**Министерство науки и высшего образования Российской Федерации**
**Федеральное государственное бюджетное образовательное учреждение высшего образования**
**«Московский государственный технический университет имени Н. Э. Баумана**
**(национальный исследовательский университет)»**
**(МГТУ им. Н. Э. Баумана)**

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

# Отчет по практикуму №2 по курсу "Архитектура ЭВМ"

**Тема** Обработка и визуализация графов в вычислительном комплексе Тераграф

**Студент** Сапожков А. М.

**Группа** ИУ7-53Б

**Оценка (баллы)**

**Преподаватель** Попов А. Ю.

Москва — 2022 г.

# Содержание

# Введение

Практикум посвящен освоению принципов представления графов и их обработке с помощью вычислительного комплекса Тераграф. В ходе практикума необходимо ознакомиться с вариантами представления графов в виде объединения структур языка C/C++, изучить и применить на практике примеры решения некоторых задач на графах. По индивидуальному варианту необходимо разработать программу хост-подсистемы и программного ядра sw_kernel, выполняющего обработку и визуализацию графов.

# 1 Выполнение лабораторной работы

## 1.1 Индивидуальное задание

**Вариант 19**

Входные данные содержатся в файле с названием simulated_blockmodel_graph_500_nodes.tsv. В каждой строке записаны два номера вершин и вес ребра.

## 1.2 Код программы

Листинг 1.1 – Изменённый фрагмент кода host_main.cpp

```
1  __foreach_core(group, core)
2  {
3      lnh_inst.gpc[group][core]->start_async(__event__(delete_graph));
4  }
5
6
7  unsigned int*
       host2gpc_ext_buffer[LNH_GROUPS_COUNT][LNH_MAX_CORES_IN_GROUP];
8  unsigned int messages_count = 0;
9  unsigned int u, v, w;
10
11 __foreach_core(group, core)
12 {
13     host2gpc_ext_buffer[group][core] = (unsigned
           int*)lnh_inst.gpc[group][core]->external_memory_create_buffer(16
           * 1048576 * sizeof(int)); //2*3*sizeof(int)*edge_count);
14     offs = 0;
15
16     std::ifstream file(argv[3], std::ios::in);
17
18     if (!file.is_open())
19     {
20         std::cout << "Cannot open input file." << std::endl;
21
22         return EXIT_FAILURE;
```

```
23      }
24
25      for (std::string line; std::getline(file, line); )
26      {
27          std::vector<std::string> tokens = split(line, '\t');
28
29          if (tokens.size() != 3)
30          {
31              std::cout << "Incorrect tokens count in file: expected
                    3, got " << tokens.size() << "." << std::endl;
32
33              return EXIT_FAILURE;
34          }
35
36          u = std::stoul(tokens[0]);
37          v = std::stoul(tokens[1]);
38          w = std::stoul(tokens[2]);
39
40          EDGE(u, v, w);
41          EDGE(v, u, w);
42          messages_count += 2;
43      }
44
45      lnh_inst.gpc[group][core]->external_memory_sync_to_device(0, 3
            * sizeof(unsigned int)*messages_count);
46  }
47  __foreach_core(group, core)
48  {
49      lnh_inst.gpc[group][core]->start_async(__event__(insert_edges));
50  }
51  __foreach_core(group, core) {
52      long long tmp =
            lnh_inst.gpc[group][core]->external_memory_address();
53      lnh_inst.gpc[group][core]->mq_send((unsigned int)tmp);
54  }
55  __foreach_core(group, core) {
56      lnh_inst.gpc[group][core]->mq_send(3 *
            sizeof(int)*messages_count);
57  }
58
59
```

```
60 __foreach_core(group, core)
61 {
62     lnh_inst.gpc[group][core]->finish();
63 }
64 printf("Data graph created!\n");
```

## 1.3 Тестирование программного обеспечения

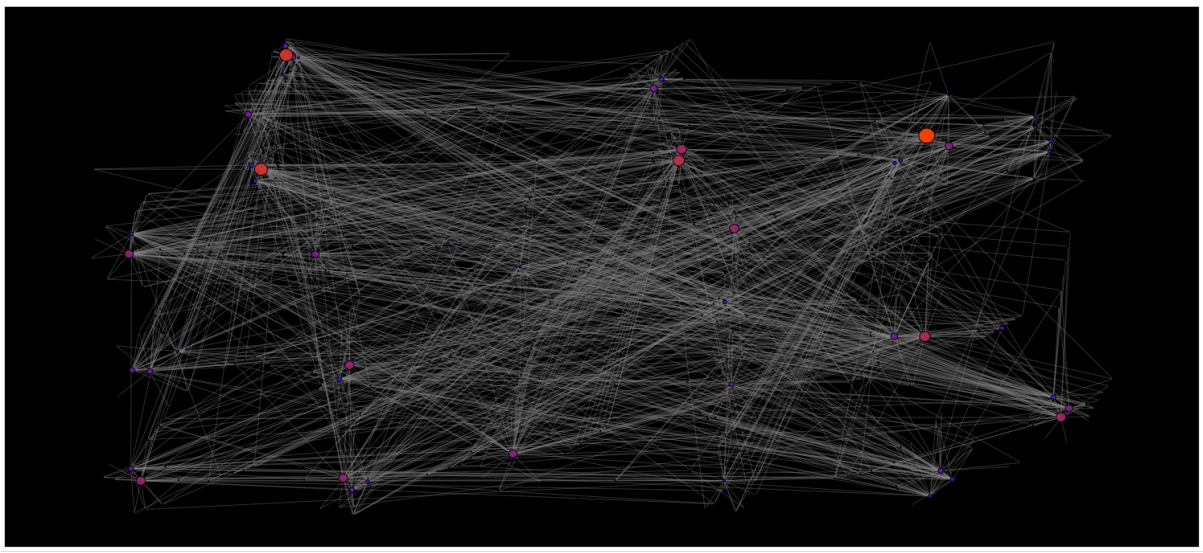

Рисунок 1.1 – Раскладка сообществ с помощью иерархического объединения и укладки в боксы

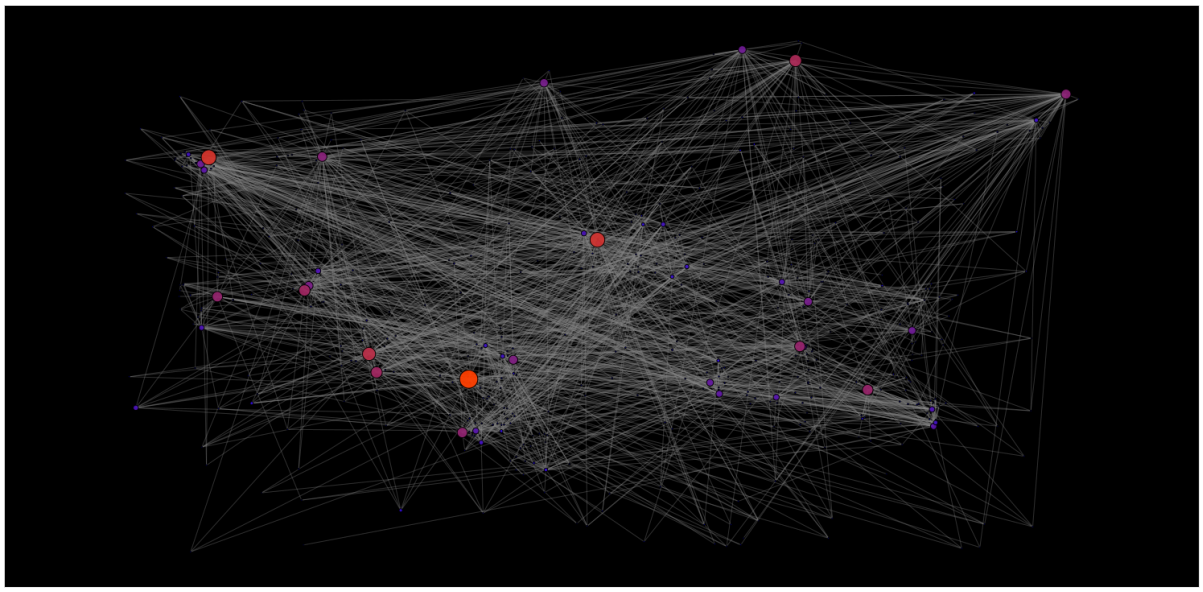

Рисунок 1.2 – Раскладка сообществ с помощью силового алгоритма Фрухтермана-Рейнгольда

# Заключение

В ходе практикума было проведено ознакомление с вариантами представления графов в виде объединения структур языка C/C++, изучены и применены на практике примеры решения некоторых задач на графах. По индивидуальному варианту была разработана программа хост-подсистемы и программного ядра sw_kernel, выполняющего обработку и визуализацию графов.