

小口研究室 研究紹介 (2022年度)

(お茶の水女子大学理学部情報科学科)

都営バスのオープンデータによる渋滞検知の精度向上のための運行特性を考慮した手法の提案 (研究担当: 畠中 希)

研究背景

- 渋滞による被害**
- ◆ 経済的損失約12兆円
 - ◆ 停車車両の排気ガスによる大気汚染
 - ◆ 救急車等の緊急車両の到着遅延
- 渋滞情報の問題点**
- ◆ 道路上の感知器等で収集しているため、感知器が無い道路では渋滞検知不可
 - ◆ 停車している車を渋滞と誤判定する可能性あり

先行研究

都営バスのリアルタイム運行データを用いた渋滞検知※ 都営バスは感知器のない道路も走行

- 手法**
- ◆ 機械学習を用いて停留所区間ごとに「渋滞」「非渋滞」の二値分類を行う

結果

- ◆ 全停留所に対して1つの学習器の場合：F1スコア 0.306
- ◆ 停留所区間ごとに個別の学習器の場合：平均F1スコア 0.399、特定の区間のF1スコア 0.675
- ◆ 個別の学習器かつ特定の時間に複数のバスが走行する場合：特定の区間で 0.742

※ 青柳宏紀, 岡田一洗, 山名早人. 都営バスのリアルタイム運行データを用いた渋滞検知. DEIM2022 第14回データ工学と情報マネジメントに関するフォーラム. 2022, pp.1-8

提案手法

先行研究の精度向上を目指す

◆ 特徴量

特徴量	説明
V_{ij}	バス <i>b_i</i> の停留所区間 <i>s</i> 走行時の速度
$V_{(i-1)j}$	1つ前のバス <i>b_{i-1}</i> の同一停留所区間 <i>s</i> 走行時の速度
C_{ij}	バス <i>b_i</i> の区間 <i>s</i> 走行時間の時間帯
<i>isEarly</i>	定刻より10秒以上早く出発したか否かを示す指標

- ◆ 運行特性を特徴量に追加
- バスは時刻表通りに運転する傾向あり
- 早く出発した場合、次の停留所に向かう際にわざと遅れて運転する可能性がある
- 特徴量に10秒以上早く出発したか否かというbool値を追加

評価実験

◆ 実験

日時：2022年12月1日～12月29日 (平日ダイヤのみ)
対象路線：池86系統の「池袋駅東口」～「渋谷駅東口」～「南池袋三丁目」

◆ 実験結果

ベースライン手法 (isEaryなし)

アルゴリズム	accuracy	precision	recall	F1スコア
ランダムフォレスト	0.869	0.784	0.497	0.594
AdaBoost	0.875	0.834	0.468	0.600
XGBoost	0.844	0.943	0.236	0.377

提案手法 (isEary追加)

アルゴリズム	accuracy	precision	recall	F1スコア
ランダムフォレスト	0.870	0.777	0.496	0.606
AdaBoost	0.874	0.829	0.468	0.598
XGBoost	0.859	0.761	0.432	0.551

F1スコアが 0.594 → 0.606 に向上 (ランダムフォレスト)

まとめと今後の展望

まとめ

- ◆ 停留所区間ごとに「渋滞」「非渋滞」の二値分類を行った
- ◆ 定刻より10秒以上早く出発したか否かという特徴量を追加
- ◆ F1スコアが 0.594 → 0.606 に向上

今後の展望

- ◆ 定刻より遅く出発したという情報も特徴量に追加

量子シミュレータのホットスポット分析と考察 (研究担当: 青木 望美)

研究背景

近年注目されている量子コンピュータは、一部の問題について従来型コンピュータと比較して非常に高速な計算を実現可能であることから盛んに研究・開発が行われている。しかし、理想的な挙動をする量子コンピュータは未だに存在していない。ここで量子シミュレータというものが存在する。量子シミュレータは従来型コンピュータ上で量子コンピュータの挙動を表現するツールであり、量子コンピュータ上で動作する量子アプリ等の開発に有用である。本研究では複数の量子シミュレータの分析を行い、性能特性の比較や高速化に向けた検討を行う。

実験概要

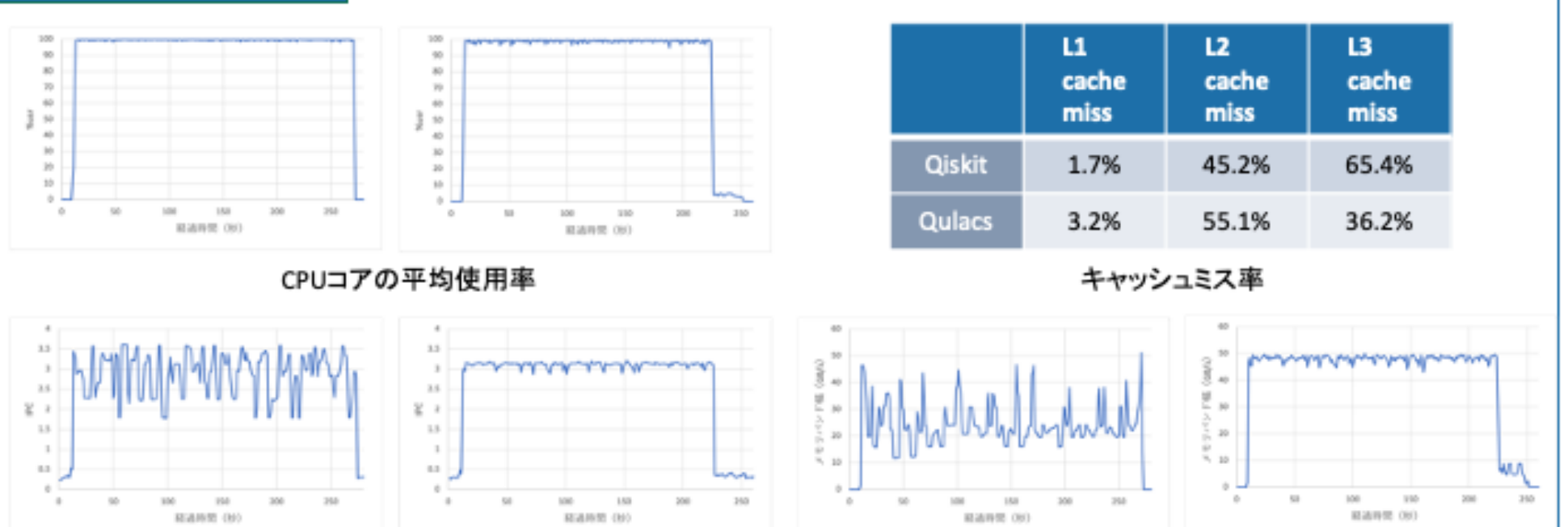
分析対象の量子シミュレータにQiskit AerとQulacsを用いる。また、用いる量子シミュレータはState Vector方式のものであり、State Vector方式のシミュレータは結果の測定前の量子状態を取得することが可能なことから、量子アプリ等開発時のデバッグ作業に有用である。分析対象の量子回路は、ランダム回路である量子体積モデル回路を使用する。また、量子ビット数は31個、深さは10とする。

この量子回路を実行し、複数の性能分析ツールを使用して性能データを収集する。

実験サーバ	Primergy RX2540 M1	sysstat	11.7.3
OS	Rocky Linux 8.6	vmstat	procs-ng 3.3.15
CPU	Intel Xeon プロセッサ E5-2697v3(2.60GHz)	perf	4.18.0-425.3.1.el8.x86_64
メモリ	128GB	pcm	202107-5.el8

実験結果

(左: Qiskit 右: Qulacs)



- ・どちらも命令実行効率が良いが、グラフの概形は異なる
- ・メモリバンド幅は特に差が見られた

- ・キャッシュミス率は、L1・L2キャッシュとL3キャッシュとで値の大小の違いが見られた

まとめ

分析結果などから、メモリバンド幅を引き出すことで性能の向上を見込むことができると考えられる。今後はさらに詳細な分析を行うとともに、高速化の具体案の検討を行う。

圧縮センシングによる準同型暗号文の通信量削減 (研究担当: 泉 湖雪)

研究背景

- クラウドサービスへの外部からの攻撃によるデータの漏洩に備える
- 個人情報を保護しながらデータ分析をしたい

→ 準同型暗号の使用が適している

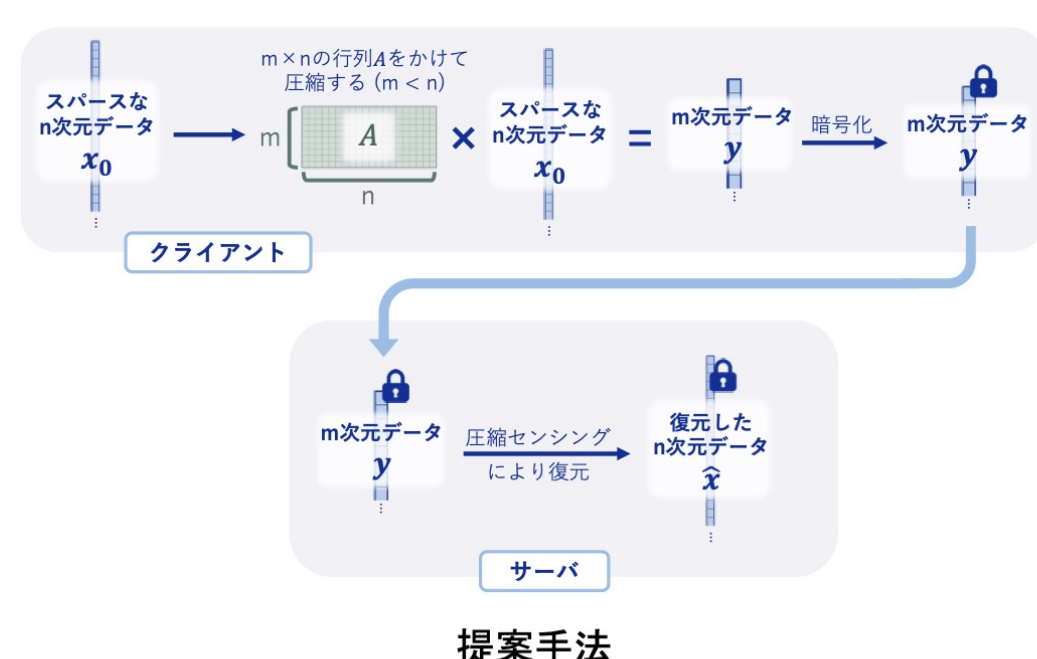


スパースな多次元データのサーバ送信時における通信量を削減する

- 準同型暗号
暗号文同士で加算や乗算ができる暗号
暗号文サイズが大きいため通信量が多い
→ 多次元データに対して圧縮センシングを使用することでクライアント・サーバ間の通信量を小さくする
- 圧縮センシング
スパース性(ゼロ成分が多いという性質)を持つデータを圧縮してから復元する方法

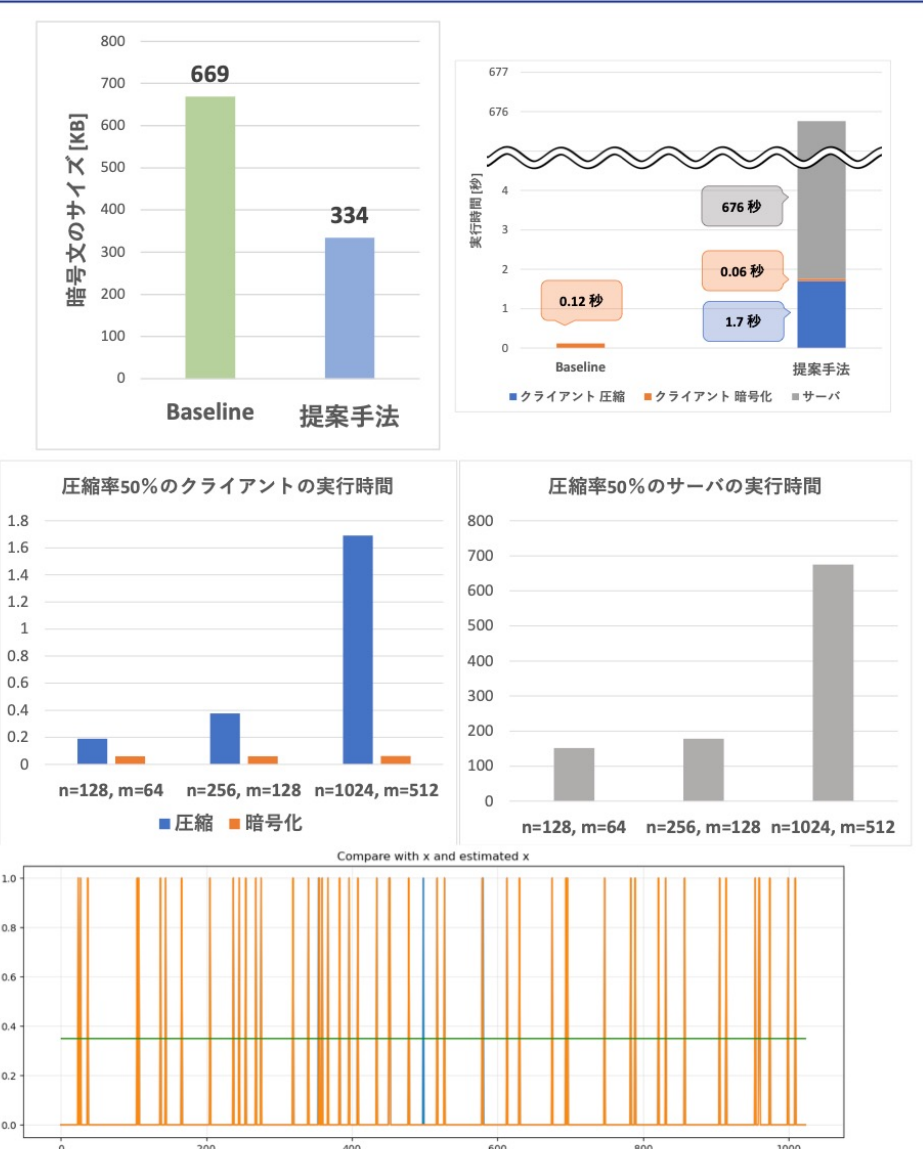
実験概要

- **Baseline**
スパースな多次元データをクライアントで準同型暗号化して、圧縮せずにサーバに送信する
- **提案手法**
圧縮センシングを利用して通信量を削減してスパースな多次元データを送信する
- **実験環境**
クライアント: Raspberry Pi
サーバ: Ubuntuのマシン
- **再構成アルゴリズム**
 $encrypt(\hat{x}) = (A^T A)^{-1} A^T \times encrypt(y)$
一般逆行列と暗号化された低次元行列との積を求めることで復元する



実験結果

- 暗号文のサイズ
提案手法ではBaselineに比べて約半分の通信量に抑えられており、通信量の削減率はnとmの比率に対応している
- 実行時間
暗号文のサイズの減少に伴い暗号化にかかる時間は減っているが、クライアントで圧縮させる際の行列計算や、特にサーバで復元する際の再構成アルゴリズムの計算に時間がかかっている
n, mの値が小さいほどクライアントでの圧縮時間やサーバでの復元時間が短くなっている
- 精度
圧縮率50% (n = 1024, m = 512)としたときのRMSEはおおよそ0.15となった
閾値により2値化すると1である部分は91%一致



今後の課題

- 実行時間の改善や他の再構成アルゴリズムの検討
- スパースな多次元データの実数への対応