



Санкт-Петербургский государственный университет
Кафедра системного программирования

Борувка. SPLA и Pregel+. Презентация 2.

Пиллюк Дмитрий Алексеевич

Санкт-Петербург
2025

На какой из библиотек `spla`(CPU(`POCL`) и GPU(`NVIDIA`)) или `Pregel`+(1, 2, 4,8 Workers) алгоритм Боровки быстрее?

- 15 запусков алгоритма
- Составить информацию о среднем значении и доверительных интервалах
- Визуализировать информацию

Характеристика вычислительной машины

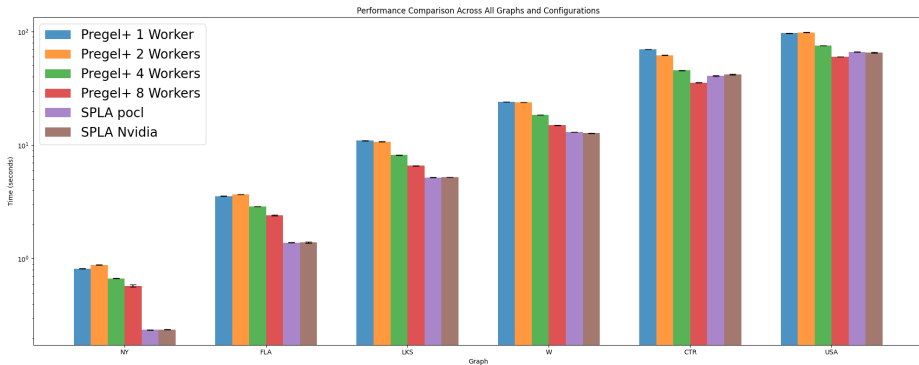
- CPU: AMD Ryzen 7 5800H
 - ▶ 8 ядер
 - ▶ hyperthreading выключен
 - ▶ L1d: 256 KiB (8 instances)
 - ▶ L2: 4 MiB (8 instances)
 - ▶ L3: 16 MiB (1 instance)
- GPU: NVIDIA GeForce GTX 1650 (4096MiB)
- RAM: 16 GB
- OS: Ubuntu 25.10

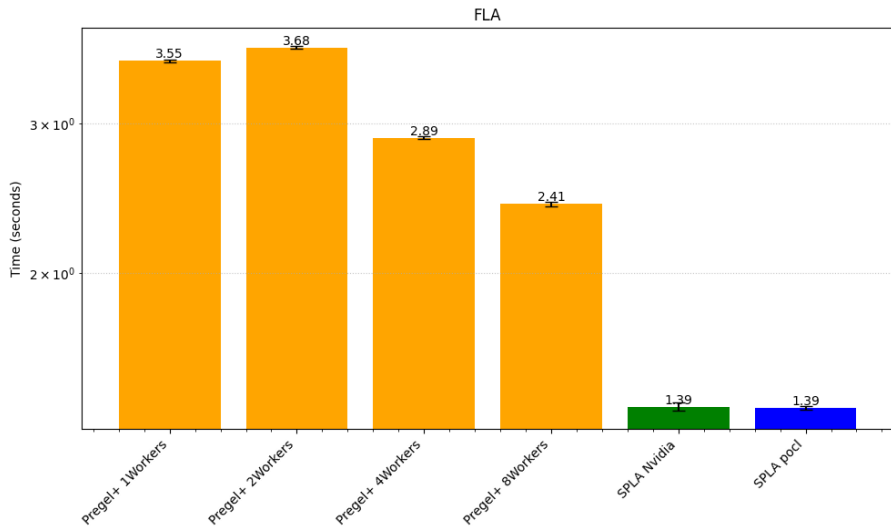
Используемое ПО:

- gcc 15.2.0
- JDK 11
- Hadoop 3.4.2
- MPICH 4.2.1

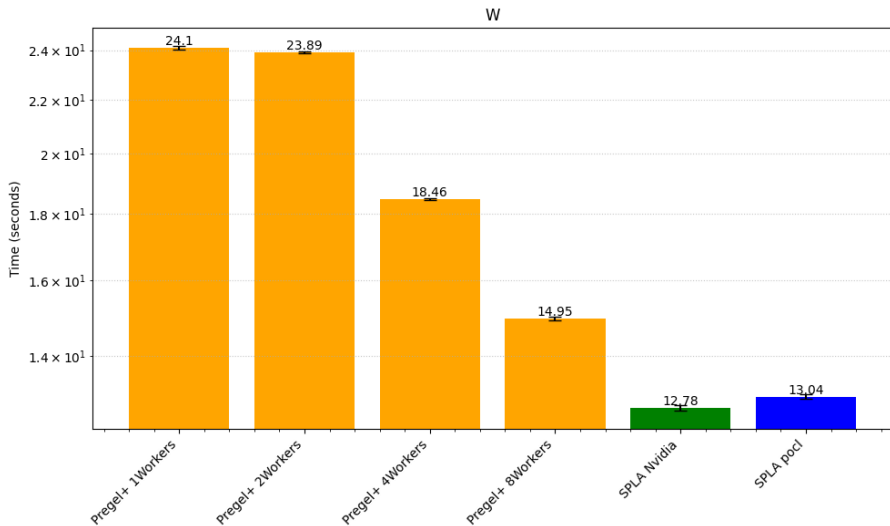
Граф	V	E
USA-road-d.USA	23.9M	58.3M
USA-road-d.CTR	14.1M	34.3M
USA-road-d.W	6.3M	15.2M
USA-road-d.LKS	2.8M	6.9M
USA-road-d.FLA	1.1M	2.8M
USA-road-d.NY	264K	734K

Результаты сравнения

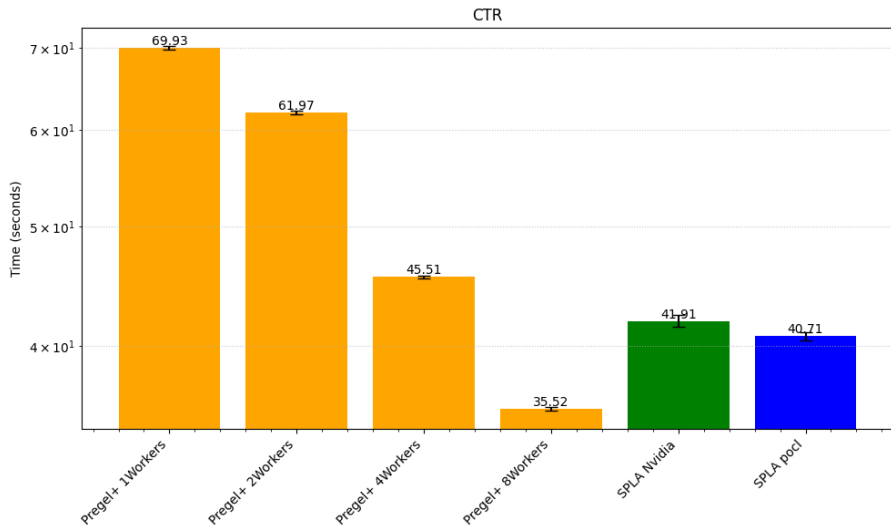




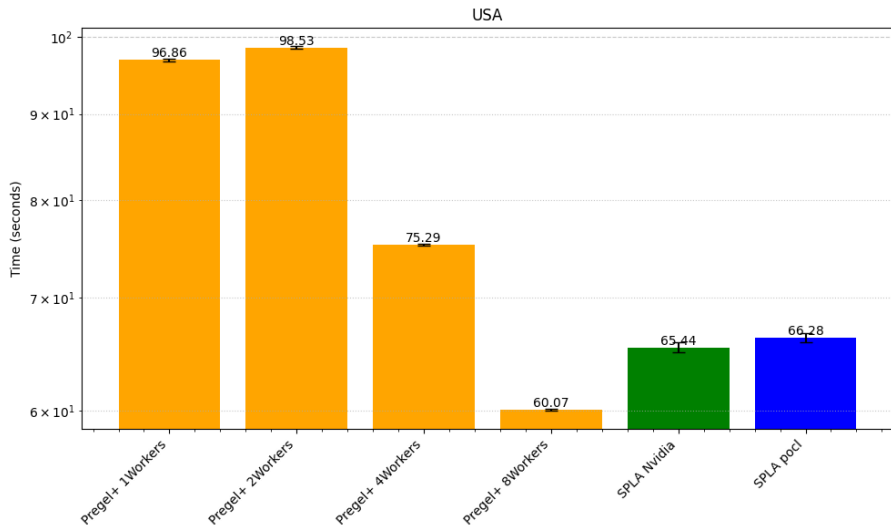
Western USA



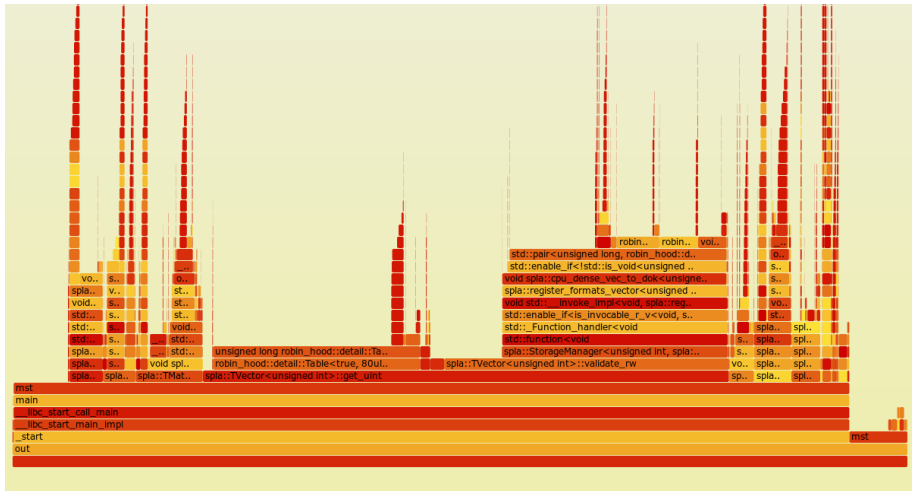
Central USA



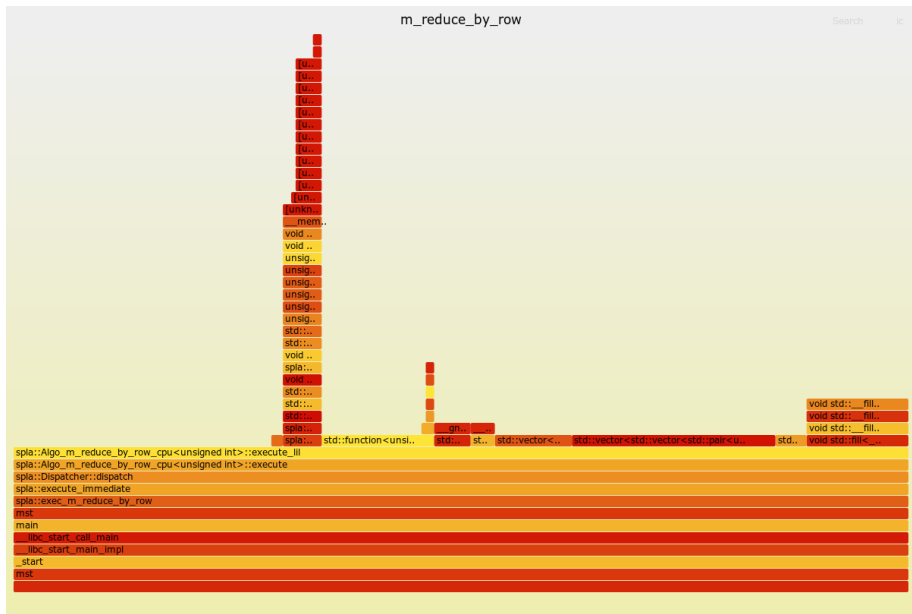
Full USA



SPLA Profiling

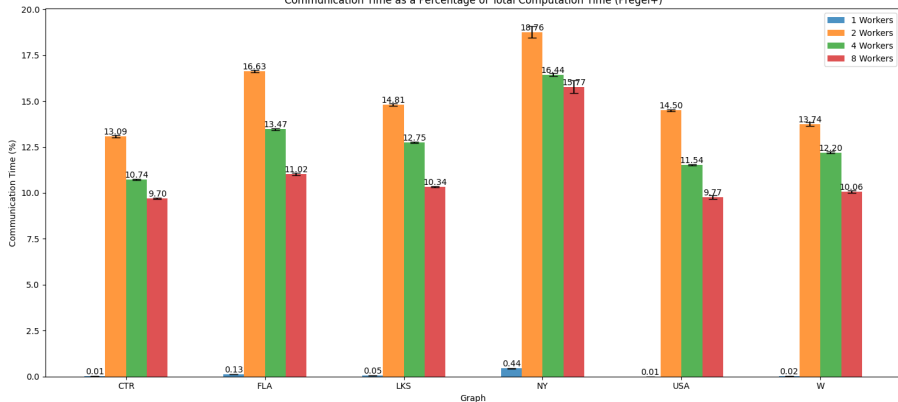


SPLA Profiling



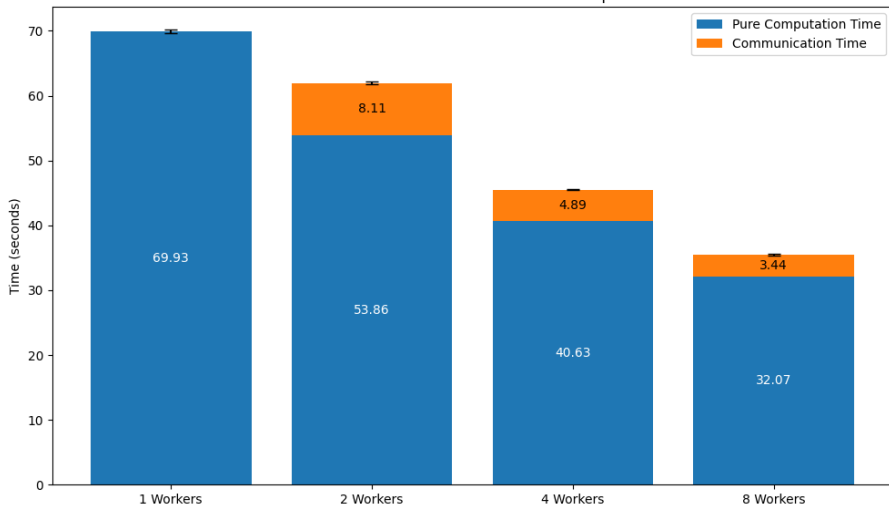
Обмен сообщениями Pregel+

Communication Time as a Percentage of Total Computation Time (Pregel+)

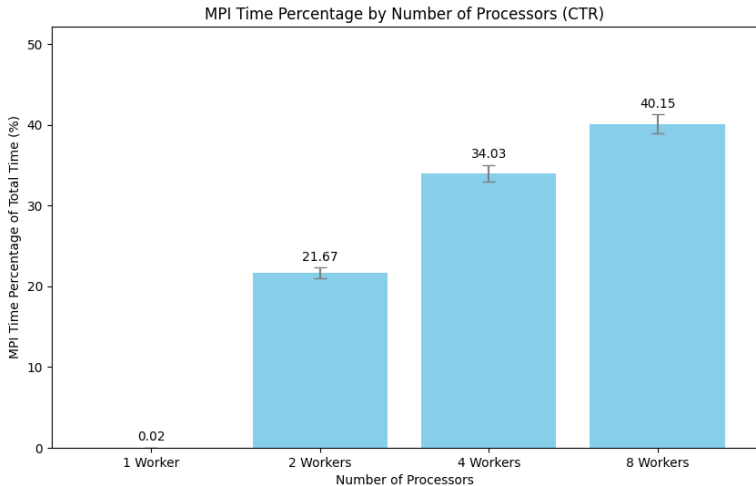


CTR Pregel+

Performance Breakdown for CTR Graph



CTR % of Mpi with Mpip



Анализ результатов

- На маленьких и средних графах из набора spla показывает лучший результат
- На больших самый быстрый Pregel+ в 8 workers
- SPLA с POCL и GPU не показали существенной разницы. Связано это с тем, что отсутствует OpenCL релизация `exes_m_reduce_by_row`
- На SPLA не хватает поддержки пользовательских типов, векторных операций с масками для реализации алгоритма максимально эффективно