Санкт-Петербургский государственный университет

Кафедра системного программирования

Оптимизация выполнения подзапросов в PosDB путем выделения из них некоррелированных частей

Яков Сергеевич Кузин, группа 24.М41-мм

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

Санкт-Петербург
2025

Постановка задачи

1. Провести обзор существующих методов оптимизации выполнения подзапросов
2. Разработать метод оптимизации выполнения подзапросов, основанный на выделении из них некоррелированных частей, и внедрить его в архитектуру PosDB
3. Осуществить сравнительный анализ производительности системы до и после внедрения предложенного метода оптимизации

PosDB

- PosDB [CGG+17] — распределенная колоночная СУБД
- Предназначена для эффективного выполнения аналитических запросов
- Не имеет инструментов для оптимизации подзапросов

Архитектура PosDB

Архитектура:

- Основа — итераторная модель Volcano [Gra94]
- Перед выполнением запроса строится его план
- Логический план $\rightarrow$ 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

Обзор

Подзапросами активно занимались ранее:

- Elhemali et al.: “Execution strategies for SQL subqueries” [EGLGJ07]
- Bellamkonda et al.: “Enhanced subquery optimizations in Oracle” [BAW+09]
- Zhao et al.: “Efficient Query Re-optimization with Judicious Subquery Selections” [ZZG23]
- Kim and Madden: “Optimizing Disjunctive Queries with Tagged Execution” [KM24]
- Bruno et al.: “Query Decorrelation in the Fabric Data Warehouse” [BGLJ25]

Ключевые моменты:

- Оптимизации дорогие и требуют переработки движка
- В некоторых случаях индустриальные практики неприменимы
- Не используется система позиций

Анализ запроса

- Коррелированный подзапрос требует неоднократного прохода по таблице employees
- Это может быть избыточно

Listing: Исходный запрос

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

Выделение некоррелированной части

- Преобразование: выделение некоррелированной части

Listing: Преобразованный запрос

```
SELECT S.name, S.salary
    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 AS E
            WHERE E.salary < S.salary
    );
```

Logical Predicate (LP)

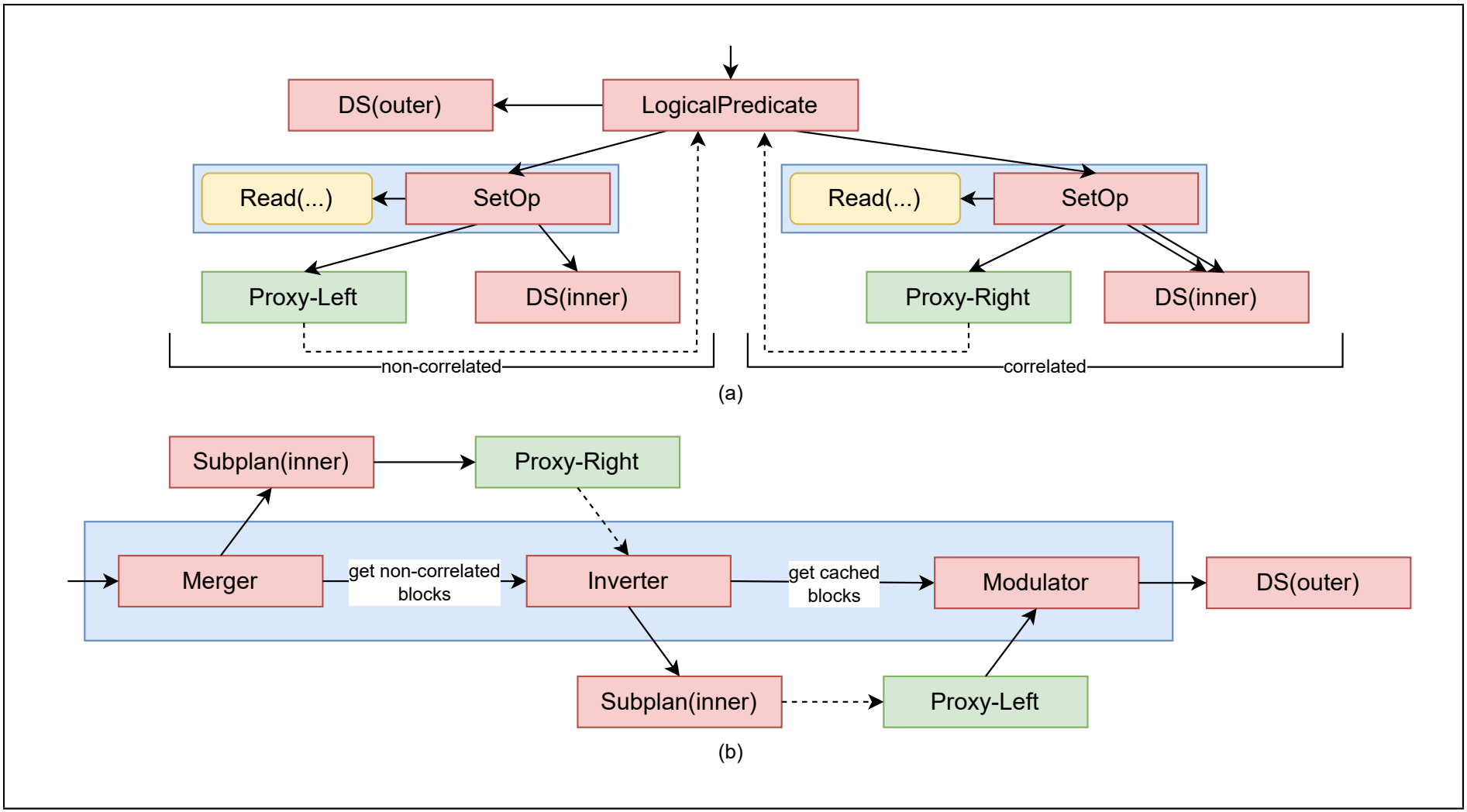

Рис.: (a) архитектура оператора LP; (b) движение данных

Эксперименты I

- Исходный запрос в PostgreSQL
- Преобразованный запрос в PostgreSQL
- Исходный запрос в PosDB
- Преобразованный запрос в PosDB с кэшированием некоррелированного подзапроса без LP
- Преобразованный запрос в PosDB с кэшированием некоррелированного подзапроса с LP

Listing: Исходный запрос

```
SELECT P.partkey
    FROM part AS P
    WHERE P.retailprice < SOME (
        SELECT 3 * L.extendedprice * L.discount * L.tax
            FROM lineitem AS L
            WHERE L.suppkey = X or P.size = L.quantity
    );
```

Эксперименты II

- Селективность коррелированного подзапроса — 40%
- Значения X обеспечивают селективность некоррелированного подзапроса в 20%, 40%, 60%, 80% и 99.9%

Listing: Преобразованный запрос

```
SELECT P.partkey
    FROM part AS P
    WHERE P.retailprice < SOME (
        SELECT 3 * L.extendedprice * L.discount * L.tax
            FROM lineitem AS L
            WHERE L.suppkey = X
    ) OR P.retailprice < SOME (
        SELECT 3 * L.extendedprice * L.discount * L.tax
            FROM lineitem AS L
            WHERE P.size = L.quantity
    );
```

Результаты экспериментов

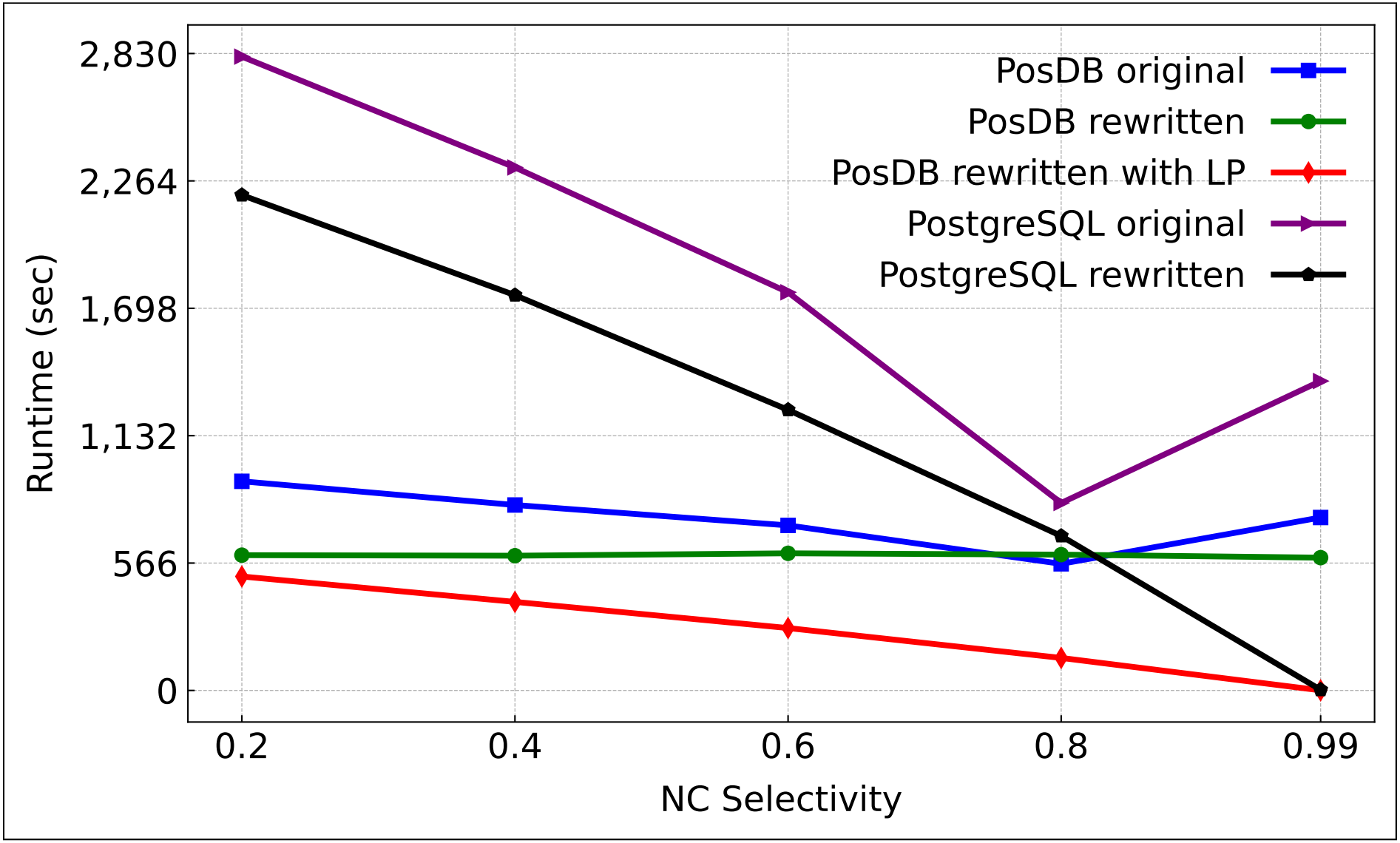

Рис.: Результаты экспериментов

Заключение

1. Проведен обзор существующих методов оптимизации выполнения подзапросов
2. Разработан метод оптимизации выполнения подзапросов, основанный на выделении из них некоррелированных частей. Данный метод успешно внедрен в архитектуру PosDB
3. Осуществлен сравнительный анализ производительности системы до и после внедрения предложенного метода оптимизации

Список литературы I

Srikanth Bellamkonda, Rafi Ahmed, Andrew Witkowski, Angela Amor, Mohamed Zait, and Chun-Chieh Lin.
Enhanced subquery optimizations in oracle.
Proc. VLDB Endow., 2(2):1366–1377, August 2009.

Nicolas Bruno, César Galindo-Legaria, and Milind Joshi.
Query decorrelation in the fabric data warehouse.
In *Companion of the 2025 International Conference on Management of Data*, SIGMOD/PODS '25, page 297–309, New York, NY, USA, 2025. Association for Computing Machinery.

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.

Mostafa Elhemali, César A. Galindo-Legaria, Torsten Grabs, and Milind M. Joshi.
Execution strategies for sql subqueries.
In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07, page 993–1004, New York, NY, USA, 2007. Association for Computing Machinery.

Список литературы II

G. Graefe.
Volcano — an extensible and parallel query evaluation system.
IEEE Trans. on Knowl. and Data Eng., 6(1):120–135, February 1994.

Albert Kim and Samuel Madden.
Optimizing disjunctive queries with tagged execution.
Proc. ACM Manag. Data, 2(3), May 2024.

Junyi Zhao, Huanchen Zhang, and Yihan Gao.
Efficient query re-optimization with judicious subquery selections.
Proc. ACM Manag. Data, 1(2), June 2023.

Приложение

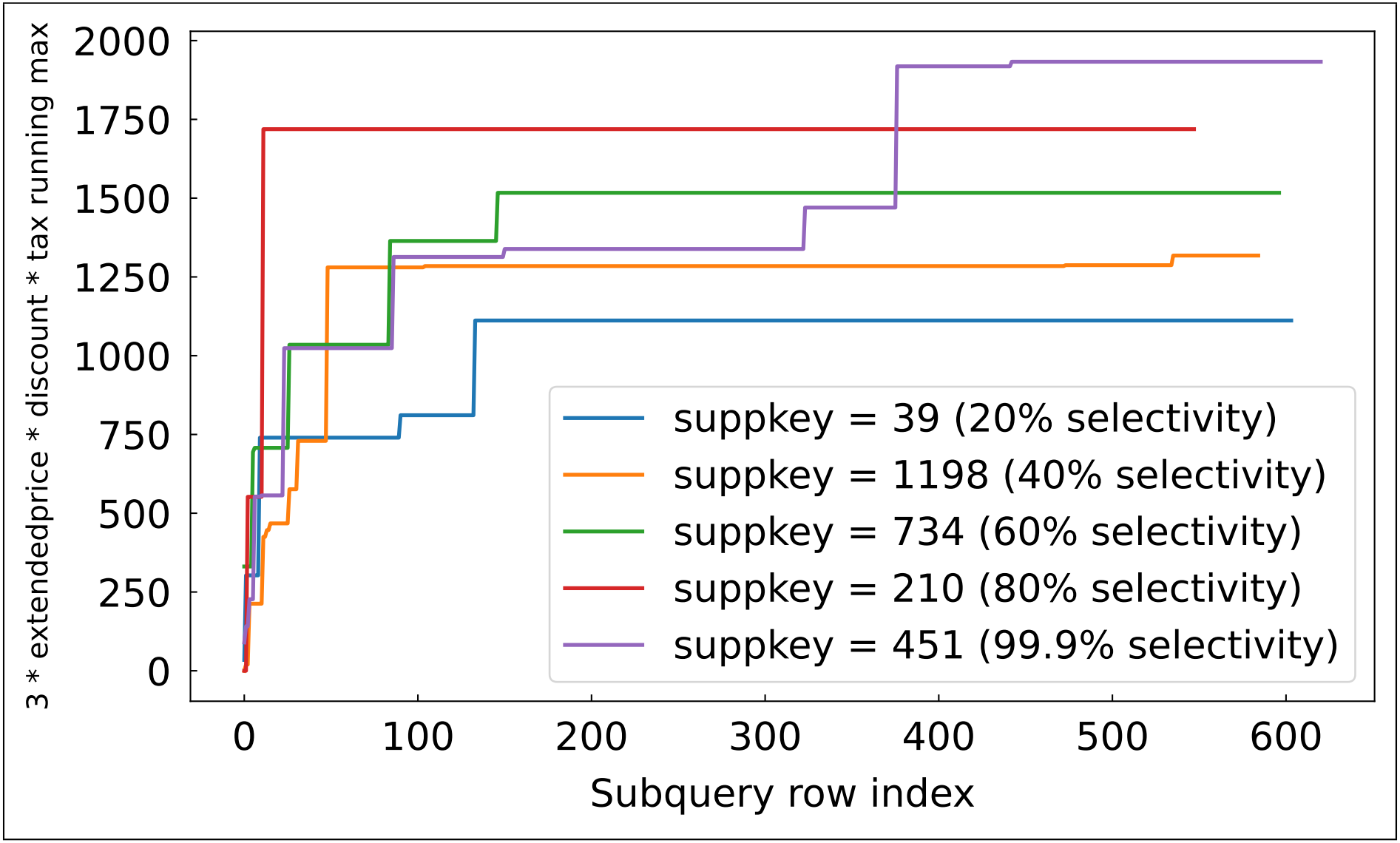

Рис.: Максимум значения выражения на разных этапах выполнения запроса