



Обобщённая разреженная линейная алгебра и высокопроизводительный анализ графов: проблемы и перспективы Graph Analytics Club

Семён Григорьев

Санкт-Петербургский Государственный Университет

10 сентября 2025

- Доцент кафедры системного программирования Санкт-Петербургского Государственного Университета
- Руководитель исследовательской группы
- Области интересов
 - ▶ **Высокопроизводительная линейная алгебра** для анализа графов
 - ★ **Обобщённая:** матрицы и вектора параметризованы типом элемента, операции над ними могут быть заданы пользователем
 - ★ **Разреженная:** специализированные структуры для хранения матриц и векторов, специализированные алгоритмы для их обработки
 - ★ В том числе, с использованием **графических ускорителей**
 - ▶ **Высокопроизводительный анализ графов**



- Email: s.v.grigoriev@mail.spbu.ru
- GitHub: [gsvgit](#)
- Google Scholar: [Semyon Grigorev](#)
- DBLP: [Semyon V. Grigorev](#)

Как создать «достаточно универсальный»
фреймворк для разработки
высокопроизводительных (параллельных)
алгоритмов анализа графов?

- API для создания алгоритмов анализа графов на основе линейной алгебры
 - ▶ Различные операции над матрицами и векторами (разреженными)
 - ▶ Параметризация алгебраическими структурами: полукольцами, моноидами и т.д.

¹https://graphblas.org/docs/GraphBLAS_API_C_v2.1.0.pdf

²<https://graphblas.org/GraphBLAS-Pointers/>

³<https://zenodo.org/record/4318870/files/graphblas-introduction.pdf>

⁴<https://github.com/GraphBLAS/LAGraph>

⁵<https://graphblas.org/>

- API для создания алгоритмов анализа графов на основе линейной алгебры
 - ▶ Различные операции над матрицами и векторами (разреженными)
 - ▶ Параметризация алгебраическими структурами: полукольцами, моноидами и т.д.
- Позволяет выражать различные алгоритмы
 - ▶ Обход в ширину, поиск кратчайших путей, достижимость, ...
 - ▶ Подсчёт треугольников, PageRank, остовные деревья, кластеризация, ...
 - ▶ RPQ, CFPQ, ...

¹https://graphblas.org/docs/GraphBLAS_API_C_v2.1.0.pdf

²<https://graphblas.org/GraphBLAS-Pointers/>

³<https://zenodo.org/record/4318870/files/graphblas-introduction.pdf>

⁴<https://github.com/GraphBLAS/LAGraph>

⁵<https://graphblas.org/>

- API для создания алгоритмов анализа графов на основе линейной алгебры
 - ▶ Различные операции над матрицами и векторами (разреженными)
 - ▶ Параметризация алгебраическими структурами: полукольцами, моноидами и т.д.
- Позволяет выражать различные алгоритмы
 - ▶ Обход в ширину, поиск кратчайших путей, достижимость, ...
 - ▶ Подсчёт треугольников, PageRank, остовные деревья, кластеризация, ...
 - ▶ RPQ, CFPQ, ...
- Подробнее
 - ▶ The GraphBLAS C API Specification¹
 - ▶ GraphBLAS Pointers²
 - ▶ Introduction to GraphBLAS³
 - ▶ LAGraph⁴

¹https://graphblas.org/docs/GraphBLAS_API_C_v2.1.0.pdf

²<https://graphblas.org/GraphBLAS-Pointers/>

³<https://zenodo.org/record/4318870/files/graphblas-introduction.pdf>

⁴<https://github.com/GraphBLAS/LAGraph>

⁵<https://graphblas.org/>

Реализации GraphBLAS-подобных API

- SuiteSparse:GraphBLAS⁶ — эталон на чистом C⁷ (Intel, Nvidia)
- Huawei's GraphBLAS⁸ — частичная реализация на C++
- CombBLAS⁹ — распределённая, частичная реализация на C++
- GraphBLAST¹⁰ — поддержка GPGPU, Cuda C, частичная реализация
- Spla¹¹ — поддержка GPGPU, OpenCL C, частичная реализация
- GraphLily¹² — подмножество GraphBLAS на FPGA
- Обёртки для различных языков: Python, Rust, ...

⁶<https://github.com/DrTimothyAldenDavis/GraphBLAS>

⁷Ведётся работа над использованием GPGPU через Cuda

⁸<https://gitee.com/CSL-ALP/graphblas>

⁹<https://github.com/PASSIONLab/CombBLAS>

¹⁰<https://github.com/gunrock/graphblast>

¹¹<https://github.com/SparseLinearAlgebra/spla>

¹²GraphLily: Accelerating Graph Linear Algebra on HBM-Equipped FPGAs

- FalkorDB (ex RedisGraph)¹³ — графовая БД, основанная на SuiteSparse:GraphBLAS
- OneSparse¹⁴ — расширение PostgreSQL, позволяющее использовать разреженную линейную алгебру (для обработки графов)
- NetworkX¹⁵ — SuiteSparse:GraphBLAS для реализации некоторых алгоритмов анализа графов
- TenSQL^{16,17} — SQL + GraphBLAS

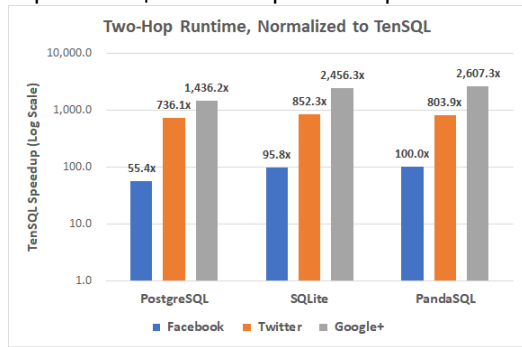
¹³<https://www.falkordb.com/>

¹⁴<https://onesparse.com/>

¹⁵<https://networkx.org/>

¹⁶<https://github.com/sandialabs/TenSQL>

¹⁷TenSQL: An SQL Database Built on GraphBLAS



И всё было бы хорошо, но . . .

Проблемы дизайна GraphBLAS¹⁹

- «Естественная» выразимость алгоритмов
 - ▶ BFS vs DFS¹⁸
- Необходимы специфичные оптимизации
 - ▶ Kernel Fusion для разреженных данных
 - ▶ В целом, нерегулярный параллелизм — это тяжело
- Неявные нули
- «Громоздкость» из-за ручных оптимизаций и необходимости тонких настроек
- (Не)универсальность
- ...

¹⁸Linear Algebraic Depth-First Search

¹⁹What did GraphBLAS Get Wrong?, John Gilbert, UC Santa Barbara, GraphBLAS BoF at HPEC, September 2022

$$\underbrace{M_1 + M_2}_{M_4} + M_3 \rightsquigarrow add(add(M_1, M_2), M_3)$$

- ✓ Stream Fusion — для «одномерных» данных
- ✓ XLA — для плотных данных
- ⚙ MLIR
 - ? Для разреженных вычислений в общем виде
 - ☹ Для обобщённых вычислений
 - ☹ Для GPGPU и других специализированных ускорителей

- Разреженный матрицы и вектора — не храним «нули»
- С API — сложно внятно описать на уровне типов, кто такие эти «нули»
 - ▶ Часто появляются выделенные значения («давайте считать, что `0: Int` не храним»)
 - ▶ Пользовательские функции: не достаточно чёткие типы, сложно обрабатывать крайние случаи
 - ▶ Переход между доменами: в одном домене выделенное значение не храним, а в другом должны хранить, потому что это «значимый» элемент, а выделенное значение другое

```
add (x: Int, y: Int): Int = x + y
```

Можем получить 0. Должны сохранить?

²⁰<https://github.com/lessthanoptimal/ejml/pull/145#issuecomment-888293732>

²¹<https://github.com/GraphBLAS/LAGraph/issues/28>

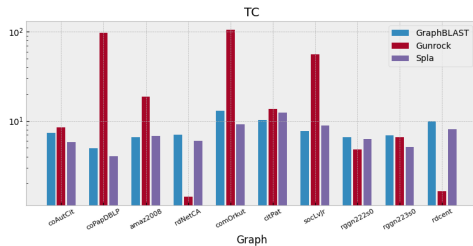
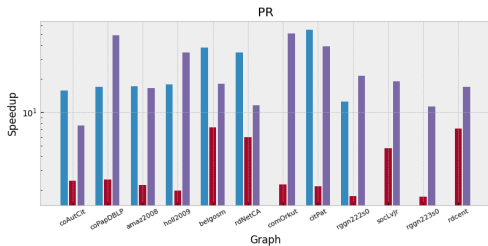
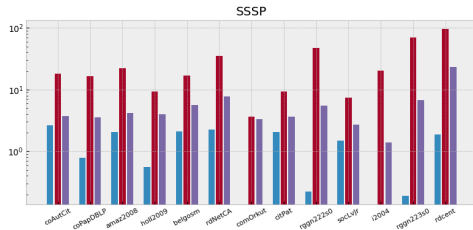
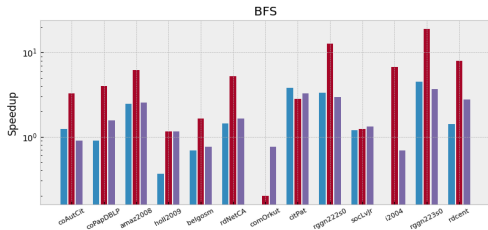
- Ручное слияние ядер (kernel fusion) — разрастается количество аргументов
 - ▶ Маска — аргумент умножения^{22,23} и не только
 - ▶ Маска может быть инвертированной или нет
- Поэлементные операции
 - ▶ `ewise_add`, `ewise_mult`, `mask`, ...²⁴
- Дескрипторы — средство тонкой ручной настройки операций
- ...

²² $mask(M_1 * M_2, M_3) \rightsquigarrow mult(M_1, M_2, M_3)$

²³ Да, как `multiply_add`

²⁴ Почему не `map2` или что-то аналогичное? Отчасти, потому что проблема с нулями.

(Не)универсальность



- Ускорение относительно LAGraph
- Графы из SuiteSparse Matrix Collection

- GraphBLAST и Spla — GraphBLAS-подобные
- Gunrock — BSP-like подход на основе BFS

- Обобщённая разреженная линейная алгебра — вполне рабочий инструмент высокопроизводительного анализа графов
 - ▶ FalkorDB, NetworkX, LAGraph, OneSparse, ...
- Пока хорошо работает только на одном узле, in memory, многоядерные CPU
 - ▶ SuiteSparse: GraphBLAS
- Не лишён недостатков
 - ▶ Если не пользуетесь готовыми прикладными алгоритмами, то думать о графах в терминах линейной алгебры — отдельный навык
 - ▶ Придумывать придумывать специальные полукольца и моноиды — нетривиальное занятие

Направления развития

- Использование (специализированных) ускорителей
 - ▶ В SuiteSparse:GraphBLAS ведётся работа по реализации некоторых ядер на Cuda
 - ▶ Spla использует GPGPU через OpenCL
- Разработка специализированных ускорителей для разреженной линейной алгебры
 - ▶ Расширения для RISC-V
 - ▶ Более «экзотические» решения²⁵
- Больше прикладных решений: анализ сетевого трафика²⁶, анализ кода²⁷, ...
- Усовершенствование API
 - ▶ Ведётся работа над C++ API²⁸
 - ▶ Попытки применить идеи из функционального программирования²⁹

²⁵ Dedicated Hardware Accelerators for Processing of Sparse Matrices and Vectors: A Survey

²⁶ Deployment of Real-Time Network Traffic Analysis Using GraphBLAS Hypersparse Matrices and D4M Associative Arrays

²⁷ Universal High-Performance CFL-Reachability via Matrix Multiplication

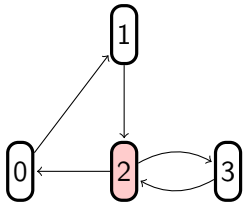
²⁸ GraphBLAS C++ API Specification v1.0

²⁹ Алгебраические типы для работы «нулями»

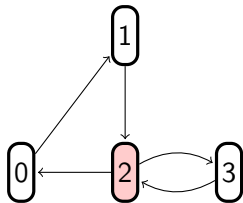
Графы для экспериментов

Name	Vertices	Edges	Avg Deg	Sd Deg	Max Deg
coAuthorsCiteseer	227.3K	1.6M	7.2	10.6	1372.0
coPapersDBLP	540.5K	30.5M	56.4	66.2	3299.0
amazon-2008	735.3K	7.0M	9.6	7.6	1077.0
hollywood-2009	1.1M	112.8M	98.9	271.9	11467.0
belgium_osm	1.4M	3.1M	2.2	0.5	10.0
roadNet-CA	2.0M	5.5M	2.8	1.0	12.0
com-Orkut	3.1M	234.4M	76.3	154.8	33313.0
cit-Patents	3.8M	33.0M	8.8	10.5	793.0
rgg_n_2_22_s0	4.2M	60.7M	14.5	3.8	36.0
soc-LiveJournal	4.8M	85.7M	17.7	52.0	20333.0
indochina-2004	7.4M	302.0M	40.7	329.6	256425.0
rgg_n_2_23_s0	8.4M	127.0M	15.1	3.9	40.0
road_central	14.1M	33.9M	2.4	0.9	8.0

Пример: обход в ширину



Пример: обход в ширину



Текущий фронт

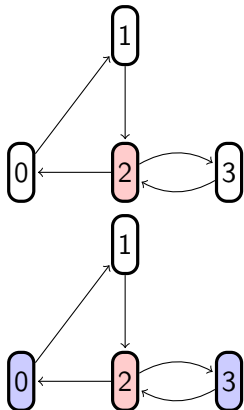
Матрица смежности

Новый фронт

Полукольцо

$$\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}$$

Пример: обход в ширину



Текущий фронт

Матрица смежности

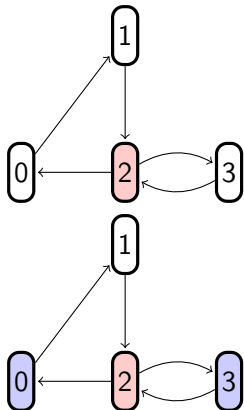
Новый фронт

$$\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}$$

Полукольцо

$$\begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & 0 \end{pmatrix}$$

Пример: обход в ширину



Текущий фронт

Матрица смежности

Новый фронт

Полукольцо

$$\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}$$
$$\begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & \cancel{1} & 0 \end{pmatrix}$$