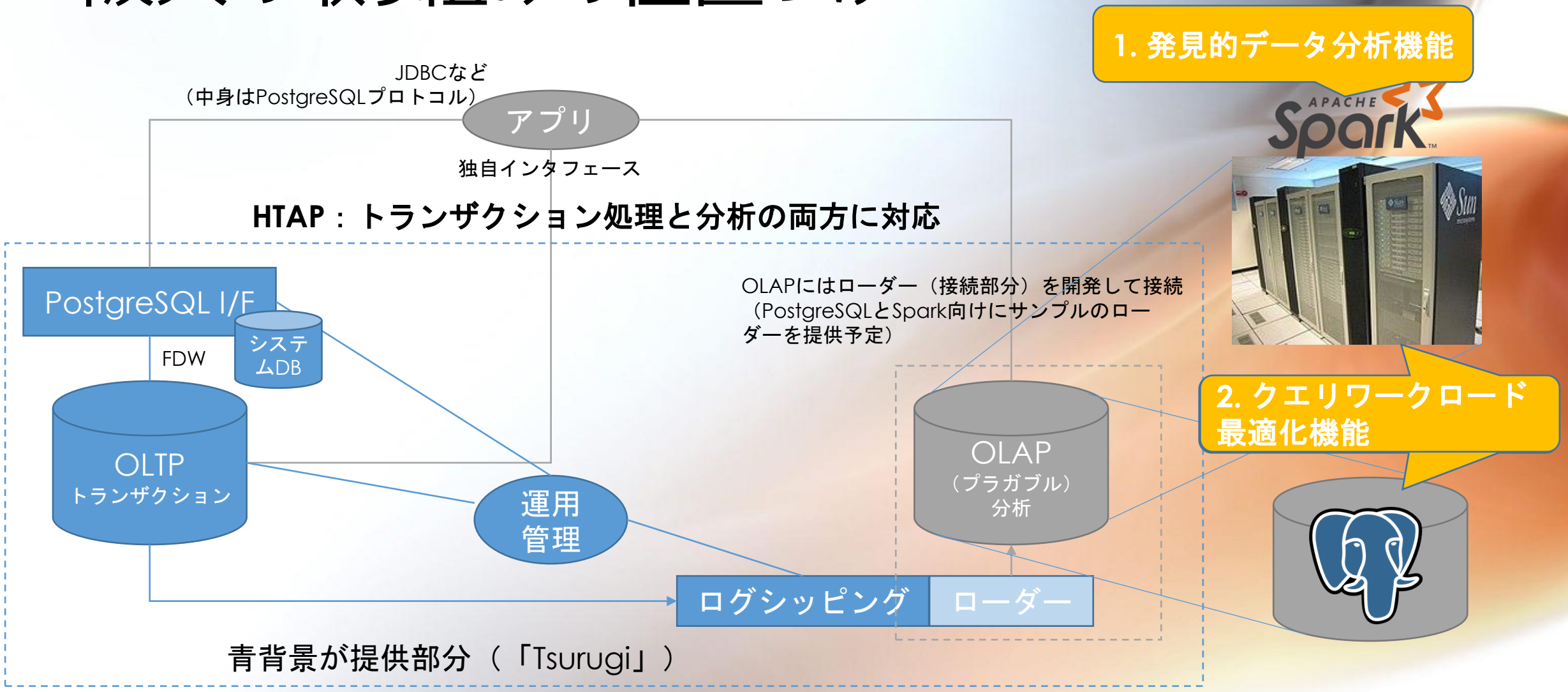
OLAP高速化技術

大阪大学　大学院情報科学研究科
教授　鬼塚真

阪大の取り組みの位置づけ



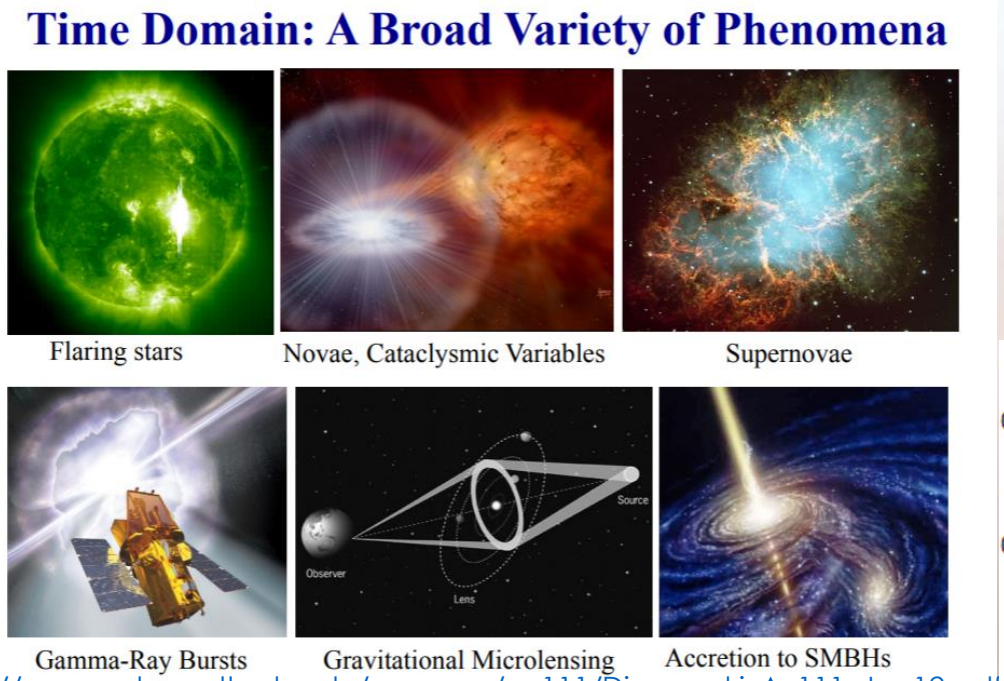
取り組みの概要

- **発見的データ分析(Exploratory data analysis)**
  - 概要: 有用性の高いデータを自動探索する技術
  - 成果の形態: Tsurugi の参考実装(OLAP on Spark)
  - 適用例: 天文台における変動天体の発見
- **クエリワークロード最適化**
  - 概要: 時間変化するワークロードに対するスキーマ最適化技術
  - 成果の形態:
    - 実体化ビュー推薦: 論文 + 参考実装(Cassandra)
    - Cardinality 推定: 論文 + 参考実装(Tsurugi, NoSQL に組み込み可能なライブラリ)
  - 適用例: 探索中(RDB/NoSQL のクエリ返却件数予測)

発見的データ分析の概要

- 発見的データ分析とは，購買データや天体観測データなどの膨大なデータを対象として，通常とは異なる特徴的なデータを発見する技術
  - 国立天文台と協力し明るさが変化する**変動天体の発見**に取り組んでいる

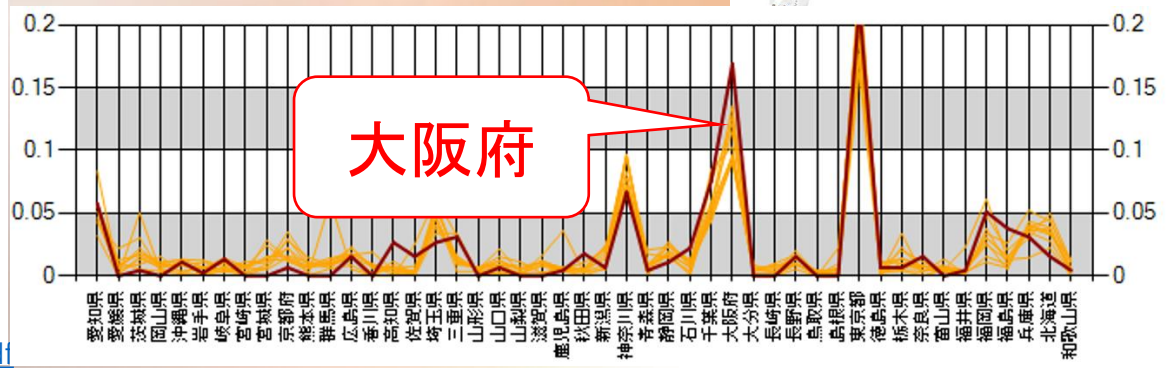
応用例1: 時系列変動天体の発見



https://www.astro.caltech.edu/~george/ay111/Djorgovski_Ay111_Jan12.pdf

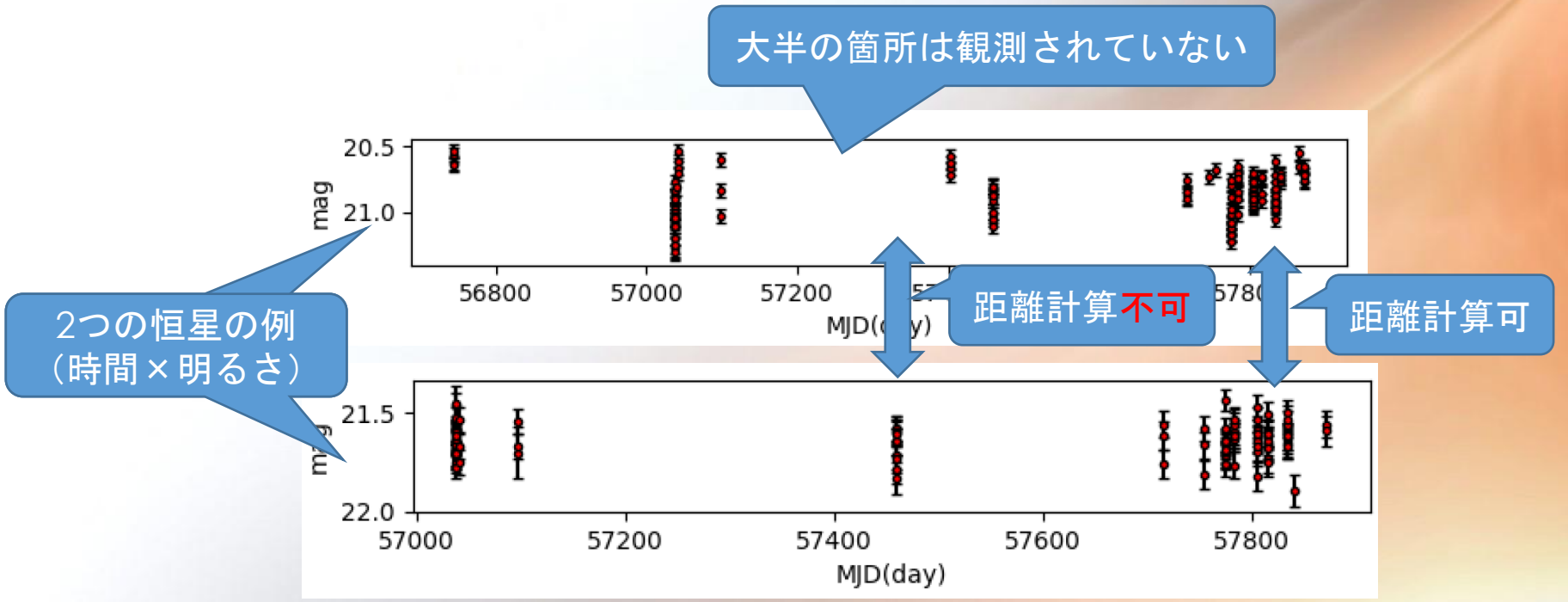
応用例2: 特徴的なデータ発見

1人1日当たりゴミ排出量 第１位 大阪府
（環境省， 2011年）



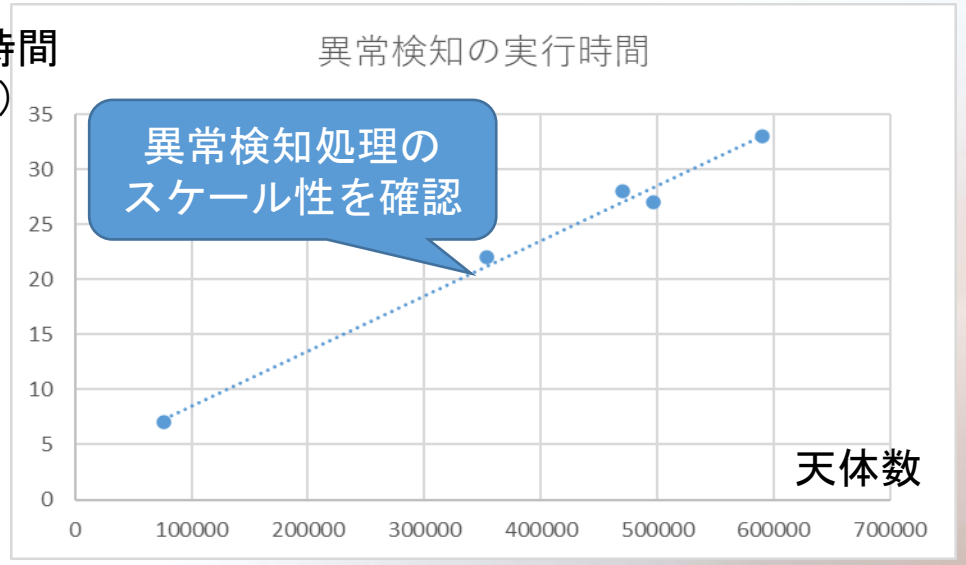
応用例: 時系列変動天体の発見

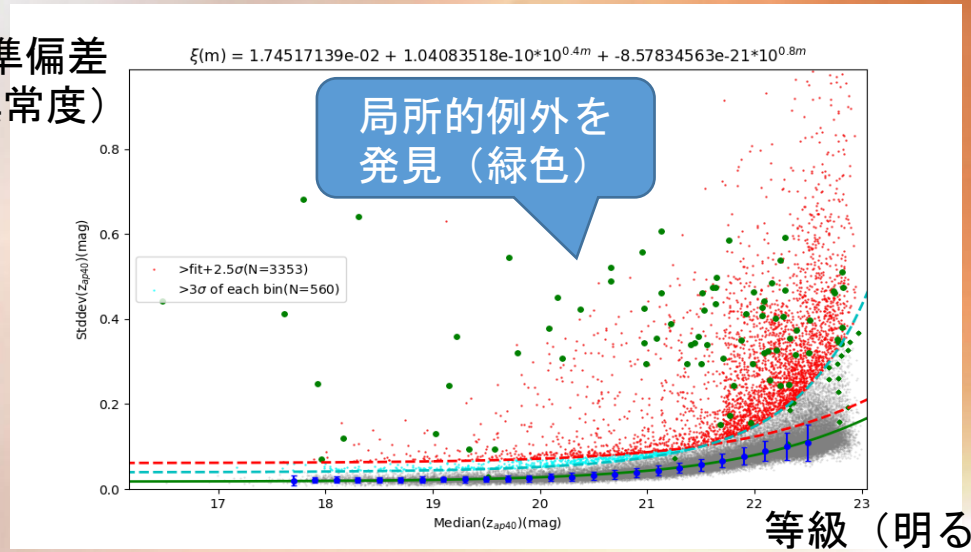
- **課題定義:** 観測点が極めて少ない時系列データに対する異常検知
- **アプローチ**: データをクラスタ化しクラスタ内で欠損値補完後に異常検知



応用例: 時系列変動天体の発見

- **アプローチ**: データをクラスタ化しクラスタ内で欠損値補完後に異常検知
- **到達点:** Spark 環境において異常検知(isolation forest)の有効性を確認
  - 性能: 高スケール性を確認(補足: 2.4億レコードを40分で分析)
  - 精度: 既存技術と同等 + 局所例外性の高い異常データの検出





発見的データ分析のまとめ

- **概要**: 大規模時系列データに対する異常検知（欠損データが多い応用）
  - 適用例: 天文台における時系列変動天体の発見
- **成果の形態**: Tsurugi の参考実装(Spark 組み込み)
- **成果の利用方法**:
  - Spark 上で本モジュールをインストール
  - 補足: 参考実装では天体観測データのスキーマを利用しているため，この部分を改造する必要がある

取り組みの概要

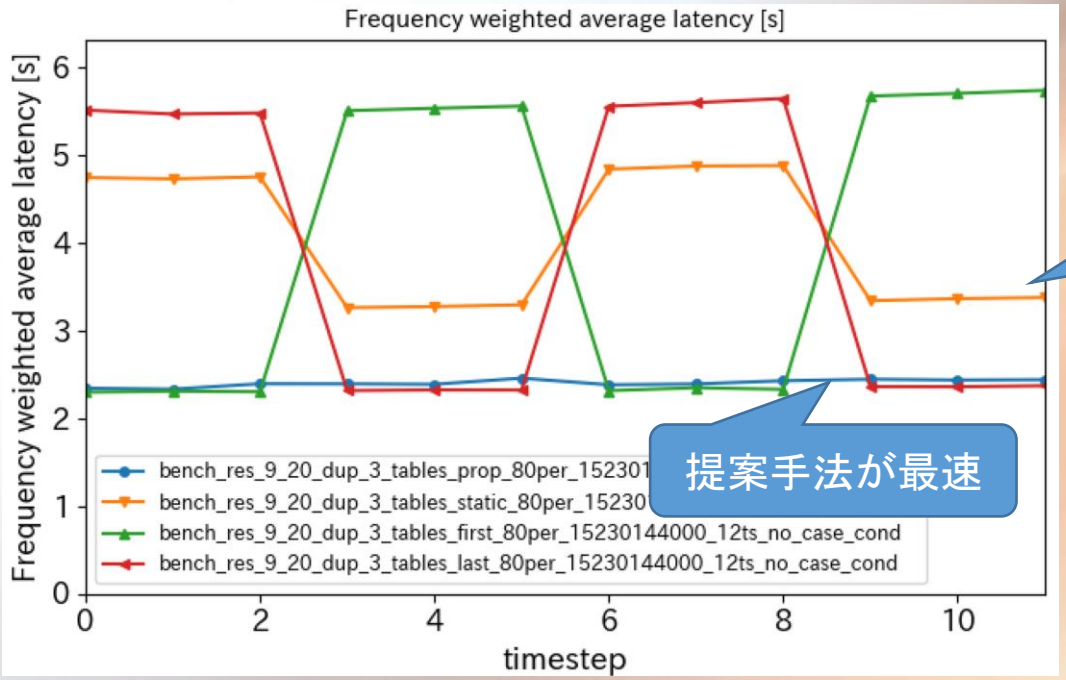
- **発見的データ分析(Exploratory data analysis)**
  - 概要: 有用性の高いデータを自動探索する技術
  - 成果の形態: Tsurugi の参考実装(OLAP on Spark)
  - 適用例: 天文台における変動天体の発見
- **クエリワークロード最適化**
  - 概要: 時間変化するワークロードに対するスキーマ最適化技術
  - 成果の形態:
    - 実体化ビュー推薦: 論文 + 参考実装(Cassandra)
    - Cardinality 推定: 論文 + 参考実装(Tsurugi, NoSQL に組み込み可能なライブラリ)
  - 適用例: 探索中(RDB/NoSQL のクエリ返却件数予測)

実体化ビュー推薦

- **課題設定:** 計画的に変化するワークロードを想定し，総コストを最小化する実体化ビューを推薦
- **アプローチ:** ワークロードコスト+DBマイグレーションコストの総和を整数計画問題で最適化
- **到達点**
  - TPC-Hを拡張（時間変化）したベンチマークで従来技術より最大2倍程度の性能向上を確認
  - 最適化問題を分割統治することで，見込みのない解候補を枝刈り（探索コストを1/10程度に削減）

実体化ビュー推薦: 評価実験

- ベンチマーク: TPC-H を拡張したベンチマークを利用
  - 2つの地域（日米）から発生する周期変化するワークロードを用いて評価
  - 到達点: 平均的なワークロードに対する最適化の約2倍の高速化を実現



Cardinality 推定

- **課題設定:** DBに関する統計情報を事前に機械学習し，SQL文に対しヒットするレコード数を予測する
- **アプローチ:：**非自己回帰モデルを活用したロバストなCardinality推定
- **到達点:**
  - 既存技術より安定的に高性能を達成(VLDB併設ワークショップ発表)
  - 部分スキーマごとに学習モデルを用意するアプローチ（大規模対応）：試行中

Cardinality 推定: 評価実験

- データ: DMV: 11.6M 件, 11 属性
- 評価クエリ: データに対してランダムに生成された2000クエリ
- 評価指標: q-error: 実際の値から何倍離れているかを示す値(1がベスト)
- 評価結果: 既存技術より安定的に高性能を達成

手法	平均 q-error
提案手法 (Transformer-based)	1.098
Naru-ResMADE-0to10	1.076
Naru-ResMADE-10to0	210.935

安定して高性能

経験に基づくパラメータにより
不安定な性能

クエリワークロード最適化のまとめ

- **概要**: 時間変化するワークロードに対するスキーマ最適化技術
  - 適用例: 探索中(RDB/NoSQL のクエリ返却件数予測)
- **成果の形態**:
  - 実体化ビュー推薦: 論文+参考実装(Cassandra組み込み済)
  - Cardinality推定: 論文+参考実装(汎用モジュール: Tsurugi, NoSQL に組み込み利用)
- **成果の利用方法**:
  - 実体化ビュー推薦:論文+参考実装を利用して，他の環境に再実装
  - Cardinality推定: Tsurugi, NoSQL の 1) フロントエンドでクエリの返却件数を推定, 2) クエリ最適化の内部(Planner)で活用

今後の予定

- **発見的データ分析(Exploratory data analysis)**
  - 2021年度: CPU利用率の向上(Koalasの導入等)
  - 2021-22年度: 欠損値補完の精度改善(周期解析技術の導入)
- **クエリワークロードの最適化**
  - 2021年度: 技術検証＆国際会議投稿(Cardinality estimation, 実体化ビュー推薦)
  - 2022年度: 大規模DB対応