Санкт-Петербургский государственный университет

Кафедра системного программирования

Команда №6

Мелашенко Ксения, Сыресенков Илья, Нурмухаметов Рафик

Санкт-Петербург
2025

GraphBLAS

- Спецификация API, определяющая стандартные строительные блоки для реализации графовых алгоритмов на языке линейной алгебры
  - Графы выражаются через разреженные матрицы смежности
  - Алгоритмы описываются матричными операциями
  - Вычисления над различными полукольцами
- **SuiteSparse::GraphBLAS** — эталонная реализация

SPLA

- Открытая библиотека обобщённой разрежённой линейной алгебры
- Реализована на C++
- Основные примитивы линейной алгебры, пользовательский типы данных и операции над ними
- OpenCL для запуска под CPU

BFS в терминах линейной алгебры

- **Представление**
  - $M \in \{0, 1\}^{n \times n}$ — матрица смежности
  - $F \in \{0, 1\}^{1 \times n}$ — текущий фронт обхода
  - $R \in \{0, 1\}^{1 \times n}$ — вектор посещённых вершин на данном этапе
- **Инициализация**
  - $F[s] = 1$
  - $R[s] = 1$
- **Алгоритм**
  - Пока $\exists v \in V : F[v] = 1$:
    - $F = FM$
    - $F = F \wedge \neg R$
    - $R = R \vee F$

SSP BFS в терминах ЛА

Вход:

- $M \in \{0,1\}^{n \times n}$ — матрица смежности графа G, $n = |V|$
- $s \in V$ — стартовая вершина

Инициализация:

- $F \in \{0,1\}^n$ — текущий фронт обхода, $F[s] = 1$
- $P \in (V \cup \{\perp\})^n$ — вектор родителей, $P[s] = s$, $P[v] = \perp$ для остальных

Алгоритм:

- Пока $\exists\, v \in V : F[v] = 1$:
  - $p = FM$ в полукольце $\langle \mathrm{ANY\_SECONDI} \rangle$
  - $p = p\langle \mathrm{mask} = (P = \perp) \rangle$
  - $P\langle \mathrm{mask} = (P = \perp) \rangle \leftarrow p$
  - $F\langle \mathrm{mask} = (p \neq \perp) \rangle \leftarrow 1_n$

Результат:

- Вектор P

SSP BFS в терминах ЛА

$$\underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{текущий фронт}} \times \underbrace{\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{матрица смежности графа}} = \underbrace{\begin{bmatrix} \perp & 3 & 3 & \perp & 3 \end{bmatrix}}_{\text{найденные родители}}$$

MS Parent BFS в терминах ЛА

- Обобщение SSP BFS на множество источников S:
  - **Матрица родителей** $P_{|S| \times |V|}$: $P[s, v]$ — родитель вершины $v \in S$ в дереве BFS, порождённом из $s \in S$
  - **Матрица фронтов** $F_{|S| \times |V|}$: $F[s, v] = 1 \Leftrightarrow v \in S$ в текущем фронте обхода из источника $s \in S$

PageRank

- **The PageRank Citation Ranking: Bringing Order to the Web**[1], Sergey Brin and Lawrence Page
- Не все страницы одинаково важны, чем больше ссылаются — тем важнее страница
- Ключевая идея: весь интернет можно представить в виде графа
- Важность страницы как вероятность посещения

[1] http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf

PageRank: Алгоритм

- $A[n \times n], A_{i,j} = 1 \Leftrightarrow$ существует ребро из вершины i в вершину j, 0 иначе
- Граф $\rightarrow$ цепь Маркова
- $P_{i,j} = \dfrac{A_{i,j}}{\sum_k A_{i,k}}$
- $P(t) = P \Rightarrow P$ однородна
- Обозначим за $p(t) = (p_0(t) \ \ldots \ p_n(t))$ вектор вероятностей нахождения в каждом из состояний в момент времени $t = 0, 1, \ldots$
- $p(t+1) = p(t)P$
- $r = \lim_{t\to\infty} p(t)$, r — искомый PageRank
- $r^{n+1} = r^n P$

PageRank: Сходимость

- Теорема: r существует и не зависит от $p(0)$ тогда и только тогда, когда допустим переход из любого состояния i в любое состояние j
- Проблема: P не удовлетворяет теореме
- Пусть E — матрица, состоящая из $\frac{1}{n}$
- Решение: $P' = \delta P + (1 - \delta)E, \delta \in (0, 1)$
- δ — коэффициент демпфирования (телепортации)
- $p(t + 1) = p(t)P'$
- $\exists r = \lim_{t \to \infty} p(t)$, r — стационарный вектор

MST: Алгоритм Борувки

Принцип работы алгоритма:

1. Каждая вершина считается отдельной компонентой (деревом)
2. Для каждой компоненты определяется ребро минимального веса, соединяющее его с другой компонентой
   - Компоненты объединяются через выбранные ребра, что приводит к уменьшению общего количества компонент
3. Процесс повторяется, пока не останется единственный компонент, охватывающий все вершины графа.

Алгоритм Борувки в терминах ЛА

- $A[n \times n]$ – матрица смежности графа, $\perp$ — отсуствие ребра
- Ребра в виде (w, i), где
  - w – вес ребра
  - i – номер компоненты, с которой это ребро соединено
- $edge[v]$ – минимальное ребро, инцидентное вершине v
- $cedge[v]$ – минимальное ребро, инцидентное одному из потомков v, если потомок – корень дерева, $+\infty$ иначе
- $parent[v]$ – номера компонентов для каждой вершины v

Алгоритм Борувки в терминах ЛА

Полукольцо $combMin = \{E \times I, comb, min\}$

- $comb((w_1, i_1), (w_2, i_2)) = (w_1, i_2)$
- $min((w_1, i_1), (w_2, i_2))$
- Свойства полукольца показать несложно
- $\perp$ — нулевой элемент

Алгоритм Борувки в терминах ЛА

- edge $=A \times$ parent
- cedge[parent[i]] $\leftarrow$ min(cedge[parent[i]], edge[i])
- Далее обновляем parent, чтобы он соответствовал новым деревьям
- Выкидываем использованные рёбра
- Повторяем пока деревьев больше одного

Эксперимент 1

- **Цель эксперимента:** для каждого алгоритма сравнить реализации на SuiteSparse::GraphBLAS и SPLA
  - Среднее время выполнения
  - Среднее пиковое потребление памяти
- Для **MS Parent BFS** число стартовых вершин будет 256

Эксперимент 2

- **Цель эксперимента:** оценить масштабируемость **MS Parent BFS** при изменении числа стартовых вершин
- $1, 4, 16, 64, 256$ вершин, выбираются случайно с фиксированным сидом
- Оцениваем замедление при увеличении числа вершин

Ход экспериментов

- Запуск алгоритмов в рамках каждого эксперимента 15 раз
- Вычисление средних значений и доверительных интервалов
- Визуализация и анализ результатов

Набор данных (SS/MS Parent BFS / Борувка)

Графы взяты из 9th DIMACS Implementation Challenge – Shortest Paths[2], где собраны реальные данные дорожных сетей США и синтетические данные для стресс-тестов алгоритмов

- Прикладные
- Связные
- Взвешенные, без отрицательных весов
- Неориентированные

[2] https://www.diag.uniroma1.it/challenge9/

Набор данных (SS/MS Parent BFS / Борувка)

Название	**Кол-во вершин**	**Кол-во ребер**
New York City	264 346	733 846
Florida	1 070 376	2 712 798
California and Nevada	1 890 815	4 657 742
Western USA	6 262 104	15 248 146
Full USA	23 947 347	58 333 344

Набор данных (PageRank)

Графы для **Pagerank** взяты из Stanford Large Network Dataset Collection[3]

- Прикладные
- Ориентированные

[3] `https://snap.stanford.edu/data/#socnets`

Набор данных (PageRank)

Название	Кол-во вершин	Кол-во ребер
Web-Stanford	281 903	2 312 497
Web-BerkStan	685 230	7 600 595
Web-Google	875 713	5 105 039
Wiki-Topcats	1 791 489	28 511 807
Sx-Stackoverflow	2 601 977	63 497 050

Оборудование

Характеристики вычислительной машины:

- ОС Ubuntu 24.04
- 11th Gen Intel Core i5-11400H
  - 6 ядер, 12 потоков
  - Частота до 4.5 ГГц
  - 12 Mb кеш-память 3 уровня
- 16 GB RAM
  - DDR4
  - 3200 МГц