

Реализация алгоритма поиска путей с КС ограничениями в графовых БД через произведение Кroneckera на платформе Nvidia CUDA

Егор Орчев

JetBrains Research, Лаборатория языковых инструментов
Санкт-Петербургский Государственный университет

19 декабря 2020

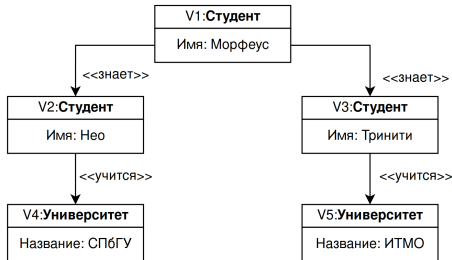


Figure: Пример графовой БД

- Графовая модель данных
- Графовые базы данных
- Запросы к графовым базам данных
- Использование КС грамматик как формализма для формирования запроса

Запросы с КС ограничениями

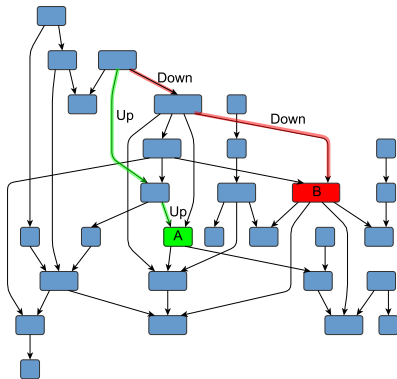


Figure: Пример графа

Навигация в графе:

- Находятся ли вершины A и B на одном уровне иерархии?
- Существует ли путь вида $Up^n Down^n$?
- Найти все такие пути $Up^n Down^n$, которые начинаются в вершине A

- Ориентированный граф с метками $\mathcal{G} = \langle V, E, L \rangle$
- $\omega(\pi) = \omega(v_0 \xrightarrow{l_0} v_1 \xrightarrow{l_1} \dots \xrightarrow{l_{n-2}} v_{n-1} \xrightarrow{l_{n-1}} v_n) = l_0 l_1 \dots l_{n-1}$
- КС Грамматика $G = \langle \Sigma, N, P, S \rangle$
- Язык $L(G) = \{w \mid S \xrightarrow{*}_G w\}$
- Семантика достижимости: $R = \{(u, v) \mid \exists u\pi v : \omega(\pi) \in L\}$
- Семантика всех путей: $\Pi = \{u\pi v \mid \omega(\pi) \in L\}$

Пример запроса

- Пример запроса:
MATCH (u) \rightarrow [: знает] \rightarrow () \rightarrow [: учится] \rightarrow (v)
RETURN u, v
- Семантика достижимости:
 $\{(V1, V4), (V1, V5)\}$
- Семантика всех путей:
 $\{(V1 \xrightarrow{\text{знает}} V2 \xrightarrow{\text{учится}} V4), (V1 \xrightarrow{\text{знает}} V3 \xrightarrow{\text{учится}} V5)\}$

Существующие алгоритмы для вычисления КС запросов

- Алгоритмы, основанные на различных техниках парсинга (CYK, LL, LR, etc.)
- Алгоритм Рустама Азимова на основе операций линейной алгебры (реализация на CPU, на GPU)
- **Алгоритм на основе произведения Кронекера** (реализация на CPU, на GPU?)

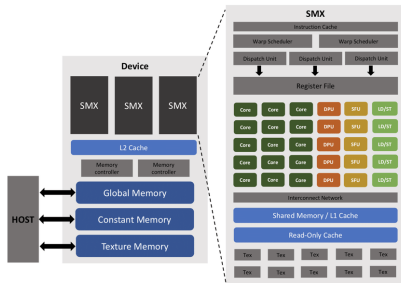


Figure: Архитектура GPU

- Неграфические вычисления общего назначения на GPU
- Хорошо подходит, когда необходимо обрабатывать множество данных фиксированным набором команд
- Можно использовать для выполнения операций линейной алгебры над разреженными данными

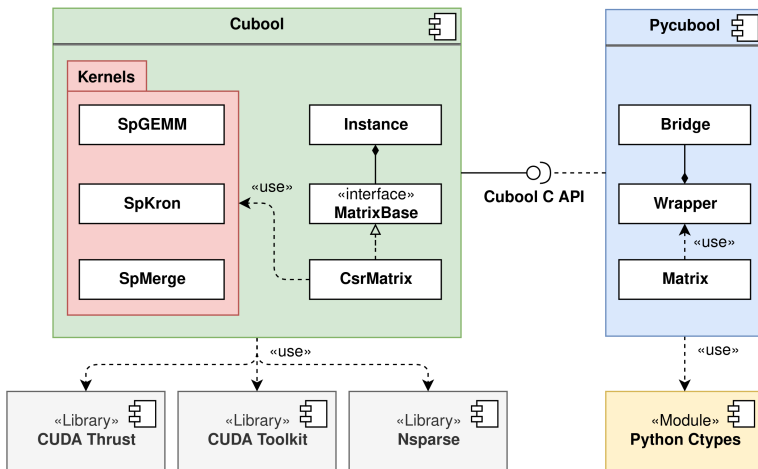


Figure: Архитектура библиотеки **Cubool**

Пример использования C API

```
1 CuBoolStatus TransitiveClosure(CuBoolInstance Inst, CuBoolMatrix A, CuBoolMatrix* T) {
2     CuBool_Matrix_Duplicate(Inst, A, T);          /** Копируем матрицу смежности A */
3
4     CuBoolSize_t total = 0;
5     CuBoolSize_t current;
6     CuBool_Matrix_Nvals(Inst, *T, &current);      /** Количество ненулевых значений */
7
8     while (current != total) {                    /** Пока результат меняется */
9         total = current;
10        CuBool_MxM(Inst, *T, *T, *T);             /** T += T * T */
11        CuBool_Matrix_Nvals(Inst, *T, &current);
12    }
13
14    return CUBOOL_STATUS_SUCCESS;
15 }
```

Figure: Вычисление транзитивного замыкания для ориентированного графа без меток с использованием **Cubool C API**

Пример использования Python API

```
1 def transitive_closure(a: pycubool.Matrix):
2     t = a.duplicate()           # Копируем матрицу смежности A
3     total = 0                   # Количество ненулевых значений результата
4
5     while total != t.nvals:     # Пока результат меняется
6         total = t.nvals
7         pycubool.mxm(t, t, t)   # t += t * t
8
9     return t
```

Figure: Вычисление транзитивного замыкания для ориентированного графа без меток с использованием **Pycubool**

Задачи на ближайшее время

- Реализация алгоритма поиска путей с КС ограничениями через произведение Кронекера с использованием примитивов библиотеки
- Экспериментальное исследование реализованного алгоритма
- Сравнение с реализацией на основе библиотеки **GraphBlast**¹
- Исследование возможности расширения библиотеки для выполнения операций в произвольном полукольце

¹<https://github.com/gunrock/graphblast>

- Почта: egororachyov@gmail.com
- Материалы презентации:
 - ▶ Arseniy Terekhov & Artyom Khoroshev & Rustam Azimov & Semyon Grigorev (2020). Context-Free Path Querying with Single-Path Semantics by Matrix Multiplication. 1-12. 10.1145/3398682.3399163.
 - ▶ Egor Orachev & Ilya Epelbaum & Rustam Azimov & Semyon Grigorev. (2020). Context-Free Path Querying by Kronecker Product. 10.1007/978-3-030-54832-2_6.
- Ссылка на проект Cubool:
<https://github.com/JetBrains-Research/cuBool>