



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 КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

**О Т Ч Е Т**

по практикуму № 2

**Название:** Обработка и визуализация графов в вычислительном комплексе  
Тераграф

**Дисциплина:** Организация ЭВМ и систем

Студент

ИУ6-72Б

(Группа)

\_\_\_\_\_  
(Подпись, дата)

С.В. Астахов

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

\_\_\_\_\_  
(И.О. Фамилия)

**Цель работы:** освоение принципов представления графов и их обработки с помощью вычислительного комплекса Тераграф. Ознакомление с вариантами представления графов в виде объединения структур языка C/C++, изучение и применение на практике примеров решения некоторых задач на графах.

### Задание

Визуализировать граф на основе tsv-файла.

Решение задачи представлено в листинге 1.

Листинг 1 — фрагмент host\_main.cpp

```
#ifdef CUSTOM_GRAPH

printf("< CUSTOM PART > \n");

unsigned int u;

__foreach_core(group, core)
{
    lnh_inst.gpc[group][core]->start_async(__event__(delete_graph));
}

unsigned int* host2gpc_ext_buffer[LNH_GROUPS_COUNT][LNH_MAX_CORES_IN_GROUP];
unsigned int messages_count = 0;

__foreach_core(group, core){
    host2gpc_ext_buffer[group][core] = (unsigned
int*)lnh_inst.gpc[group][core]->external_memory_create_buffer(16 * 1048576 * sizeof(int));
    offs = 0;

    int num1, num2;
    std::string line;
    std::ifstream myfile("/iu_home/iu6031/Astakhov/worksp/lab41/btwc-dijkstra-
xrt/host/src/kronecker_var01.txt");
    if(myfile.is_open())
    {
        /*
        [?] issue: why segmaentation fails if (messages_count > 1900) [?]
        */
        while((getline(myfile,line)) && (messages_count<10000))
        {
            // printf(">>> alloating N_%d \n", messages_count);
            num1 = std::atoi(line.substr(0, line.find("\t")).c_str());
            num2 = std::atoi(line.substr(line.find("\t")+1).c_str());
            //printf("key: %d val: %d\n", num1, num2);
            EDGE(num1, num2, 1);
            EDGE(num2, num1, 1);
            messages_count++;
        }
    }
}
```

```

        }
        myfile.close();
    }
    printf(">>> external_memory_sync_to_device \n");
    Inh_inst.gpc[group][core]->external_memory_sync_to_device(0, 3 * sizeof(unsigned
int)*messages_count);
}

__foreach_core(group, core)
{
    Inh_inst.gpc[group][core]->start_async(__event__(insert_edges));
}
__foreach_core(group, core) {
    long long tmp = Inh_inst.gpc[group][core]->external_memory_address();
    Inh_inst.gpc[group][core]->mq_send((unsigned int)tmp);
}
__foreach_core(group, core) {
    printf(">>> mq_send(2 * sizeof(int)*messages_count)");
    Inh_inst.gpc[group][core]->mq_send(2 * sizeof(int)*messages_count);
}

__foreach_core(group, core)
{
    Inh_inst.gpc[group][core]->finish();
}
printf("Data graph created!\n");
printf("</ CUSTOM PART >");

#endif

```

Результат визуализации графа представлен на рисунке 1.

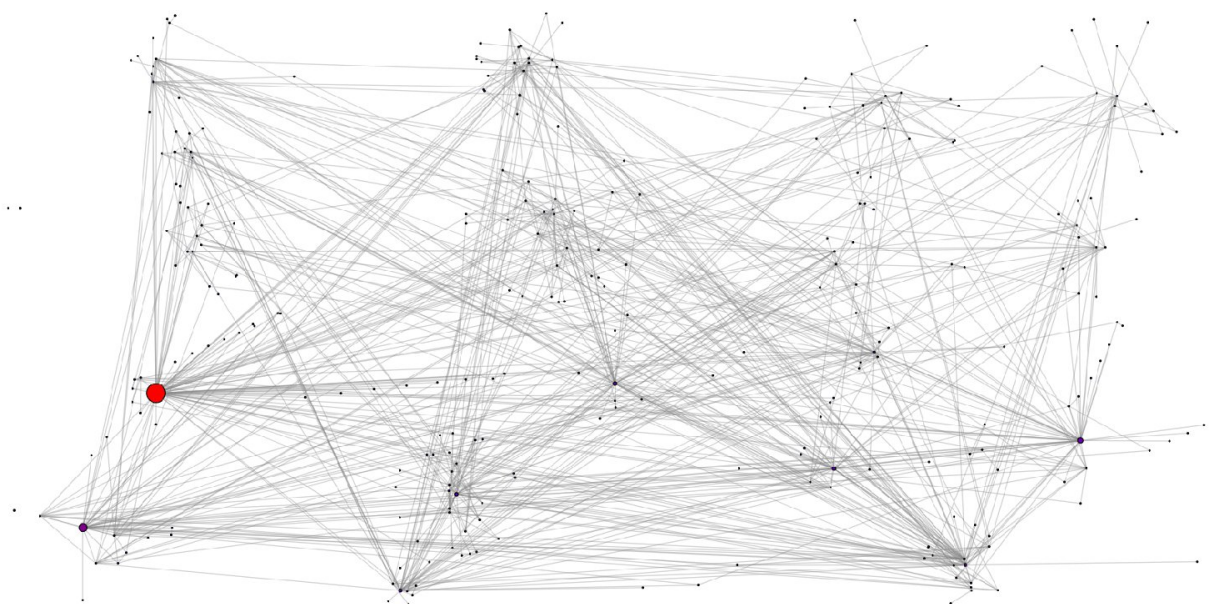


Рисунок 1 — результаты визуализации графа

**Вывод:** в результате выполнения лабораторной работы были освоены принципы представления графов и их обработки с помощью вычислительного комплекса Тераграф. Произошло ознакомление с вариантами представления графов в виде объединения структур языка C/C++, изучение и применение на практике примеров решения некоторых задач на графах.