

小口研究室 研究紹介 (2024年度)

(お茶の水女子大学理学部情報科学科)

量子回路シミュレータの性能特性比較と量子回路実行時間推定モデルの検証 (研究担当:青木 望美)

研究背景と目的

量子回路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、性能特性の比較を行うとともに、量子回路実行時間の推定を可能にするモデルを提案し、その検証を行う。

実験概要

3. 実験概要

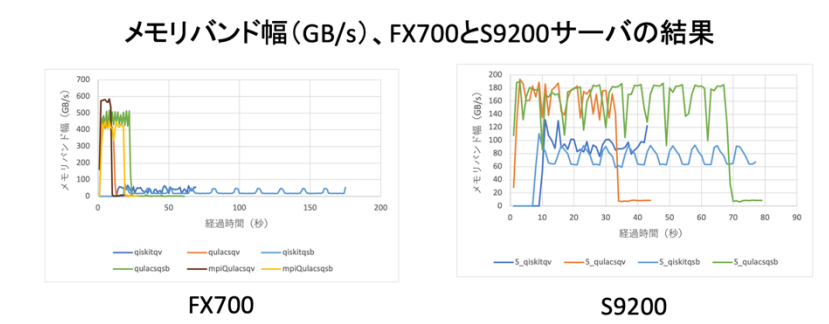
量子回路シミュレータ: Qiskit Aer, Qulacs, mpiQulacs
性能分析: 量子ビット数: 30、回路の深さ: 10、QV回路とQSB回路対象
モデル化実験量子ビット数: 1~29、回路の最大深さ: 10 (FX700サーバでのみ実施)、1量子ゲート対象

サーバ	Primergy RX2540 M1	サーバ	FX700	サーバ	Intel S9200WK
OS	Rocky Linux 8.6	OS	Rocky Linux 8.5	OS	Rocky Linux 8.6
CPU	Intel Xeon プロセッサー E5-2697v3(2.60GHz)	CPU	Fujitsu A64FX (2.0GHz)	CPU	Intel Xeon Platinum 9242(2.30 GHz)
CPUコア数	14	CPUコア数	48	CPUコア数	48
L1d/L2/L3 キャッシュ	32 KB(core)/256 KB(core)/35 MB	L1d/L2 キャッシュ	64KB(core)/8MBx4(CMG)	L1d/L2/L3 キャッシュ	32 KB(core)/1024 KB(core)/36 MB
メモリ容量	128 GB	メモリ容量	HBM2 32 GB	メモリ容量	384 GB

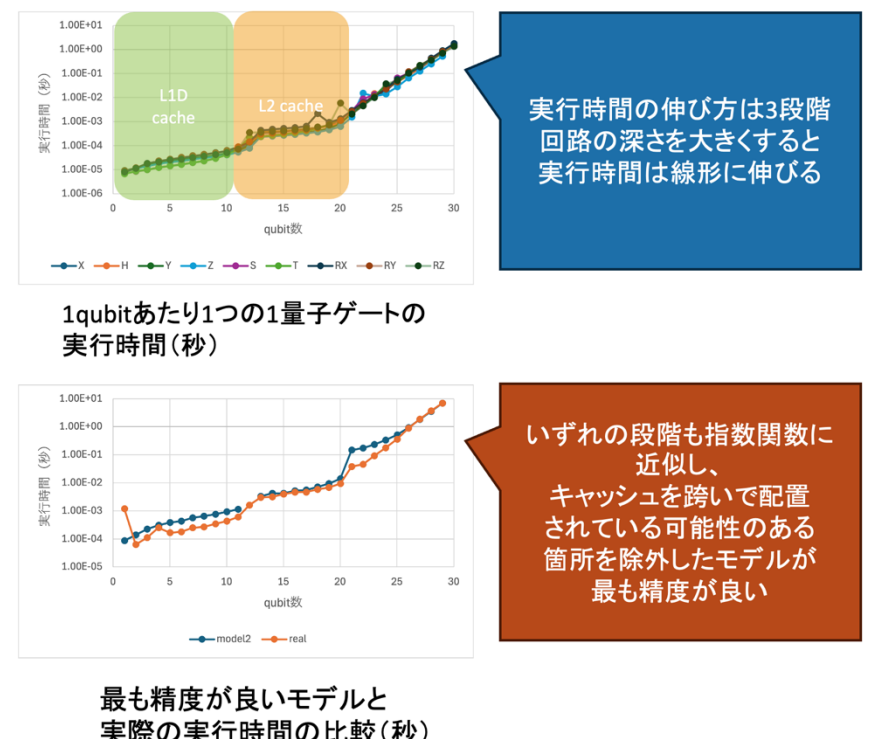
実験結果

性能特性比較

量子回路実行時間（秒）							
QV回路	RX2540	FX700	S9200	QSB回路	RX2540	FX700	S9200
Qiskit	129.1	77.5	34.0	Qiskit	280.7	223.5	70.3
Qulacs	119.0	49.5	44.1	Qulacs	233.8	57.0	81.6
mpiQulacs	-	19.5	-	mpiQulacs	-	27.4	-



モデル化



まとめ

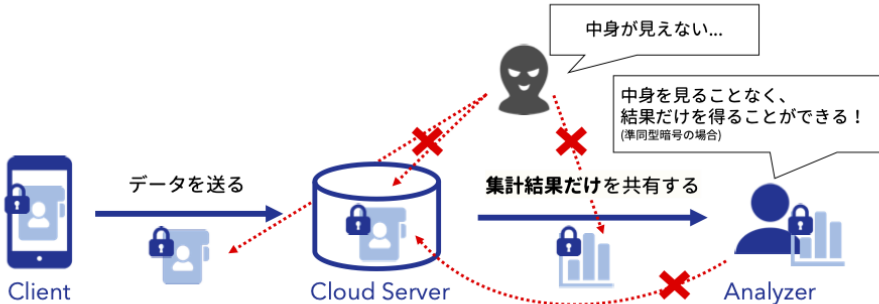
量子回路シミュレータ間での性能の差は実行環境によって異なることがわかった。メモリバンド幅と実行時間には関係性が見られる環境と見られない環境があり、この点のさらなる分析は今後の課題である。量子回路実行時間のモデル化では、指数関数に近似し、キャッシュを跨いでいる可能性のあるqubit数を除外することで精度が良くなることがわかった。
今後は量子回路シミュレータのさらに詳細な分析を行い、モデルは2量子ゲート対応に拡張することで汎用性を高める。

辞書を用いた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



準同型暗号とは？

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

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

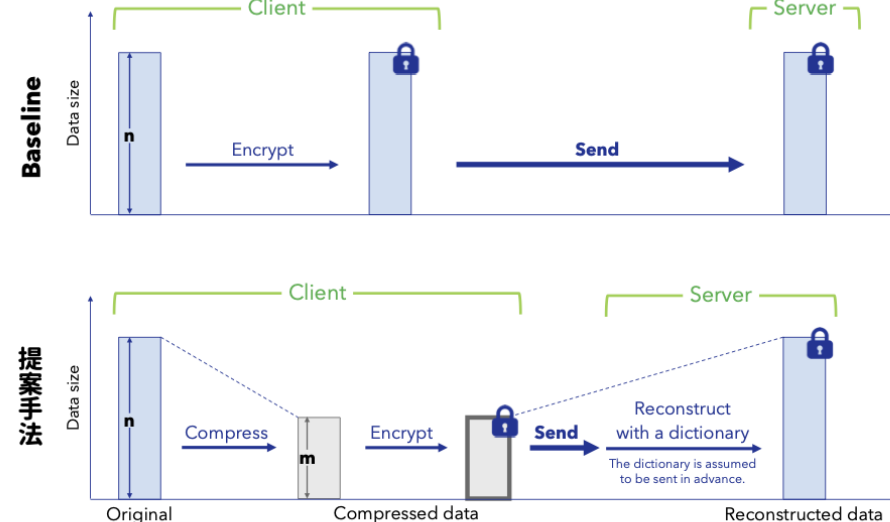
圧縮センシングを用いて
多次元データのサーバ送信時
における通信量を削減する

課題

- 圧縮センシングではデータがスパースではない場合には復元精度は低下する
→ 辞書学習 (最適なスパース変換をデータから学習する) を用いる
- 準同型暗号は比較演算が不得意 & 復元アルゴリズムに含まれる乗算の数が多いほど復元時間が長くなる
→ 復元アルゴリズムには、比較演算を必要としない一般逆行列による L_2 ノルムの最小化を使用する

$$y = Ax \text{ であるとき } x = (A^T A)^{-1} A^T y \text{ である}$$

実験概要

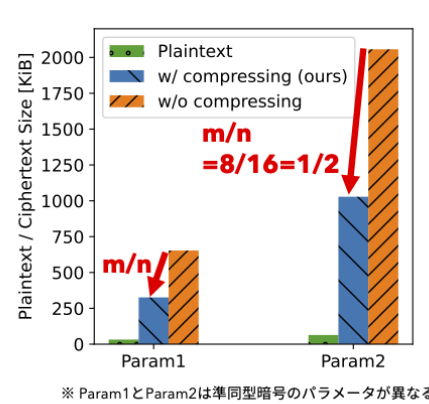


- SEALライブラリを使用してBaselineと提案手法をC++で実装し、データセットを利用して評価
- 評価指標: 通信量・復元精度・実行時間

評価

通信量

Baselineに比べて、提案手法では約半分の通信量に抑えられている。
 n/m の値を大きくすると通信量削減量は増加するが、精度の低下にもつながる。



実行時間

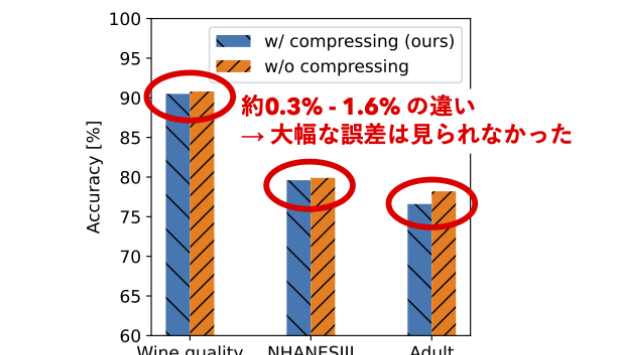
圧縮行列のサイズと実行時間: 復元アルゴリズムでの準同型演算回数は m に比例するため、長いベクトルよりも短いベクトルを複数復元する (n, m を小さくする) ほうが高速であり適している。

圧縮の有無と実行時間: ロジスティック回帰分析、画像生成それぞれの実行時間を以下に示す。
 n, m の大きさを小さくするほど、トータルの平文の長さは同じでも、圧縮や復元に要する時間は短くなる。また、 n, m の値が小さい場合、連結作業の増加の影響で暗号化に要する時間が増加する。

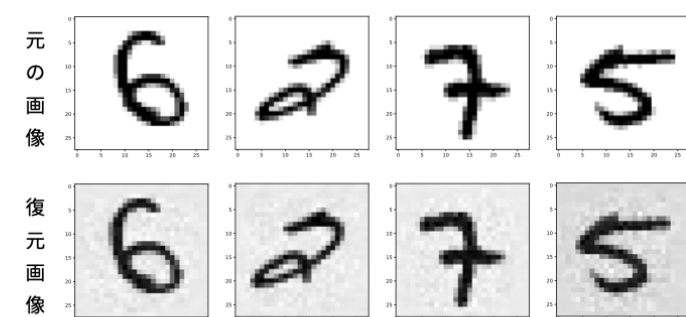
		Client		Server
		Compression	Encryption	Reconstruction
ロジスティック回帰分析 ($n, m = (16, 8)$)	w/ compression (ours)	0.0195 [ms]	0.71 [ms]	1.128 [s]
	w/o compression	—	0.34 [ms]	—
画像生成 ($n, m = (1024, 512)$)	w/ compression (ours)	72.625 [ms]	8 [ms]	94.5 [s]
	w/o compression	—	15.25 [ms]	—

有用性

ロジスティック回帰分析: 提案手法による復元結果とBaselineに対してロジスティック回帰分析を行い、2値分類結果を比較する。提案手法は0.3%~1.6%程度の低下であった。提案手法により復元したデータは予測に用いることが十分に可能であるといえる。



画像生成: 提案手法による復元結果とBaseline (それぞれ画素データ) に対して画像を生成し、誤差をPSNR (値が大きいほど画像の劣化が少なく精度が高い) で評価する。MNIST画像データセットを用いた結果を以下に示す。復元画像のPSNRの平均は約32.2であり、これは一般的に「拡大すると劣化がわかるが、通常の使用では問題ない品質」とされる。



IoT通信のためのJPEG圧縮と物体検出精度の性能評価 (研究担当:伊藤 千紗)

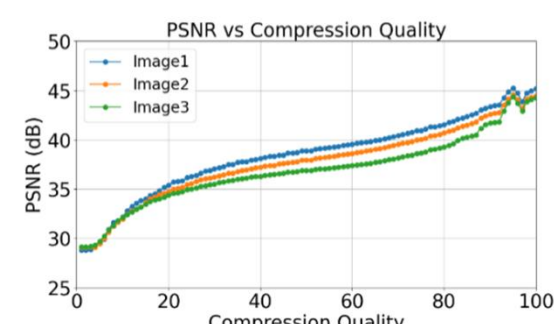
背景/目的



- 課題 | 画像データに対する転送手法の検討が不十分
- データサイズが大きく、可逆圧縮では効果がほぼ見られない
 - 圧縮品質を下げることでデータサイズを小さくできるが、機械学習モデルに与える影響を抑える必要がある
- 画像データに対してJPEG圧縮を適用した通信性能と物体検出精度を調査

実験結果

JPEG圧縮品質を変えた画像ごとのPSNR値の推移



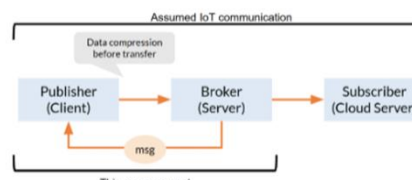
- 使用する画像のPSNR値は25-50dB
- 品質10の画像でも、PSNR値は30以上と比較的品質が良い
- 品質を下げて画質の低下は一般的な範囲

実験概要

使用データ



構成図

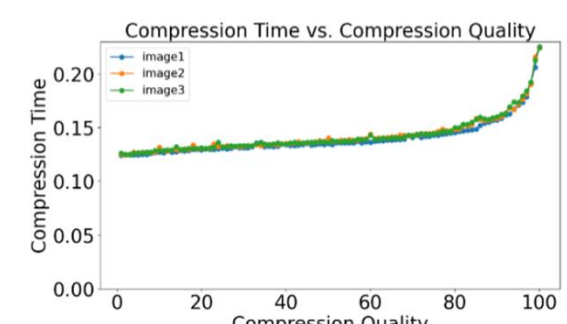


※JPEG圧縮品質は1-100を指定

実験環境

Client: Raspberry Pi 4	
Model	Raspberry Pi 4 Computer Model B 4GB RAM
OS	Ubuntu 20.04 LTS
CPU	ARMv7 Processor rev 3 (v7l), 4.237, 2065 MHz
Memory	4GB
Virtual CPU Cores	16
Virtual Memory	24.19GB
Virtual Disk Size	100GB
Server: mda VM環境	
OS	Ubuntu 20.04 LTS
Virtual CPU Cores	16
Virtual Memory	24.19GB
Virtual Disk Size	100GB
Software	
SNPE (TensorFlow)	v1.8.1
Monogito	v1.8.9

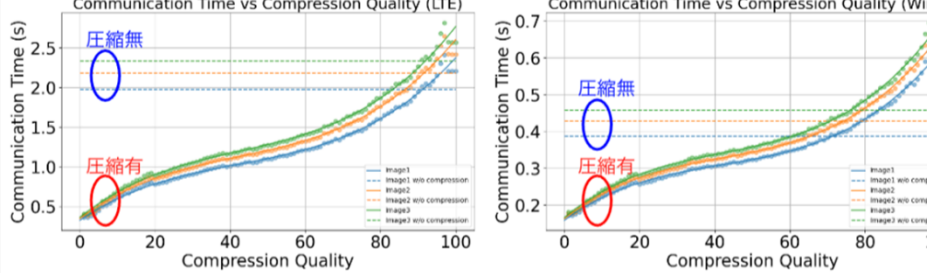
JPEG圧縮品質を変えて圧縮時間を比較



- 圧縮品質が低いほど圧縮時間は短い
- 圧縮品質を下げることで比較的短時間でデータサイズを削減することが可能

通信時間計算モデル

- 概要 | 圧縮品質に対する通信時間の3次関数モデルを作成
- 転送時間はクライアント-サーバ間の往復通信スループットから算出
 - スループット測定環境はLTE環境/有線環境



結果

- 圧縮品質を下げることで圧縮無の場合と比べて全体の通信時間を削減可能
- 特にLTEのような帯域幅の限られた環境で効果的
- 品質30-50ではグラフの傾きが緩やか
- LTE環境では品質90、有線環境では品質70程度の場合、通信時間が圧縮無とほぼ同等
- 最適な圧縮品質は通信環境によって異なり、通信環境に応じて圧縮品質を制御することが重要

圧縮品質を変えた物体検出精度の比較

- 概要 | JPEG圧縮品質ごとにYOLOv5を用いて物体検出モデルを作成
- 216枚の船舶上の画像データを使用
 - 訓練データ: デストデータ = 151: 65
 - 学習に用いる画像を変えて2通りで実験
 - 圧縮前画像で学習・圧縮後画像で評価
 - 圧縮後画像で学習・評価

	圧縮無	q=80	q=70	q=60	q=50	q=40	q=30	q=20	q=10	q=5	q=1
mAP	0.476	0.458	0.451	0.467	0.465	0.463	0.456	0.391	0.369	0.241	0.222
圧縮前画像で学習	0.476	0.464	0.444	0.498	0.478	0.485	0.481	0.488	0.492	0.413	0.328
圧縮後画像で学習	0.476	0.464	0.444	0.498	0.478	0.485	0.481	0.488	0.492	0.413	0.328

結果

- 圧縮前画像で学習 q=30以上、圧縮後画像で学習 q=10以上ではmAPはほぼ一定であり、高い精度を保っている
- 品質10以上の場合にはPSNR値が30以上であるためだと考えられる
- 上記以外では精度が大幅に低下したが、圧縮後画像で学習した場合は精度の低下を抑えられている
- 圧縮後画像で学習したモデルを用いることで、JPEG圧縮品質を10まで下げた場合でも作業中の人や他の船を検出する精度に影響は小さい

IoT通信でJPEG画像データを扱う際、センサ側で必要に応じて圧縮品質を制御し、データサイズを削減してから通信を行うことで物体検出精度への影響を抑えつつ通信性能を高めることができると考えられる

まとめと今後の課題

- IoT通信を想定した環境で性能向上に効果的なデータ圧縮処理を調査し、圧縮品質を下げることで圧縮無の場合と比べて全体の通信時間を削減できると、圧縮後画像で学習したモデルを用いることで圧縮品質を10まで下げた場合でも物体検出精度への影響は小さいことを示した
- 今後は他の実験で得られた画像データに対してや、PNGなど別の不可逆圧縮アルゴリズムについて通信性能と学習精度を調査

本研究はJST CREST、PM2019M1およびPM2022M2の助成を受けたものです。また、本研究成果はデータ活用社会創成プラットフォーム mode を利用して得られたものです。