



# Анализ графов и разреженная линейная алгебра на GPGPU в экосистеме RISC-V

Рабочая группа “Развитие экосистемы ПО на RISC-V”

Семён Григорьев

Санкт-Петербургский Государственный Университет

29 августа 2025

- **RISC-V CPU + Imagination Technologies GPU**: самая распространённая (из доступных) конфигурация
- **RISC-V CPU + AMD GPU**: утверждается, что работает и есть официальная поддержка ([Milk-V Pioneer](#), [Milk-V Megrez](#), [RuyiBook](#))
- **RISC-V CPU + Intel GPU**: [ходят слухи, что можно](#), но официальной поддержки пока нет
- **RISC-V CPU + Nvidia GPU**: [анонсировано](#)
- **RISC-V GPU**: [Vortex](#)

Нетривиальная задача из-за нерегулярного параллелизма

- NVGraph
- Gunrock
- Различные системы, которые что-то умеют отдавать на ГПУ
- Частые решения отдельных задач

- SuiteSparse:GraphBLAS
  - ▶ Cuda
  - ▶ Пока не анонсировано
- GraphBLAST
  - ▶ Cuda
  - ▶ Очень небольшое подмножество
- Spla
  - ▶ OpenCL
  - ▶ В разработке

Реализация подмножества GraphBLAS-подобного API на C++ и OpenCL C

- Переносимое решение между GPGPU разных вендоров
- Запускается на CPU благодаря POCL
- На «взрослых» GPGPU конкурентно с аналогами

---

<sup>1</sup><https://github.com/SparseLinearAlgebra/spla>

# Экспериментальное исследование

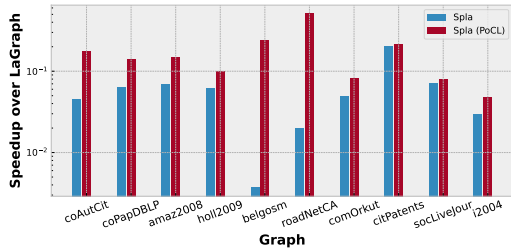
- SuiteSparse matrix collection: матрицы разных размеров и разной степени разреженности
- Оборудование
  - ▶ RISC-V
    - ★ **SoC**: SPACEMIT K1/M1, Octa-core X60™(RV64GCVB), RVA22, RVV1.0 1600MHz с векторами размером 2048 битов
    - ★ **RAM**: LPDDR4X, 16GB
- Замеряется время решения прикладной задачи (работы алгоритма)
- Подгрузка и конвертация данных не замеряется
- Ускорение относительно LaGraph

## Графы для экспериментов

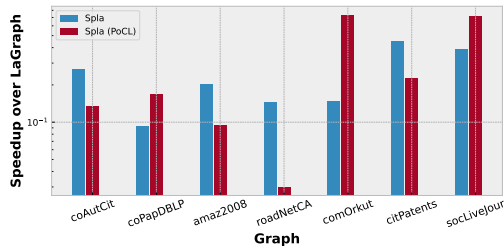
Name	Vertices	Edges	Avg Deg	Sd Deg	Max Deg
coAuthorsCiteseer	227.3K	1.6M	7.2	10.6	1372.0
coPapersDBLP	540.5K	30.5M	56.4	66.2	3299.0
amazon-2008	735.3K	7.0M	9.6	7.6	1077.0
hollywood-2009	1.1M	112.8M	98.9	271.9	11467.0
belgium_osm	1.4M	3.1M	2.2	0.5	10.0
roadNet-CA	2.0M	5.5M	2.8	1.0	12.0
com-Orkut	3.1M	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

# Результаты на SpacemiT M1 CPU, IMG BXE-2-32 GPU

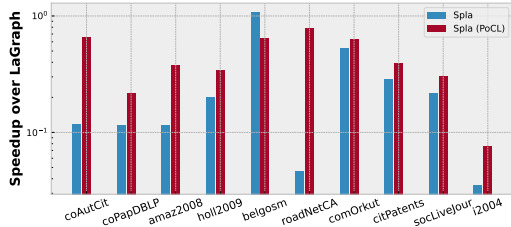
## BFS



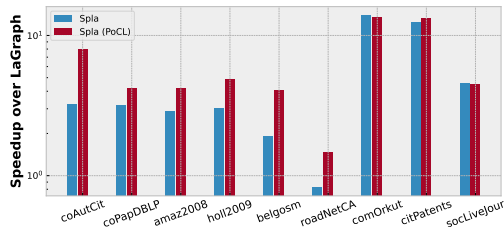
## Triangle Count (TC)



## Single Source Shortest Path (SSSP)

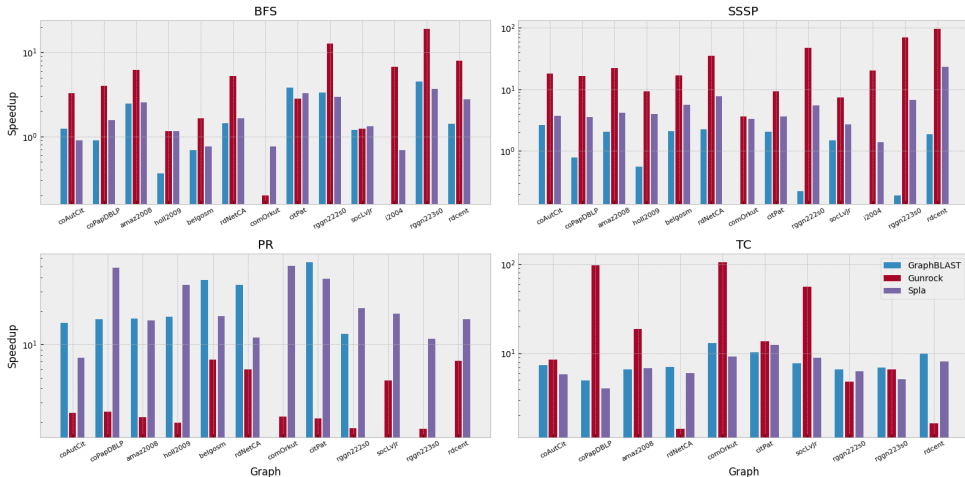


## PageRank (PR)



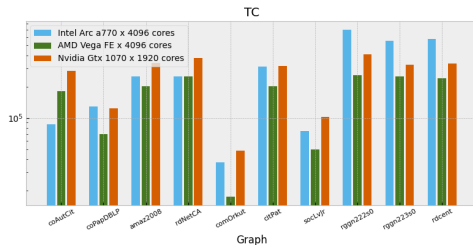
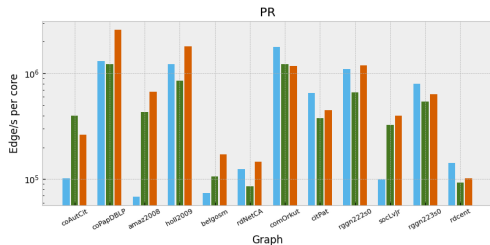
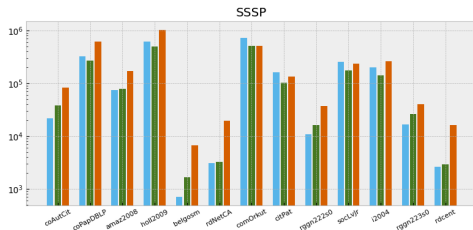
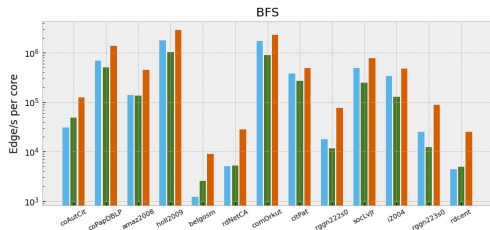


# Результаты на Nvidia GTX 1070



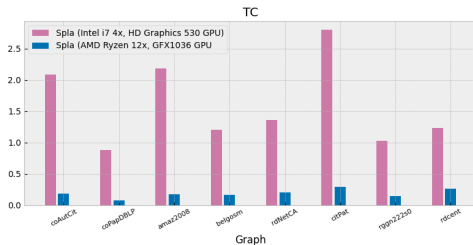
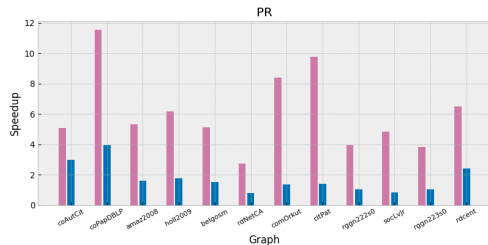
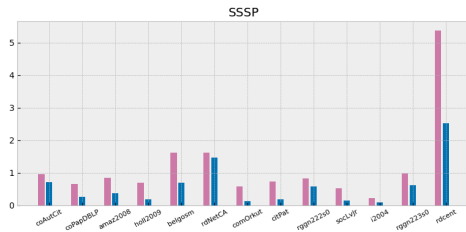
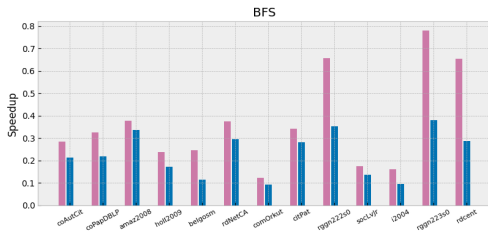
GPGPU не всегда полезен

# Результаты ГПН от Intel и AMD



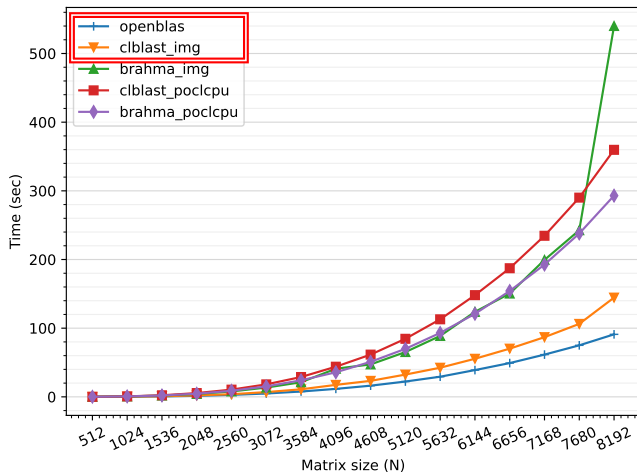
OpenCL, конечно, переносим, но всегда есть «но»

# Результаты на встроенной графике



Драйвер — это важно!

# Результаты умножения матриц на GPGPU от Imagination Technologies



GPGPU от Imagination Technologies (пока) не совсем для вычислений

- Набор инструкций, основанный на RISC-V ISA
- Поддержка OpenCL через POCL
  - ⚙️ Spla должен запускаться
- Проблемы со сбросом регистров
  - ▶ Типичные оптимизации не работают
  - ▶ Issue 1
  - ▶ Issue 2
- В целом, есть подозрение, что мало регистров
- Для ПЛИС с HBM
  - ❓ Бонус для обработки слабоструктурированных данных

- Обработка графов на GPGPU возможна и иногда полезна
  - ▶ Для некоторых алгоритмов полезнее, чем для некоторых других
  - ▶ Сильно зависит от структуры графа
- OpenCL работает и для GPU и для CPU
  - ▶ В том числе, для RISC-V
- Пока нет поддержки Cuda, выбор инструментов для обработки графов на GPU под RISC-V очень ограничен
  - ▶ Фактически, только Spla
  - ▶ В любом случае, OpenCL даёт хоть какую-то переносимость и независимость от вендора (и даже наличия GPU как такового)

- Запуск Spla на Vortex
- Анализ производительности Spla на различных CPU и GPGPU
- Оптимизация Spla
- Расширение возможностей Spla
- Запуск различных GPU (AMD, Intel) на платах с RISC-V