



Санкт-Петербургский государственный университет
Кафедра системного программирования

Оптимизация логических выражений внутри подзапросов в PosDB

Яков Сергеевич Кузин, группа 24.M41-мм

Научный руководитель: Г.А. Чернышев, ассистент кафедры ИАС

Санкт-Петербург
2025

Постановка задачи

- 1 Провести обзор внутренней архитектуры PosDB
- 2 Предложить метод оптимизации подзапросов с логическими выражениями. Разработать и внедрить прототип, реализующий этот метод
- 3 Рассмотреть практические примеры использования разработанного прототипа в различных сценариях работы с базами данных



- PosDB [CGG⁺17] — распределенная колоночная СУБД
- Предназначена для эффективного выполнения аналитических запросов
- Не имеет инструментов для оптимизации как логических выражений, так и подзапросов в целом

Архитектура:

- Основа — итераторная модель Volcano [Gra94]
- Перед выполнением запроса строится его план
- Логический план → физический план
- Операторы: позиционные и кортежные

Расширение функциональности:

- Создание нового оператора: добавление узла, физического оператора, системы преобразования первого во второго
- Внутренние контракты и система тестирования

Анализ запроса

- Коррелированный подзапрос требует неоднократного прохода по таблице employees
- Если почти все работники работают в отделе “Dep1”, это избыточно

Listing: Подзапрос после IN: исходный

```
SELECT S.name, S.age
FROM students AS S
WHERE S.name IN (
    SELECT E.name
    FROM employees AS E
    WHERE E.department = 'Dep1' OR
        E.age < S.age
);
```

Выделение некоррелированной части I

- Преобразование: выделение некоррелированной части
- Редкий полный проход по таблице employees
- Кеширование результата первого подзапроса

Listing: Подзапрос после IN: преобразованный

```
SELECT S.name, S.age
  FROM students AS S
 WHERE S.name IN (
      SELECT E.name
        FROM employees AS E
       WHERE E.department = 'Dep1'
    ) OR S.name IN (
      SELECT E.name
        FROM employees
       WHERE E.age < S.age
    );
```

Выделение некоррелированной части II

Преобразование применимо и к другим запросам, которые:

- Содержат как коррелированную, так и некоррелированную части
- Содержат подзапросы, образованными операторами IN, SOME, ALL, EXISTS

Структура преобразования зависит от запроса:

- Возможно дублирование некоррелированной части во втором подзапросе
- Возможна смена исходной логической связки (AND/OR) на обратную

Complex Subquery Predicate

- Прототип
- Три дочерних узла и два прокси-узла
- Движение данных: $ds \rightarrow csp \rightarrow lproxy \rightarrow left \xrightarrow{csp} rproxy \rightarrow right$

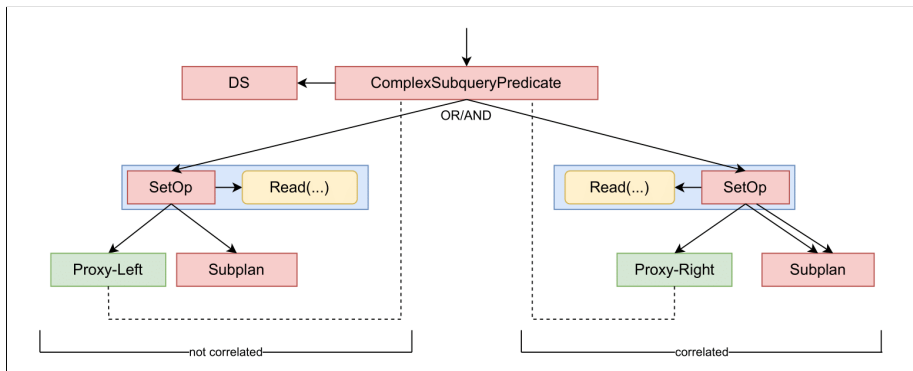


Рис.: Архитектура CSP

Описание:

- В каждом сравнении два запроса: основной и преобразованный
- Для преобразованного запроса два плана выполнения: с CSP и без него
- Сравнение на разных объемах данных

Исходные данные:

- Сгенерированные исходные данные для демонстрации потенциала прототипа

Эксперименты II

Listing: Q1

```
SELECT books1.title
FROM books1
WHERE books1.rating >= ALL (
    SELECT liked_books1.rating
    FROM liked_books1
    WHERE liked_books1.pages > 400 AND
        liked_books1.book_id != books1.id
);
```

Listing: Q2

```
SELECT books2.title
FROM books2
WHERE EXISTS (
    SELECT writers2.name
    FROM writers2
    WHERE writers2.age < 30 AND
        writers2.rating > 8 AND
        writers2.name = books2.author_name
);
```

Таблица: Результаты тестирования CSP + ALL

# Любимых книг	Q1' + CSP	Q1'	Q1	Ускорение
1000	12	3484	4076	287х, 340х
2000	13	6695	8089	515х, 622х
3000	13	9938	11975	764х, 921х

Таблица: Результаты тестирования CSP + EXISTS

# Писателей	Q2' + CSP	Q2'	Q2	Ускорение
500	720	47699	46813	66х, 65х
550	857	60316	60255	70х, 70х
600	1026	76285	76180	74х, 74х

- 1 Проведен обзор внутренней архитектуры PosDB
- 2 Предложен метод оптимизации подзапросов с логическими выражениями. Разработан и внедрен прототип, реализующий этот метод
- 3 Рассмотрены практические примеры использования разработанного прототипа в различных сценариях работы с базами данных

Список литературы



George A. Chernishev, Viacheslav Galaktionov, Valentin D. Grigorev, Evgeniy Klyuchikov, and Kirill Smirnov.

PosDB: A distributed column-store engine.

In Alexander K. Petrenko and Andrei Voronkov, editors, *Perspectives of System Informatics - 11th International Andrei P. Ershov Informatics Conference, PSI 2017, Moscow, Russia, June 27-29, 2017, Revised Selected Papers*, volume 10742 of *Lecture Notes in Computer Science*, pages 88–94. Springer, 2017.



G. Graefe.

Volcano — an extensible and parallel query evaluation system.

IEEE Trans. on Knowl. and Data Eng., 6(1):120–135, February 1994.

Listing: Q1'

```
SELECT books1.title
FROM books1
WHERE books1.rating >= ALL (
    SELECT liked_books1.rating
    FROM liked_books1
    WHERE liked_books1.pages > 400
) AND books1.rating >= ALL (
    SELECT liked_books1.rating
    FROM liked_books1
    WHERE liked_books1.book_id != books1.id
);
```

Listing: Q2'

```
SELECT books2.title
  FROM books2
 WHERE EXISTS (
    SELECT writers2.name
      FROM writers2
     WHERE writers2.age < 30 AND
           writers2.rating > 8
 ) AND EXISTS (
    SELECT writers2.name
      FROM writers2
     WHERE writers2.age < 30 AND
           writers2.rating > 8 AND
           writers2.name = books2.author_name
 );
```