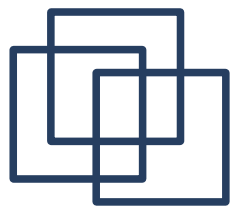


Использование GPU для ускорения поиска в ширину на графах

Черноскутов Михаил

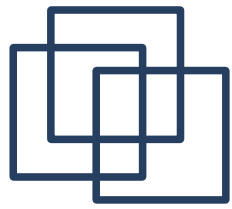
Отдел вычислительной техники ИММ УрО РАН

E-mail: mach@imm.uran.ru



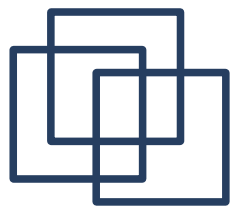
Актуальность

- GPU используются для вычислений общего назначения: «Computing Intensive» задачи
- Рост популярности «Data Intensive» задач
 - Биоинформатика
 - Информационная безопасность
 - Анализ социальных сетей и т. д.
- Пример: обход графа в ширину



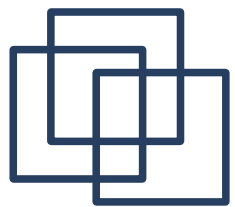
Обход графа в ширину

- Широкое применение
- Простота реализации
- Универсальная метрика
 - Используется как вычислительное ядро теста Graph500
- Описание
 - Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ. - 2-е изд. - М.: «Вильямс», 2006. - 1296 с.

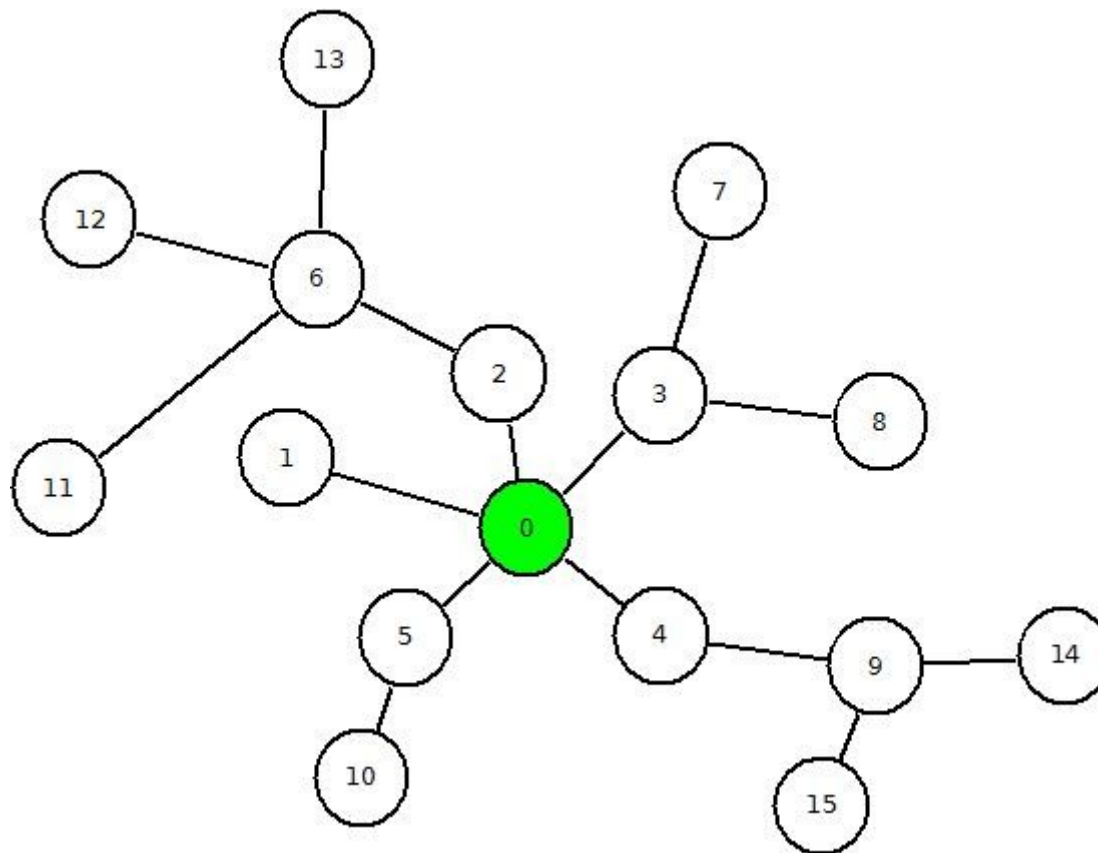


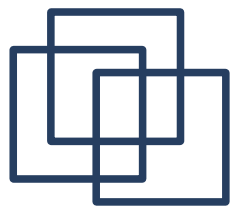
Цель и задачи

- Цель
 - Оценить эффективность использования GPU при решении задачи обхода графа в ширину
- Задачи
 - Реализовать на CPU и GPU
 - Сравнить производительности CPU и GPU
 - Сделать выводы

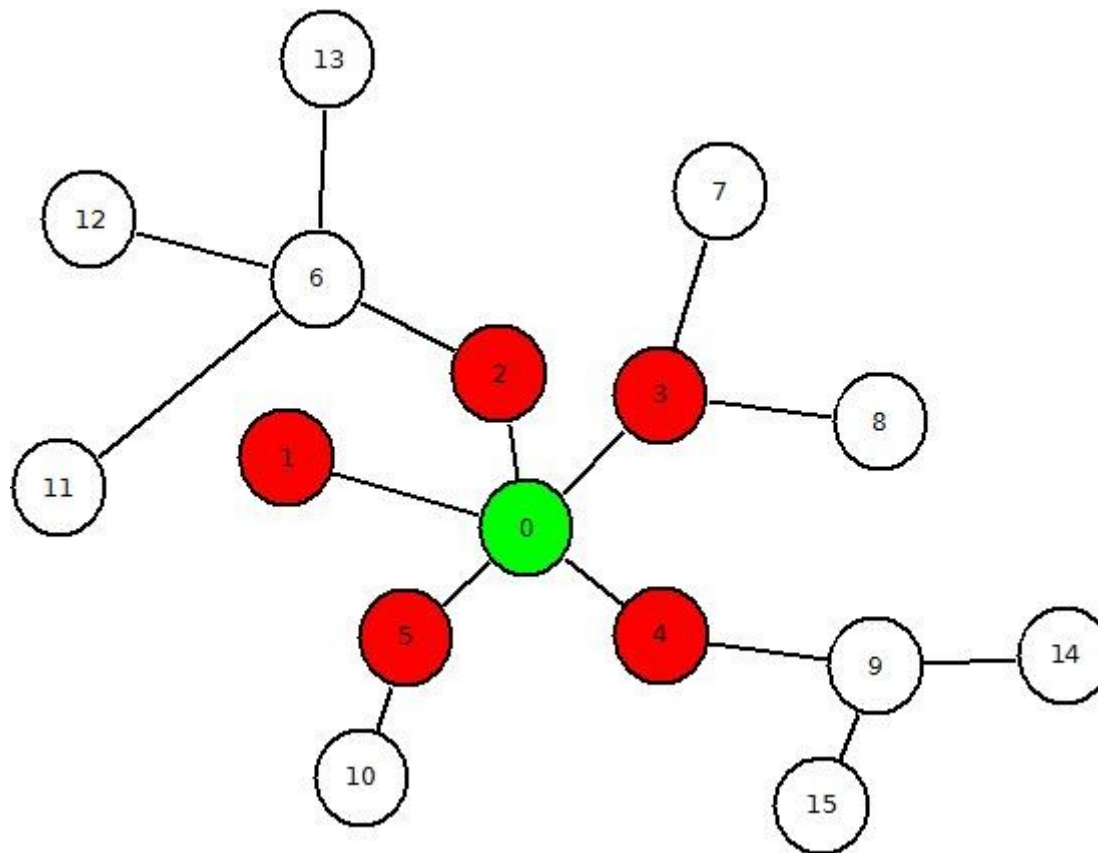


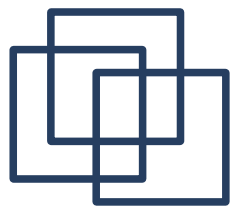
Обход графа в ширину



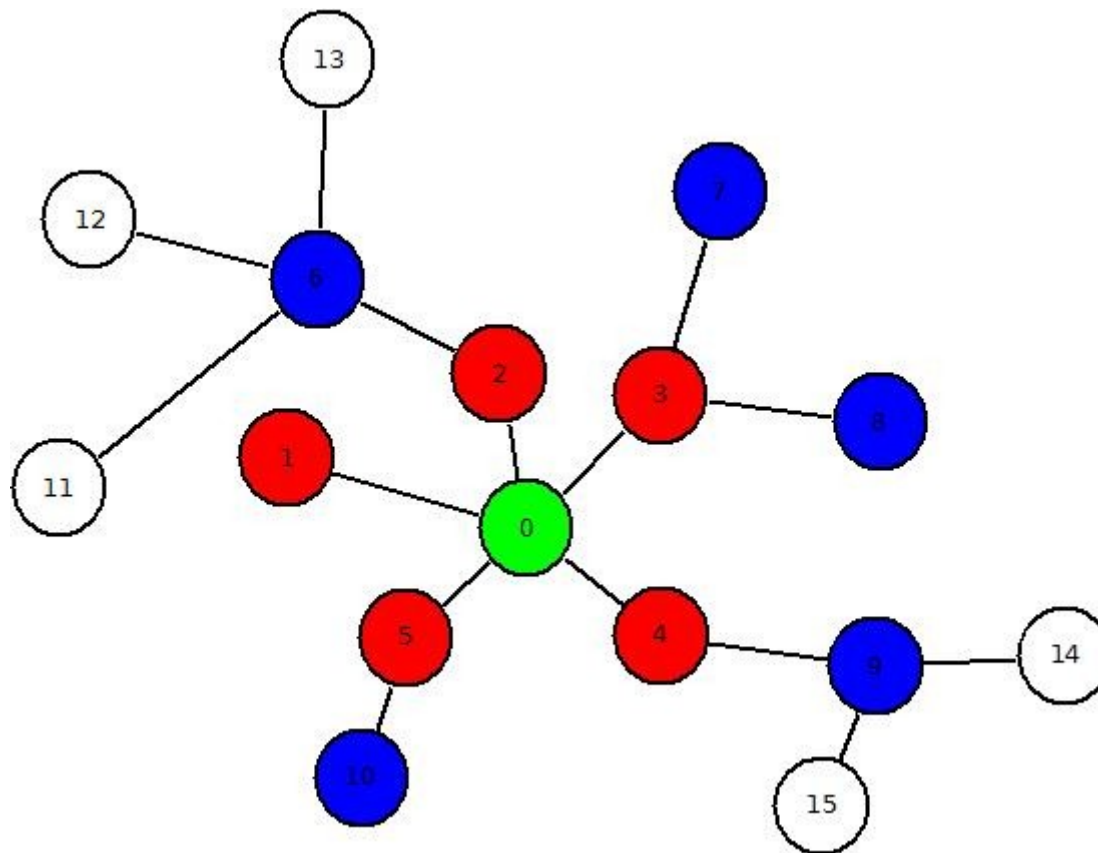


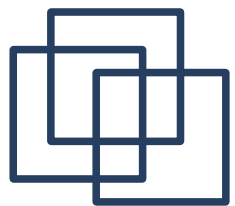
Обход графа в ширину



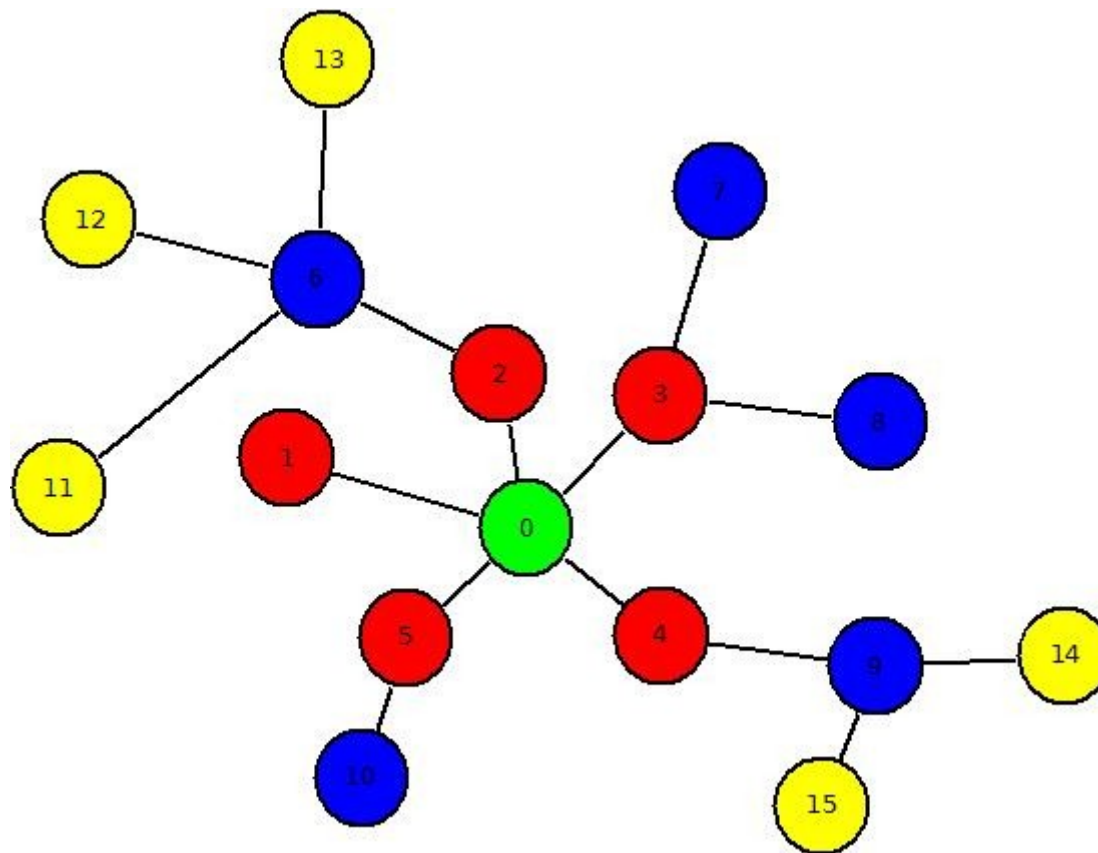


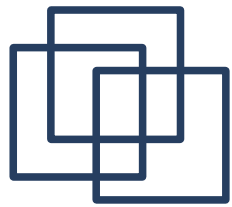
Обход графа в ширину





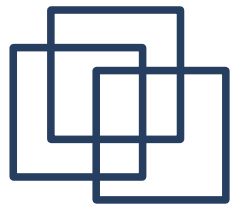
Обход графа в ширину





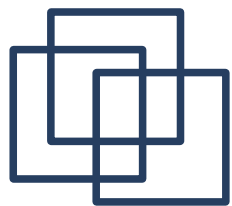
Обеспечение корректности тестирования

- Использование одинаковых принципов распределения нагрузки по вычислительным элементам
- Обеспечение одинакового времени доступа ко всем элементам графа



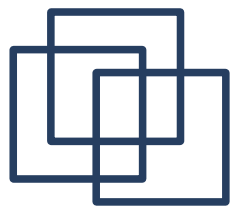
Требования к тесту

- Граф должен полностью заполнять память одного GPU
- Для CPU и GPU используется одинаковый объем памяти
- Необходимо обеспечить произвольный доступ в память
 - Выбор соседей у вершин графа случайным образом



Входные и выходные данные для алгоритма

- Входные данные
 - Граф с неразмеченными вершинами
 - Выходные данные
 - Граф с размеченными вершинами
 - Характеристики графа
 - У каждой вершины имеется от 5 до 70 исходящих ребер
 - Число входящих ребер определяется случайным образом
 - Распределение вершин
 - GPU: 1 нить — 1 вершина
 - CPU: 1 ядро — диапазон вершин
-



Используемые графы

Количество исходящих из вершины ребер, шт.	Количество вершин, шт.	Размер графа, ГБ
5	66 060 288	2,21
15	35 651 584	2,52
25	23 068 672	2,49
35	16 777 216	2,44
45	13 631 488	2,49
55	11 534 336	2,54
65	9 437 184	2,43

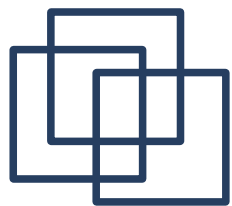
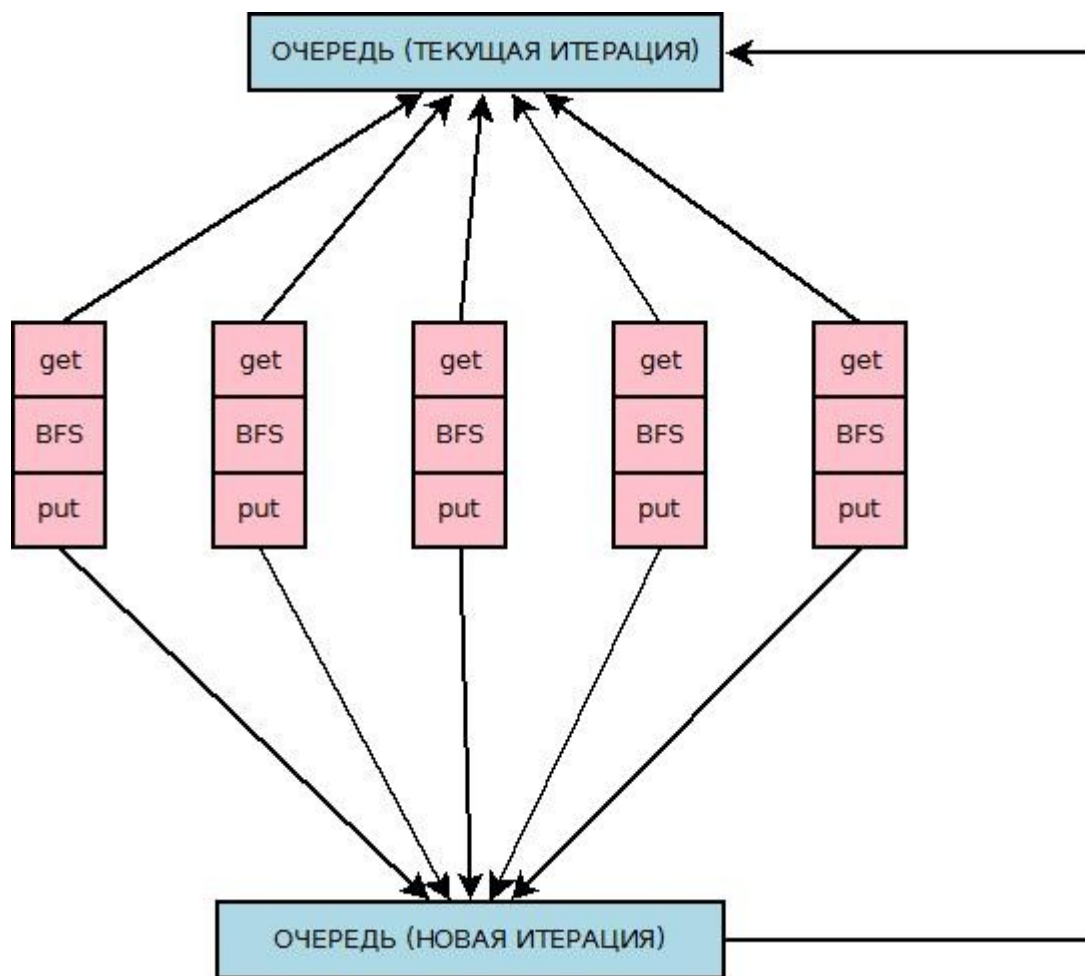
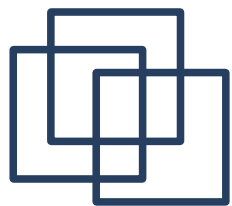


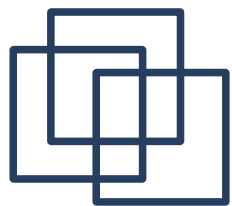
Схема обхода графа



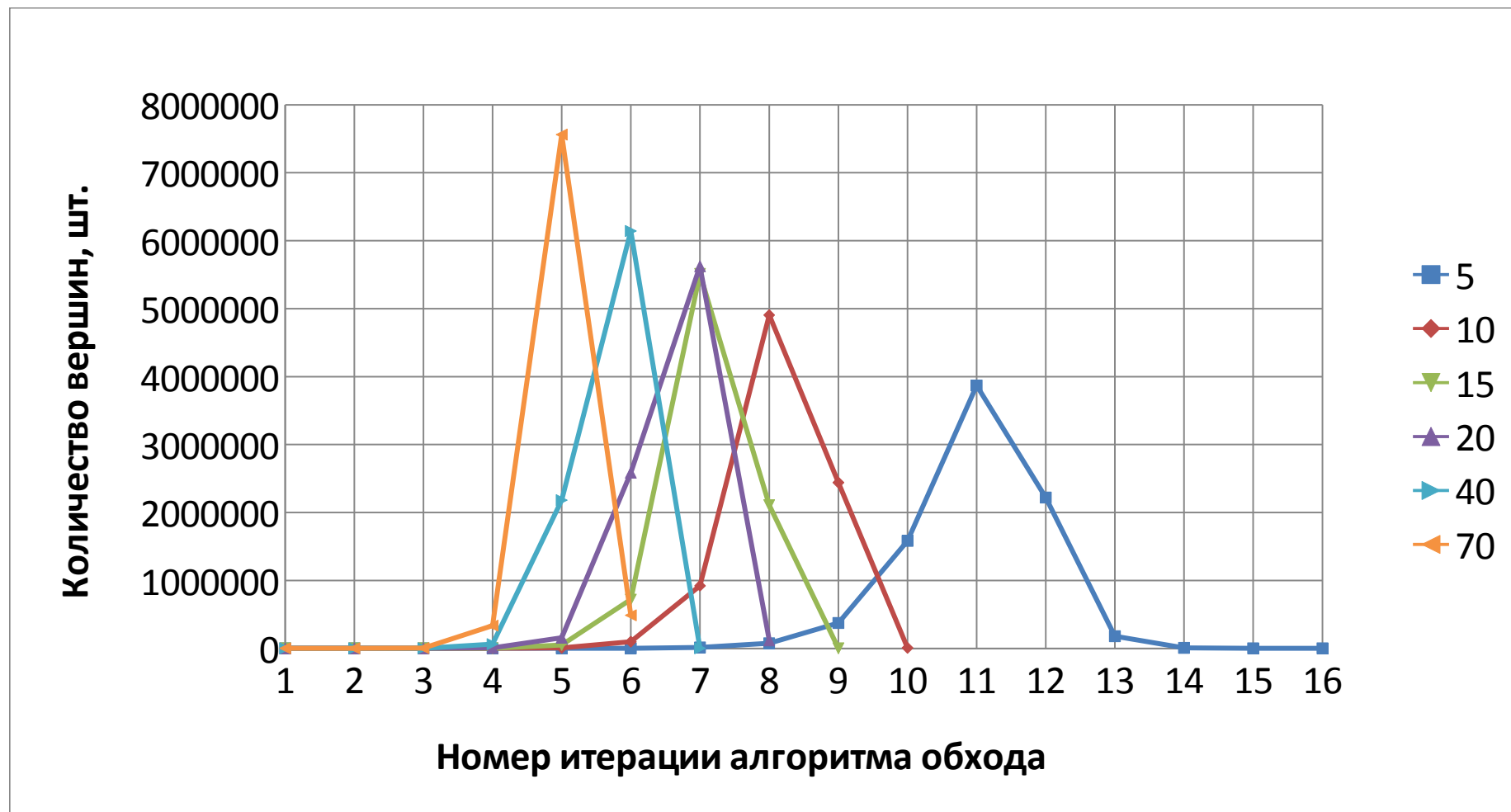


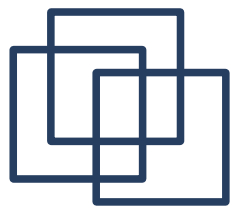
Аппаратное обеспечение

- GPU Nvidia Tesla M2050
 - 448 ядер
 - 2.5 ГБ ОЗУ
- CPU Intel Xeon X5675
 - 6 ядер
 - 48 ГБ ОЗУ

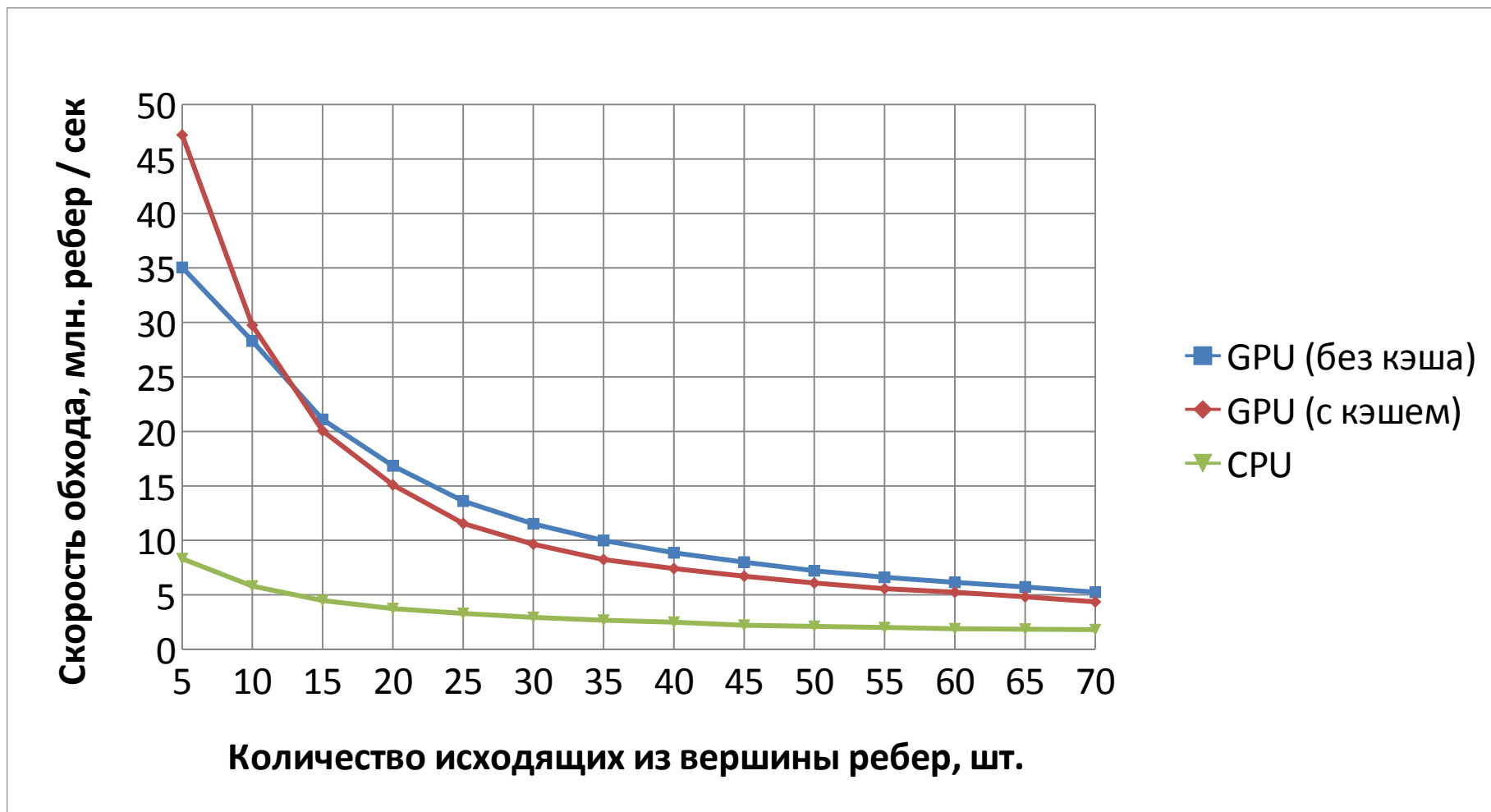


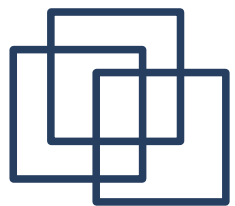
Особенности задачи обхода графа



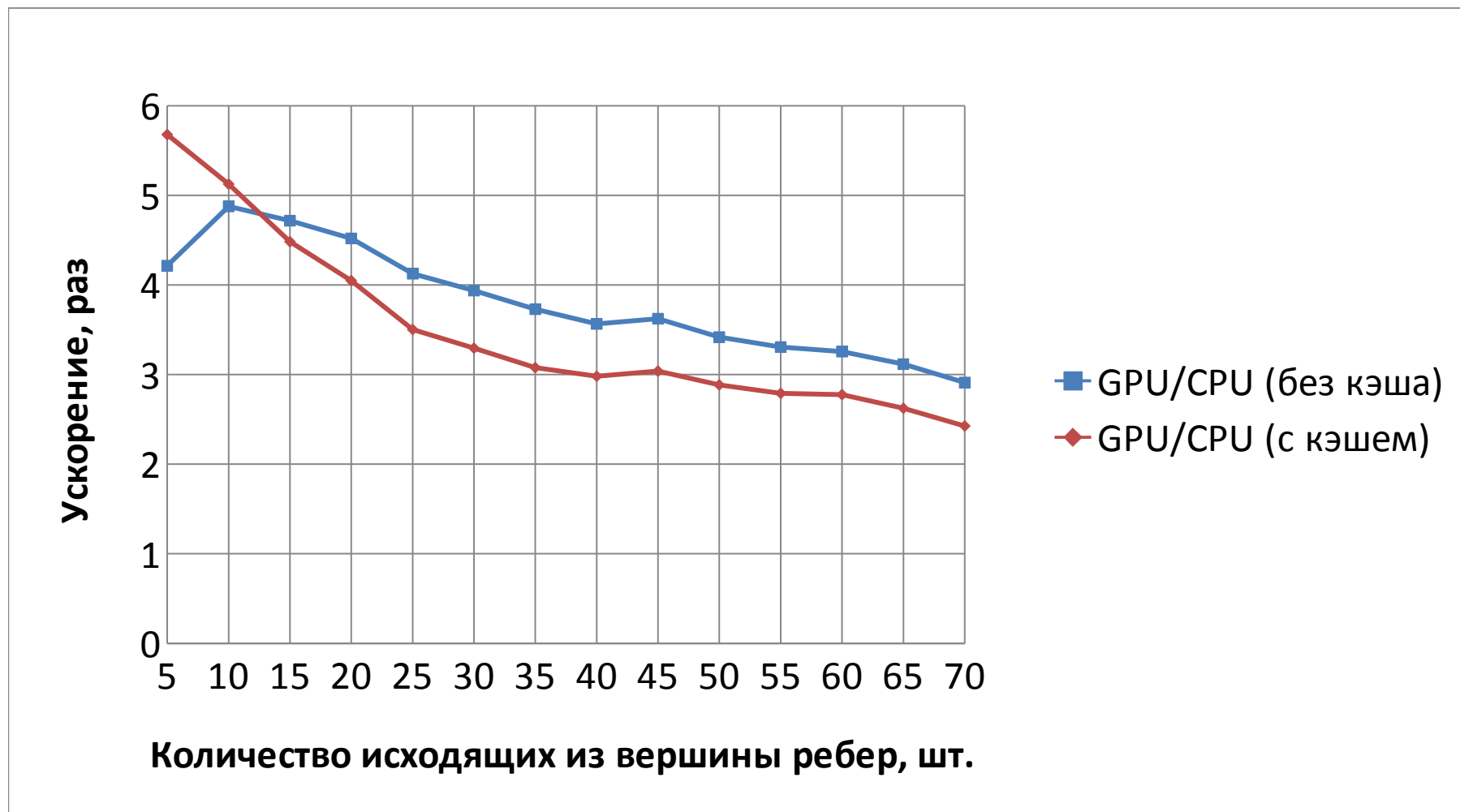


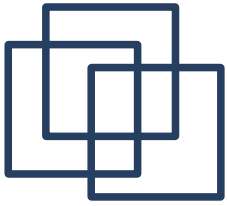
Результаты эксперимента





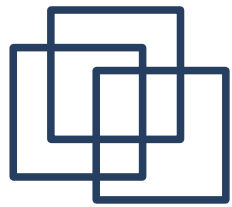
Достигнутое ускорение





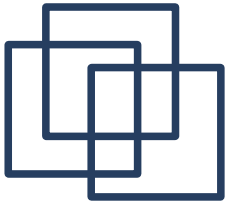
Выводы

- Реализован алгоритм обхода графов в ширину для CPU и GPU
 - Проведено тестирование реализации алгоритма для графов с различным числом вершин и ребер
 - GPU опережает CPU по производительности от 2.4 до 5.7 раз на графах, заполняющих всю доступную память на GPU
 - Кэш GPU не дает выигрыша в производительности для задач с произвольным доступом в память
-



Направления дальнейшей работы

- Исследование поведения алгоритма при обходе графов с неоднородной структурой
 - Балансировка нагрузки на потоки
- Сравнение эффективности Nvidia Kepler и Intel MIC



Вопросы?