|  |  |
| --- | --- |
|  | **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**

**Отчет**

|  |  |
| --- | --- |
| **по практикуму №** | 2 |

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

**Дисциплина:** Организация ЭВМ и систем

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ6-72Б |  |  | С.В. Астахов | |
|  | (Группа) |  | (Подпись, дата) | | (И.О. Фамилия) |
|  |  |  |  | |  |
| Преподаватель |  |  |  | |  |
|  |  |  | (Подпись, дата) | | (И.О. Фамилия) |

Москва, 2022

**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");  lnh\_inst.gpc[group][core]->external\_memory\_sync\_to\_device(0, 3 \* sizeof(unsigned int)\*messages\_count);  }  \_\_foreach\_core(group, core)  {  lnh\_inst.gpc[group][core]->start\_async(\_\_event\_\_(insert\_edges));  }  \_\_foreach\_core(group, core) {  long long tmp = lnh\_inst.gpc[group][core]->external\_memory\_address();  lnh\_inst.gpc[group][core]->mq\_send((unsigned int)tmp);  }  \_\_foreach\_core(group, core) {  printf(">>> mq\_send(2 \* sizeof(int)\*messages\_count)");  lnh\_inst.gpc[group][core]->mq\_send(2 \* sizeof(int)\*messages\_count);  }  \_\_foreach\_core(group, core)  {  lnh\_inst.gpc[group][core]->finish();  }  printf("Data graph created!\n");  printf("</ CUSTOM PART >");  #endif |

Результат визуализации графа представлен на рисунке 1.

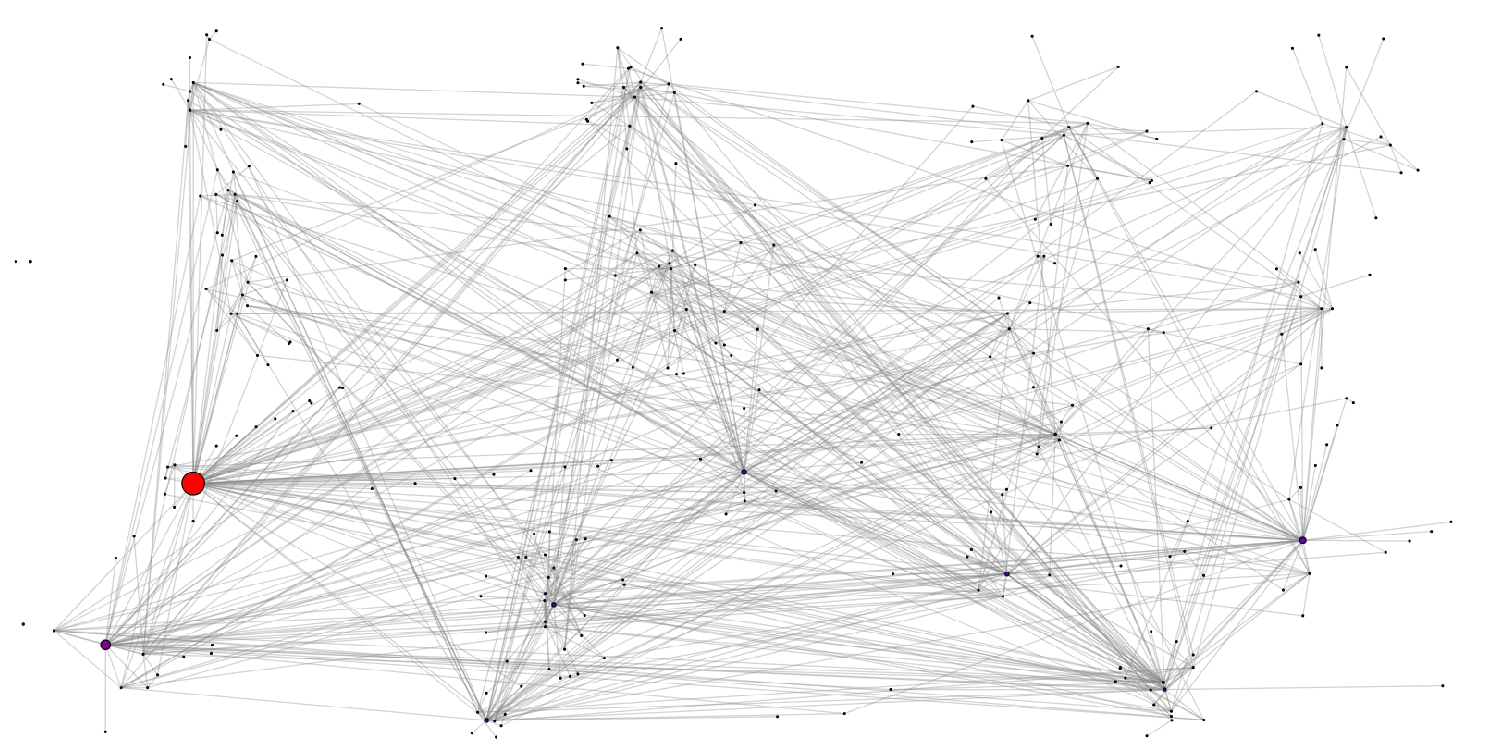


Рисунок 1 — результаты визуализации графа

**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.