

## Задание 2. Алгоритм Борувки. Масштабируемость по памяти

Папченко Анастасия, группа 538

### 1 Компиляция и запуск на POLUS

1. `cd 2022-plgp-mst`
2. `module load SpectrumMPI/10.1.0`
3. `g++ graph_tools.o mst_reference.o gen_valid_info.o -o gen_valid_info -O3 -lrt`
4. `g++ validation.o graph_tools.o -o validation -O3 -lrt`
5. `g++ main.o mst_reference.o graph_tools.o -o mst_reference -O3 -lrt`
6. `mpic++ -std=c++11 -O3 -Wall -o mst.o -c mst.cpp`
7. `mpic++ main.o mst.o graph_tools.o -o mst -O3 -lrt`

`mpisubmit.pl -p <p> -t <t> -w 00:20 ./mst -in <name>`

<p> - количество процессов; <t> - количество процессов на процессе  
<name> - название графа.

### 2 Генерация и тестирование графов

**Генерация обычного графа:**

`./gen_random -n <n> -m <m> -out <name>`

**Генерация RМAТ 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 сохранения или тестирования.

## 3 Спецификация системы

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

## 4 Результаты выполнения и описание

Генерация графа - `./gen_RMAT -s 10`

- `MPI_Scatter` / `MPI_Scatterv`: Используются в `partition_graph()` для распределения по процессам;
- `MPI_Bcast`: рассылка флага объединения и массива компонент `parent`
- `MPI_Gather`: в `MST()` используется для сбора количества рёбер от каждого процесса к процессу 0;
- `MPI_Send` / `MPI_Recv`: используются в `MST()` для отправки локальных минимальных рёбер (структур типа `Edge`) от всех процессов к процессу 0;

Ниже на рисунках 1 - 3 показаны результаты выполнения алгоритма на 2, 4 и 8 процессах соответственно, по 2 процесса на каждый процесс. На рисунке 4 результаты проверки валидности.

```
MST 63 ... finished. time is 0.0008 secs
algorithm iterations finished.
rmat-10: vertices = 1024 edges = 32768 trees = 1 nIters = 64 MST performance min = 35.7405 avg = 41.0151
max = 41.9214 MTEPS
Performance = 41.0151 MTEPS
```

Рис. 1: `mpisubmit.pl -p 2 -t 2 -w 00:20 ./mst -in rmat-10`

```

... finished. Time is 0.0005 secs
algorithm iterations finished.
rmat-10: vertices = 1024 edges = 32768 trees = 1 nIters = 64 MST performance min = 57.4638 avg = 68.3203
max = 69.9227 MTEPS
Performance = 68.3203 MTEPS

```

Рис. 2: `mpisubmit.pl -p 4 -t 2 -w 00:20 ./mst -in rmat-10`

```

algorithm iterations finished.
rmat-10: vertices = 1024 edges = 32768 trees = 1 nIters = 64 MST performance min = 58.3251 avg = 77.6166
max = 80.9606 MTEPS
Performance = 77.6166 MTEPS

```

Рис. 3: `mpisubmit.pl -p 8 -t 2 -w 00:20 ./mst -in rmat-10`

```

[edu-cmc-sq122-28@polus-ib 2022-plgp-mst]$ ./validation -in_graph rmat-10 -in_result rmat-10.mst -in_valid rmat-10.vinfo
reading trees from      rmat-10.mst ...      finished
reading weights from    rmat-10.vinfo ...    finished
starting validation ...      ok

```

Рис. 4: Проверка на валидность.