

Экспериментальный анализ реализации

Алан Гамаонов
23 сентября 2020 г

Железо:

CPU: Intel Core i7-7700HQ, 2.80GHz x 8
RAM: 16 ГБ DDR5

О тестах:

Источник данных: refinedDataForRPQ
Количество итераций при измерении: 5
Точность: 6 знаков после запятой
Единицы измерения: Секунды

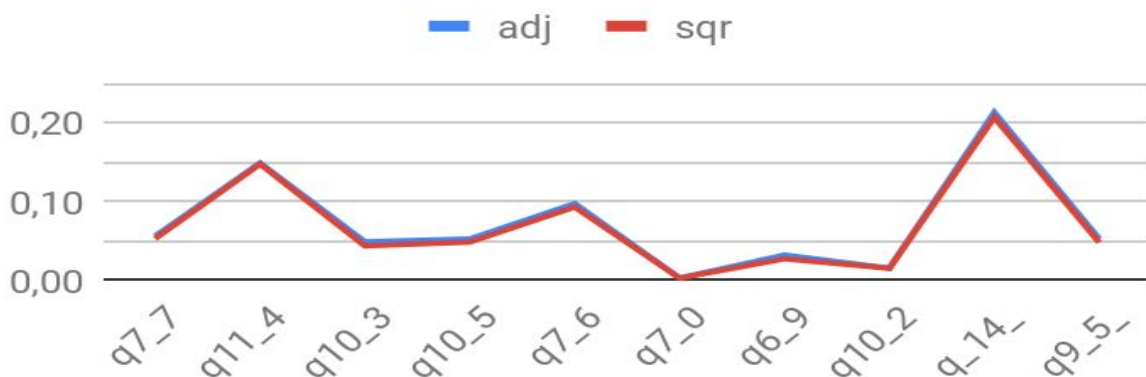
Замеры:

Сравнение эффективности реализации транзитивного замыкания посредством умножения на матрицу смежности и возведения в квадрат. Я брал первые 10 регекспов из соответствующей графу папки

Здесь и далее:

adj - умножение на матрицу смежности
sqr - возведение в квадрат

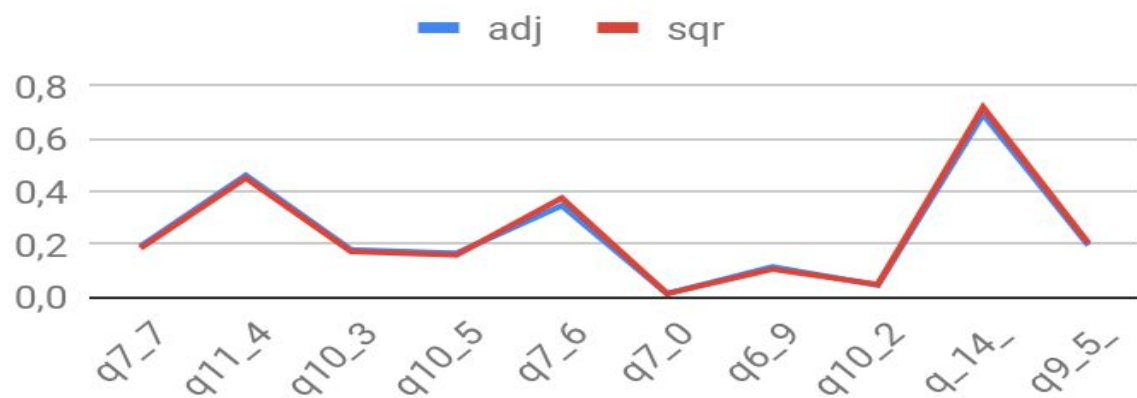
LUBM300



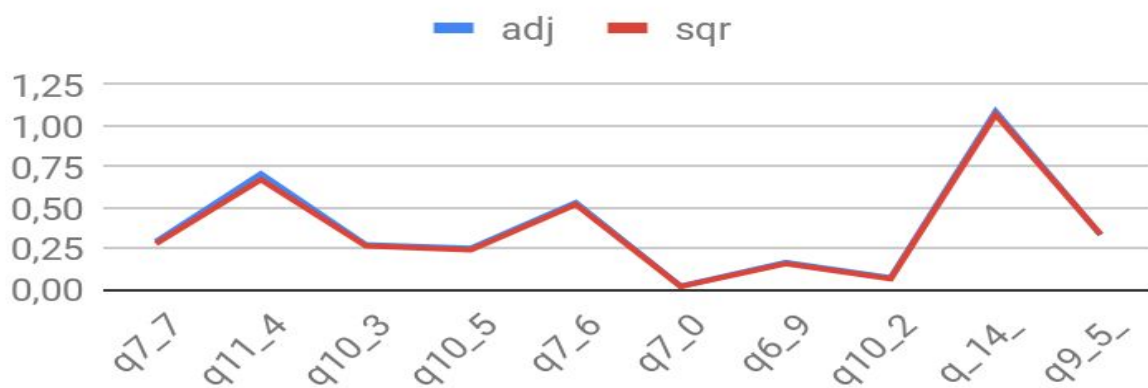
LUBM500



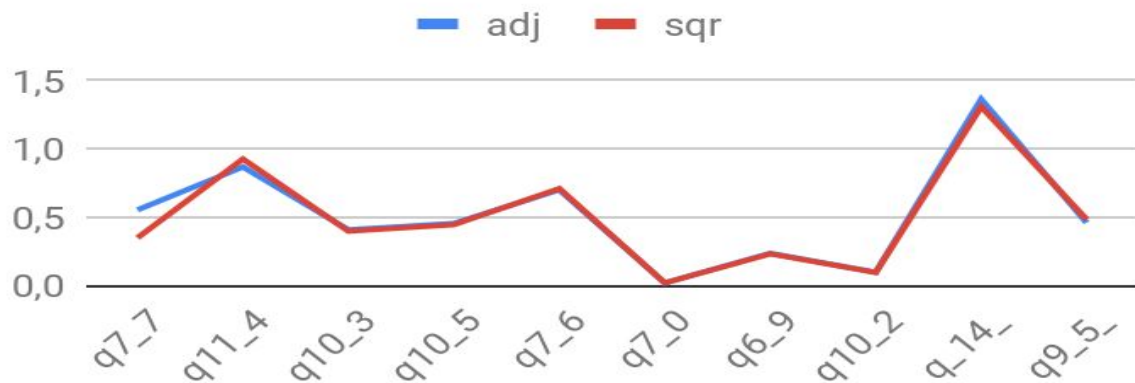
LUBM1M



LUBM1.5M



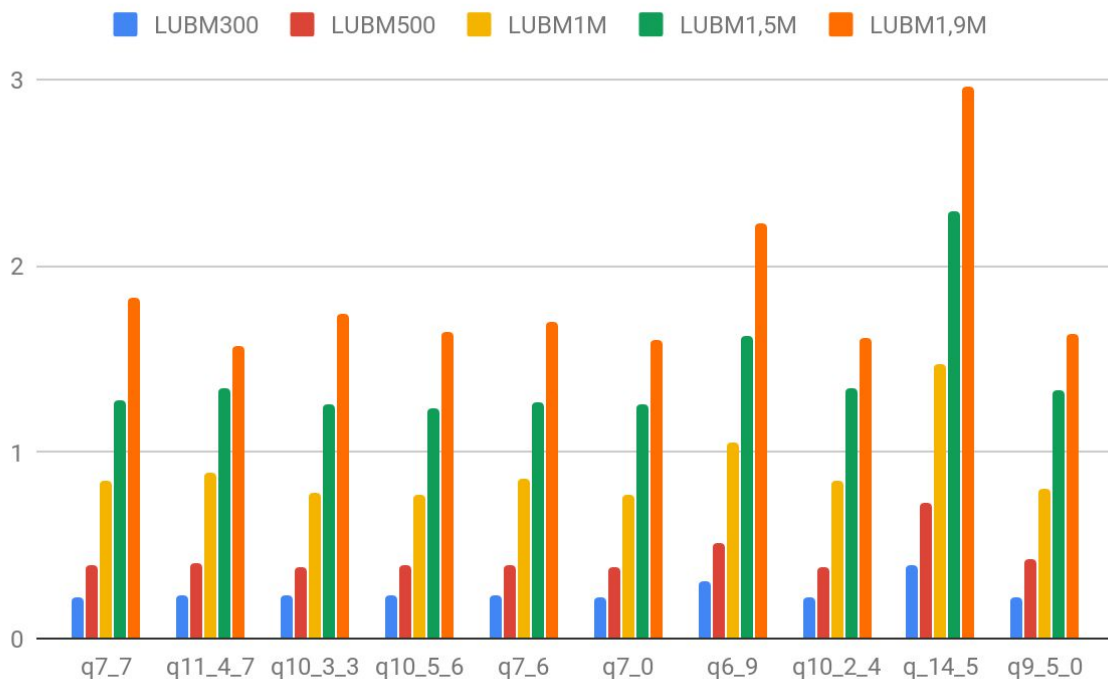
LUMB1.9M



Вывод:

Пусть на графиках не очень хорошо видно, но, тем не менее, в среднем, возведение в квадрат показывает лучший результат на больших графах, в то время как умножение на матрицу смежности быстрее работает на малых. Мне кажется, что маленький разброс по времени связан с мощностью моей машины и на более слабых устройствах разница будет ощутимее

График выполнения запросов:



Вывод же во всех случаях занял $<10^{-6}$ времени

Таблица контрольных значений:

(Количество достижимых пар для некоторых запросов)

Граф	Запрос	Количество пар
LUBM300	q6_9	233411
LUBM300	q10_4_1	395481
LUBM500	q1_1	74299
LUBM500	q10_3_9	292942
LUBM1M	q4_3_2	426362
LUBM1M	q11_4_0	2104492
LUBM1.5M	q10_5_0	2785826
LUBM1.5M	q9_4_8	1628864
LUBM1.9M	q_16_7	3554548
LUBM1.9M	q9_2_3	101212