小口研究室　研究紹介　(2018年度)
(お茶の水女子大学理学部情報科学科)

災害時に有用なインターネット非依存ローカルシステムの提案と実装
（研究担当：田中　有彩）

研究背景

- 近年、日本では多くの自然災害が発生
  - → インターネットが切れてしまう恐れ
  - → 各避難所状況を災害対策本部が把握したい
- 現場のニーズ
  - → サブネットが集まった時に、繋げたい

大規模災害時に有用な情報通信システムの構築を目指す

目的

- 想定ネットワーク：それぞれ異なるサブネットの発生
  - ⇨ ルータをローカルに動かし“NAT越え”を行い、事前設定等の必要なく、システムの実現を目指す

2つのシステムを提案

① 避難者同士のチャットシステム

◆ 通信基盤
- Wi-Fi AP 兼 NATルータ (以降 エッジサーバ) を使用
- 基地局や基幹ネットワークなどの影響を受けないプライベートネットワークをエッジサーバごとに構築
  - ⇨ プライベートネットワークを通して通信開始
- NAT越え技術を使用
  - ⇨ エッジサーバにSTUN/TURN・シグナリングサーバを搭載

◆ 端末上の情報共有アプリケーション

※使用中の図

② 避難所状況共有システム

◆ 避難所状況可視化アプリ SheRepo2(株式会社FIXER)
- 災害発生時の避難所の状況を避難者・救護者・避難所管理者が共有できるアプリケーション
- Microsoftのクラウドサービスであるを使用
- 主な機能　①避難所からの報告機能　②報告の表示機能

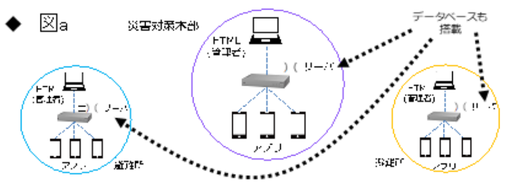
◆ 既存の取組の問題点：**クラウドサービスを使用**

◆ 提案策(図a)
- エッジサーバとしてDBを搭載したWi-Fi機能付きAPを各避難所に設置
- 各避難所において、APに繋がった各端末がエッジサーバへ災害情報を送信
- エッジサーバのDB間が自動的に同期
  - → ローカル環境のサーバとして機能させ、エッジコンピューティングを構成

災害対策本部がすべての各避難所状況を把握することが可能に

◆ システム概要
- 入力側(図b)：各避難所内にいるユーザが避難所の状況を報告
- 管理側(図c)：各避難所の状況を集計
  - 地図上で避難所と状況を把握できるようにする

◆ 図a



◆ 図b

◆ 図c

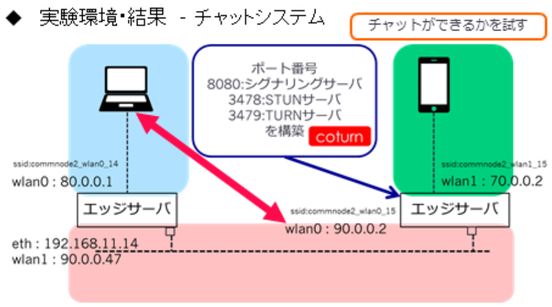
実験

提案したシステムの有用性の確認の為、
それぞれ実験を行った

◆ 使用するエッジサーバ
- 実システムの構築とシステム解析・連携をサポートする汎用的なプラットフォーム
- ネットワークを自律的に構築
- DTN技術より実システム間での情報同期を可能とする
  - → 避難所状況共有システム
- シグナリング・STUN/TURNを搭載
  - → チャットシステム

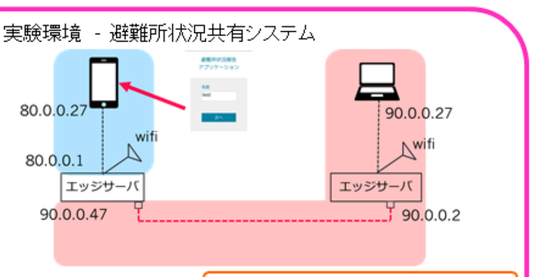
◆ 実験環境・結果 - チャットシステム

チャットができるかを試す



図に示す、矢印間のTCPによる通信を確認

◆ 実験環境 - 避難所状況共有システム



エッジサーバ間でデータの同期ができるかを試す

◆ 実験結果 - 避難所状況共有システム

エッジサーバ間のデータ共有を確認

まとめ・今後の課題

◆まとめ	NAT越えによる避難者同士のチャットシステム・DTN技術とエッジコンピューティングによる避難所状況共有システムの考案 それぞれのシステムを用いた、実機による実験
◆今後の課題	STUN/TURNが使用されない原因の究明、親玉NATサーバによる管理

完全準同型暗号を用いた頻出パターンマイニング委託システムの分散処理等による高速化手法の実装と評価
（研究担当：山本　百合）

研究背景

◆ 頻出パターンマイニング委託システム

大型の計算リソースを所有する外部機関にデータを委託し，頻出パターンマイニングを外部機関が代理で計算するシステム

◆ 個人的なデータの保護の必要性

購買履歴や生活中の動作ログに対して、暗号化によるデータのプライバシー保護が必要

提案手法（頻出パターンマイニング）

◆ 完全準同型暗号の利用

加法と乗法で準同型性が成立する完全準同型暗号を利用し、データを暗号化した状態で頻出パターンマイニングを行う

2 ⊗ 3 ⇨ 6　　5 ⊕ 3 ⇨ 8

完全準同型暗号は**計算量が大きい**ため、サーバ側の計算時間が長い ⇨ **サーバ側の処理の高速化を目指す**

◆ マスタ・ワーカ型分散処理

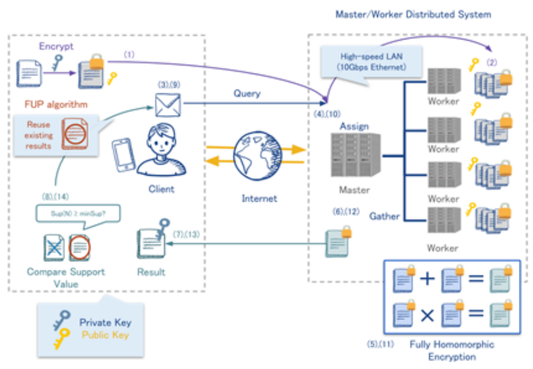
分散処理によってサーバサイドの計算を高速化
マスタとワーカに役割を分担し，インターネットを経由する通信回数を抑える

◆ Aprioriアルゴリズム

「頻出ではないアイテムセットを含むアイテムセットは頻出ではない」を規則とすることで高速に頻出パターンマイニングを行うAprioriアルゴリズムを適用

◆ FUPアルゴリズム

Aprioriアルゴリズムは、データベース更新時に再計算が必要
データベース更新前の結果を再利用することにより，
候補アイテムセットを削減し、頻出パターンマイニング計算の高速化を行うFUPアルゴリズムを適用



◆ 実装方法

C++で実装、マスタとワーカの制御にOpen MPIを使用
完全準同型暗号計算にHElibライブラリを使用
お茶の水女子大学内クラスタにサーバプログラムを、
AWS EC2にクライアントプログラムを設置し、
SSHポートフォワーディングで通信を行う

提案手法（動作ログデータへの応用）

◆ 生活中動作ログデータへの応用

活動量計で取得した人間の生活中の動作ログデータ解析では、プライバシー保護の観点からセキュリティ管理が求められる
⇨ 完全準同型暗号を用いたデータマイニングシステムが有用

◆ 頻出な生活中動作パターンの発見

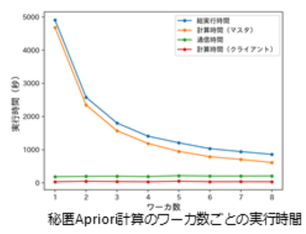
ある患者の期間ごとの頻出な動作パターンをAprioriで抽出し、最終的に期間ごとの生活状態の分類を行う
⇨ データの前処理を行い、提案手法のシステムに適用

実験と考察

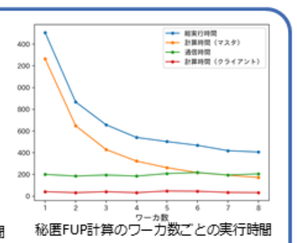
◆ 秘匿Apriori計算の実験
- 分散台数が増えるにつれて計算時間が短縮
- 実行時間は徐々に横ばいになる
- 通信時間とクライアント側の復号計算はワーカ数による変化は小さい

◆ 秘匿FUP計算の実験
- 分散台数が増えるにつれて計算時間が短縮
- Apriori計算時と比較して、計算時間が短縮

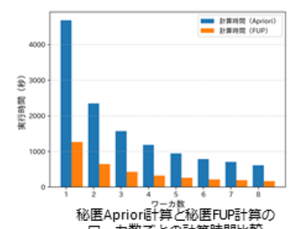
◆ Amdahlの法則による考察
- 高速化率＝逐次実行時間/並列実行時間
- 高速化率 $\leqq \dfrac{1}{(1\text{-並列実行時間の割合}) + \dfrac{\text{並列実行時間の割合}}{\text{ワーカ数}}}$
- ワーカ数が最大の時の秘匿FUP計算の高速化率は、分散化前の6.7倍程度



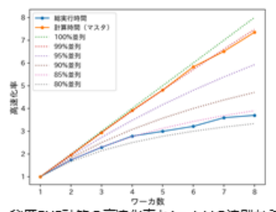
秘匿Apriori計算のワーカ数ごとの実行時間



秘匿FUP計算のワーカ数ごとの実行時間



秘匿Apriori計算と秘匿FUP計算のワーカ数ごとの計算時間比較



秘匿FUP計算の高速化率とAmdahlの法則から導かれる理想的な高速化率の比較

今後の課題

- 生活中動作ログのデータマイニング計算への応用（データの前処理や統計手法の工夫）
- クライアント側のプログラムの軽量化

大規模災害時のインターネット非接続環境における情報共有システムの提案及び仮想環境下の実装
（研究担当：Yu Hui）

研究背景

- インターネットを介した情報伝達手段の発展
- 災害時の情報共有手段が不可欠
  - 被災者に対し：
    - 災後の生活に便利性を提供
    - 家族の安否確認が来た場合、安心させる
  - 支援者に対し：
    - 被災程度を把握させる
    - 災害の関連情報を効率よく拡散させる

研究目的

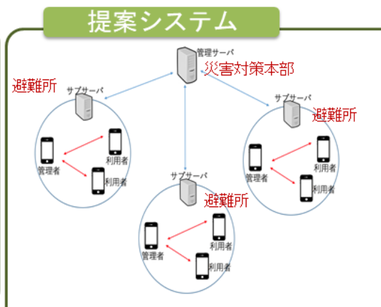
災害時に:

インフラが使えるが、サービスが提供できないローカル環境が存在可能

ローカル環境でも使える情報共有システムを構築

- インターネットに依存しすぎない
- 情報支援サービスを提供できる
- 災害時の劣悪な環境に対応できる接続と情報伝送の対策

提案システム



— システム機能 —

- **ユーザ**：管理者と利用者の２種類を想定
  - それぞれ支援者と被災者に対応する
  - 情報の支援とサービスは端末機で提供させる
- **サーバ**：管理サーバとサブサーバの２種類を想定それぞれ災害対策本部と避難所に設置
  - サーバを分散し、分散化システムを構成
    - サブサーバ（ワーカ）の役割：
    - 通信可能な範囲で被災者と支援者に情
    - 収集手段を提供
    - 管理サーバ（マネージャ）の役割：
    - サブサーバの追加と削除を管理
    - 共有情報を管理し、サブサーバに情報を共有

— 接続認証手法 —

インフラが機能するネットワーク環境を利用

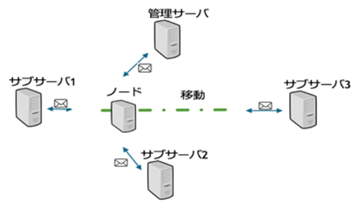
- 目標：
  - I. 接続セキュリティが守れる
  - II. 支援情報を共有
- 概要：
  - I. 管理サーバは認証したサブサーバを「Server-List」に追加し、ピアサーバ同士に知らせる（「Server-List」に追加したサーバはピアサーバである）
  - II. 管理サーバの稼働している場合：
    - I. システムの接続認証を管理（ピアサーバの間での直接接続、サブサーバの追加・削除）
    - II. 情報を共有させる
  - III. 管理サーバの遮断された場合：
    - I. 「Server-List」によって、ピアサーバの間で直接接続を行う
    - II. 追加、削除ができなく、情報共有ができる

— DTNを用いた情報共有手法 —

インフラが使えなく、通信が途絶えた

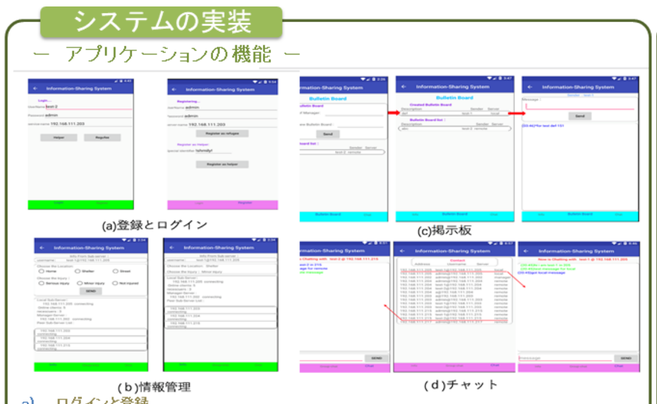
- 概要：

ノードサーバは移動と蓄積機能付きの端末機として、
サーバ同士の間で巡回し、情報を運搬・伝送・共有させる



システムの実装

— アプリケーションの機能 —



(a)登録とログイン　(c)掲示板

(b)情報管理　(d)チャット

a) ログインと登録
- I. ユーザは管理者（支援者）と利用者（被災者）になる
- II. 管理者として登録する場合に、識別子が必要

b) 情報管理
- I. 登録したサブサーバの情報
- II. 被災状況を報告する仕組み
- III. 通信状況を表示

c) 掲示板
- I. 一対多の配信
- II. 重要な情報を掲示板で載せ、利用者は自分自身の状況によって、情報集を選ぶ
- III. 一つの掲示板の発信者が唯一（1人の管理者は１つの掲示板を作り、情報を発信）
- IV. システム内のユーザに情報を公開

d) チャット
- I. 一対一の通信
- II. システム内の全ての管理者は連絡先リストに自動的に追加
- III. 同じサブサーバの管理者であっても、異なるサブサーバの管理者であっても、チャットができる

システムの実装

— 接続認証手法 —

仮想空間の実験環境

— DTNを用いた情報共有手法 —

実験結果

I. アップロード時間：（Upload）
ノードが送信準備のリクエストを受け取った時から、ファイルがノードサーバに格納されるまでの時間

II. ダウンロード時間：（Download）
目的地のサブサーバが送信準備のリクエストを受け取った時から、ファイルが格納されるまでの時間

III. 伝送時間：（Total）

まとめと今後の課題

— まとめ —

I. 災害時の大規模情報配信の需要に対応するため、システム機能を提案・作成
- I. アプリケーション機能を提案
- II. 接続認証手法を提案
- III. DTNを用いた情報共有手法を提案

II. 仮想環境下で、実装実験を行なった

— 今後の課題 —

I. 現実的な環境で、評価実験を行う

http://ogl.is.ocha.ac.jp/

oguchi@is.ocha.ac.jp