



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, искусственный интеллект 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 Романов С. К.

Группа ИУ7-55Б

Оценка (баллы) _____

Преподаватель Дубровин Е.Н.

Москва — 2022 г.

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 Задание

1.1 Постановка задачи

Граф, вариант 15.

Для чтения и обработки графа из файла, был добавлен следующий код (Листинги 1 и 2)

1.2 Дополнение к коду

Листинг 1 – Загрузка данных

```
1
2 #ifdef FILE_GRAPH
3
4     __foreach_core(group, core)
5     {
6         Inh_inst.gpc[group][core]
7             ->start_async(__event__(delete_graph));
8     }
9
10
11     unsigned int* host2gpc_ext_buffer    [LNH_GROUPS_COUNT]
12                                           [LNH_MAX_CORES_IN_GROUP];
13     unsigned int edge_count = get_edge_count(SRC_FILE);
14     unsigned int messages_count = 0;
15     unsigned int u, v, w;
16     printf("%d\n", edge_count);
17     __foreach_core(group, core)
18     {
19         host2gpc_ext_buffer[group][core] =
```

```

20         (unsigned int*)Inh_inst.gpc[group][core]
21         ->external_memory_create_buffer
22             (2 * 3 * sizeof(int) * edge_count);
23
24     std::ifstream fin(SRC_FILE);
25     for (int edge = 0; edge < edge_count; ++edge)
26     {
27         fin >> u >> v >> w;
28         EDGE(u, v, w);
29         EDGE(v, u, w);
30         messages_count += 2;
31     }
32     fin.close();
33
34
35     Inh_inst.gpc[group][core]->
36         external_memory_sync_to_device
37             (0, 3 * sizeof(unsigned int)*messages_count);
38 }
39 __foreach_core(group, core)
40 {
41     Inh_inst.gpc[group][core]
42     ->start_async(__event__(insert_edges));
43 }
44 __foreach_core(group, core) {
45     long long tmp = Inh_inst.gpc[group][core]
46     ->external_memory_address();
47     Inh_inst.gpc[group][core]->mq_send((unsigned int)tmp);
48 }
49 __foreach_core(group, core) {
50     Inh_inst.gpc[group][core]->mq_send(3 *
        sizeof(int)*messages_count);

```

Листинг 2 – Обработка данных

```
1 int get_edge_count(std::string filename)
2 {
3     std::ifstream fin(filename);
4     printf("%d\n", fin.is_open());
5     int a1, a2;
6     int count = 0;
7     while (fin >> a1 >> a2)
8         ++count;
9     fin.close();
10    return count;
11 }
```

Сам граф представлен на рисунке 1

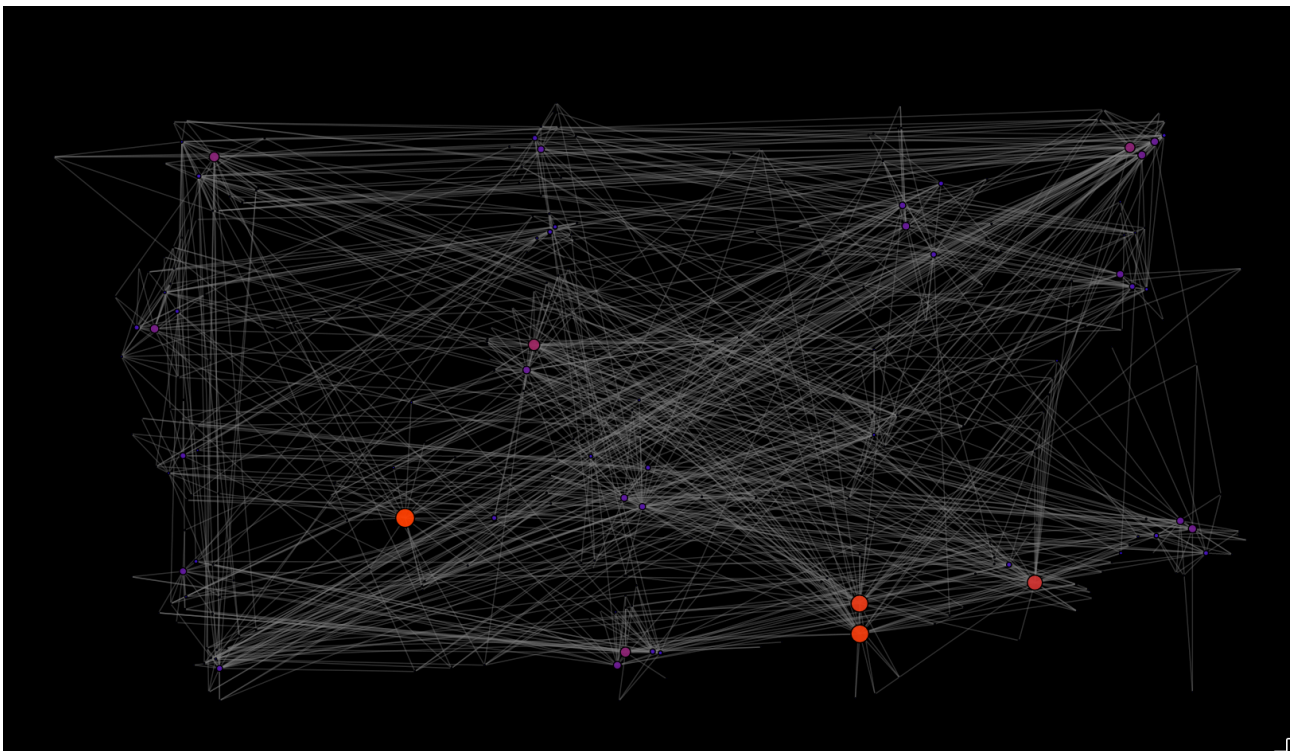


Рисунок 1 – Граф, вариант 15.

ЗАКЛЮЧЕНИЕ

В данной лабораторной работе были освоены принципы представления графов и их обработке с помощью вычислительного комплекса Тераграф. Были получены практические навыки, связанные с представлением графов в виде объединения структур языка C/C++, Были изучены на практике примеры решения некоторых задач на графах