Отчет по проведению эксперимента

Гордиенко Егор, 371 группа, 23.09.2020

В ходе эксперимента был произведен анализ производительности пересечения автоматов, реализованного через тензорное умножение булевых матриц. Для поиска транзитивного замыкания сравнивались два способа: возведение в квадрат и умножение на матрицу смежности.

В качестве исходных данных были взяты датасеты LUBM300, LUBM500, LUBM1M, LUBM1.5M, LUBM1.9M, где графы взяты без изменений, а регулярные запросы изменены для корректного прочтения библиотекой pyformlang.
К сожалению, на других датасетах замеры произвести не удалось (например, на geospecies при вычислении одной из матриц замыканий количество значений становилось слишком большим, и программа зависала).

Замеры производились на компьютере со следующими характеристиками:
процессор Inter(R) Core i7-7700HQ CPU @ 2.80GHz, 16.0 Gb RAM DDR4, Ubuntu 18.04 (WSL) under Windows 10.

Время, потраченное на вычисление тензорного произведения и транзитивного замыкания, бралось среднее за 5 подходов. Для замеров использовалась библиотека time, данные округлялись до миллисекунд.

В качестве контрольных цифр проверялось количество получившихся достижимых пар после замыкания.

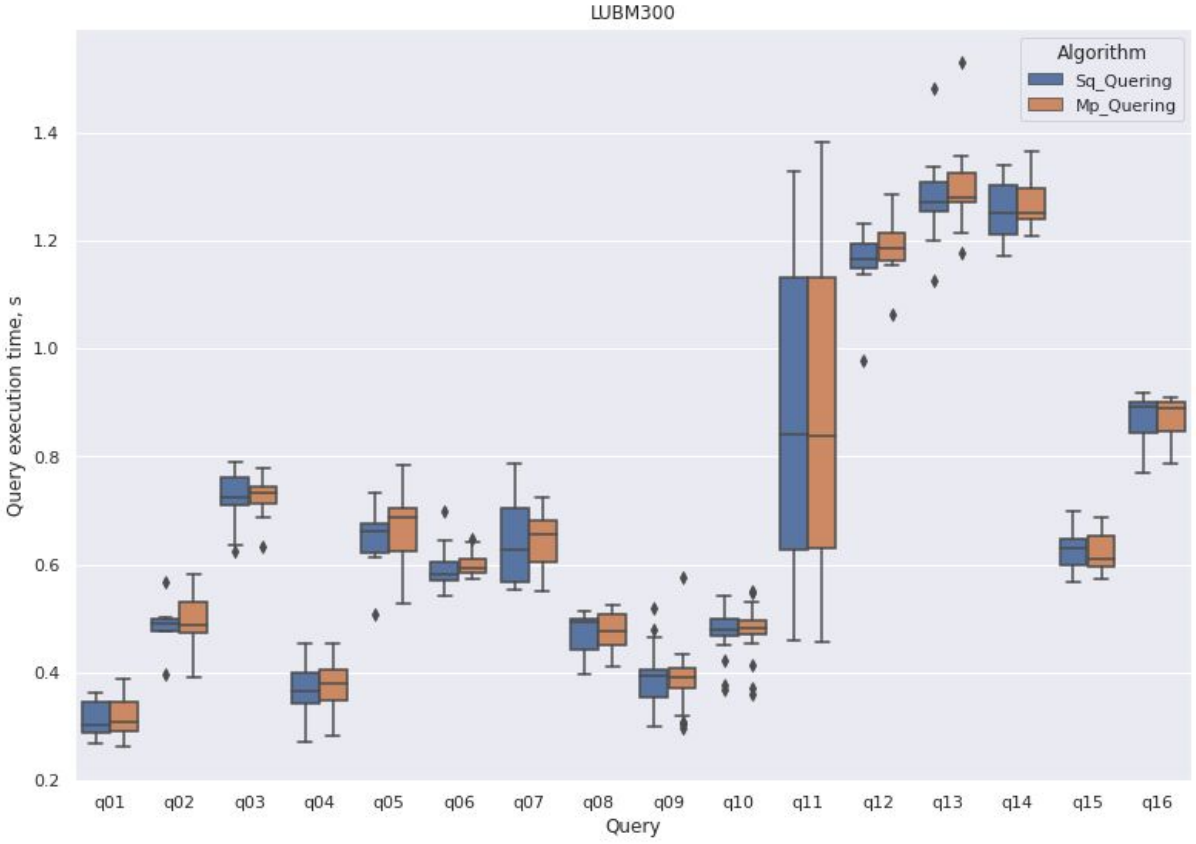
Данные собирались в следующем формате:

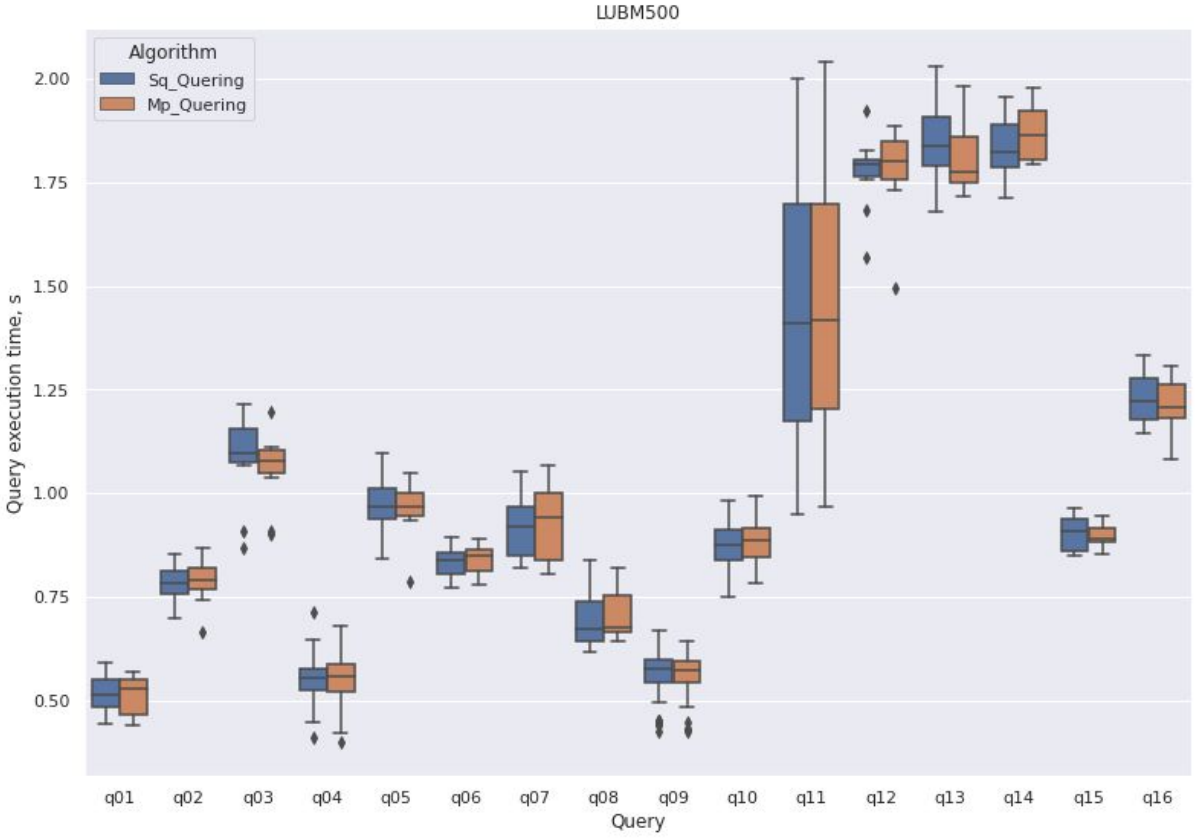
	Dataset	Regex	Sq_Pairs	Sq_Quering	Sq_Infer	Mp_Pairs	Mp_Quering	Mp_Infer
0	LUBM300	q10_2_0	129045	0.501	0.0	129045	0.472	0.0
1	LUBM300	q10_2_1	688617	0.524	0.0	688617	0.549	0.0
2	LUBM300	q10_2_2	132436	0.504	0.0	132436	0.550	0.0
3	LUBM300	q10_2_3	496394	0.541	0.0	496394	0.526	0.0
4	LUBM300	q10_2_4	58559	0.422	0.0	58559	0.413	0.0

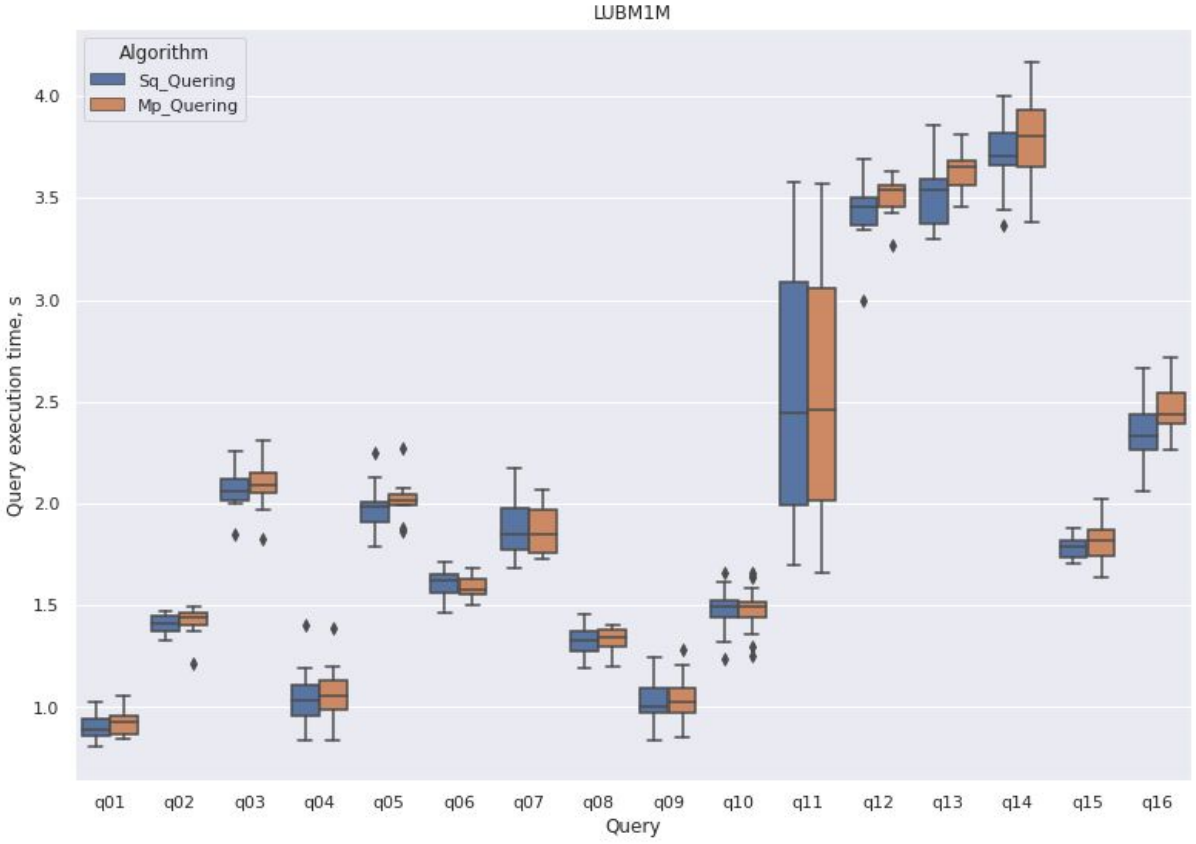
Sq - squaring - получение транзитивного замыкания возведением в квадрат.
Mp - multiplying - получение транзитивного замыкания домножением на матрицу смежности.
Pairs - контрольные цифры, совпадают у обоих способов вычисления замыкания.
Quering - время, потраченное на вычисление запроса (тензорное произведение + транзитивное замыкание), указано в секундах.
Infer - время, потраченное на вывод пар (терминал, кол-во ребер с терминалом), указано в секундах. Во всех запросах оно оказалось меньше миллисекунды.

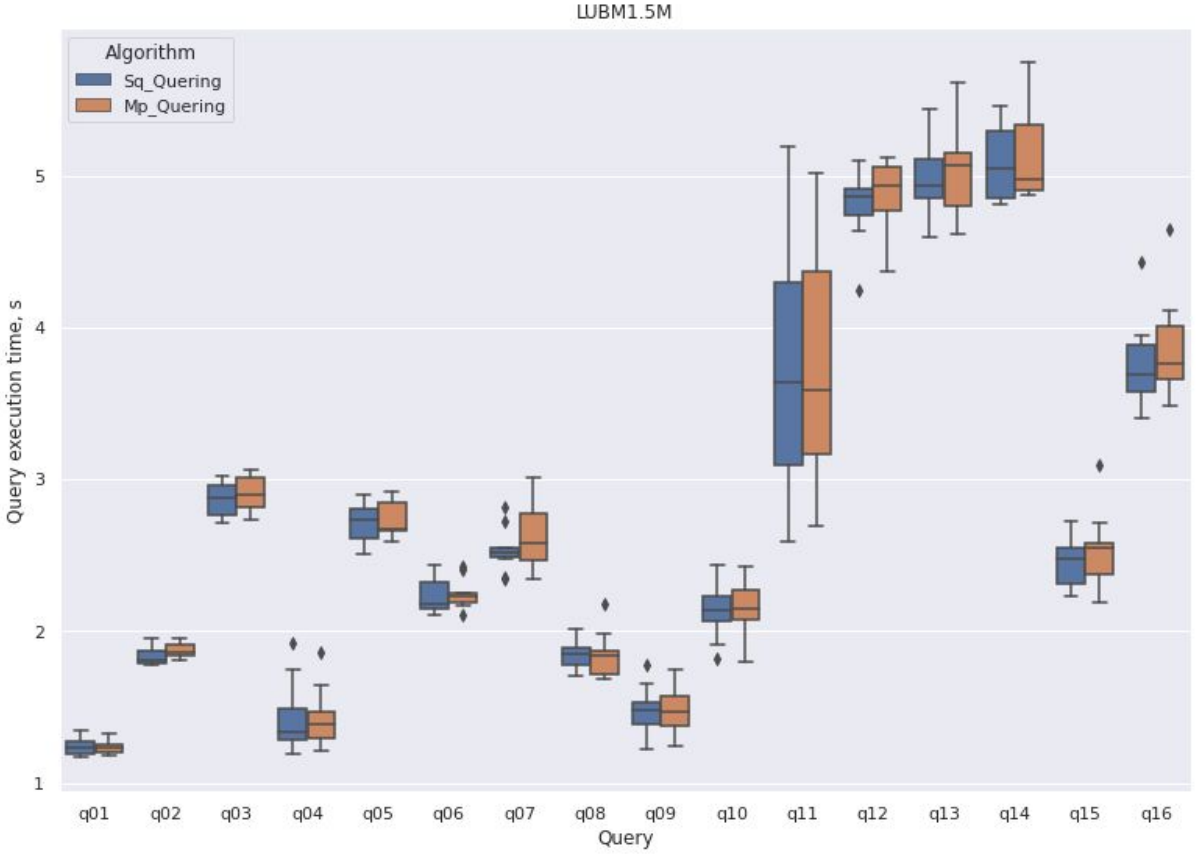
Для отображения на графике регулярные выражения были сгруппированы по первым индексам.

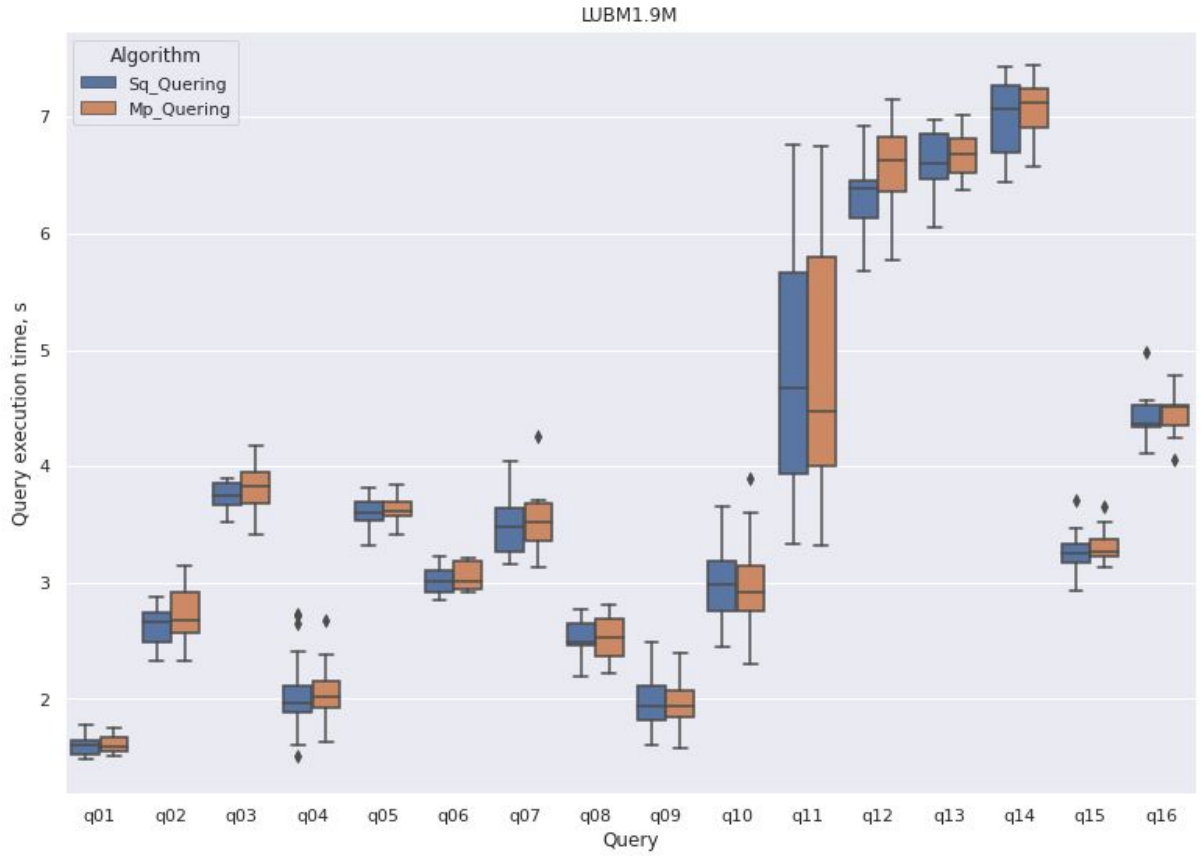
Графики представлены в форме boxplot:


LUBM300
Algorithm
Sq_Quering
Mp_Quering
Query execution time, s
1.4
1.2
1.0
0.8
0.6
0.4
0.2
q01
q02
q03
q04
q05
q06
q07
q08
q09
q10
q11
q12
q13
q14
q15
q16
Query


LUBM500
Algorithm
Sq_Quering
Mp_Quering
Query execution time, s
2.00
1.75
1.50
1.25
1.00
0.75
0.50
q01
q02
q03
q04
q05
q06
q07
q08
q09
q10
q11
q12
q13
q14
q15
q16
Query


LUBM1M
Algorithm
Sq_Quering
Mp_Quering
Query execution time, s
4.0
3.5
3.0
2.5
2.0
1.5
1.0
q01
q02
q03
q04
q05
q06
q07
q08
q09
q10
q11
q12
q13
q14
q15
q16
Query


LUBM1.5M
Algorithm
Sq_Quering
Mp_Quering
Query execution time, s
5
4
3
2
1
q01
q02
q03
q04
q05
q06
q07
q08
q09
q10
q11
q12
q13
q14
q15
q16
Query



Выводы: в результате эксперимента видно, что время вычисления запроса увеличивается пропорционально возрастанию вершин в исходном графе. Что касается двух различных способов получения транзитивного замыкания, то больших различий в производительности не видно - медианы находятся практически на одном уровне. В большинстве случаев возведение в квадрат немного выигрывает у домножения на матрицу смежности, но есть и обратные случаи. Возможно, это связано с тем, что данные недостаточно большие для того, чтобы увидеть явную разницу.