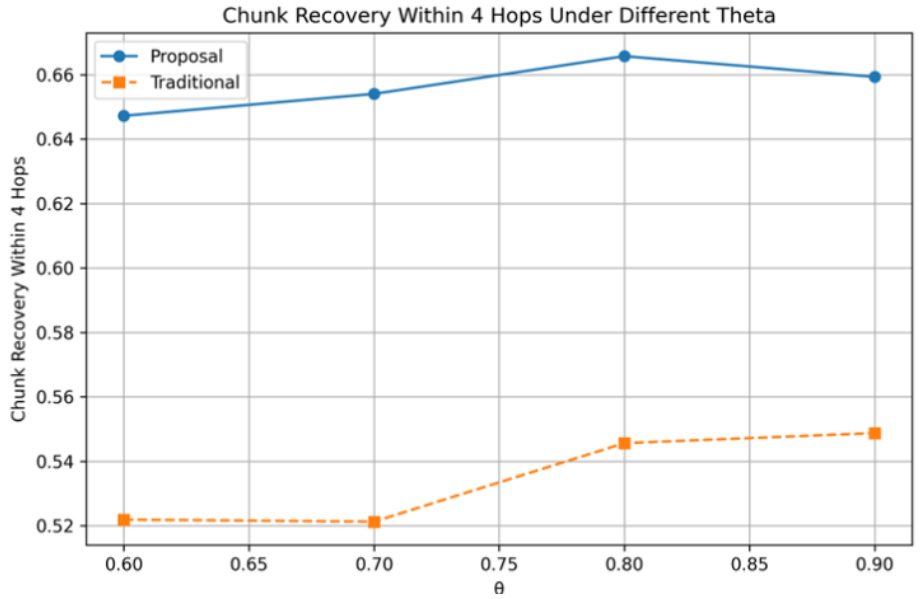
性能評価

- 一般シミュレーション結果

- 異なるθ値で:
  - 遅延 10～25% 減少
  - 平均ホップ数 10%～25% 減少
  - 4ホップ以内成功率 10～20% 向上







まとめ

- 提案方式：レポジトリ導入＋NID付与アルゴリズム
- 大規模障害下でも高い復元成功率を維持
- 提案手法は従来手法に比べて，災害時・平常時の両方で有効性を確認
  - 特に低人気度ファイルに対しての効果がいい
- 今後の方針：
  - ネットワークトポロジーのサイズを増大