リレーショナルデータベース上のXMLビューに対する外部関数を考慮した問合せ処理

筑波大学
川田純　石川佳治　北川博之

発表手順

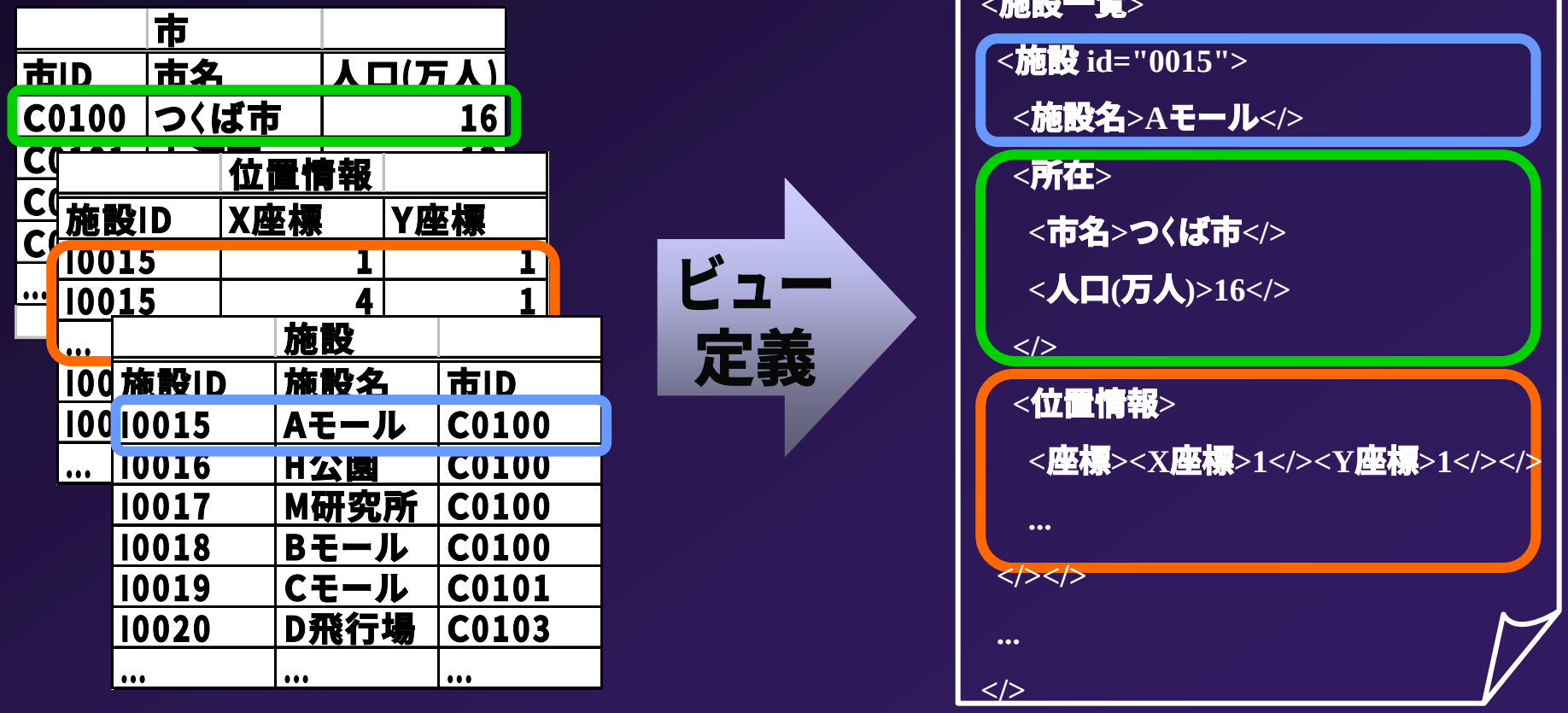
1. 背景と目的
2. 外部関数処理へのアプローチ
3. 提案処理方式
4. 問合せ変換
5. 評価実験
6. まとめ

背景

- データ交換共通フォーマットとしてXMLが広く利用
  - インターネットの急激な普及
  - 構造化文書データ表現としてのXMLの汎用性
  - ➡大量のXMLデータ管理・運用技術への要求
- XML形式によるデータベース出力が重要
  - 大量のデータがリレーショナルデータベースに格納
  - データを用途に応じて加工する必要性
  - データベースシステムにおける確立された大量データ管理技術の有効利用

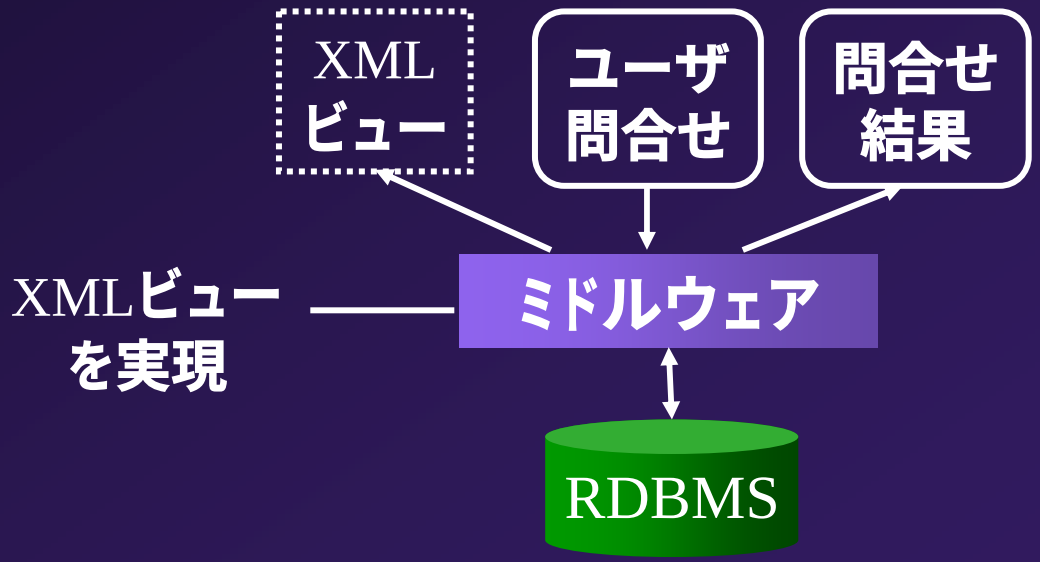
リレーショナルデータベース上のXMLビュー

- XMLビューを用いた変換方式
  - リレーショナルデータベースからXML文書への変換定義のみが存在し，実体のXML文書は存在しない



リレーショナルデータベース上のXMLビュー

- リレーショナルデータベース上にXMLビューを定義
  - RDBMSとミドルウェアの連携
    - ➡RDBMSの問合せ処理能力を最大限に活用
  - XML問合せ言語を用いて，ユーザが必要な部分だけXML文書化
    - ➡問合せ結果に不要なXML部分文書生成コストを削減



ユーザ定義の外部関数

- XMLに対する処理要求は多様
  - タグの階層構造で表されるXML文書構造操作
  - XML文書の記述に対する記述内容操作
    - 比較的単純な語句に関するパターンマッチ
    - ドメインに依存した条件を外部関数によって指定

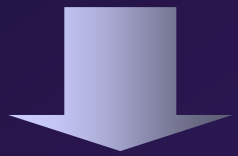
```
<結果>{
  for $施設 in view("施設一覧")/施設一覧/施設
  where isWider( $施設/位置情報, 10, "km“ )
  return <施設>$施設/施設名</>
}</>
```

- Java等の汎用プログラミング言語
- 述語として指定
- 引数にXMLフラグメント

外部関数を用いて面積10km^{2}以上の施設名を取得

本研究の目的

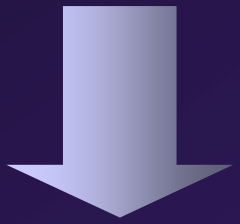
- XMLビューに対する外部関数を用いた問合せ処理の支援
  - XQuery:外部関数の検討，X²QL:外部関数の導入
  - XQuery，X²QL処理系の実現



- XMLビュー問合せ処理に関する先行研究のアプローチを拡張
- 2種類の外部関数処理方式を考案

既存アプローチ

- 既存アプローチではXML文書要素の**タグ付け演算**以外の問合せ処理をSQL問合せとしてRDBMSに委譲
- 外部関数問合せは対象XMLフラグメントの存在が前提
  - ➡外部関数処理は単純にRDBMSへ委譲不可能



外部関数処理の前に，予めリレーションタプルにタグ付けを行いXMLフラグメントを生成した後，外部関数処理を行う必要

外部関数処理へのアプローチ

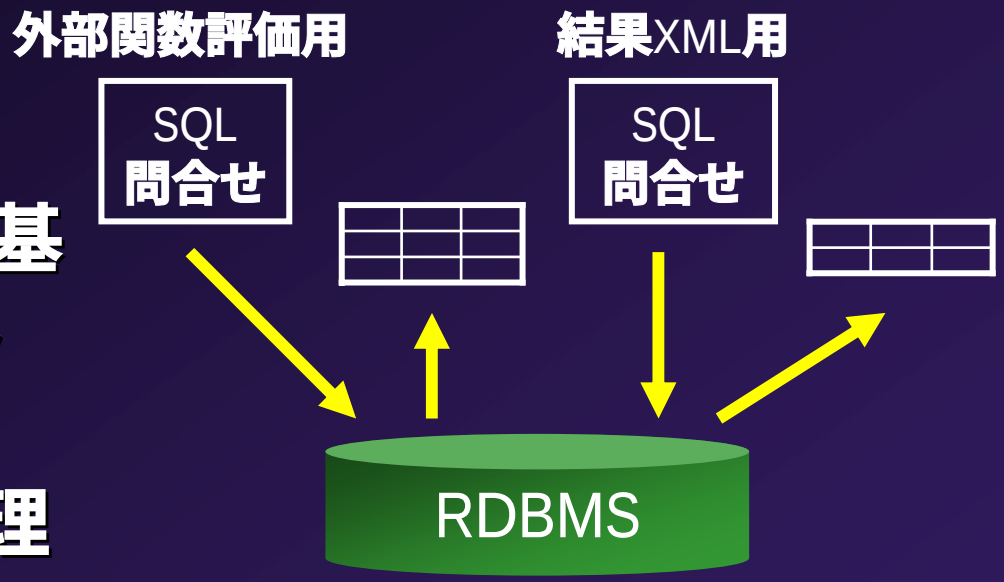
- XMLビューに関する先行研究の外部関数処理へのアプローチ
  - SilkRoute（AT & T）
    - 現時点で未検討
  - XPERANT（IBM）
    - RDBMSにXMLフラグメント生成機能と外部関数評価機能を追加する簡単なアイデアのみ

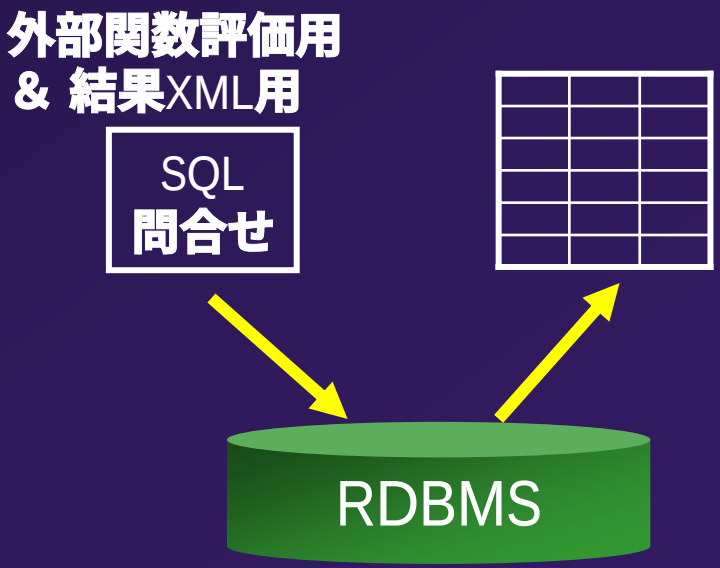
- **本研究のアプローチ**
  - ミドルウェアにおける外部関数処理方式
    - RDBMSに依存しない問合せ処理による高い汎用性の実現
  - 問合せ内容に応じた外部関数処理方式を提案
    - **分割処理方式，一括処理方式**

発表手順

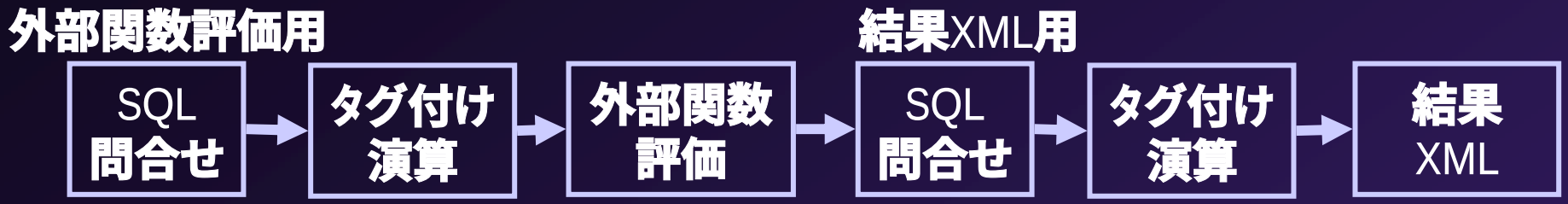
1. 背景と目的
2. 外部関数処理へのアプローチ
3. 提案処理方式
4. 問合せ変換
5. 評価実験
6. まとめ

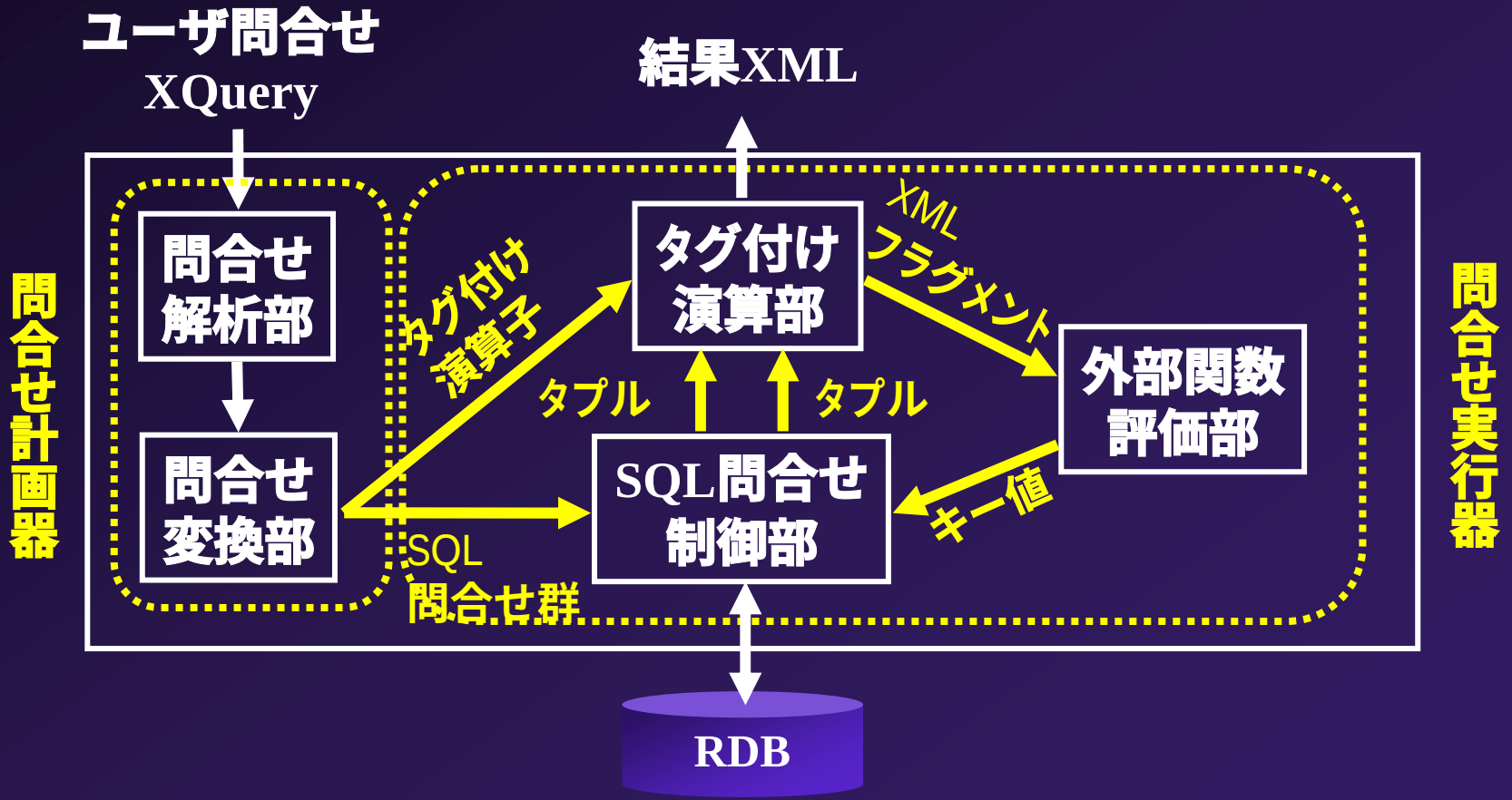
- 分割処理方式
  - 外部関数の評価結果に基づいて，タプルの絞込みが可能
  - 複数回のSQL問合せ処理によるコストが大きい
- 一括処理方式
  - 処理が簡単なため，オーバヘッドが小さい
  - 無駄なタプルを多く取得してしまうため，問合せ処理コストが大きい





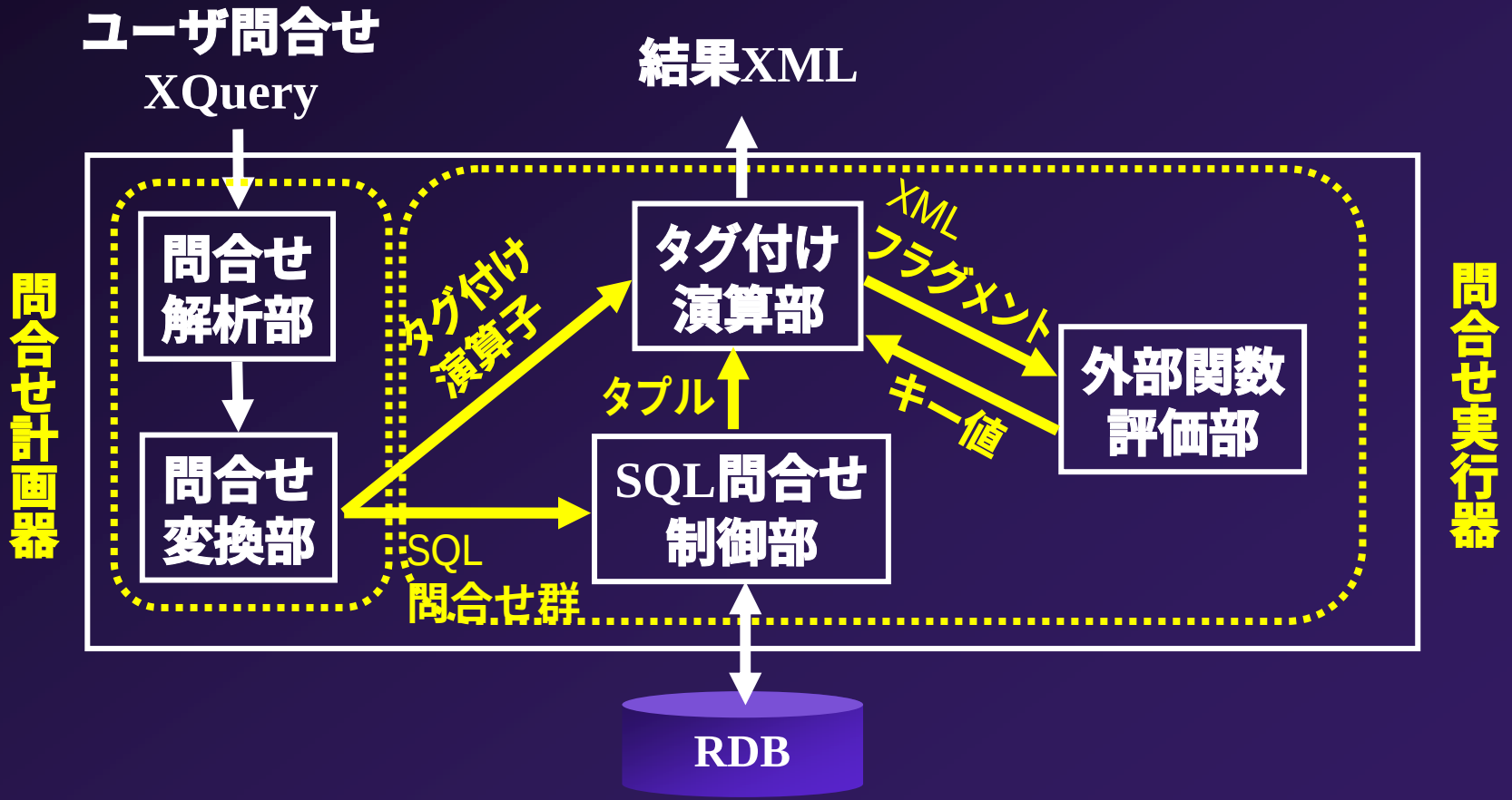
分割処理方式


外部関数評価用
SQL 問合せ
タグ付け 演算
外部関数 評価
結果XML用
SQL 問合せ
タグ付け 演算
結果 XML


ユーザ問合せ
XQuery
結果XML
問合せ 解析部
問合せ 変換部
タグ付け 演算部
外部関数 評価部
SQL問合せ 制御部
タグ付け 演算子
XML フラグメント
タプル
タプル
キー値
SQL 問合せ群
問合せ計画器
問合せ実行器
RDB

一括処理方式

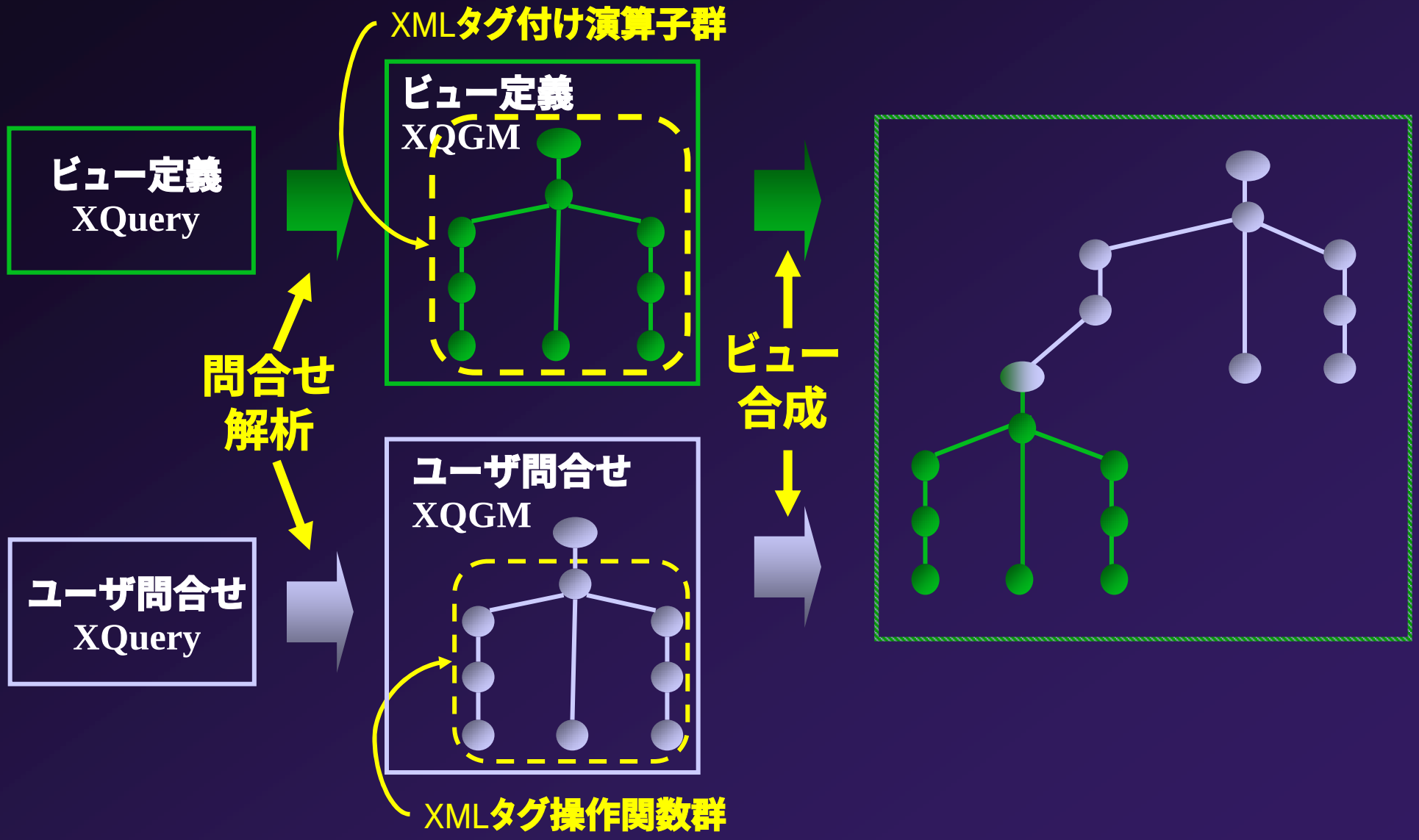
外部関数評価用
＆ 結果XML用



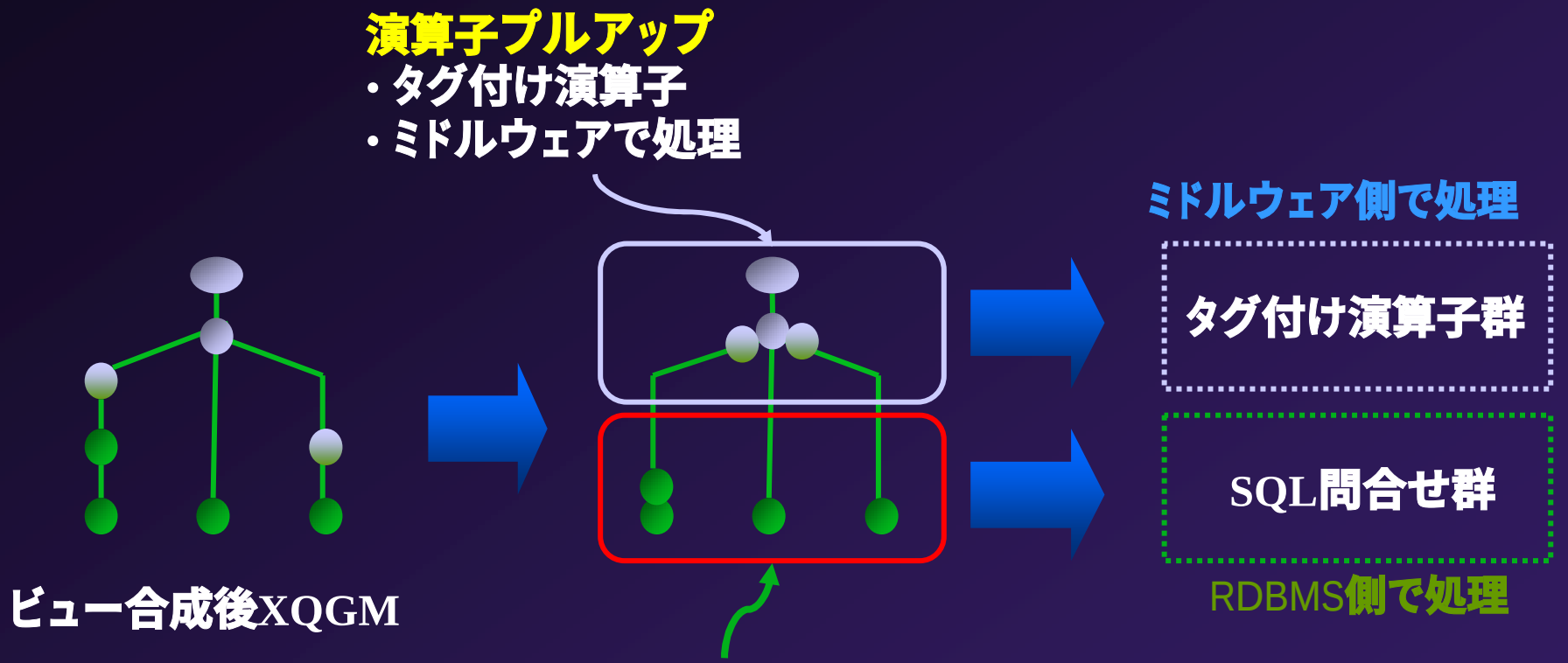
発表手順

1. 背景と目的
2. 外部関数処理へのアプローチ
3. 提案処理方式
4. **問合せ変換**
5. 評価実験
6. まとめ

XPERANTシステムの基本アプローチ

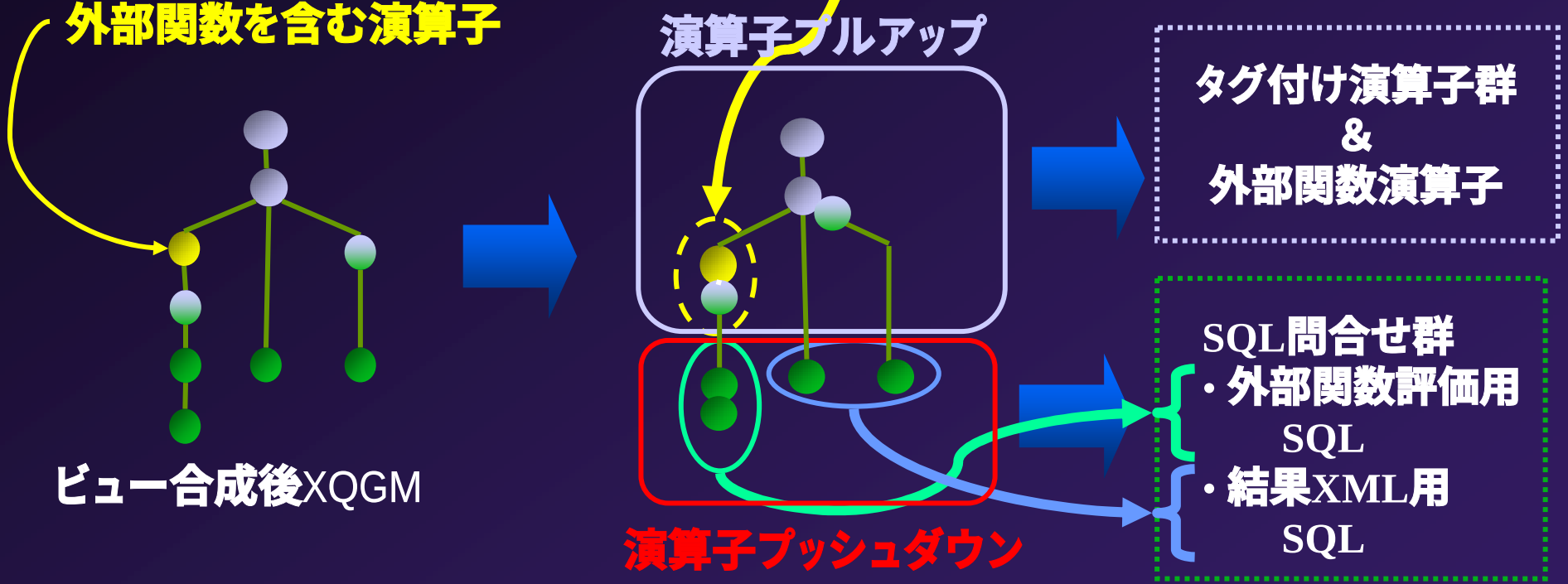


XPERANTシステムの基本アプローチ



XPERANTシステムの拡張

外部関数演算はプッシュダウン不可

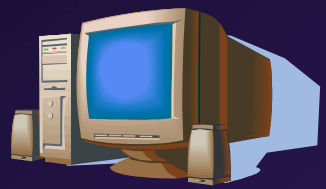


発表手順

1. 背景と目的
2. 外部関数処理へのアプローチ
3. 提案処理方式
4. 問合せ変換
5. 評価実験
6. まとめ

評価実験

- 各処理方式の異なる問合せ条件下の処理効率を比較
- 実験環境



•Celeron 300MHz
•メモリ 196MB
•OS Windows2000

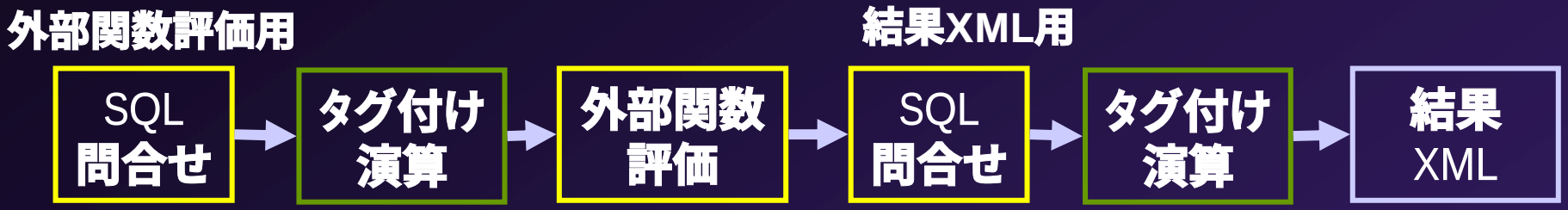
ミドルウェア
Java: JDK1.2
JDBC2.0

RDB
PostgreSQL7.1

- リレーショナルテーブル
  - 市 *X* タプル，施設 *4X* タプル，位置情報 *16X* タプル
- 外部関数
  - 選択率をパラメータ指定可能な形でシミュレート

実験内容

- 分割処理方式



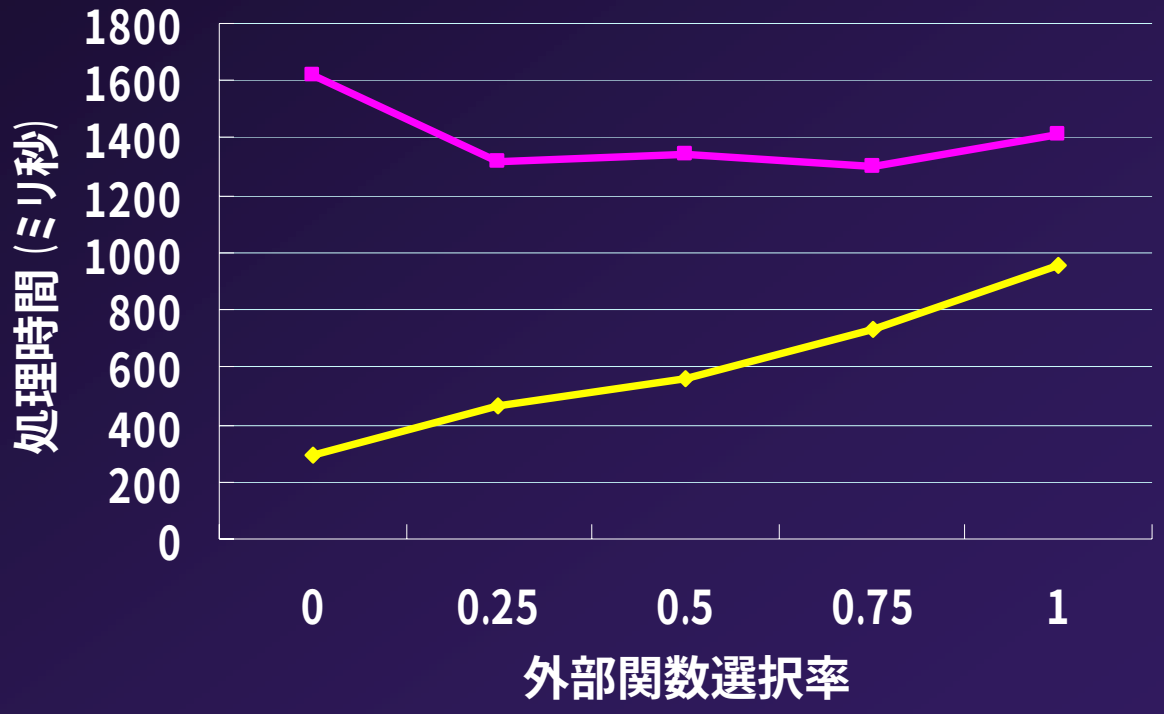
- 一括処理方式

共通の処理コスト　異なる処理コスト

- 各処理方式で異なる処理コストの合計を比較
  - リレーションタプル数と外部関数選択率をパラメータとして問合せ処理能力を比較

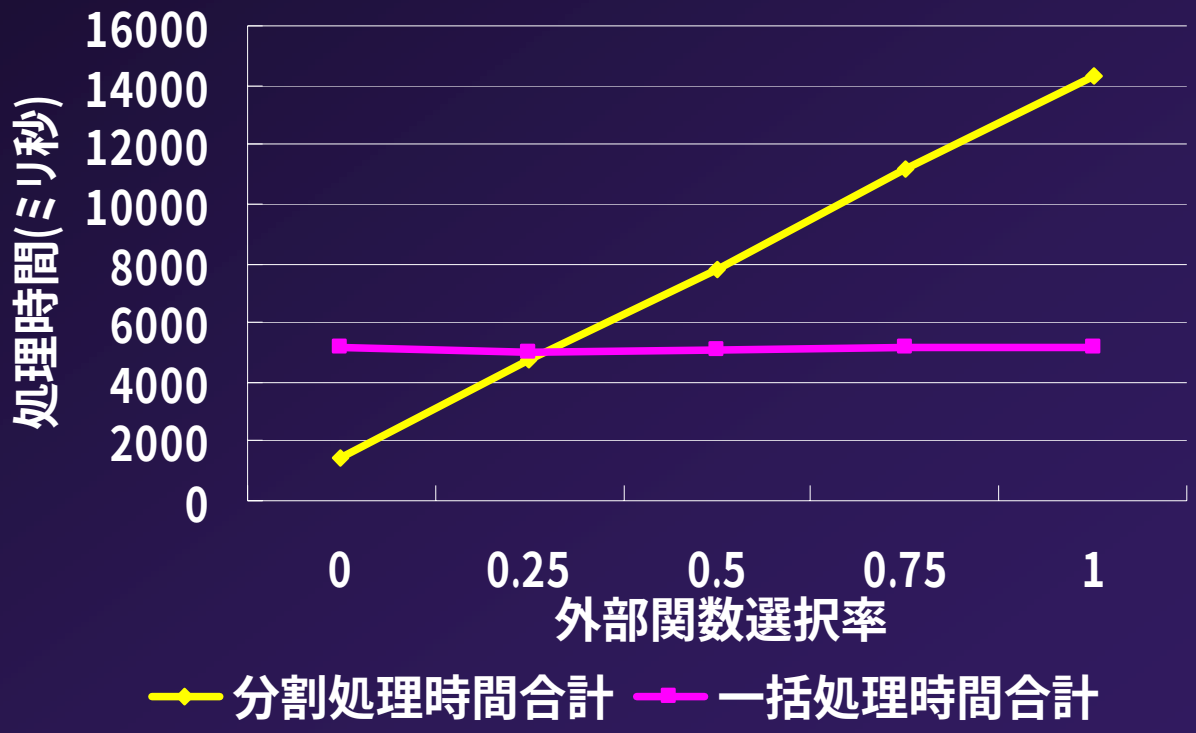
実験結果グラフ（1）

（市テーブル，施設テーブル，位置情報テーブル）
＝（１００タプル，４００タプル，１６００タプル）



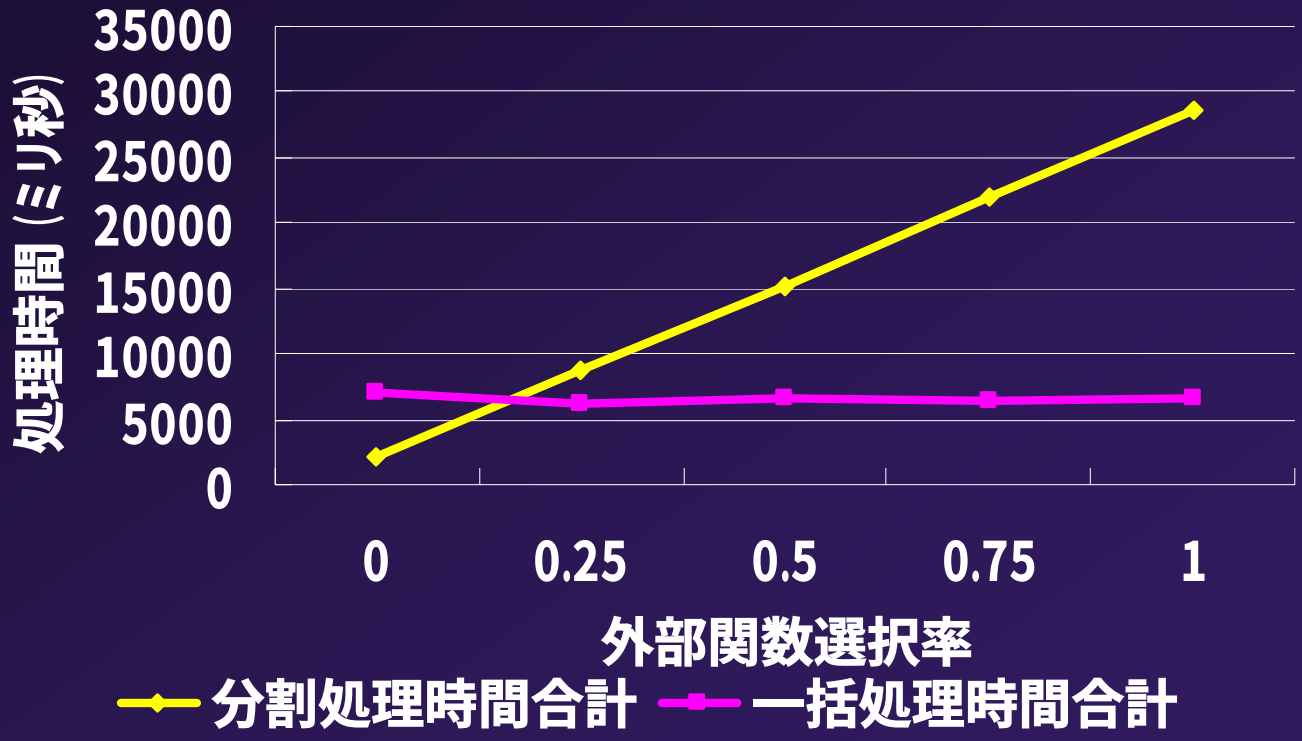
実験結果グラフ（2）

（市テーブル，施設テーブル，位置情報テーブル）
＝（５００タプル，２０００タプル，８０００タプル）



実験結果グラフ(3)

(市テーブル, 施設テーブル, 位置情報テーブル)
＝(７００タプル, ２８００タプル, １１２００タプル)



実験に関する考察

- 分割処理方式
  - 処理対象タプル数が少なく外部関数選択率が低い場合は処理コスト小
  - 処理対象タプル数が多く外部関数選択率が高い場合処理コスト大
    - 外部関数評価結果による大規模なキー値リスト生成コスト
    - 大量のキー値による選択条件を持つ結果XML生成問合せコスト
- 一括処理方式
  - 処理対象タプル数や外部関数選択率に関わらず安定
- 各処理方式で有効な問合せ処理条件は異なる

まとめ

- 2種類の外部関数処理方式の考案
  - 分割処理方式
  - 一括処理方式
- 各処理方式の評価実験

今後の課題

- 複合処理方式について検討
  - 問合せ内容に応じて2種類の処理方式の使い分け
- 実システムにおける詳細な評価