



Санкт-Петербургский государственный университет

# Сложение разреженных матриц с использованием Brahma.FSharp

Кирилл Анатольевич Гарбар, группа 20.Б07-мм

6 июня 2022 г.

**Научный руководитель:** к.ф.-м.н. С.В. Григорьев, доцент кафедры информатики

Санкт-Петербург  
2021

- **Матрицы и векторы** — способ параллельной обработки и анализа графов
- Стандарт **GraphBLAS** определяет строительные блоки алгоритмов над графами на языке линейной алгебры
- **GraphBLAS-sharp** — попытка реализовать GraphBLAS на языке **F#**
  - ▶ **Brahma.FSharp** — транслятор из **F#** в **OpenCL**

# Существующие решения

Критерии сравнения:

- Платформа для вычислений
- Язык программирования

Название	CPU/GPU	Язык
SuiteSparse	Да/Нет	C
GraphBLAST	Нет/Да	C++
CUSP	Нет/Да	C++
pggraphblas	Да/Нет	C
GraphBLAS-sharp	Да/Да	F#
Math.NET Numerics	Да/Нет	C#

$$\begin{pmatrix} 1 & 2 & 0 \\ 0 & 4 & 0 \\ 0 & 2 & 6 \end{pmatrix}$$

- COO — координатный формат
  - ▶ массив\_значений = [1, 2, 4, 2, 6]
  - ▶ массив\_номеров\_столбцов = [0, 1, 1, 1, 2]
  - ▶ массив\_номеров\_строк = [0, 0, 1, 2, 2]
- CSR — Йельский формат
  - ▶ массив\_значений = [1, 2, 4, 2, 6]
  - ▶ массив\_номеров\_столбцов = [0, 1, 1, 1, 2]
  - ▶ сжатый\_массив\_номеров\_строк = [0, 2, 3]

$$\begin{pmatrix} \text{None} & 2 & 0 \\ 4 & \text{None} & \text{None} \\ \text{None} & 2 & \text{None} \end{pmatrix}$$

- Пример(как происходит в **SuiteSparse**) — в результате операций над матрицей взвешенного графа в одной из ячеек оказалось нулевое значение. Возможно два варианта:
  - ▶ 0 — вес ребра графа — **явный ноль**.
  - ▶ 0 — отсутствие ребра между вершинами, в таком случае 0 следует удалить — **неявный ноль**
- Решения
  - ▶ Явная фильтрация — неудобно, непроизводительно
  - ▶ **Option** типы

Целью работы является реализация операций разреженной линейной алгебры для **GraphBLAS-sharp** с использованием **Brahma.FSharp**

Поставленные задачи:

- Реализовать поэлементное сложение разреженных матриц, представленных в **CSR** формате
- Реализовать поддержку **option** типов в операции сложения элементов матриц, а также добавить возможность сложения матриц разного типа.
- Произвести сравнение производительности с **SuiteSparse** и **CUSP**, а также с уже реализованным сложением в координатном формате и с математической библиотекой **Math.NET Numerics**

# Поэлементное сложение матриц в CSR формате

- Представить матрицу в координатном формате
  - ▶ Построить массив номеров строк по сжатому массиву номеров строк
- Произвести поэлементное сложение матриц в координатном формате
- Представить результирующую матрицу в **CSR** формате
  - ▶ Построить битмап уникальных вхождений в массив номеров строк
  - ▶ Посчитать количество ненулевых элементов в каждой строке
  - ▶ С помощью префиксной суммы, по массиву, полученному в прошлом пункте, получить сжатый массив номеров строк

# Option типы и сложение матриц разных типов

- Слияние значений из обеих матриц
  - ▶ Вместо слияния в один массив распределить элементы по двум
  - ▶ Создать соответствующий битмап принадлежности элемента к левой или правой матрице
- Поэлементное сложение
  - ▶ Если есть пара элементов — произвести сложение
  - ▶ Иначе, сложить элемент из одной матрицы с нулём другой матрицы
- Фильтрация нулей
  - ▶ **Some value** — положить в результирующий массив
  - ▶ **None** — игнорировать



## Гибкость полученного решения

Написав соответствующую бинарную операцию, можно выразить:

- Поэлементное сложение
- Поэлементное умножение — в **SuiteSparse** для этого написан отдельный алгоритм
- Операции, использующие матрицу как маску

```
let float32Mul =  
  fun (values: AtLeastOne<float32 , float32 >) ->  
    let res =  
      match values with  
      | Both (f, s) -> f * s  
      | _ -> 0f  
    if res = 0f then None else Some res
```

Листинг 1: Операция поэлементного умножения в GraphBLAS-sharp

## Постановка эксперимента

- BenchmarkDotNet
- SpBench
- Matrix Market format
- Intel Core i7-4790 CPU, 3.60GHz, 4 cores, 8 threads, 32GB DDR4 RAM, GTX 2070

Название	Размер	Количество элементов	Заполненность
wing	62 032	243 088	0,0063%
luxembourg_osm	114 599	119 666	0,0009%
amazon0312	400 727	3 200 440	0,0019%
amazon-2008	735 323	5 158 388	0,0009%
web-Google	916 428	5 105 039	0,0001%
webbase-1M	1 000 005	3 105 536	0,0003%
cit-Patents	3 774 768	16 518 948	0,0001%

Таблица: Матрицы, на которых производилось сравнение

## Сравнение с SuiteSparse, CUSP и Math.NET

Название	GraphBLAS-sharp	SuiteSparse	CUSP	Math.NET Numerics
wing	$1,8 \pm 0,1$	$1,9 \pm 0,1$	$0,5 \pm 0,2$	$5,5 \pm 0,2$
luxembourg_osm	$2,9 \pm 0,3$	$1,9 \pm 0,5$	$0,5 \pm 0,1$	$286,2 \pm 2,2$
amazon0312	$17,0 \pm 0,8$	$28,9 \pm 0,2$	$2,8 \pm 0,1$	—
amazon-2008	$12,2 \pm 0,8$	$50,1 \pm 2,4$	$3,5 \pm 0,1$	—
web-Google	$18,4 \pm 0,6$	$58,8 \pm 0,7$	$3,6 \pm 0,1$	—
webbase-1M	$70,7 \pm 1,0$	$72,9 \pm 0,4$	$24,6 \pm 2,1$	—
cit-Patents	$54,6 \pm 1,2$	$157,4 \pm 1,2$	$8,5 \pm 1,2$	—

**Таблица:** Результаты сравнения библиотек на сложение в CSR формате, GTX 2070, среднее  $\pm$  стандартное отклонение, мс. Отсутствие данных означает, что среднее время превышает 100 секунд

## Сравнение поэлементного умножения в GraphBLAS-sharp и SuiteSparse

Название	GraphBLAS-sharp CSR	SuiteSparse
wing	$2,5 \pm 0,4$	$1,0 \pm 0,1$
luxembourg_osm	$2,6 \pm 0,3$	$1,4 \pm 0,3$
amazon0312	$13,0 \pm 1,0$	$23,0 \pm 0,9$
amazon-2008	$9,1 \pm 0,8$	$35,2 \pm 4,0$
web-Google	$14,7 \pm 0,8$	$43,9 \pm 0,2$
webbase-1M	$55,4 \pm 1,2$	$31,0 \pm 1,6$
cit-Patents	$47,9 \pm 0,9$	$107,9 \pm 0,4$

**Таблица:** Сравнение результатов поэлементного умножения матриц, GTX 2070, среднее  $\pm$  стандартное отклонение, мс.

- Достигнута приемлемая производительность, несмотря на реализацию на F# вместо C/C++
- Операция сложения достаточно гибкая, чтобы с её помощью произвести другие поэлементные операции, такие как поэлементное умножение

- Реализовано поэлементное сложение разреженных матриц, представленных в **CSR** формате
- Реализована поддержка **option** типов в операции сложения элементов матриц, а также добавлена возможность сложения матриц разного типа
- Произведено сравнение производительности с **SuiteSparse** и **CUSP**, а также с уже реализованным сложением в координатном формате и с математической библиотекой **Math.NET Numerics**

Репозиторий с реализацией:

<https://github.com/YaccConstructor/GraphBLAS-sharp/tree/net5>

Имя пользователя: **kirillgarbar**