



# Реализация алгоритма поиска путей с контекстно-свободными ограничениями для графовой базы данных Neo4j

**Автор:** Погожельская Влада Владимировна, 18.Б11-мм  
**Научный руководитель:** к. ф.-м. н., доцент Григорьев С. В.

Санкт-Петербургский государственный университет  
Кафедра системного программирования

06 июня 2020г.

- **Подсчет количества треугольников в графе** — количество уникальных троек вершин  $u, v, w$  в неориентированном графе:  $(u, v), (u, w), (v, w) \in E$ , где  $E$  — множество ребер графа
- **Область применения:** анализ социальных сетей (обнаружение сообществ и степени сплоченности между ними, коэффициент кластеризации и транзитивности)
- Для решения задачи применяются методы линейной алгебры

**Целью** данной работы является сравнительный анализ существующих на данный момент решений задачи подсчета количества треугольников, основанных на линейной алгебре

**Задачи:**

- Выполнить обзор существующих решений задачи о подсчете количества треугольников, основанных на линейной алгебре
- Выполнить реализацию выбранного в результате обзора алгоритма подсчета количества треугольников в графе с помощью выбранной библиотеки
- Провести экспериментальное исследование реализованного алгоритма и уже существующих в выбранной библиотеке

# Обзор существующих решений

- $A$  — матрица смежности входного графа
- $U$  — верхнетреугольная,  $L$  — нижнетреугольная:  $A = L + U$
- $(*)$  — умножение матриц,  $(.*)$  — поэлементное умножение,  $(')$  — транспонирование

## Алгоритмы

- Базовая версия матричного алгоритма —  $ntri = \frac{1}{6} trace(A^3)$
- Burkhardt algorithm —  $ntri = \frac{1}{6} sum(sum((A^2). * A))$
- Cohen algorithm —  $ntri = \frac{1}{2} \sum_j (\sum_i ((L * U). * A))$
- Sandia algorithm —  $ntri = \sum_j (\sum_i ((U * U). * U))$
- SandiaDot algorithm —  $ntri = \sum_j (\sum_i ((L * U'). * L))$

## Библиотеки

GraphBLAS — стандарт для разработки графовых алгоритмов в терминах линейной алгебры

- **GraphBLAS Template Library (GBTL)** — реализация GraphBLAS для языка C++, поддерживающая реализация алгоритма Cohen
- **GraphBLAST** — реализация GraphBLAS для GPU, алгоритм не может быть реализован
- **SuiteSparse:GraphBLAS** — одна из самых полных реализаций стандарта, содержащая все вышеописанные алгоритмы подсчета количества треугольников, кроме базового

- Вопрос о практической применимости базового алгоритма не исследован до конца
- В качестве библиотеки для реализации был выбран SuiteSparse

Дан неорграф  $G = (V, E)$ ,  $A$  — матрица смежности размера  $N \times N$

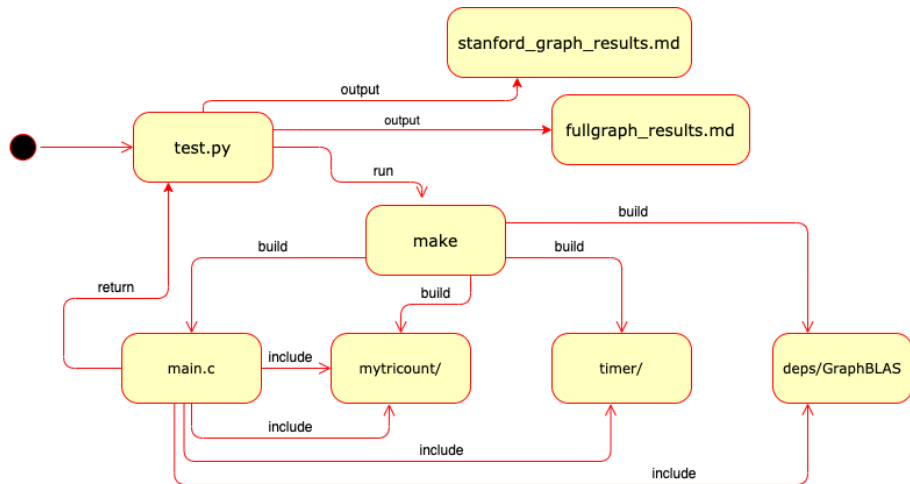
- $A^n[i][j]$  — количество различных путей в графе из  $i$  в  $j$  длины  $n$
- $A^3[i][i]$  — количество треугольников, проходящих через вершину  $i$
- Число треугольников в графе :  $\frac{1}{6} \text{trace}(A^3)$

# Особенности реализации с помощью SuiteSparse:GraphBLAS

- Функция перемножения двух разреженных матриц  $GrB\_m \times m$  использует полукольцо
- В SuiteSparse скалярное сложение в стандартном умножении матриц заменяется моноидом
- Полукольцо ( $GrB\_Semiring$ ) состоит из моноида и оператора «умножения», вместе эти операции определяют операцию матричного умножения
- В данной реализации был использован оптимизированный метод умножения матриц — Gustavson's method



# Архитектура решения



# Сравнительный анализ

Результаты замеров на полных графах, время указано в секундах

<b>N</b>	<b>Naive</b>	<b>Burkhardt</b>	<b>Cohen</b>	<b>Sandia</b>	<b>SandiaDot</b>
10	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
50	0.002	< 0.001	< 0.001	< 0.001	< 0.001
100	0.002	0.002	< 0.001	< 0.001	< 0.001
200	0.014	0.017	0.002	0.001	0.003
300	0.048	0.021	0.007	0.004	0.011
400	0.112	0.048	0.017	0.010	0.025
500	0.217	0.107	0.033	0.017	0.048
600	0.392	0.167	0.064	0.029	0.082
700	0.611	0.261	0.089	0.045	0.134
800	0.924	0.402	0.144	0.067	0.201
900	1.356	0.589	0.203	0.094	0.284
1000	1.915	0.820	0.251	0.129	0.382

# Сравнительный анализ

Результаты замеров на реальных графах, время указано в секундах

Name	nodes $\times 10^6$	edges $\times 10^6$	Naive	Burkhardt	Cohen	Sandia	SandiaDot
loc-brightkite_edges	0.06	0.21	5.880	0.050	0.030	0.018	0.016
amazon0302	0.40	2.35	2.220	0.111	0.063	0.034	0.035
roadNet-PA	1.09	1.54	0.351	0.045	0.075	0.053	0.032
amazon0505	0.41	2.44	28.143	0.480	0.270	0.095	0.111
soc-Epinions1	0.08	0.41	33.430	0.146	0.060	0.035	0.052
email-EuAll	0.27	0.36	NaN	0.333	0.111	0.019	0.040
loc-gowalla_edges	0.20	0.95	NaN	0.484	0.303	0.116	0.097
soc-Slashdot0902	0.08	0.50	50.605	0.168	0.075	0.039	0.057
soc-Slashdot0811	0.08	0.47	47.451	0.152	0.068	0.035	0.053

Были выполнены следующие задачи:

- Выполнен обзор существующих решений задачи о подсчете количества треугольников, основанных на линейной алгебре, и использованных для этого инструментов
- Выполнена реализация выбранного алгоритма подсчета количества треугольников в графе с помощью библиотеки GraphBLAS: SuiteSparse
- Проведено экспериментальное исследование реализованного алгоритма и уже существующих в данной библиотеке