

Экспериментальный отчёт

Автор Дулетов Дмитрий

Дата 22.09.2020

Данные:

Графы: LUBM300, LUBM500, LUBM1M, mappingbased_properties_en (для краткости обозначаемый 8M)

Были использованы все запросы для каждого графа, изменённые для прочтения pyformlang.

Оборудование:

Замеры проводились на ПК с процессором Intel® Core™ i5-7400 CPU @ 3.00 GHz, оперативной памятью 8Gb DDR4.

ОС: Windows 10

RAM, выделенная для docker: 4800Mb

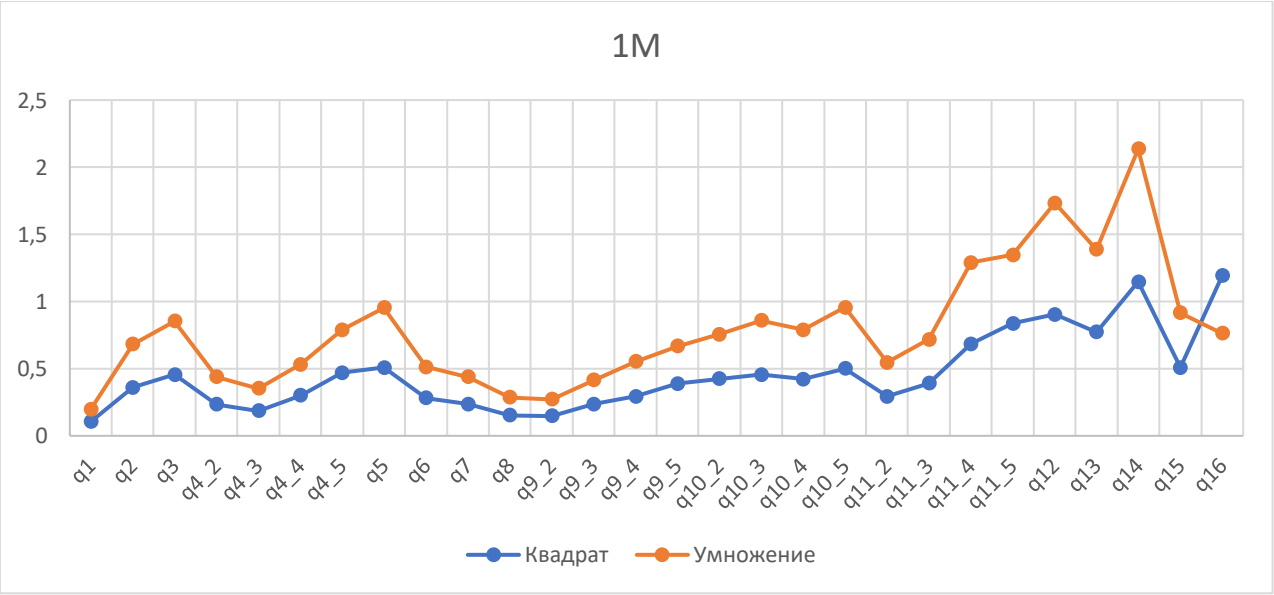
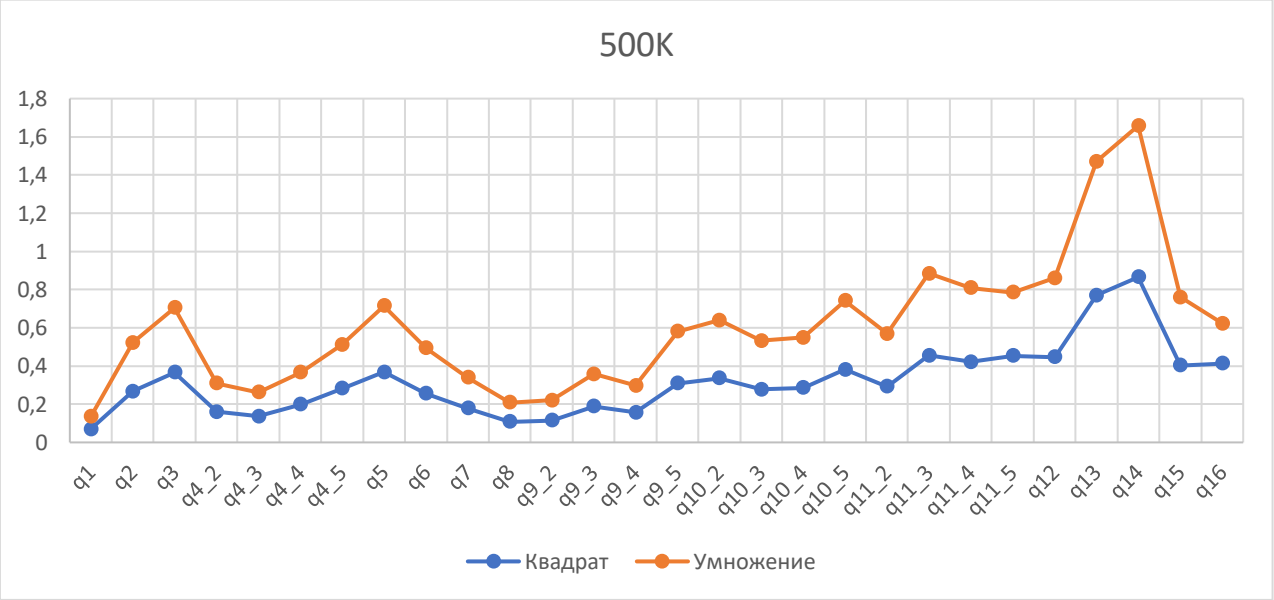
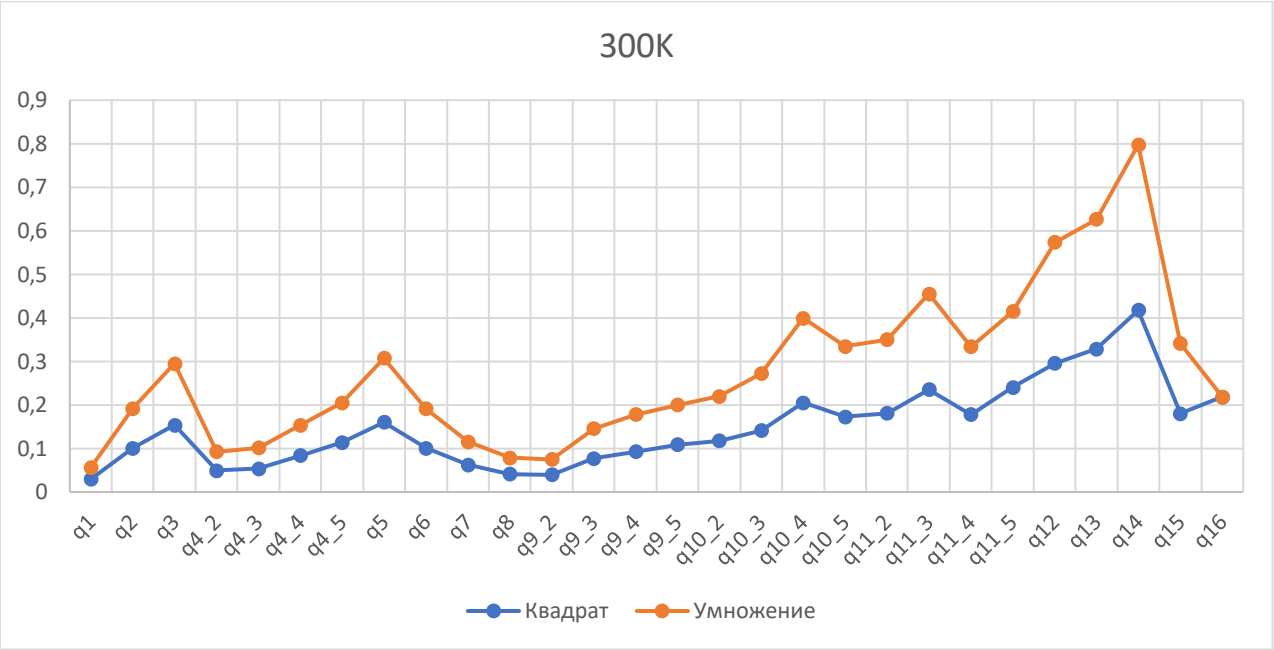
Метод:

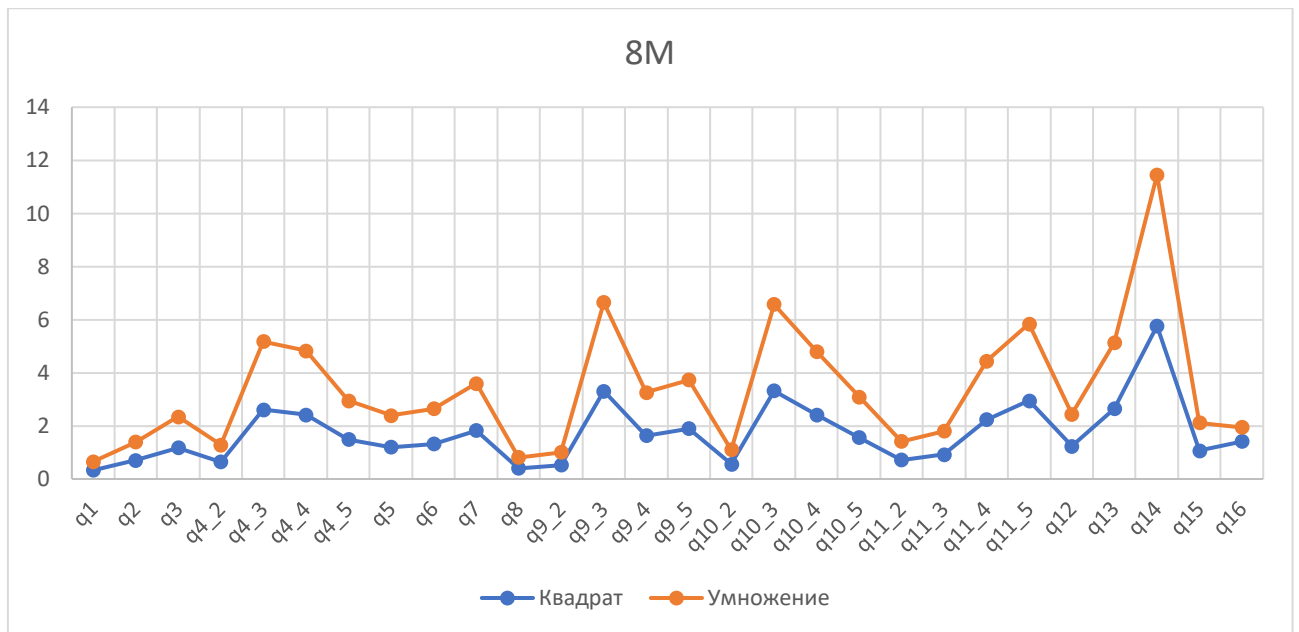
Все замеры проводились по 5 раз, бралось среднее значение. Для замеров использовалась функция `time.time()` с погрешностью 0.01 сек.

На графиках представлены результаты, сгруппированные по номеру q , для большей наглядности.

Результаты:

Сравнение эффективности реализации транзитивного замыкания при помощи возведения матрицы в квадраты (Квадрат) и домножения на матрицу смежности (Умножение).



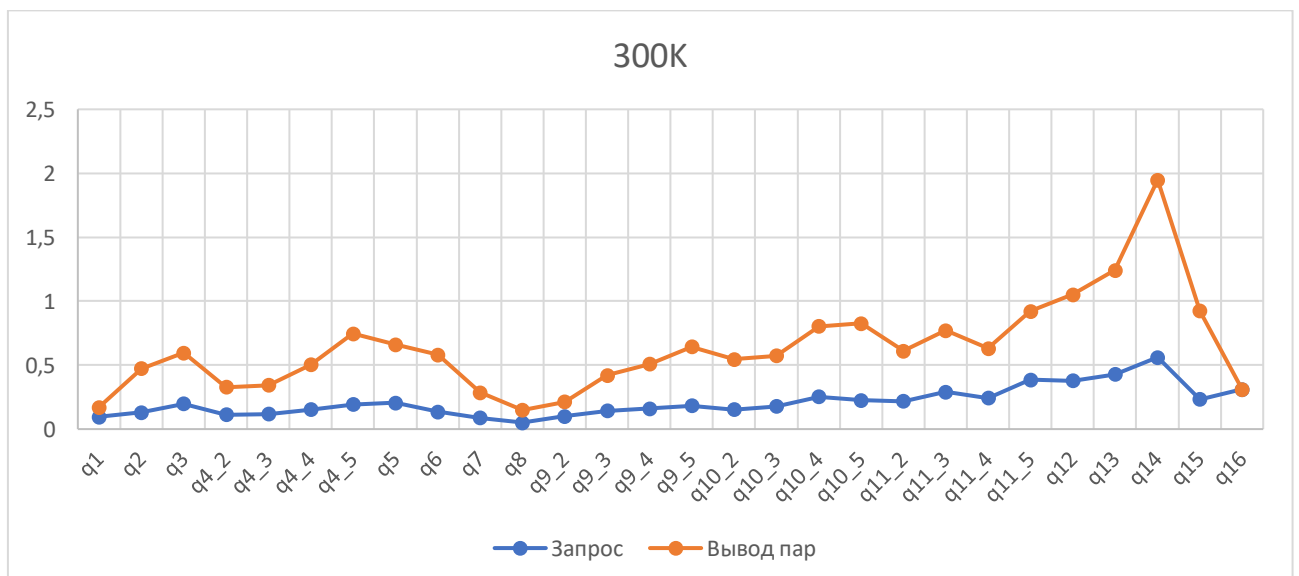


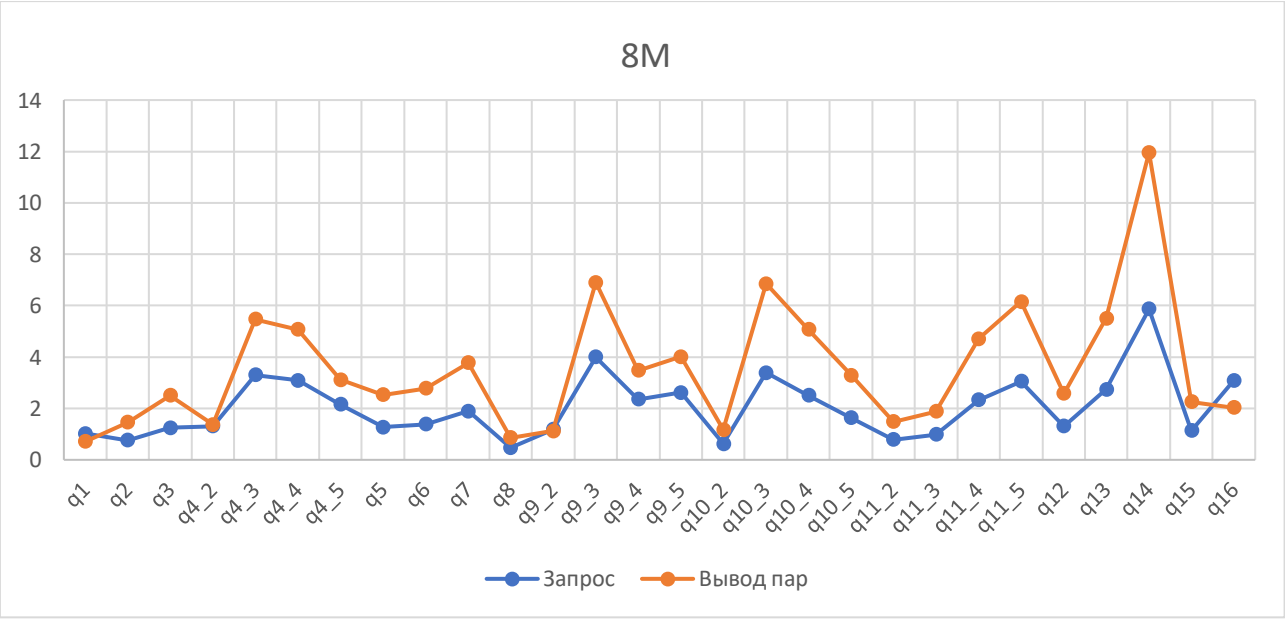
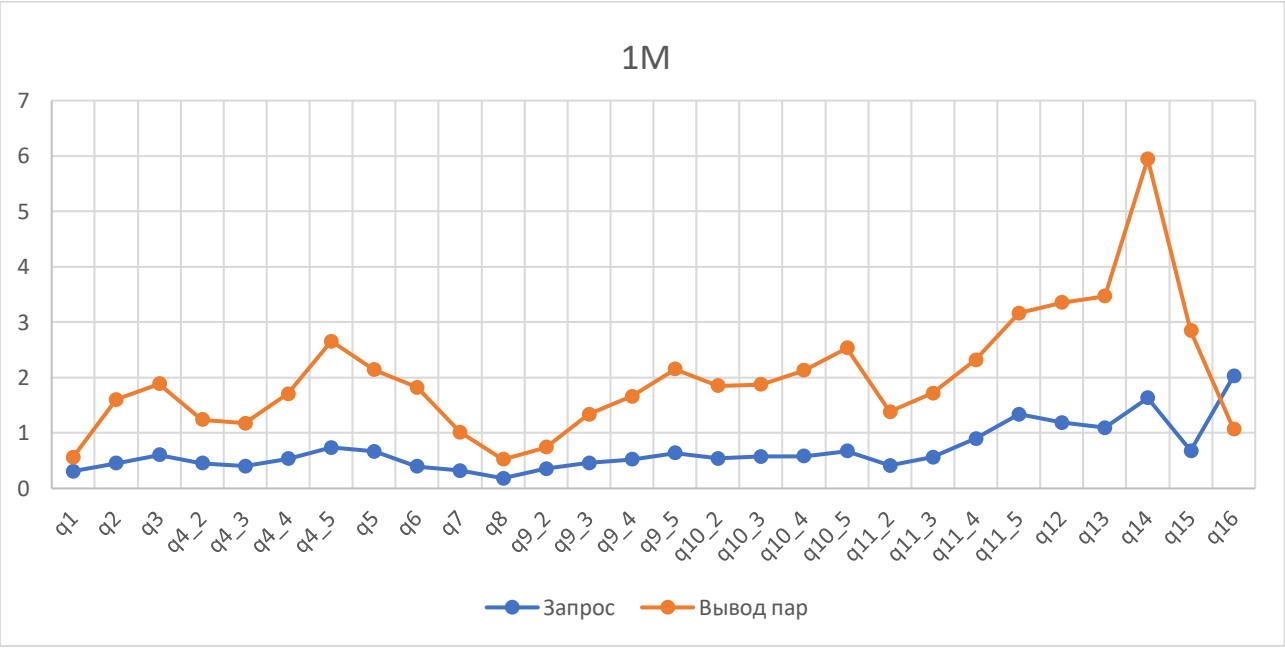
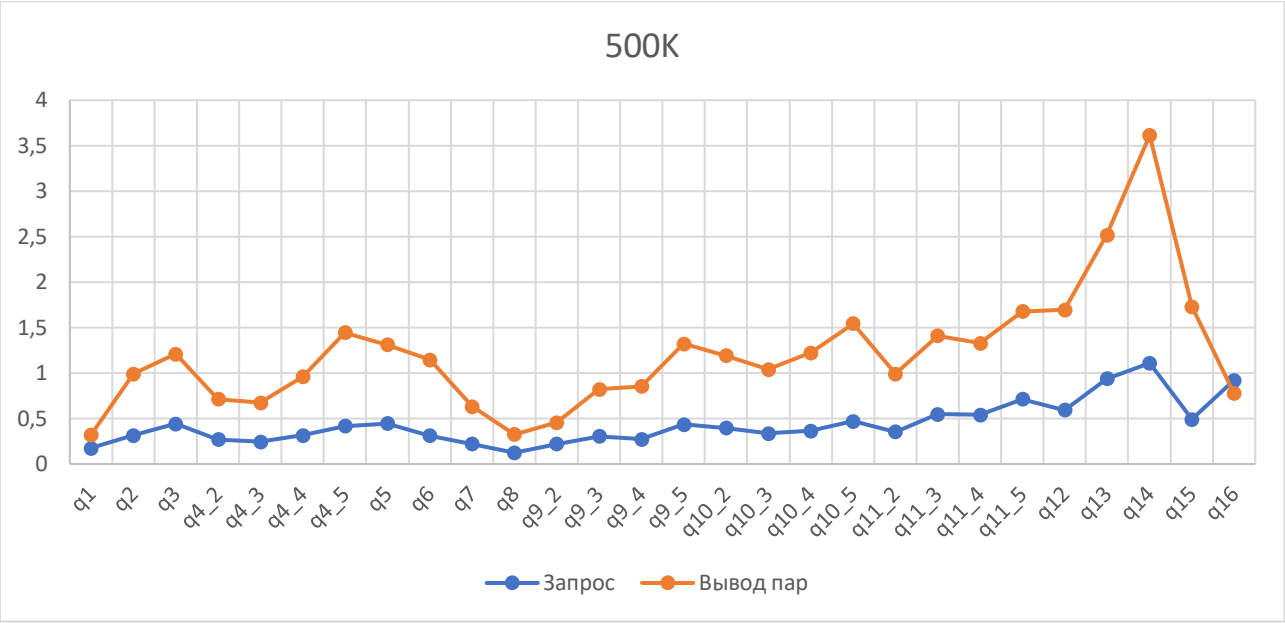
Вывод:

Как видно из представленных графиков, в моём случае возведение матрицы в квадрат показало лучшие результаты. Я думаю, это связано с тем, что для вычисления результата нужно меньше операций и, возможно, с тем, что операция возведения в квадрат лучше оптимизирована в библиотеке. Так же можно заметить, что с увеличением числа вершин время растёт как линейная функция.

Дополнительные результаты:

Время вычисления запроса (Запрос) и вывода пар (Вывод пар) для каждого графа





Контрольные цифры:

Таблица с количеством достижимых пар для некоторых запросов:

Граф	Запрос	Кол-во пар
LUBM300	q1_0	7797
LUBM300	q4_2_0	238449
LUBM300	q_15_0	257234
LUBM500	q1_0	12993
LUBM500	q4_2_0	397850
LUBM500	q_15_0	429174
LUBM1M	q1_0	26098
LUBM1M	q4_2_0	794757
LUBM1M	q_15_0	854228
mappingbased_properties	q1_0	3102
mappingbased_properties	q4_2_0	38879
mappingbased_properties	q_15_0	502933