ソフトウェア品質シンポジウム2023
（SQiP2023）

MITSUBISHI ELECTRIC
Changes for the Better

# ワークフローモデルの構築による AI推論フローの 処理割当て手法の提案

伊藤 弘毅

Mail: Ito.Hiroki@dr.MitsubishiElectric.co.jp

2023/9/8

三菱電機株式会社

# 自己紹介

- 伊藤 弘毅(いとう ひろき)
- 三菱電機株式会社
  設計システム技術センター
  ソフトウェア技術推進部
- 担当業務
  - モデルベース開発の適用支援
    - 2022年 SQiPシンポジウム「パターンランゲージによるモデルベース開発初学者に対する知識共有の試み」を発表
  - **IoT・AIシステムの設計支援**
    - 2022年度 SQiP研究会
      研究コース5「人工知能とソフトウェア品質」に参加
      →研究会の活動成果と追加実験の内容を紹介します

# 会社紹介

**三菱電機グループは、重電システム、産業メカトロニクス、情報通信システム、電子デバイス、家庭電器などの製造・販売・サービスを事業目的としています。**

交通システム

FAシステム

半導体・デバイス

公共システム

自動車機器

空調・冷熱システム

エネルギーシステム

情報通信システム

住宅用設備・
生活家電

ビルシステム

宇宙システム

# 会社紹介(設計システム技術センター ソフトウェア技術推進部)

ミッション

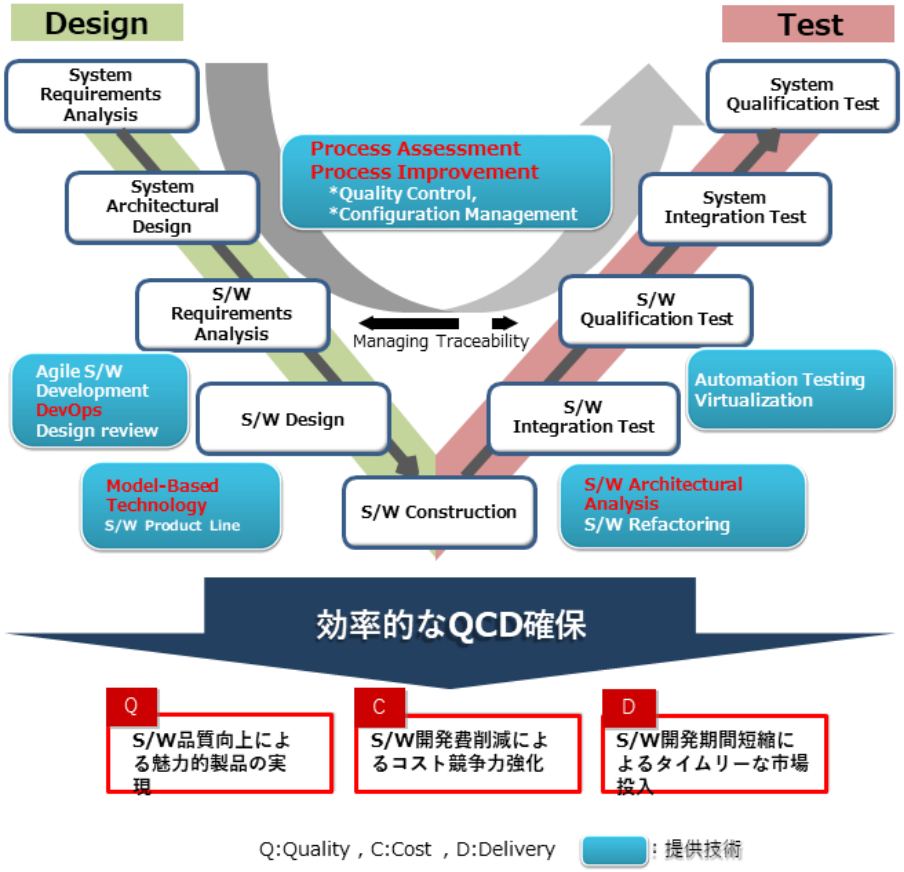
- ソフトウエア設計技術と開発プロセスの革新

ソフトウエア設計技術の開発と実用化とソフトウエア開発プロセスの改善とソフトウエア製品開発への適用支援を実施します。

活動実績

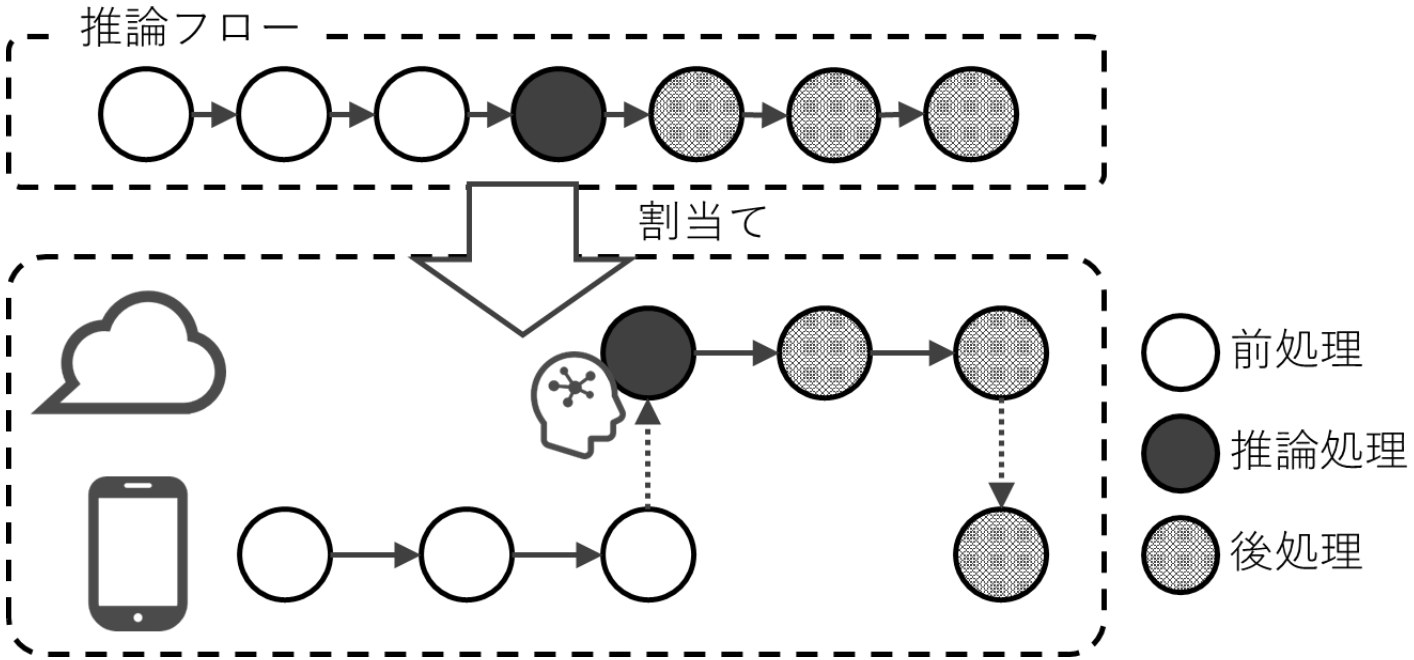
- 開発プロセス全体にわたる改善に貢献

ソフトウエア要求分析手法の構築、ソースコードの静的解析・構造分析ツールを活用した構造改善手法の構築及びシミュレーター開発による試験環境の整備など、ソフトウエア開発プロセス全体にわたる改善活動を推進しています。また、国際標準に準拠したプロセス改善、ガイドライン制定、設計上流でのデザインレビュー技法開発、セキュアな開発プロセス導入も推進し、製品に組み込まれるソフトウエアの品質と生産性の向上に貢献しています。



# 背景

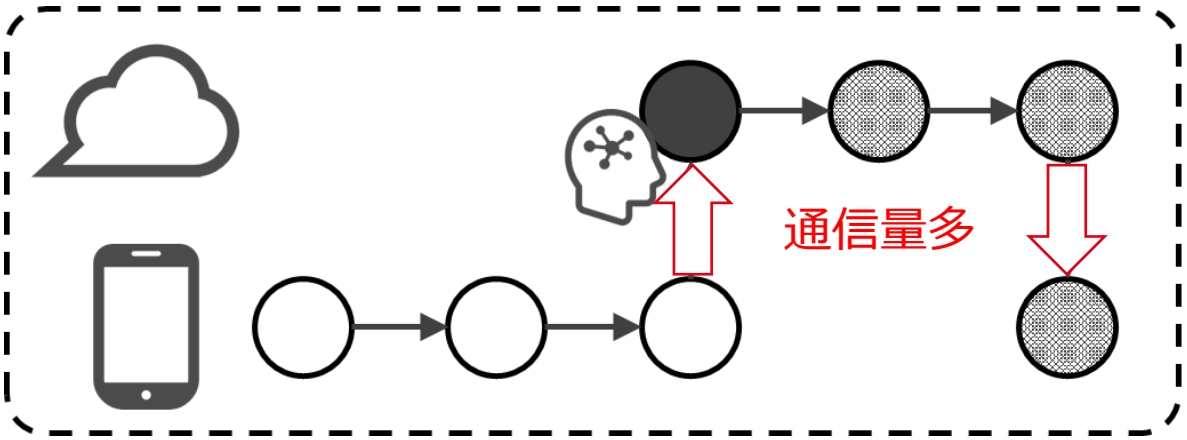
- エッジデバイス上で推論するエッジAIが選択肢に




それぞれの処理をクラウドとエッジの
どちらで実行させたらよいかがポイントになります

# 処理を割当てる時の課題

- どのようにクラウドとエッジに処理を割当てるのか
  - 処理時間、通信量、クラウド課金額など、いろいろな観点で要求品質を満たすか確認する必要がある

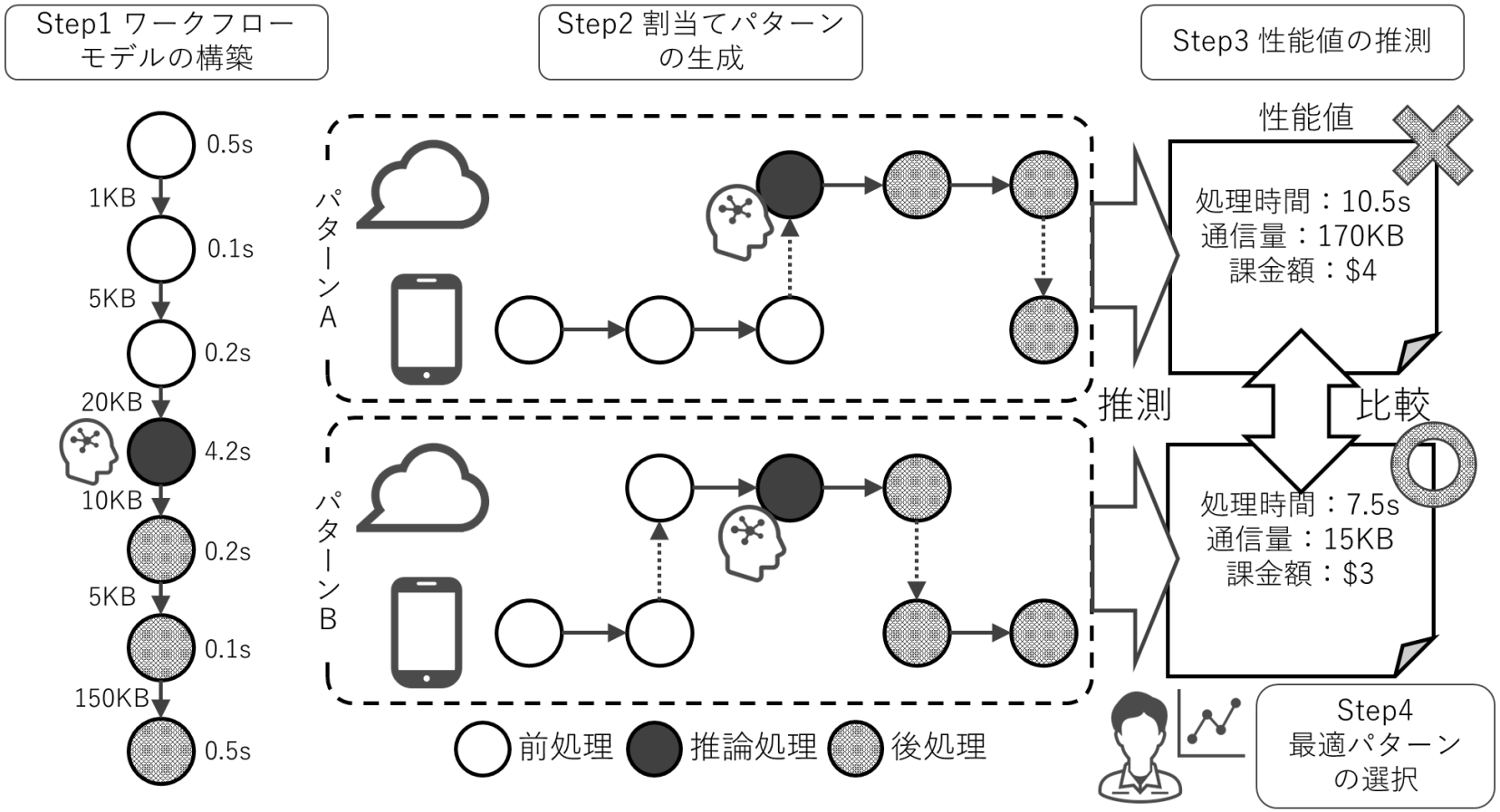



処理時間：7.0s
通信量：10MB
課金額：
$4000/month

あまり検討をしないで決めてしまうと
大量の通信や高額な課金に悩まされる可能性が・・・

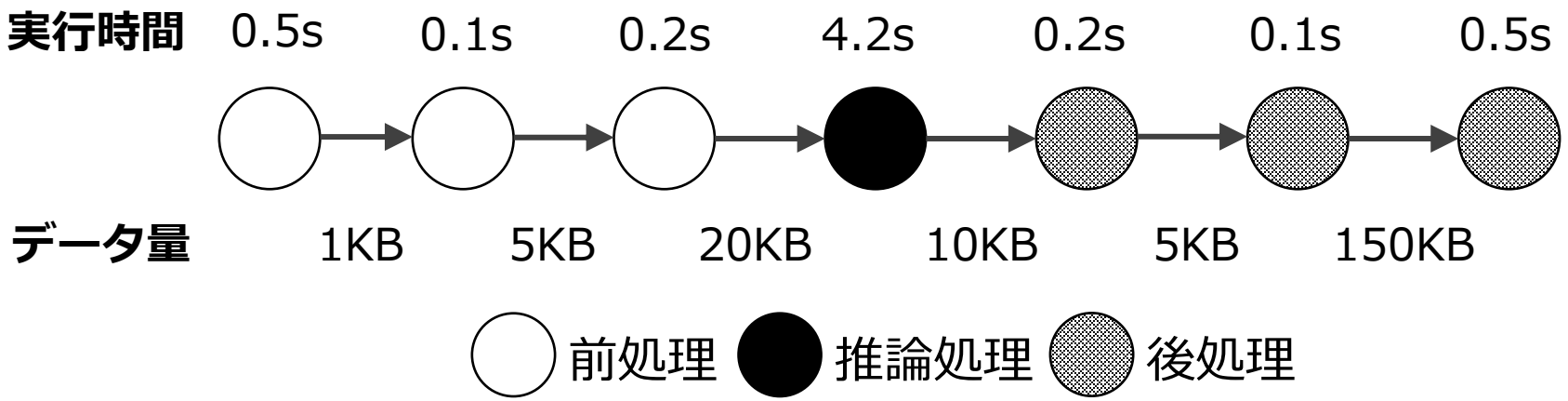
# 手法の全体像

- 4つのステップで実現する



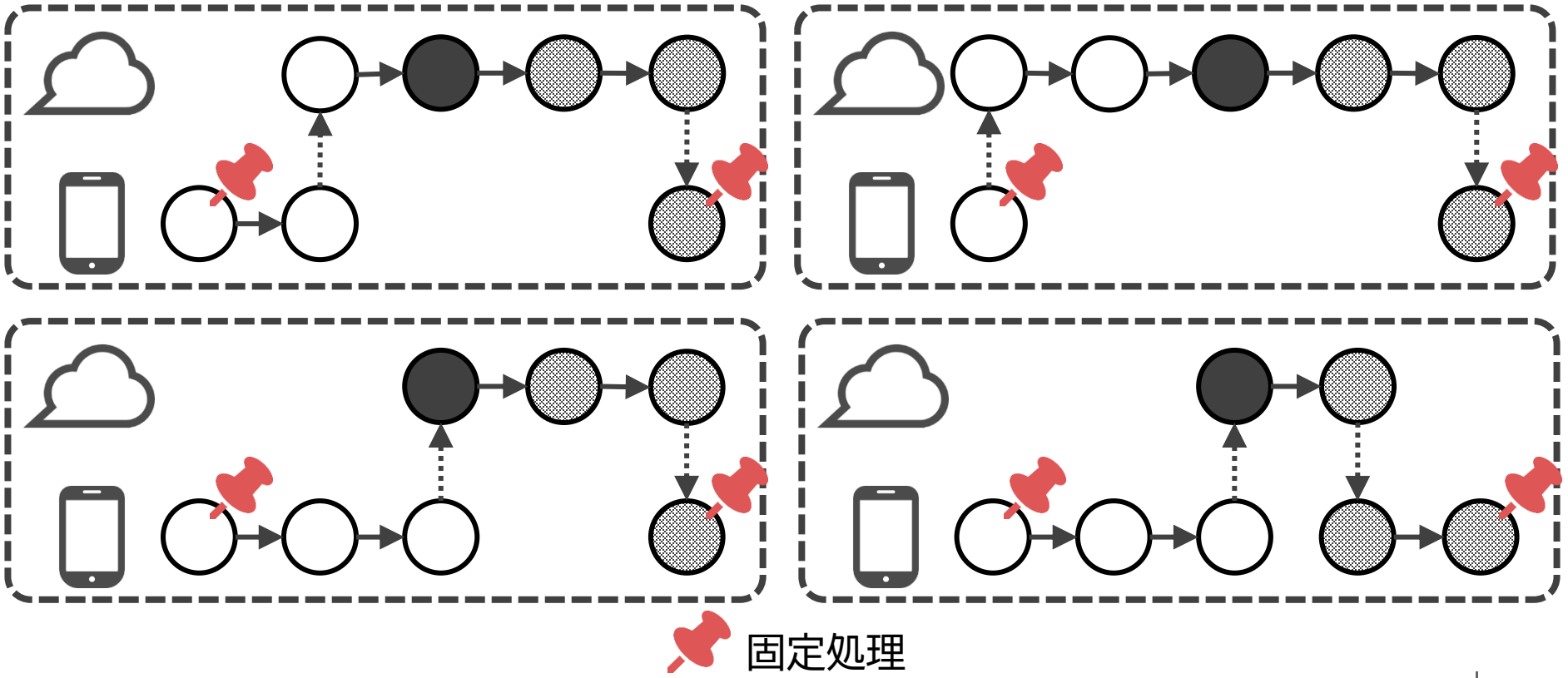
# Step1：ワークフローモデル

- 推論フローを表現するモデルを構築
  - 処理をノード、データの流れをエッジで表現
- ワークフローモデルにパラメータを設定
  - ノード：各処理の実行時間
  - エッジ：処理間で渡されるデータサイズ



# Step2:割当てパターンの生成

- 処理をクラウドまたはエッジに配分
  - 実行装置を固定する処理を決める
  - 取りうる全ての配分の組合せを割当てパターンとして抽出



# Step3：性能値の予測

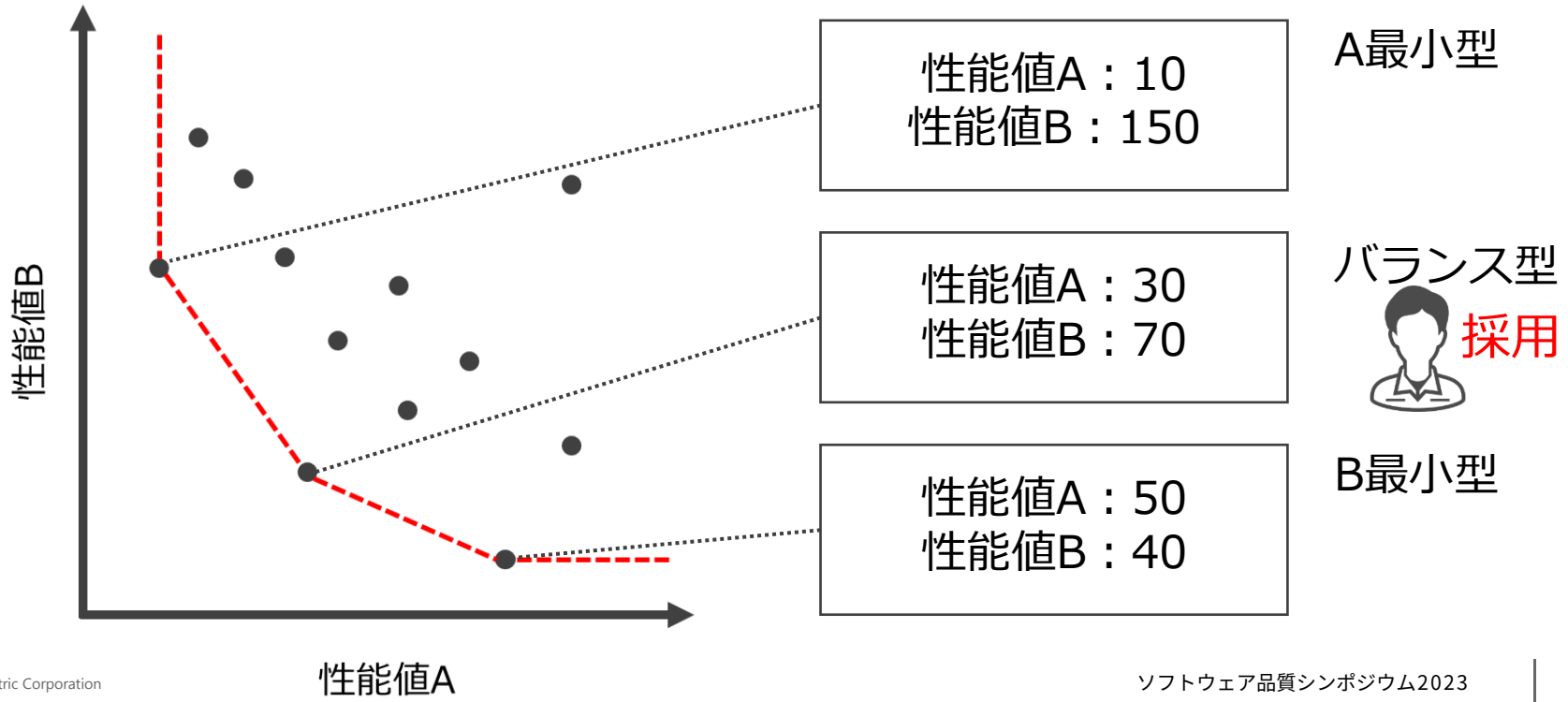
- 性能値として**処理時間**と**通信量**、**クラウド課金額**を推測
- 処理時間
  - 実測に用いたマシンと実機マシンの性能比から算出

$$\text{推測処理時間} = \text{実測処理時間} \times \frac{\text{実測マシン性能値}}{\text{実機マシン性能値}}$$

  - マシン性能値：以下のCPU指標値から算出式を定義する
    ※処理の特性により、重要な指標値が変わるため
    ex.）コア数・クロック周波数・ベンチマーク
- 通信量
  - 通信時間 $= \frac{\text{通信データサイズ}}{\text{通信速度}}$
- クラウド課金額
  - 使用サービスの料金体系に、推測した処理時間と通信量を反映し算出

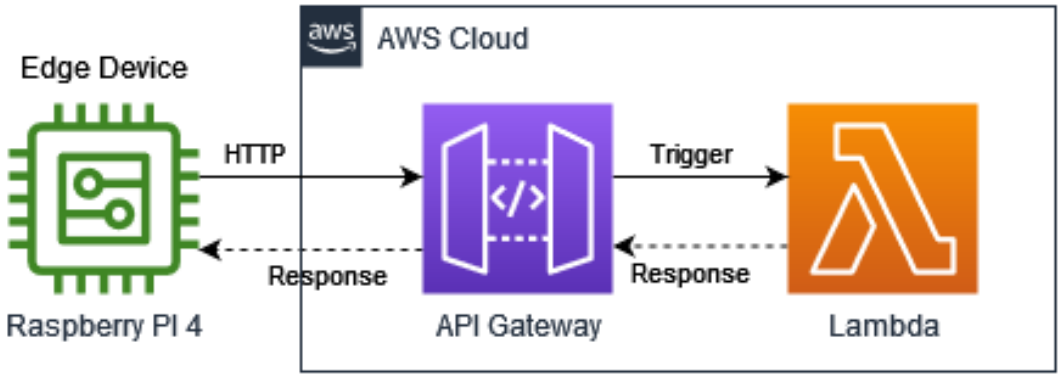
# Step4：最適パターンの選択

- 推測した性能値をグラフにプロットし、
  パレートフロントを導出
  - 他のどの解にも優越されない解を繋げたトレードオフ曲線
  - パレートフロントを見ていずれかの割当てパターンを採用



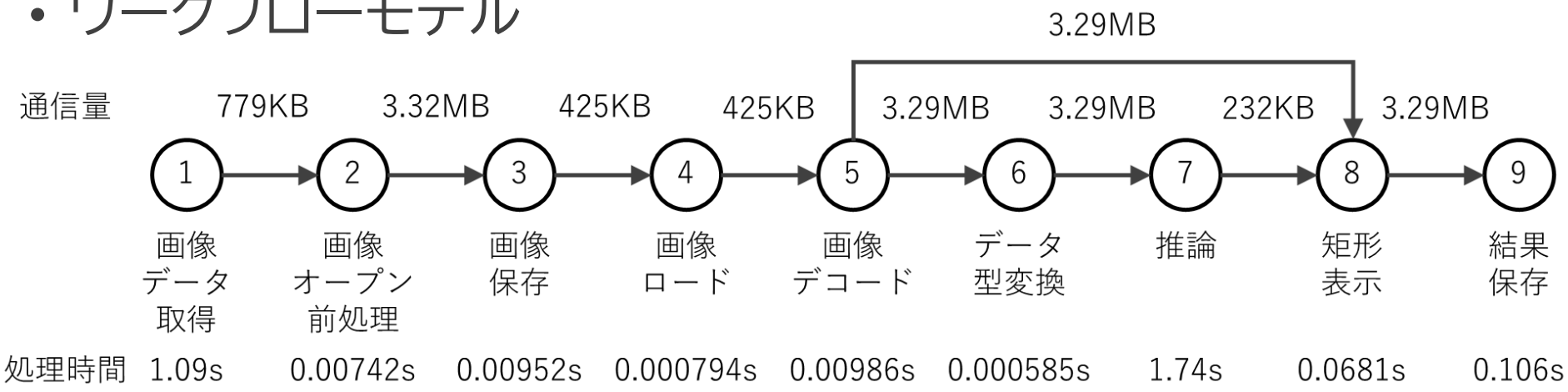
# 実験

- 本手法を3つの推論フローに適用
  - ワークフローモデルを構築
  - 処理時間と課金額でプロットし、パレートフロントから割当てパターンの候補を導出
- 推論フロー
  - （a）物体検知
  - （b）画風変換
  - （c）音声認識＋感情分析
- エッジとクラウドで構成されるシンプルな構成



# (a)物体検知

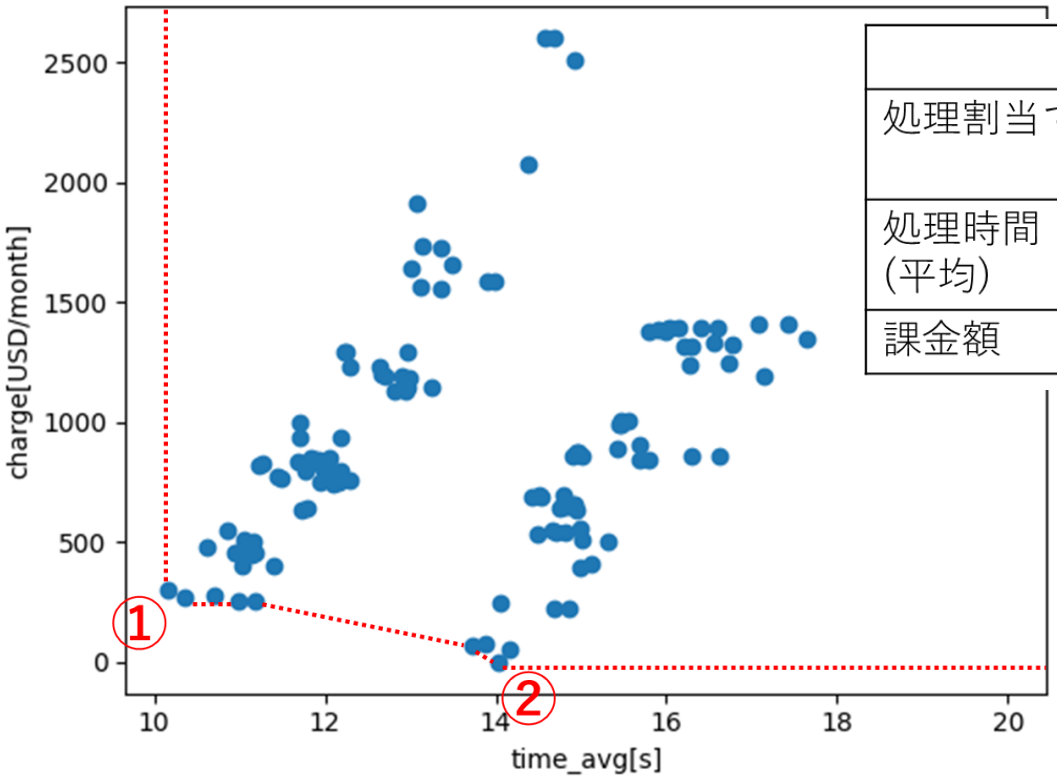
- 機能
  - 入力画像に含まれる物体を検知して領域を出力する
  - 例:机・椅子・傘
- ワークフローモデル




分岐なくシーケンシャルに実行されるシンプルなフロー

# (a)物体検知の実験結果

- 全128割当てパターンをプロット
- パレートフロント→2つの最適解の候補を導出
  - パターン①:割当てられる全ての処理がクラウドの場合
  - パターン②:すべての処理がエッジの場合



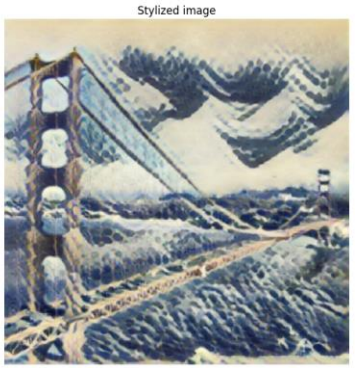

| | パターン① | パターン② |
|---|---|---|
| 処理割当て | クラウド：2～8<br>エッジ：1、9 | クラウド：なし<br>エッジ：1～9 |
| 処理時間<br>(平均) | 10.15s | 14.04s |
| 課金額 | 312USD/month | 0USD/month |

割と当たり前の結果・・・？
それでも他のパターンと
比較ができることはメリット？

# (b)画風変換

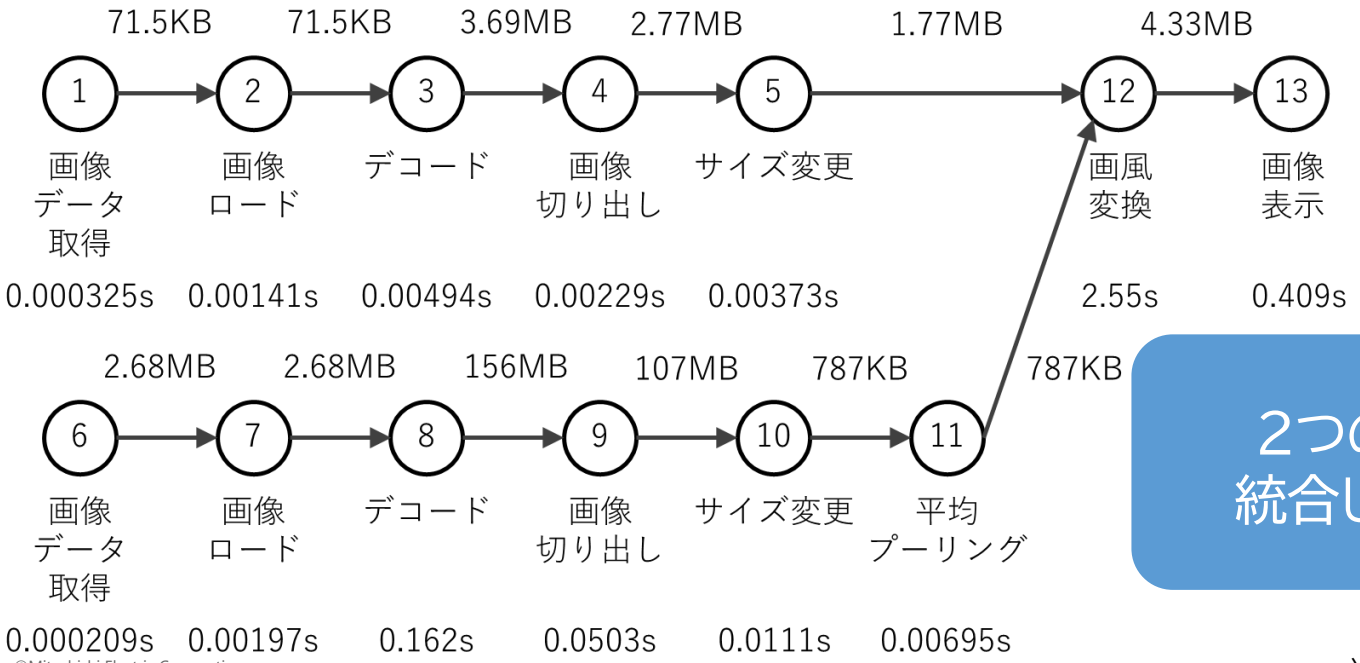
- 機能
  - 入力画像の画風を
    変換して出力する
  - 変換対象の画像とスタイル画像を入力する






https://www.tensorflow.org/hub/tutorials/tf2_arbitrary_image_stylization

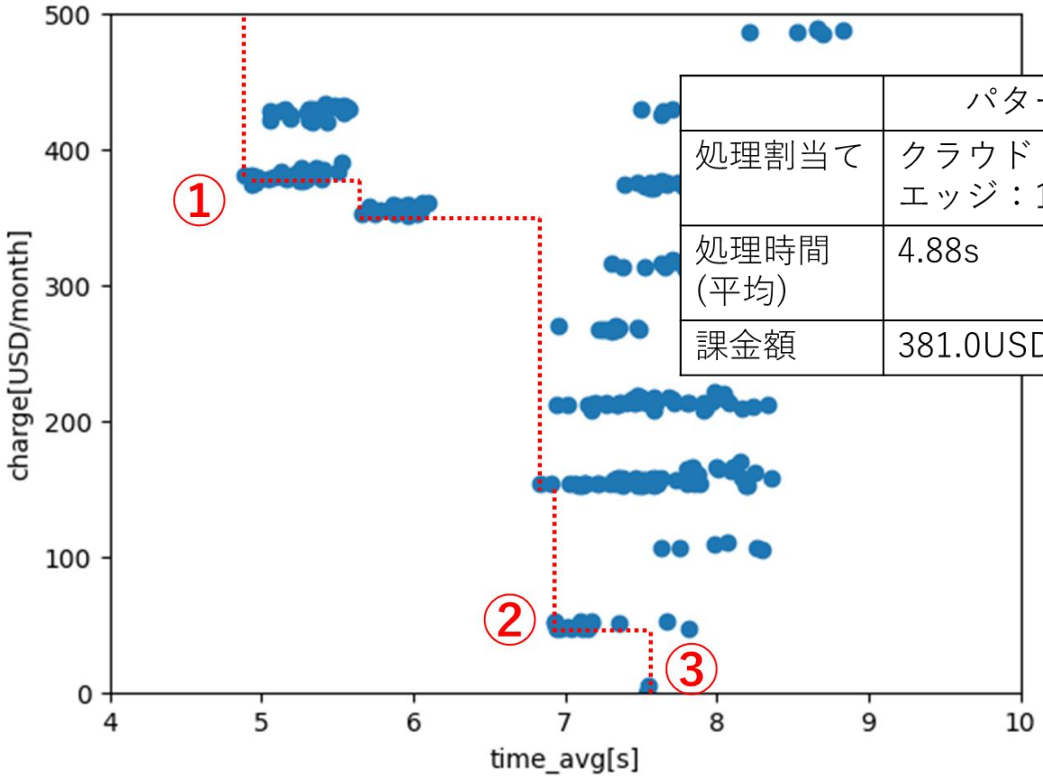
- ワークフローモデル




2つの画像処理結果を
統合して推論するフロー

# (b)画風変換の実験結果

- 全2048割当てパターンをプロット
- パレートフロント→3つの最適解の候補を導出
  - パターン①③:割当てる全ての処理がクラウドorエッジ
  - パターン②:入力の画像処理のみエッジ




| | パターン① | パターン② | パターン③ |
|---|---|---|---|
| 処理割当て | クラウド：2～12<br>エッジ：1,13 | クラウド：6～11<br>エッジ：1~5,12,13 | クラウド：なし<br>エッジ：1～13 |
| 処理時間(平均) | 4.88s | 6.95s | 7.53s |
| 課金額 | 381.0USD/month | 47.4USD/month | 0USD/month |

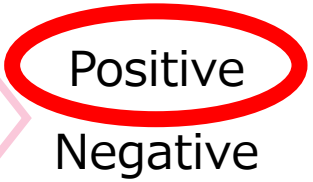
一部をエッジで実行する
中間解の候補が出てきた
（ただ有力候補は①か③？）

# (c)音声認識＋感情分析

- 機能
  - 音声データから
    発言をテキスト化
  - テキストを入力に、発言のネガポジ分析をする

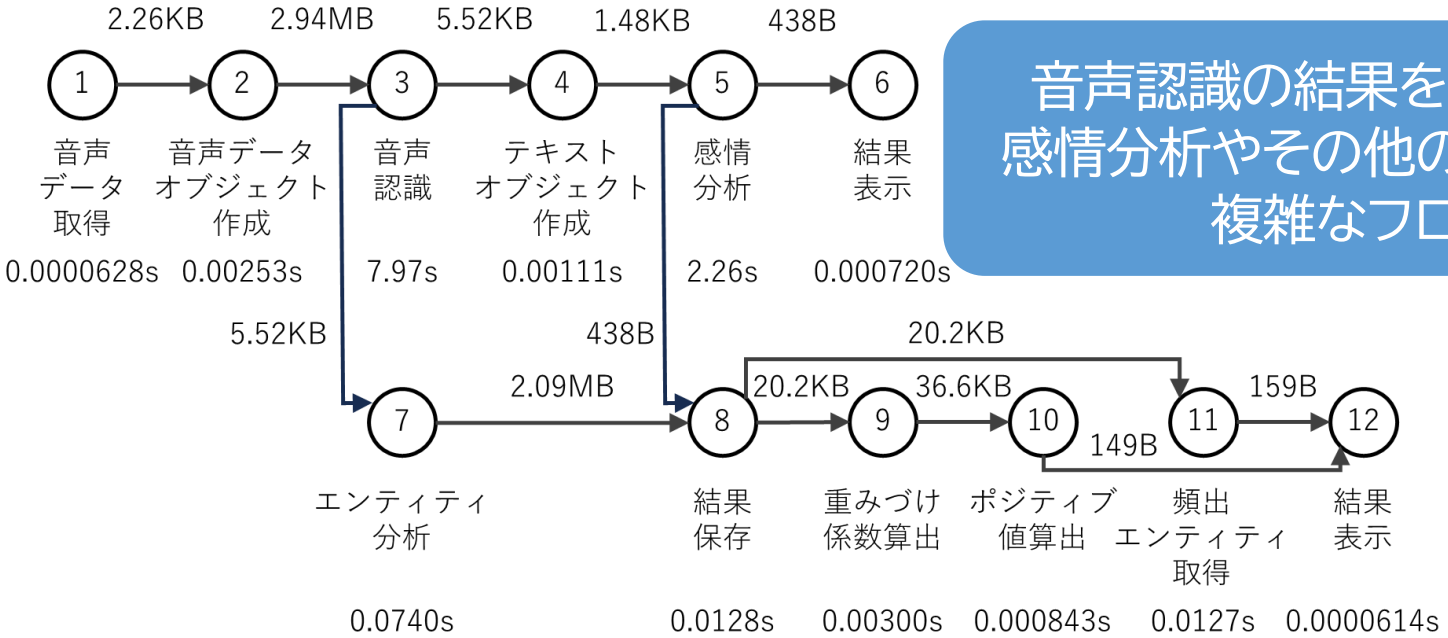
音声データ


音声認識
(実際は英語)




感情(ネガポジ)
分析

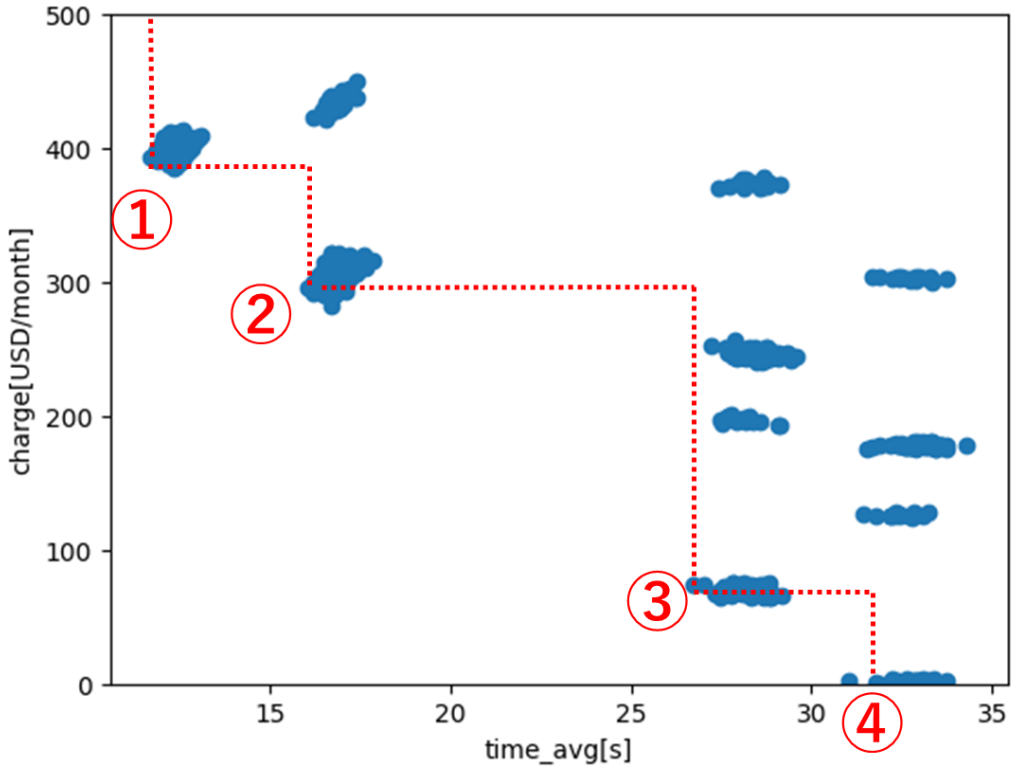
- ワークフローモデル




音声認識の結果をうけとって
感情分析やその他の分析をする
複雑なフロー

# (c)音声認識＋感情分析の実験結果

- 全512割当てパターンをプロット
- パレートフロント→4つの最適解の候補を導出
  - パターン②:音声認識以外をエッジで実行
  - パターン③:音声認識をエッジで実行




| | パターン① | パターン② |
|---|---|---|
| 処理割当て | クラウド：2~5,7~11<br>エッジ：1,6,12 | クラウド：2~4,7~11<br>エッジ：1,5,6,12 |
| 処理時間<br>(平均) | 11.7s | 16.1s |
| 課金額 | 393.0USD/month | 295.3USD/month |

| | パターン③ | パターン④ |
|---|---|---|
| 処理割当て | クラウド：4~5,7~8<br>エッジ：1~3,6,9~12 | クラウド：なし<br>エッジ：1~12 |
| 処理時間<br>(平均) | 27.0s | 31.8s |
| 課金額 | 74.0USD/month | 0USD/month |

複雑なワークフローになっても
候補を提示することができた！

# 考察：推測精度

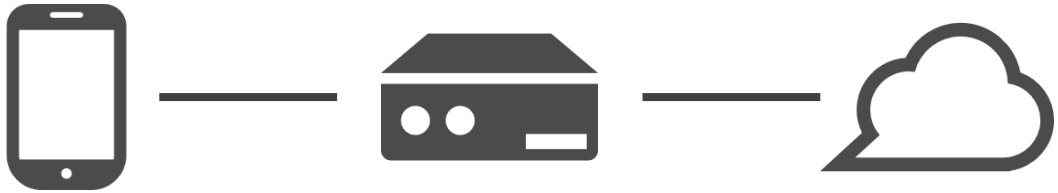
- 測定値と実測値を比較：誤差あり
  - 処理7：エッジでCPU資源が枯渇し性能劣化した可能性
  - 通信時間：トラフィックの輻輳を考慮していない

| | | 総時間 | 処理1 | 処理2 | 処理3 | 処理4 | 処理5 | 処理6 | 処理7 | 処理8 | 処理9 | 通信時間 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| パターン① | 予測値 | 3.88 | 1.12 | 0.126 | 0.0097 | 0.00138 | 0.0173 | 0.000997 | 1.99 | 0.117 | 0.107 | 0.396 |
| | 実測値 | 6.38 | 0.991 | 0.0595 | 0.0070 | 0.00064 | 0.00724 | 0.00051 | 1.93 | 0.045 | 0.118 | 3.51 |
| パターン② | 予測値 | 7.54 | 1.08 | 0.466 | 0.0097 | 0.00509 | 0.0609 | 0.00350 | 5.39 | 0.420 | 0.101 | — |
| | 実測値 | 13.96 | 0.906 | 0.122 | 0.0224 | 0.00167 | 0.0293 | 0.00173 | 12.5 | 0.224 | 0.117 | — |

モデルの精度向上が今後の課題

# 考察:スケーラビリティ

- 実験ではクラウド―エッジの構成だったが・・・
- 間にゲートウェイを挟んだ構成にすると



ゲートウェイ

- 割当てパターンの数が一気に増える
  - (b)画風変換はパターンが177147(=$3^{11}$)通りに
  - 解析完了までに5時間かかった・・・

全探索ではなく有望な解を探索するなど
アルゴリズムを工夫する必要がある

# 考察:効率性

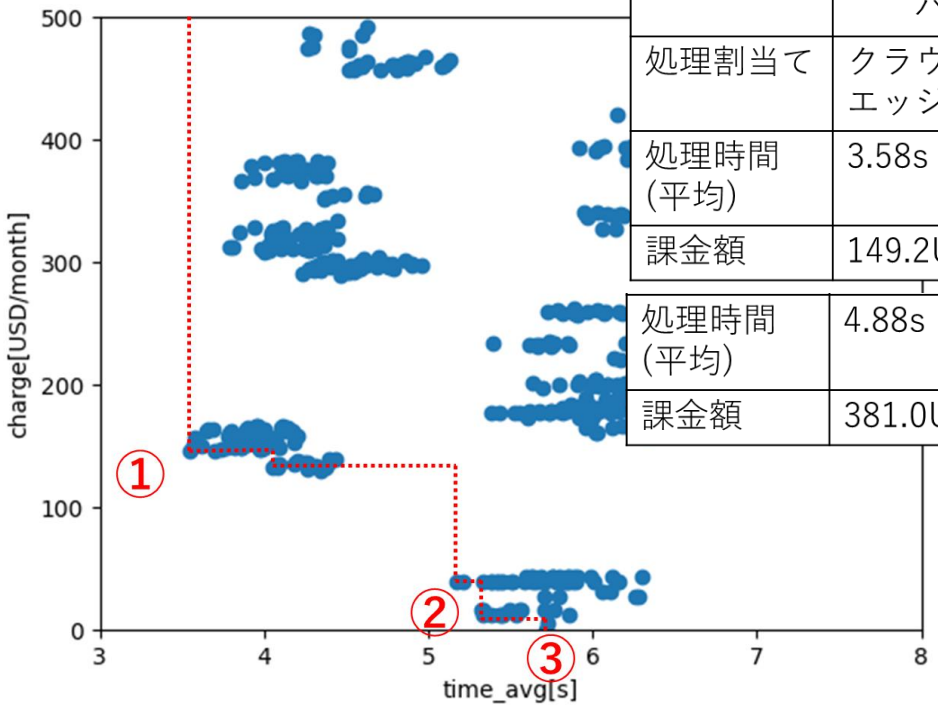
- 実験期間:推測→約2日、実測→約7日
- 実測に7日もかかったのは・・・
  - パッケージサイズ
    AWS Lambdaのモジュールは通常解凍後250MB以内の制限あり
    →制限解除のためにコンテナを利用する必要あり
  - クラウドサービスの連携
    AWS Lambdaだけではなく、エッジデバイスからの通信を
    受け取るAPI Gatewayを構築し、サービスを接続する必要がある
  - クラウドのアカウント準備、ロールの設定
    クラウドサービスを利用する準備として様々な設定が必要
  - エッジデバイスの調達
    試験装置をすぐに手配できるとは限らない。シェアの可能性も

(a)物体検知の
推論フロー

**実機を使うと実装上の課題を解決する必要がある・・・
推測の方がコストが低くすむ**

# 考察:手法の応用

- (b)画風変換の入力画像サイズを小さくしたときの割当て候補を導出
- パレートフロント
  - **最適解の候補は変化なし。処理時間と課金額の値は減少**
    →要求により採用する最適解が変わる可能性あり




| | パターン① | パターン② | パターン③ |
|---|---|---|---|
| 処理割当て | クラウド：2～12<br>エッジ：1,13 | クラウド：6～11<br>エッジ：1~5,12,13 | クラウド：なし<br>エッジ：1～13 |
| 処理時間(平均) | 3.58s | 5.33s | 5.71s |
| 課金額 | 149.2USD/month | 11.8USD/month | 0USD/month |
| 処理時間(平均) | 4.88s | 6.95s | 7.53s |
| 課金額 | 381.0USD/month | 47.4USD/month | 0USD/month |

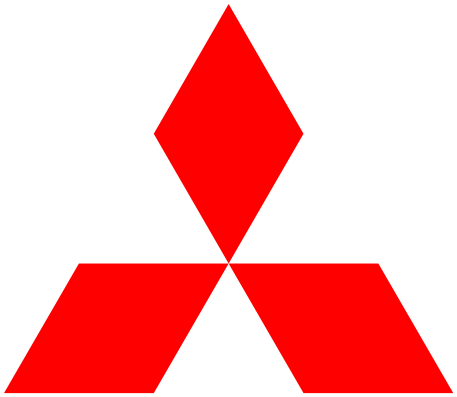
サイズ小

サイズ大

仕様変更時の影響を分析できる
他の影響分析にも
応用できる可能性もある

# おわりに

- 「ワークフローモデルの構築によるAI推論フローの処理割当て手法」をご紹介
  - IoTシステム上にAIの推論フローを実現する時の処理時間やクラウド課金額等を机上で推測
  - パレートフロントを導出することにより最適な処理割当ての候補を導出
- 今後の展望
  - 処理時間と通信時間の推測精度向上
  - スケーラビリティを考慮した最適化アルゴリズムの検討
  - ユースケースに応じた提案手法の応用可能性の検討
- 謝辞
  - 本研究にあたりSQiP研究会 研究コース5の石川冬樹主査、栗田太郎副主査、徳本晋副主査には丁寧に指導を賜りました。深く御礼を申し上げます



MITSUBISHI
ELECTRIC

*Changes for the Better*