



Анализ графов и разреженная линейная алгебра в экосистеме RISC-V

Рабочая группа “Развитие экосистемы ПО на RISC-V”

Семён Григорьев

Санкт-Петербургский Государственный Университет

30 мая 2025

Мат-Мех СПбГУ, лаборатория YADRO СПбГУ

- Семён Григорьев (s.v.grigoriev@spbu.ru)
- Владимир Кутуев (vladimir.kutuev@gmail.com)
- Родион Суворов
- Анастасия Мигунова

- Линейная алгебра, графы, GraphBLAS
- Реализации и примеры использования GraphBLAS
- SuiteSparse:GraphBLAS на RISC-V
- Направления работ, выводы

- **Анализ больших графов:** графовые БД, анализ кода, поиск уязвимостей, анализ трафика, анализ транзакции, банковская аналитика, социальные сети. . .
 - ▶ Важна производительность
 - ▶ Разнообразные алгоритмы

- **Анализ больших графов:** графовые БД, анализ кода, поиск уязвимостей, анализ трафика, анализ транзакции, банковская аналитика, социальные сети. . .
 - ▶ Важна производительность
 - ▶ Разнообразные алгоритмы
- Путь к унифицированной параллельной обработке графов
 - ▶ Граф \iff **матрица** смежности
 - ▶ Метки на рёбрах \iff **полукольца**, моноиды, . . .
 - ▶ Линейная алгебра \iff **параллелизм** по данным

- **Анализ больших графов:** графовые БД, анализ кода, поиск уязвимостей, анализ трафика, анализ транзакции, банковская аналитика, социальные сети. . .
 - ▶ Важна производительность
 - ▶ Разнообразные алгоритмы
- Путь к унифицированной параллельной обработке графов
 - ▶ Граф \iff **матрица** смежности
 - ▶ Метки на рёбрах \iff **полукольца**, моноиды, . . .
 - ▶ Линейная алгебра \iff **параллелизм** по данным
- **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, ПЛИС**

- API для создания алгоритмов анализа графов на основе разреженной обобщённой 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://graphblas.org/>

- API для создания алгоритмов анализа графов на основе разреженной обобщённой 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, ...
 - ▶ ...

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://graphblas.org/>

GraphBLAS⁴

- API для создания алгоритмов анализа графов на основе разреженной обобщённой 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, ...
 - ▶ ...
- Подробнее
 - ▶ The GraphBLAS C API Specification¹
 - ▶ GraphBLAS Pointers²
 - ▶ Introduction to GraphBLAS³

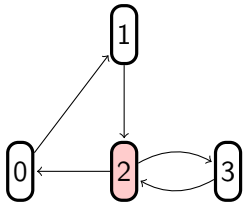
¹https://graphblas.org/docs/GraphBLAS_API_C_v2.1.0.pdf

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

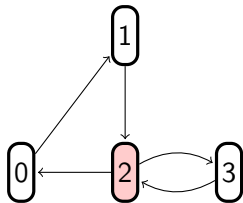
³<https://zenodo.org/record/4318870/files/graphblas-introduction.pdf>

⁴<https://graphblas.org/>

Пример: обход в ширину



Пример: обход в ширину



Текущий фронт

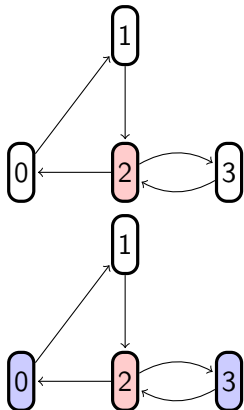
Матрица смежности

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

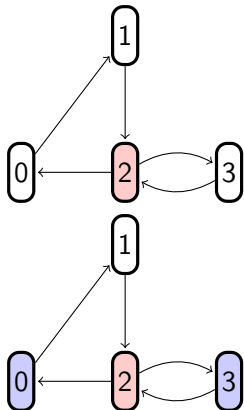
Новый фронт

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

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

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

Полукольцо

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

Реализации GraphBLAS-подобных API

- SuiteSparse:GraphBLAS⁵: эталон на чистом C
- Huawei's GraphBLAS⁶: частичная реализация на C++
- CombBLAS⁷: распределённая, частичная реализация на C++
- GraphBLAST⁸: поддержка GPGPU, Cuda C, частичная реализация
- Spla⁹: поддержка GPGPU, OpenCL C, частичная реализация
- Обёртки для различных языков: Python, Rust, ...
- Экспериментальные реализации от IBM, Postgres, ...
- ...

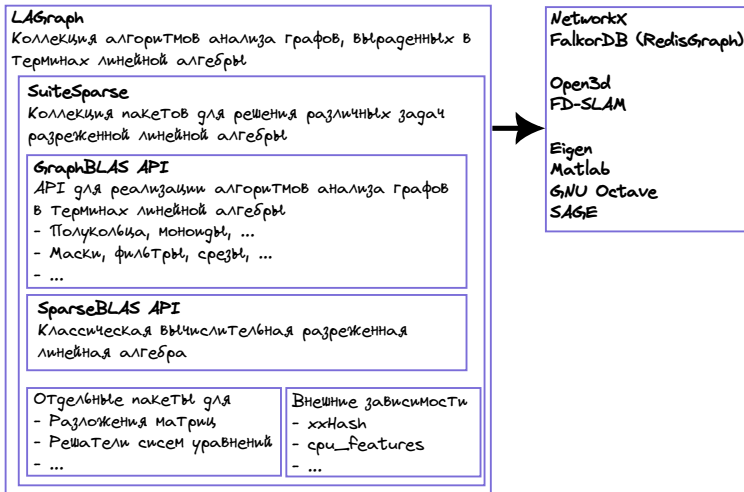
⁵<https://github.com/DrTimothyAldenDavis/GraphBLAS>

⁶<https://gitee.com/CSL-ALP/graphblas>

⁷<https://github.com/PASSIONLab/CombBLAS>

⁸<https://github.com/gunrock/graphblast>

⁹<https://github.com/SparseLinearAlgebra/spla>



¹⁰<https://github.com/DrTimothyAldenDavis/SuiteSparse>

SuiteSparse на RISC-V

- Было

- ▶ Alpine linux + chroot¹¹
- ▶ Сборка и тестирование в эмуляторе (qemu)¹²
- ▶ Продолжительность workflow в GitHub CI: 2 часа 20 минут

¹¹ До недавнего времени не было RISC-V

¹² Не для всех компонент

¹³ Для всех компонент

¹⁴ Позже выяснилось, что про них знали и ошибка в GCC а не в SuiteSparse

¹⁵ https://github.com/DrTimothyAldenDavis/SuiteSparse/pull/955#discussion_r2103092266

¹⁶ Соответствующий реквест: <https://github.com/DrTimothyAldenDavis/SuiteSparse/pull/949>

• Было

- ▶ Alpine linux + chroot¹¹
- ▶ Сборка и тестирование в эмуляторе (qemu)¹²
- ▶ Продолжительность workflow в GitHub CI: 2 часа 20 минут

• Стало

- ▶ Кросс-тулчейн + MultiArch
- ▶ Кросс-сборка и тестирование в эмуляторе (qemu-user)¹³
- ▶ Продолжительность workflow в GitHub CI: 40 минут
- ▶ Выявлены и локализованы ошибки под x390s и ppc64le¹⁴

¹¹ До недавнего времени не было RISC-V

¹² Не для всех компонент

¹³ Для всех компонент

¹⁴ Позже выяснилось, что про них знали и ошибка в GCC а не в SuiteSparse

¹⁵ https://github.com/DrTimothyAldenDavis/SuiteSparse/pull/955#discussion_r2103092266

¹⁶ Соответствующий реквест: <https://github.com/DrTimothyAldenDavis/SuiteSparse/pull/949>

- Было

- ▶ Alpine linux + chroot¹¹
- ▶ Сборка и тестирование в эмуляторе (qemu)¹²
- ▶ Продолжительность workflow в GitHub CI: 2 часа 20 минут

- Стало

- ▶ Кросс-тулчейн + MultiArch
- ▶ Кросс-сборка и тестирование в эмуляторе (qemu-user)¹³
- ▶ Продолжительность workflow в GitHub CI: 40 минут
- ▶ Выявлены и локализованы ошибки под x390s и ppc64le¹⁴

- Предложенное нами решение для кросс-сборки начали использовать в GNU Octave¹⁵

¹¹ До недавнего времени не было RISC-V

¹² Не для всех компонент

¹³ Для всех компонент

¹⁴ Позже выяснилось, что про них знали и ошибка в GCC а не в SuiteSparse

¹⁵ https://github.com/DrTimothyAldenDavis/SuiteSparse/pull/955#discussion_r2103092266

¹⁶ Соответствующий реквест: <https://github.com/DrTimothyAldenDavis/SuiteSparse/pull/949>

- Используется AVX и Neon
 - ▶ В явном виде — в очень ограниченном количестве функций
- Много кода написано так, будто авторы полагаются на автоматическую векторизацию компилятором
- Определение доступных расширений при компиляции и во время исполнения¹⁷

¹⁷ `cpu_features` и JIT

¹⁸ Соответствующая серия реквестов в GraphBLAS:

- <https://github.com/DrTimothyAldenDavis/GraphBLAS/pull/381>
- <https://github.com/DrTimothyAldenDavis/GraphBLAS/pull/422>
- <https://github.com/DrTimothyAldenDavis/GraphBLAS/pull/421>

- Используется AVX и Neon
 - ▶ В явном виде — в очень ограниченном количестве функций
- Много кода написано так, будто авторы полагаются на автоматическую векторизацию компилятором
- Определение доступных расширений при компиляции и во время исполнения¹⁷
- Мы поддержали RVV1.0 в умножении плотной матрицы на разреженную
 - ▶ Одна из немногих функций, векторизованных вручную с использованием AVX и Neon

¹⁷ `cpu_features` и JIT

¹⁸ Соответствующая серия реквестов в GraphBLAS:

- <https://github.com/DrTimothyAldenDavis/GraphBLAS/pull/381>
- <https://github.com/DrTimothyAldenDavis/GraphBLAS/pull/422>
- <https://github.com/DrTimothyAldenDavis/GraphBLAS/pull/421>

Экспериментальное исследование

- SuiteSparse matrix collection: матрицы разных размеров и разной степени разреженности
- Оборудование
 - ▶ X86_64
 - ★ **CPU:** Intel Core i7-12700H 800MHz с векторами размером 1024 битов
 - ★ **RAM:** LPDDR4, 16GB
 - ★ **Compiler:** GCC 14.2.0
 - ▶ RISC-V
 - ★ **SoC:** SPACEMIT K1/M1, Octa-core X60™(RV64GCVB), RVA22, RVV1.0 1600MHz с векторами размером 2048 битов
 - ★ **RAM:** LPDDR4X, 16GB
 - ★ **Compiler:** GCC 14.2.0 (cross)
- Сравнивали изменение величины среднего времени выполнения 400 запусков умножения матриц

Результаты экспериментального исследования¹⁹

№	Matrix name	Rows number	Nonzeros	AVX2 (ms.)	No AVX2 (ms.)	RVV (ms.)	No RVV (ms.)	AVX speedup (%)	RVV speedup (%)
1	olafu	16146	515651	5327.7	6629.7	43080.7	52940.1	19.6	18.6
2	fd18	16428	63406	476.4	482.0	2212.6	2181.2	1.2	-1.4
3	sme3Da	12504	874887	4236.9	5124.9	32008.0	42763.8	17.3	25.2
4	stokes64	12546	74242	508.8	564.4	2629.7	2814.1	9.8	6.6
5	sinc12	7500	294986	632.6	864.0	5970.1	8593.8	26.8	30.5
6	fd12	7500	28462	90.4	92.3	484.1	555.3	2.0	12.8
7	bcsstk15	3948	60882	87.8	117.9	1271.5	1770.8	25.6	28.2
8	tols4000	4000	8784	17.1	18.2	184.0	203.5	5.9	9.6
9	ex36	3079	53843	28.5	41.0	574.2	584.8	30.5	1.8
10	iprob	3001	9000	25.2	34.7	279.3	344.9	27.5	19.0
11	MISKnowledgeMap	2427	28511	31.3	38.5	401.6	490.0	18.8	18.0
12	LeGresley_2508	2508	16727	10.4	12.1	106.5	97.7	14.3	-8.9
13	reorientation_2	1544	9408	5.6	9.8	117.9	125.4	42.7	6.0
14	netscience	1589	2742	1.5	2.8	31.4	28.5	47.0	-10.0
15	mcfe	765	24382	2.3	5.5	51.1	65.3	58.8	21.8
16	orbitRaising_3	761	3256	0.6	1.6	10.5	13.1	63.0	19.5

¹⁹Во всех экспериментах стандартное отклонение не превосходит 5%

Результаты экспериментального исследования¹⁹

№	Matrix name	Rows number	Nonzeros	AVX2 (ms.)	No AVX2 (ms.)	RVV (ms.)	No RVV (ms.)	AVX speedup (%)	RVV speedup (%)
1	olafu	16146	515651	5327.7	6629.7	43080.7	52940.1	19.6	18.6
2	fd18	16428	63406	476.4	482.0	2212.6	2181.2	1.2	-1.4
3	sme3Da	12504	874887	4236.9	5124.9	32008.0	42763.8	17.3	25.2
4	stokes64	12546	74242	508.8	564.4	2629.7	2814.1	9.8	6.6
5	sinc12	7500	294986	632.6	864.0	5970.1	8593.8	26.8	30.5
6	fd12	7500	28462	90.4	92.3	484.1	555.3	2.0	12.8
7	bcsstk15	3948	60882	87.8	117.9	1271.5	1770.8	25.6	28.2
8	tols4000	4000	8784	17.1	18.2	184.0	203.5	5.9	9.6
9	ex36	3079	53843	28.5	41.0	574.2	584.8	30.5	1.8
10	iprob	3001	9000	25.2	34.7	279.3	344.9	27.5	19.0
11	MISKnowledgeMap	2427	28511	31.3	38.5	401.6	490.0	18.8	18.0
12	LeGresley_2508	2508	16727	10.4	12.1	106.5	97.7	14.3	-8.9
13	reorientation_2	1544	9408	5.6	9.8	117.9	125.4	42.7	6.0
14	netscience	1589	2742	1.5	2.8	31.4	28.5	47.0	-10.0
15	mcfe	765	24382	2.3	5.5	51.1	65.3	58.8	21.8
16	orbitRaising_3	761	3256	0.6	1.6	10.5	13.1	63.0	19.5

¹⁹Во всех экспериментах стандартное отклонение не превосходит 5%

Результаты экспериментального исследования¹⁹

№	Matrix name	Rows number	Nonzeros	AVX2 (ms.)	No AVX2 (ms.)	RVV (ms.)	No RVV (ms.)	AVX speedup (%)	RVV speedup (%)
1	olafu	16146	515651	5327.7	6629.7	43080.7	52940.1	19.6	18.6
2	fd18	16428	63406	476.4	482.0	2212.6	2181.2	1.2	-1.4
3	sme3Da	12504	874887	4236.9	5124.9	32008.0	42763.8	17.3	25.2
4	stokes64	12546	74242	508.8	564.4	2629.7	2814.1	9.8	6.6
5	sinc12	7500	294986	632.6	864.0	5970.1	8593.8	26.8	30.5
6	fd12	7500	28462	90.4	92.3	484.1	555.3	2.0	12.8
7	bcsstk15	3948	60882	87.8	117.9	1271.5	1770.8	25.6	28.2
8	tols4000	4000	8784	17.1	18.2	184.0	203.5	5.9	9.6
9	ex36	3079	53843	28.5	41.0	574.2	584.8	30.5	1.8
10	iprob	3001	9000	25.2	34.7	279.3	344.9	27.5	19.0
11	MISKnowledgeMap	2427	28511	31.3	38.5	401.6	490.0	18.8	18.0
12	LeGresley_2508	2508	16727	10.4	12.1	106.5	97.7	14.3	-8.9
13	reorientation_2	1544	9408	5.6	9.8	117.9	125.4	42.7	6.0
14	netscience	1589	2742	1.5	2.8	31.4	28.5	47.0	-10.0
15	mcfe	765	24382	2.3	5.5	51.1	65.3	58.8	21.8
16	orbitRaising_3	761	3256	0.6	1.6	10.5	13.1	63.0	19.5

¹⁹Во всех экспериментах стандартное отклонение не превосходит 5%

- SuiteSparse:GraphBLAS (и SuiteSparse целиком) готов к использованию на RISC-V
 - ▶ Кросс-сборка есть
 - ▶ Тесты проходят
 - ▶ Нативно собирается

- SuiteSparse:GraphBLAS (и SuiteSparse целиком) готов к использованию на RISC-V
 - ▶ Кросс-сборка есть
 - ▶ Тесты проходят
 - ▶ Нативно собирается
- Есть простор для оптимизаций
 - ▶ Детальный анализ векторных расширений и их работы в конкретном случае
 - ▶ Изучение поддержки автоматической векторизации компилятором
 - ▶ Векторизуемость других функций

- Анализ векторизации: что имеет смысл векторизовывать, какими средствами это лучше делать
- Сравнительный анализ на прикладных алгоритмах
- Исследование применимости матричных расширений
- Исследование других реализации GraphBLAS и GraphBLAS-подобных решений
 - ▶ CombBLAS — для распределённой аналитики
 - ▶ Spla — поддержка RISC-V CPU и GPGPU (IMG, Vortex)