



**УНИВЕРСИТЕТ  
ЛОБАЧЕВСКОГО**

Институт Информационных технологий, математики и механики  
Кафедра Высокопроизводительных вычислений и системного  
программирования

# **Кроссплатформенная реализация маскированного умножения разреженных матриц**

Устинов А.В., Пирова А.Ю., Мееров И.Б.

Москва, 2024 г.

# GraphBLAS

- ❑ Алгоритмы на графах применяются для анализа сетей в различных предметных областях (химия, биология, медицина, транспортные и социальные сети)
- ❑ Трудности реализации алгоритмов на графах:
  - Многообразие форматов хранения графов
  - Низкая производительность
  - Сложность эффективной параллельной реализации
- ❑ Частичное решение – [стандарт GraphBLAS](#)
- ❑ Стандарт определяет набор матрично-векторных операций для реализации алгоритмов на графах
- ❑ Существующие реализации: SuiteSparse:GraphBLAS (CPU), GBTL (CPU), GraphBLAST (GPU)

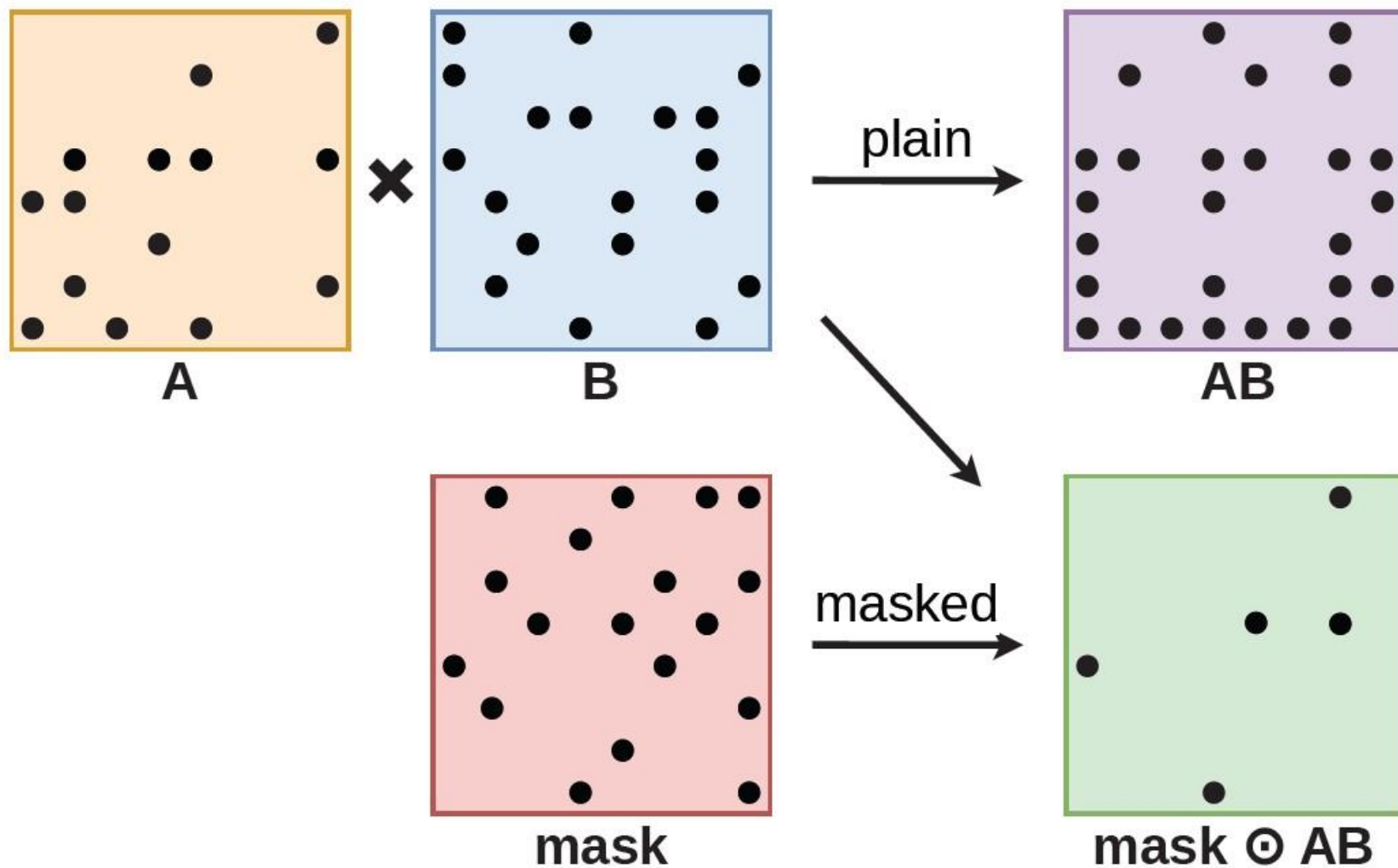


# MaskedSpGEMM

- ❑ Маскированное умножение разреженных матриц (MaskedSpGEMM) – умножение матриц с обращением в ноль тех элементов произведения, которые равны нулю в маске
- ❑ Важный компонент для ряда алгоритмов на графах:
  - Поиск в ширину
  - Подсчёт числа треугольников
  - Кластеризация графа
- ❑ Эффективные реализации используют структуру маски в процессе умножения, чтобы уменьшить число операций и затраты памяти
- ❑ **Цель работы** – разработка кроссплатформенной реализации алгоритма маскированного умножения разреженных матриц и его приложений к алгоритмам на графах



# MaskedSpGEMM. Пример



# Приложения маскированного умножения матриц. Подсчёт числа треугольников

- Были реализованы два алгоритма:
  - Алгоритм подсчёта числа треугольников в графе (Гилберт, 2015)
  - Алгоритм Брандеса вычисления степени посредничества (Брандес, 2001)
- Подсчёт числа треугольников используется в анализе социальных сетей для выявления сообществ и вычисления характеристик сетей
- Применение маскированного умножения матриц в подсчёте числа треугольников:
  - Число треугольников равно сумме элементов матрицы  $L \odot L^2$ , где  $L$  – нижнетреугольная часть матрицы смежности



# Степень посредничества вершины

- Степень посредничества используется в анализе социальных и транспортных сетей, в биоинформатике для определения важности вершин (больше – важнее):

$$C_B(v) = \sum_{s,t \in V \setminus \{v\}} \frac{\sigma_{st}(v)}{\sigma_{st}}, \quad (1)$$

где  $\sigma_{st}$  - число всех кратчайших путей из  $s$  в  $t$ ,  $\sigma_{st}(v)$  - число кратчайших путей из  $s$  в  $t$ , проходящих через  $v$

- Для вычисления степени посредничества используется алгоритм Брандеса
- Основной компонент алгоритма - поиск в ширину из нескольких вершин, который можно реализовать как маскированное умножение разреженных матриц



# Методика проведения экспериментов

- ❑ Параметры тестового окружения (узел кластера Лобачевский):

Процессор	Два 8-ядерных процессора Intel Xeon CPU E5-2660 2.2 GHz
Память	64 GB
Графический процессор	Nvidia A100 40 ГБ
Операционная система	Linux CentOS 6.4
Компиляторы	Intel C++ Compiler 2023, CUDA Toolkit 11.2

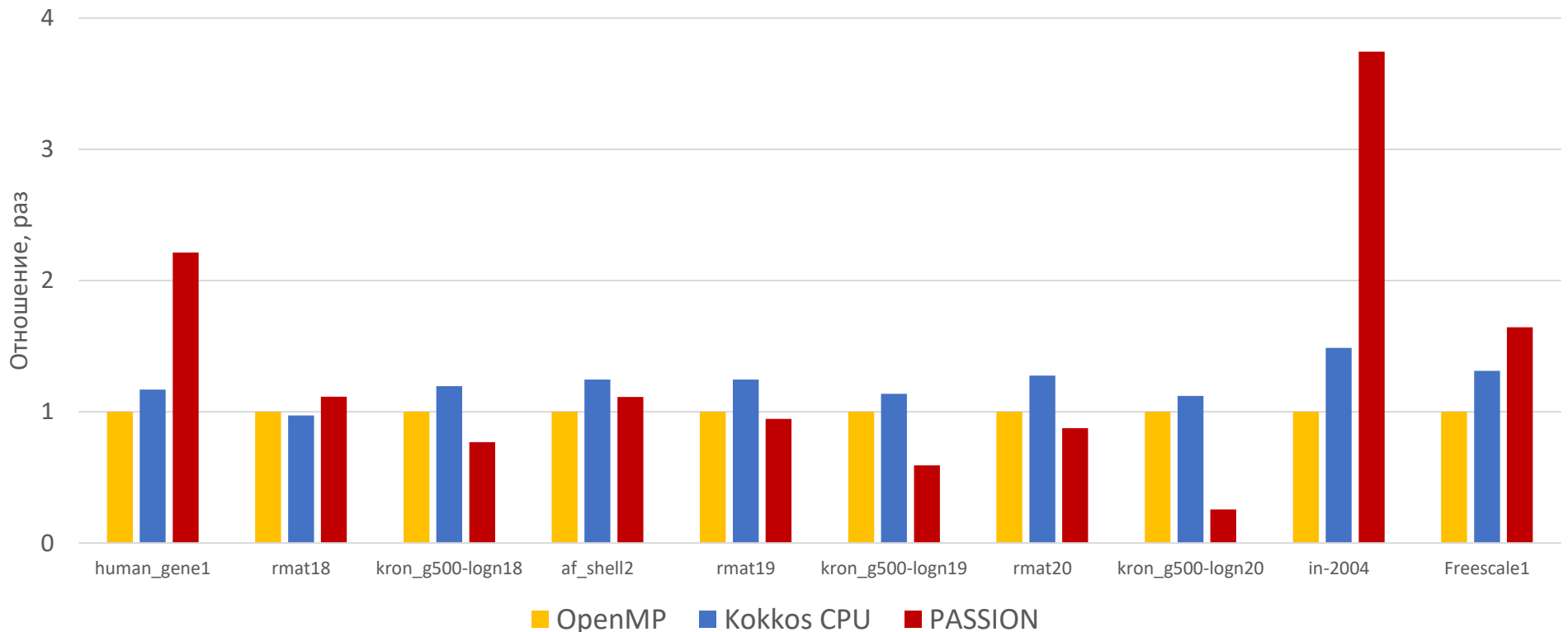
- ❑ Эксперименты проводились на CPU и GPU
- ❑ Сравнивались реализации:
  - Эталонная (PASSION) от авторов алгоритма маскированного умножения матриц: <https://github.com/PASSIONLab>
  - Собственная реализация с использованием OpenMP (OpenMP)
  - Собственная реализация с использованием Kokkos (Kokkos)
- ❑ В тестах использовались матрицы размерности от 20 тыс. до 8 млн. строк из коллекции SuiteSparse и генератора R-MAT



# Вычислительные эксперименты.

## Подсчёт числа треугольников на CPU

- ❑ Относительное время работы на 16 потоках
- ❑ Kokkos отстаёт от OpenMP не более чем в 1,5 раза, на ряде тестов обгоняет эталонную реализацию

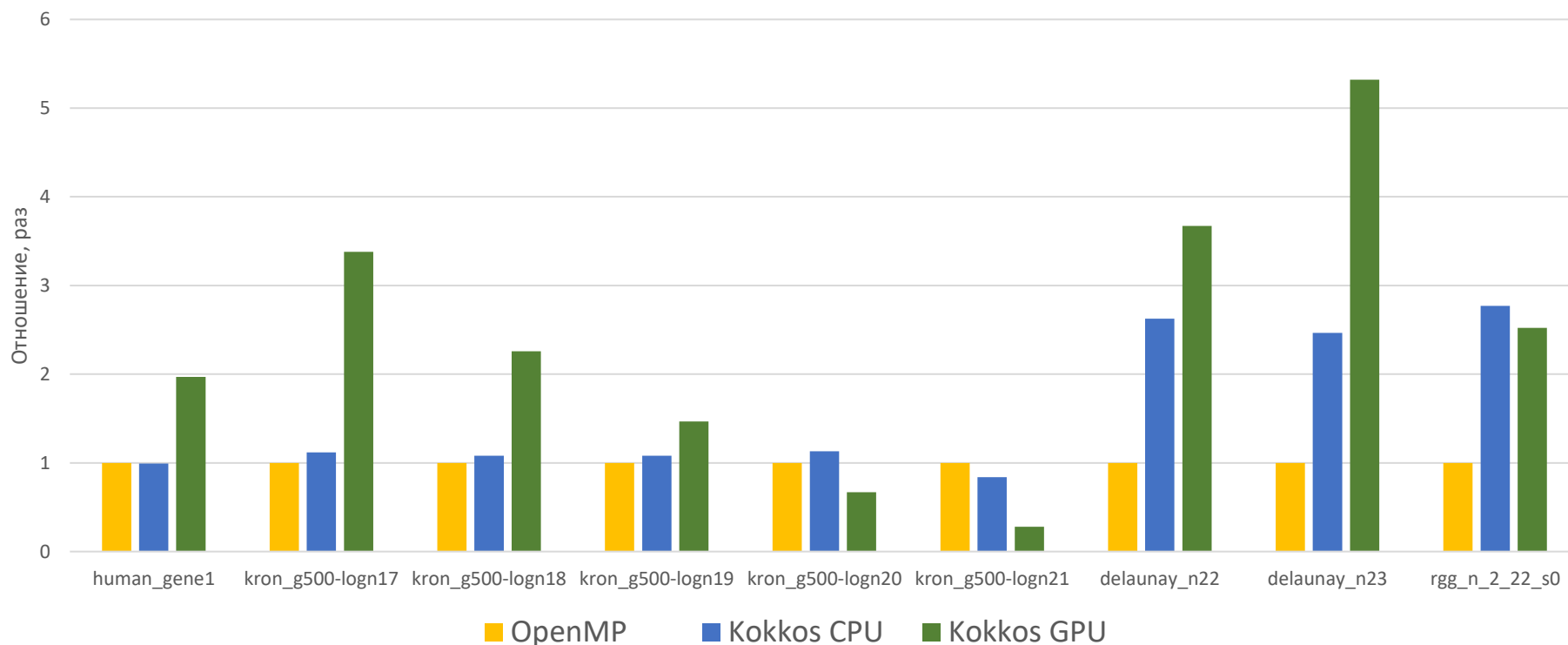




# Вычислительные эксперименты.

## Подсчёт числа треугольников на различных платформах

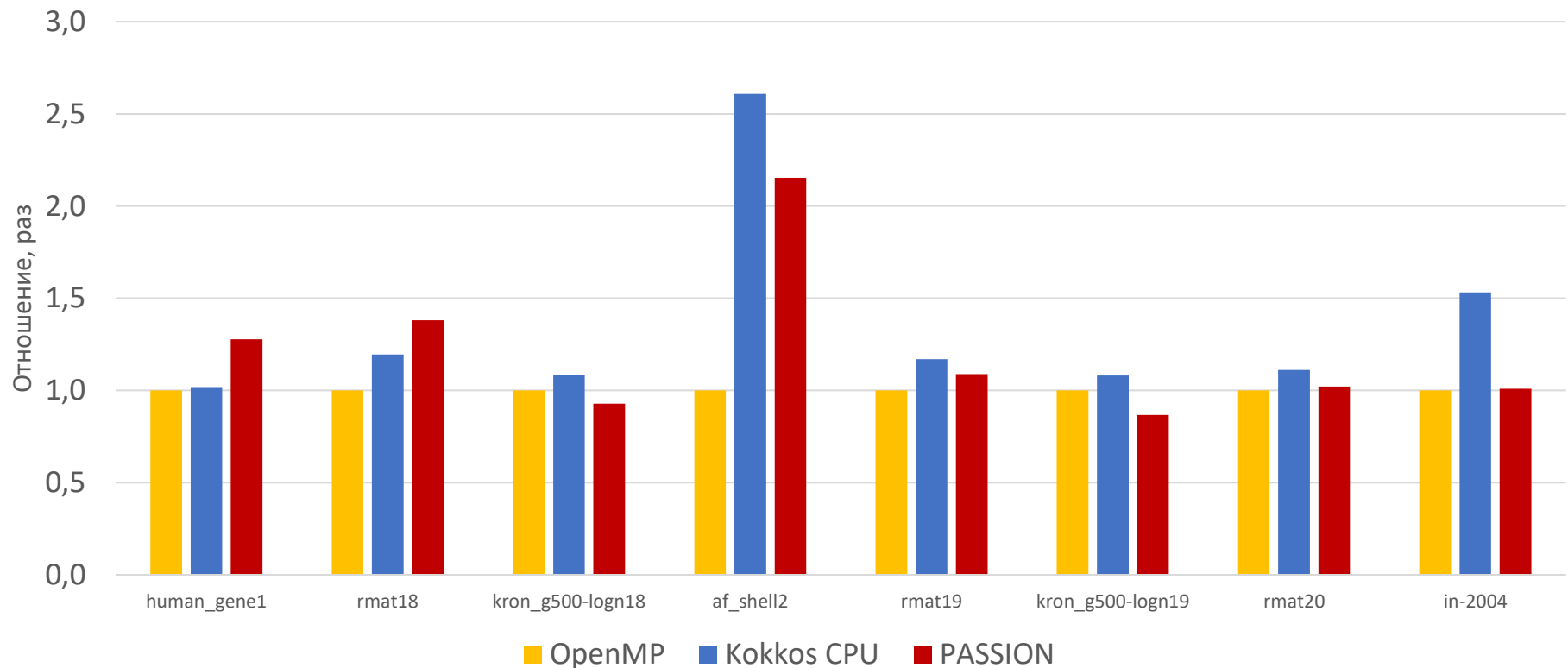
- ❑ Относительное время работы
- ❑ Kokkos на GPU работает быстрее других реализаций на графах Кронекера большой размерности



# Вычислительные эксперименты.

## Алгоритм Брандеса на CPU

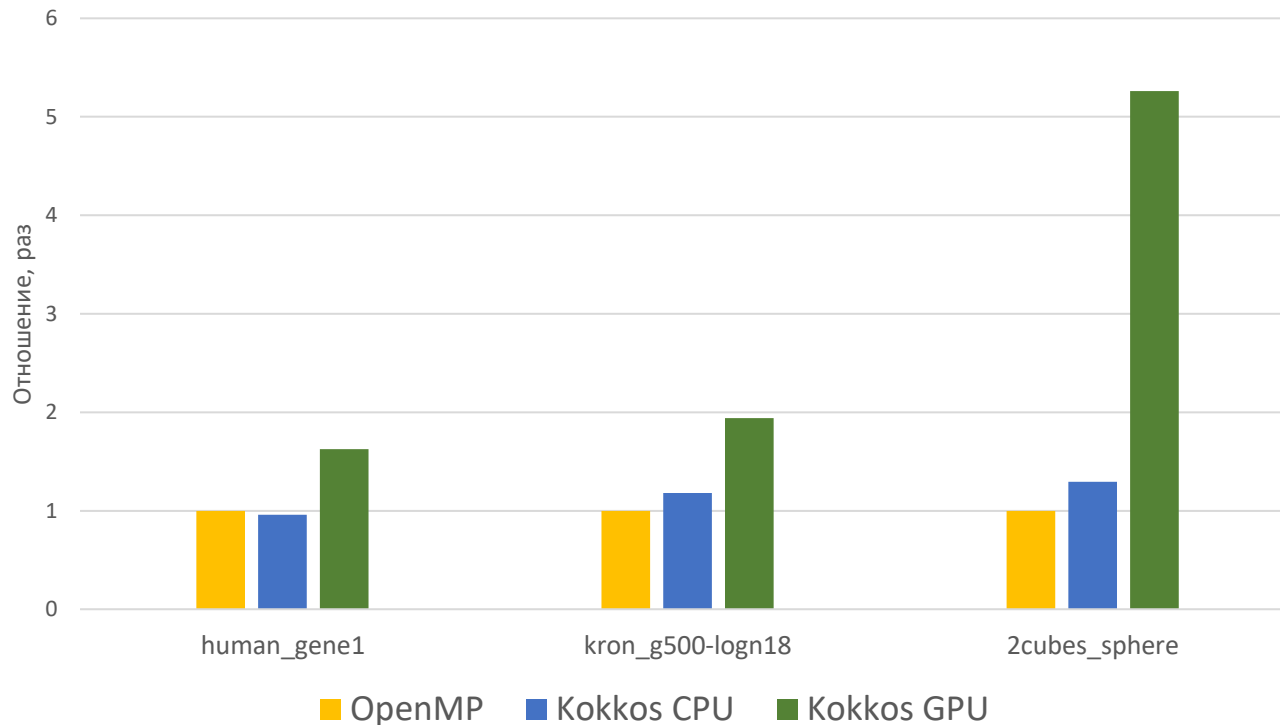
- ❑ Относительное время работы на 16 потоках
- ❑ Kokkos отстаёт от OpenMP не более чем в 1,5 раза на большинстве тестов, обгоняет PASSION на 2 графах из 8



# Вычислительные эксперименты.

## Алгоритм Брандеса на различных платформах

- ❑ Относительное время работы
- ❑ Kokkos на GPU работает значительно медленнее, чем на CPU



# Заключение

- Результаты:
  - Разработана кроссплатформенная реализация алгоритма маскированного умножения разреженных матриц на Kokkos
  - Реализованы алгоритмы подсчёт треугольников и вычисления степени посредничества, использующие маскированное умножение
  - Kokkos на CPU показывает производительность, сравнимую с OpenMP, во многих случаях оказывается быстрее эталонной реализации
  - Kokkos на GPU работает быстрее других реализаций на малом числе тестов
- Направления дальнейших исследований:
  - Изучение вопроса об экономии памяти в реализации алгоритма маскированного умножения
  - Возможность интегрирования маскированного умножения в библиотеку Kokkos Kernels



Спасибо за внимание!

