Санкт-Петербургский государственный университет

Кафедра системного программирования

Теория графов, презентация 2

Луконенко Никита Игоревич

Санкт-Петербург
2025

Эксперимент

Цель: сравнение среднего времени работы различных реализаций алгоритмов.
Шаги

- Провести 20 запусков
- Построить доверительные интервалы
- Сравнить результаты
- Вопросы:
  - Вопрос: на какой из трех библиотек (GBTL, LAGraph или SPLA) окажется быстрее реализация алгоритма Прима?
  - Вопрос: на какой из двух библиотек (SPLA и Suite Sparse(LAGraph)) окажется быстрее реализация алгоритма Sandia?

Характеристики машины:

- Процессор: Apple Silicon M1 (8 ядер, четыре высокопроизводительных, четыре энергоэффективных)
- ОЗУ: 16 GB
- ОС: macOS Sequoia

Данные для Прима

Таблица: Данные[1], используемые в эксперименте

Название графа	Количество узлов	Количество ребер
New York City	264,346	733,846
San Francisco Bay Area	321,270	800,172
Colorado	435,666	1,057,066
Internet	124,651	207,214

- Используются в статье[2] для бенчмаркинга при сравнении алгоритмов

[1] https://www.diag.uniroma1.it/challenge9/download.shtml

[2] https://userweb.cs.txstate.edu/ mb92/papers/sc23b.pdf

Данные

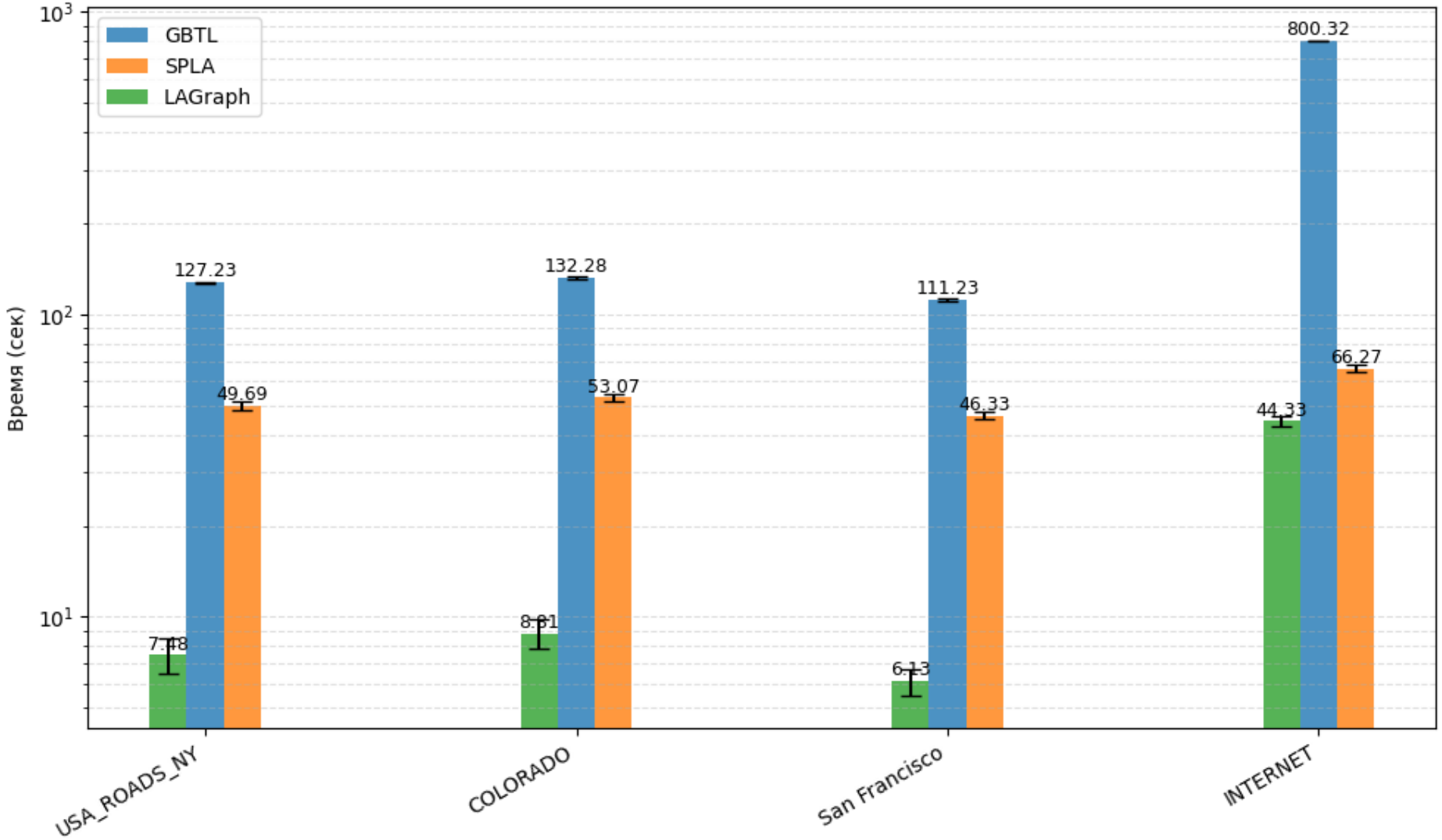
Таблица: Данные[3], используемые в эксперименте

Название графа	Количество узлов	Количество ребер
New York City	22,387	43,237
San Francisco Bay Area	18,336	40,632
Colorado	17,843	46,538
Internet	72,672	49,281

- Графы оказались слишком большими
- Было принято решение их обрезать до указанных размеров

[3] https://www.diag.uniroma1.it/challenge9/download.shtml

Результаты


GBTL
SPLA
LAGraph
Время (сек)
10^3
10^2
10^1
127.23
49.69
7.48
132.28
53.07
8.81
111.23
46.33
6.13
800.32
66.27
44.33
USA_ROADS_NY
COLORADO
San Francisco
INTERNET

Результаты

Граф	Прим LAGraph(сек)	Прим GBTL(сек)	Прим SPLA(сек)
USA_ROADS_NY	7.48	127.23	49.69
COLORADO	8.81	132.28	53.07
USA_road_d_BAY(San Francisco)	6.13	111.23	46.33
INTERNET	44.33	800.32	66.27

Результаты

Для выяснения причин существенной разницы во времени работы алгоритмов был запущен профилировщик

- Название профилировщика: xctrace
- Граф: INTERNET
- Библиотека: GBTL

	Название метода	Занимаемое время в процентах	Занимаемое время
1	apply	47%	6 min 19 sec
2	assign	23%	2 min 7 sec
3	eWiseMult	13%	1 min 49 sec
4	eWiseAdd	9%	1 min 18 sec
5	extract	7%	1 min 2 sec

Результаты

- Название профилировщика: xctrace
- Граф: INTERNET
- Библиотека: SPLA

	Название метода	Занимаемое время в процентах	Занимаемое время
1	set_uint	31%	21.08 sec
2	exec_v_eadd_fdb	21%	14.09 sec
3	allocator	19%	13.62 sec
4	read	18%	12.93 sec
5	exec_v_assign_masked	9%	6.70 sec

Результаты

- Сводная таблица времени работы методов в различных библиотеках

Библиотека	1		2		3	
GBTL	apply	6 min 19 sec	assign	2 min 7 sec	eWiseMult	1 min 49 sec
SPLA	set_uint	21.08 sec	exec_v_eadd_fdb	14.09 sec	allocator	13.62
LAgraph	GrB_Vector_removeElement	30.41 sec	GrB_Col_extract	4.09 sec	GrB_Vector_eWiseAdd	324 ms

Результаты

Прим оказался быстрее при использовании библиотеки LAGraph на всех графах
Причины:

- Метод apply (при применении маски) в библиотеке gbtl
- Метод set_uint в библиотеке SPLA (при обновлении весов)
- "Под капотом" apply скрывает системные вызовы для работы с памятью, это тормозит исполнение
- При увеличении плотности графа это становится более заметно

Данные для Sandia

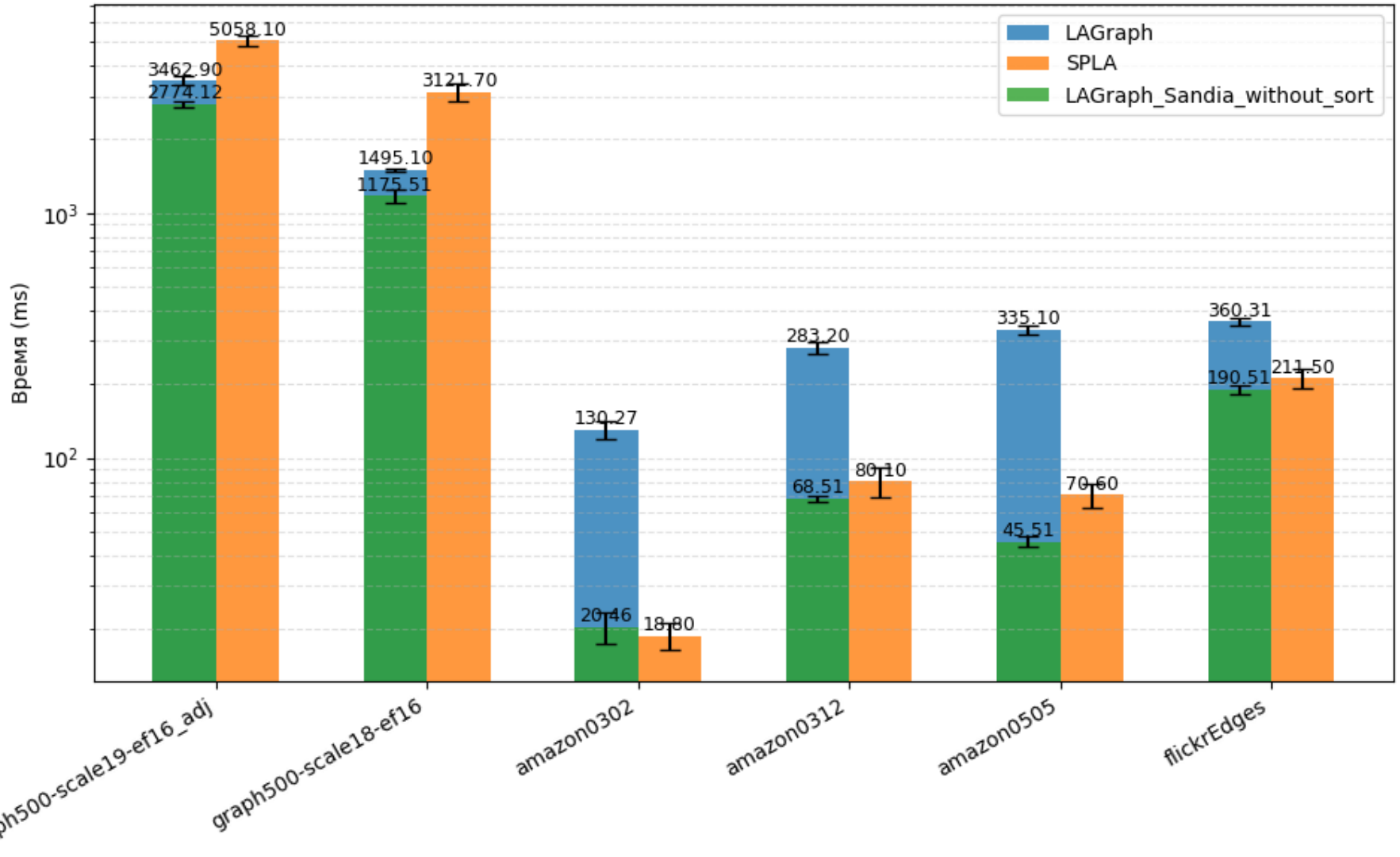
Таблица: Данные[4], используемые в эксперименте

Название графа	Количество узлов	Количество ребер
graph500-scale19-ef16_adj	335,318	7,729,675
graph500-scale18-ef16_adj	174,147	3,800,348
Amazon0302	262,111	1,234,877
Amazon0312	400,727	2,349,869
Amazon0505	410,236	2,439,437
flickrEdges_adj	105,938	2,316,948

- Используются в статье[5] для бенчмаркинга при сравнении алгоритмов
- Есть и синтетика, и "натуральные" графы

[4] https://www.diag.uniroma1.it/challenge9/download.shtml
[5] https://www.osti.gov/servlets/purl/1466485

Результаты


LAGraph
SPLA
LAGraph_Sandia_without_sort
Время (ms)
10^3
10^2
3462.90
2774.12
5058.10
1495.10
1175.51
3121.70
130.27
20.46
18.80
283.20
68.51
80.10
335.10
45.51
70.60
360.31
190.51
211.50
graph500-scale19-ef16_adj
graph500-scale18-ef16
amazon0302
amazon0312
amazon0505
flickrEdges

Результаты

Граф	Sandia SuiteSparse(LAGraph) (ms)	Sandia SPLA (ms)
graph500-scale19-ef16_adj	3462.90	5058.10
graph500-scale18-ef16_adj	1495.10	3121.70
Amazon0302	130.27	19.10
Amazon0312	283.20	80.10
Amazon0505	335.10	70.60
flickrEdges_adj	360.31	211.50

Результаты

Sandia оказался быстрее при использовании библиотеки Suite Sparce (LAGraph) на больших графах, но при уменьшении их размера проигрывает

- Проигрыши SuiteSparse(LAGraph) были в тех местах, где предварительная сортировка ребер занимала много времени
- Это позволяет реализовать преимущество на больших графах, но с маленькими может занимать больше времени, чем сам алгоритм