

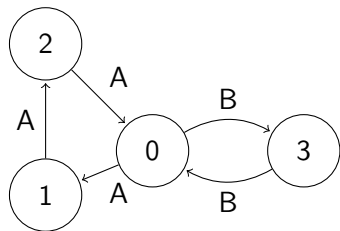


# Реализация операций над разреженными булевыми матрицами на OpenCL

**Автор:** Влаев Никита Владиславович, ОП ПИ 18.Б11-мм  
**Научный руководитель:** к.ф.-м.н., доцент кафедры информатики СПбГУ Григорьев С.В.

Санкт-Петербургский государственный университет  
Кафедра системного программирования

24 апреля 2021г.



1. Граф

$$s \rightarrow A s B$$

$$s \rightarrow A B$$

2. Грамматика для  $\{A^n B^n\}$

**Путь в графе** — слово из меток на рёбрах

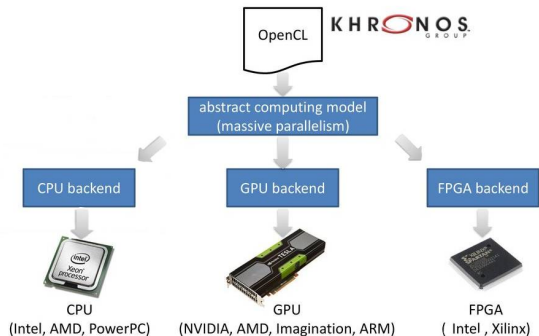
**CFPQ** — задача поиска путей с КС ограничениями.

Применения:

- Статический анализ кода (например, Taint Tracking)
- Контекстно-свободные запросы к графовым базам данных
- Биоинформатика — предсказания вторичной структуры РНК

- Существует множество алгоритмов CFPQ, на данный момент матричный [2] и тензорный [1] алгоритмы CFPQ самые перспективные
- Эти алгоритмы основаны на операциях линейной алгебры с разреженными булевыми матрицами
- Есть реализации булевой алгебры на CPU и GPU(CUDA)

## OpenCL as a portable language



Фреймворк для разработки многопоточных кроссплатформенных приложений

# Линейная алгебра на OpenCL

- Операции линейной алгебры для разреженных булевых матриц на OpenCL позволяют:
  - ▶ Создавать переносимые реализации алгоритмов CFPQ
  - ▶ Расширить выбор аппаратуры
- **cISPARSE** — библиотека разреженной линейной алгебры на OpenCL, булевы матрицы не поддерживаются

# Постановка задачи

**Целью** данной работы является реализация операций линейной алгебры для работы с разреженными булевыми матрицами на GPGPU и сравнение ее с другими реализациями

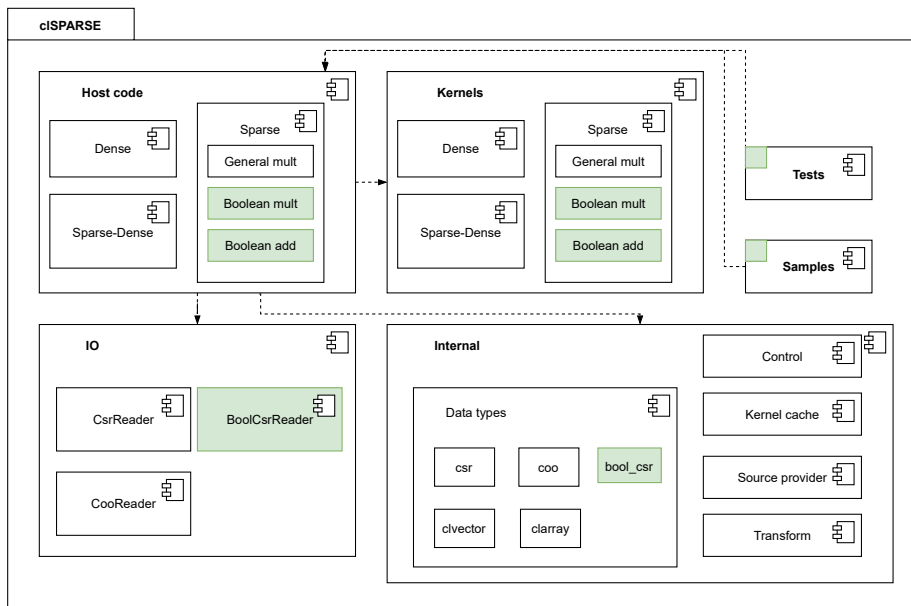
## Задачи

- 1 Реализовать операции линейной алгебры для разреженных булевых матриц на основе cSPARSE
  - 1 Ввод-вывод
  - 2 Поэлементное сложение
  - 3 Матричное умножение
- 2 Провести сравнительный анализ полученной реализации с другими

Реализации операций линейной алгебры:

- CPU
  - ▶ SuiteSparse
- CUDA-специфичные
  - ▶ CUSP
  - ▶ cuSPARSE
  - ▶ Cubool
- OpenCL
  - ▶ Clbool

# Архитектура





## Сравнительный анализ: матрицы

№	Matrix $M$	# Rows	Nnz of $M$	Nnz of $M^2$	Nnz of $M + M^2$
0	wing	62,032	243,088	714,200	917,178
1	luxembourg_osm	114,599	239,332	393,261	632,185
2	amazon0312	400,727	3,200,400	14,390,544	14,968,909
3	amazon-2008	735,323	5,158,388	25,366,745	26,402,678
4	web-Google	916,428	5,105,039	29,710,164	30,811,855
5	roadNet-PA	1,090,920	3,083,796	7,238,920	9,931,528
6	roadNet-TX	1,393,383	3,843,320	8,903,897	12,264,987
7	belgium_osm	1,441,295	3,099,940	5,323,073	8,408,599
8	roadNet-CA	1,971,281	5,533,214	12,908,450	17,743,342
9	netherlands_osm	2,216,688	4,882,476	8,755,758	13,626,132

## Сравнительный анализ: сложение

Оборудование: Ubuntu 20.04, Intel core i7-6700 CPU, 3.4GHz, DDR4 64Gb RAM, Geforce GTX 1070 GPGPU 8Gb RAM.

M	Clbool	cSPARSE-Bool	Cubool	Cusp	cuSPARSE	SuiteSparse
0	$1.89 \pm 0.32$	$1.96 \pm 1.04$	$1.12 \pm 0.02$	$1.54 \pm 0.20$	$2.40 \pm 0.04$	$3.91 \pm 2.06$
1	$1.63 \pm 0.32$	$3.18 \pm 1.05$	$1.84 \pm 0.10$	$1.11 \pm 0.21$	$0.86 \pm 0.04$	$1.86 \pm 0.29$
2	$23.53 \pm 0.63$	$14.12 \pm 1.79$	$12.14 \pm 0.46$	$16.26 \pm 0.40$	$23.74 \pm 0.04$	$37.12 \pm 0.13$
3	$36.68 \pm 0.88$	$21.34 \pm 2.53$	$20.06 \pm 2.67$	$29.86 \pm 1.65$	$27.75 \pm 1.65$	$65.02 \pm 1.35$
4	$43.28 \pm 2.89$	$22.78 \pm 2.01$	$24.20 \pm 0.90$	$32.08 \pm 1.34$	$88.48 \pm 0.28$	$77.68 \pm 1.66$
5	$12.47 \pm 0.13$	$27.96 \pm 1.73$	$16.69 \pm 0.59$	$11.07 \pm 0.28$	$11.62 \pm 0.03$	$36.50 \pm 1.01$
6	$15.29 \pm 0.05$	$35.35 \pm 1.89$	$19.86 \pm 0.65$	$14.42 \pm 0.98$	$17.43 \pm 0.03$	$46.33 \pm 3.07$
7	$10.43 \pm 0.14$	$36.17 \pm 1.77$	$19.55 \pm 0.12$	$9.83 \pm 0.24$	$11.57 \pm 0.99$	$28.44 \pm 1.00$
8	$22.31 \pm 0.11$	$50.07 \pm 2.25$	$31.61 \pm 1.01$	$18.90 \pm 0.40$	$20.50 \pm 0.82$	$65.74 \pm 1.39$
9	$18.91 \pm 2.08$	$55.38 \pm 2.14$	$30.65 \pm 1.89$	$14.75 \pm 0.31$	$18.15 \pm 0.03$	$50.29 \pm 0.93$

# Сравнительный анализ: умножение

Оборудование: Ubuntu 20.04, Intel core i7-6700 CPU, 3.4GHz, DDR4 64Gb RAM, Geforce GTX 1070 GPGPU 8Gb RAM.

M	Clbool-Hash	Clbool	cSPARSE-Bool	cSPARSE	Cubool	cuSPARSE	Cusp	SuiteSparse
0	1.88±0.34	4.23±0.36	1.57±0.28	2.02±0.34	1.78±0.06	20.04±0.04	5.39±0.12	7.92±0.27
1	2.06±0.36	6.41±0.36	2.31±0.30	2.64±0.34	2.50±0.04	1.72±0.05	3.79±0.07	3.72±1.57
2	52.02±0.04	62.39±0.06	25.19±0.98	32.73±1.57	24.19±0.82	408.71±0.54	108.98±0.45	258.04±8.39
3	81.43±0.29	96.62±0.44	40.25±1.36	51.19±2.01	34.63±1.53	184.64±10.23	172.74±0.36	378.94±15.28
4	126.36±0.34	Out of memory	140.54±4.12	165.03±8.30	42.19±1.26	4726.33±0.52	Out of memory	712.54±20.07
5	14.02±0.15	34.56±0.18	19.16±1.08	23.03±1.60	18.26±0.19	37.24±0.13	42.38±0.17	67.36±1.57
6	16.64±0.12	41.90±0.20	23.62±1.24	28.43±1.88	22.73±0.18	46.40±0.15	51.77±0.30	81.27±1.80
7	16.62±0.23	38.61±0.23	22.28±0.99	25.31±1.47	23.37±0.17	26.54±0.07	33.09±0.20	58.46±1.76
8	23.10±0.06	58.99±1.62	32.99±1.77	39.26±2.52	32.09±0.34	65.14±4.47	76.72±3.23	116.88±3.99
9	24.57±0.11	57.07±0.13	33.89±1.39	38.51±2.12	35.30±0.22	50.68±0.14	51.28±0.35	93.47±2.09

- Сложение
  - 1 На матрицах с  $\text{Nnz of } M + M^2 > 10000000$  реализация не уступает CPU
  - 2 На OpenCL альтернативный алгоритм из clBool показывает лучшие результаты, чем представленный
- Умножение
  - 1 Представленный алгоритм превосходит по скорости CPU
  - 2 Алгоритм превышает по скорости аналогичную реализацию с нуля на OpenCL в clBool
  - 3 Алгоритм с использованием хеш-таблиц в clBool превосходит по эффективности другие реализации на OpenCL
- Реализации на CUDA более эффективные на NVIDIA

- 1 Реализованы операции ввода-вывода для разреженных булевых матриц на основе cSPARSE
- 2 Реализована операция матричного умножения разреженных булевых матриц на основе cSPARSE
- 3 Реализована операция поэлементного сложения разреженных булевых матриц на основе cSPARSE
- 4 Проведен сравнительный анализ полученной реализации с другими

Реализация: <https://github.com/nikitavlaev/cSPARSE/tree/dev>

- 1 Реализация произведения Кронекера
- 2 Интеграция в алгоритмы CFPQ



Context-Free Path Querying by Kronecker Product / Egor Orachev, Ilya Epelbaum, Rustam Azimov, Semyon Grigorev. — 2020. — 08. — P. 49–59. — ISBN: 978-3-030-54831-5.



Context-Free Path Querying with Single-Path Semantics by Matrix Multiplication / Arseniy Terekhov, Artyom Khoroshev, Rustam Azimov, Semyon Grigorev // Proceedings of the 3rd Joint International Workshop on Graph Data Management Experiences Systems (GRADES) and Network Data Analytics (NDA). — GRADES-NDA'20. — New York, NY, USA : Association for Computing Machinery, 2020. — Access mode: <https://doi.org/10.1145/3398682.3399163>.