Задание 1. Алгоритм Борувки.

Папченко Анастасия, группа 538

1 Постановка задачи

2 Компиляция и запуск

1. cd 2022-plgp-mst
2. g++ graph_tools.o mst_reference.o gen_valid_info.o -o gen_valid_info -O3 -lrt
3. g++ validation.o graph_tools.o -o validation -O3 -lrt
4. g++ main.o mst_reference.o graph_tools.o -o mst_reference -O3 -lrt
5. mpic++ -O3 -Wall -o mst.o -c mst.cpp
6. mpic++ main.o mst.o graph_tools.o -o mst -O3 -lrt

mpirun -np <p> ./mst -in <name>
<p> - количество процессов; <name> - название графа.

Флаг –oversubscribe когда <p> привышает количестко процессоров (mpirun –oversubscribe -np 8 ./mst -in rmat-10)

3 Генерация и тестирование графов

Генерация обычного графа:
./gen_random -n <n> -m <m> -out <name>

Генерация RMAT графов:
./gen_RMAT -s <s>

Запуск эталонного тестирования:
./mst_reference -in <name>

<n> - количество вершин; <m> - количество ребер; <s> - степень двойки == количество вершин; <name> - имя файла для сохранениия или тестирования.

4 Спецификация системы

Процессор: AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz
Число вычислительных ядер: 4

5 Результаты выполнения

Генерация графа - ./gen_RMAT -s 10

Возьмем для примера $RMAT$ граф с количеством вершин 2^{10}. Результат выполнения ленивого алгоритма (./mst_reference -in rmat-10) представлен на рисунке 1. Алгиритм Борувки с использованием функций send/recv написан в файле mst.cpp. На рисунках 2, 3 и 4 показан результат запуска алгоритма на 1ом, 2х и 4х процессах соответственно. На рисунке 5 демонстрируется проверка программы на валидность (./validation -in_graph rmat-10 -in_result rmat-10.mst -in_valid rmat-10.vinfo)

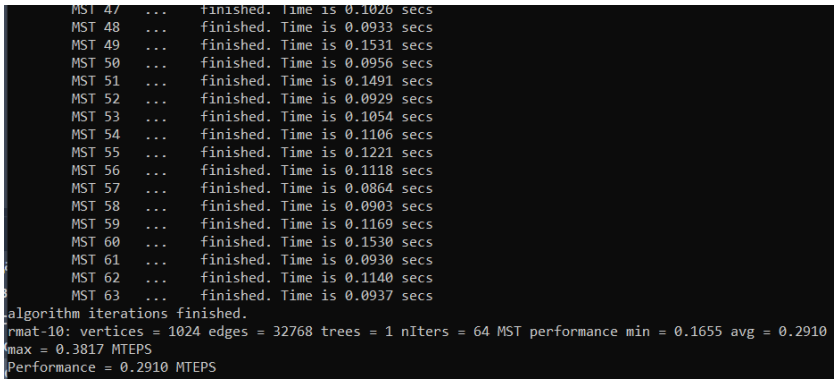

Рис. 1: Результат выполнения ленивого алгоритма.

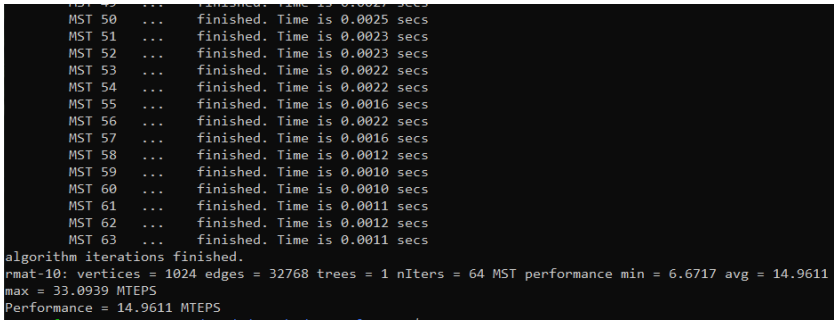

Рис. 2: Запуск mst.cpp на 1ом процессе.

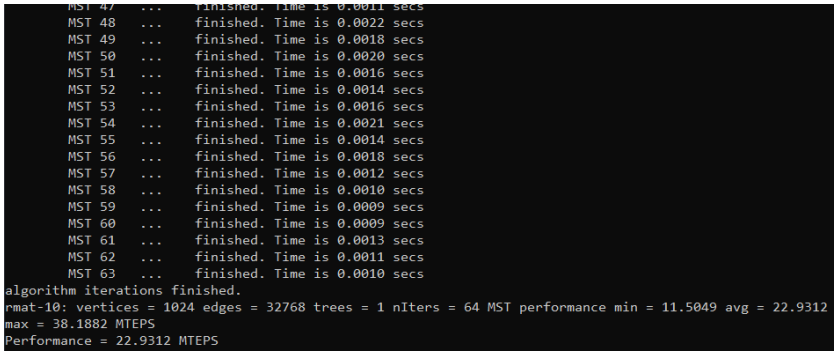

Рис. 3: Запуск mst.cpp на 2х процессах.

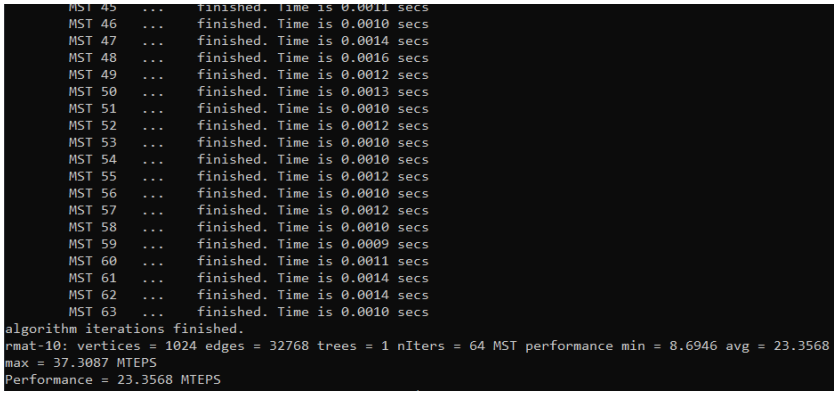

Рис. 4: Запуск mst.cpp на 4х процессах.

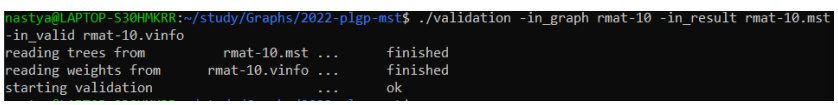

Рис. 5: Проверка на валидность.