



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

Дисциплина: Организация ЭВМ и систем

Название: Обработка и визуализация графов в вычислительном
комплексе Тераграф

Преподаватель

(Подпись, дата)

Ибрагимов

(И.О. Фамилия)

Студент гр. ИУ7-52Б

(Подпись, дата)

Фам М. Х.

(И.О. Фамилия)

Москва, 2023

Цель работы: 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.

Ход работы

Пример 4. Использование языка python и библиотеки gpc64io для приемо-передачи данных между хост-подсистемой и sw_kernel

Пример демонстрирует основные механизмы инициализации гетерогенных ядер gpc и взаимодействие хост-подсистемы с Graph Processor Core, используются аппаратные очереди. Для хост подсистемы используется библиотека gpc64io

```
[8]: gpc.start_handler("echo_mq")
test = 0x123456789
gpc.mq_send_uint64(test)
if test == gpc.mq_receive_uint64():
    print("Echo uint64_t test result: True")
else:
    print("Echo uint64_t test result: False")

Echo uint64_t test result: True

[9]: arr=array.array('Q',(i for i in range(0,TEST_PACK_SIZE)))
buf_out=bytearray(arr)
buf_in=bytearray(arr)
write_thread = gpc.mq_send_buf(buf_out)
read_thread = gpc.mq_receive_buf(len(buf_in),buf_in)
gpc.join(write_thread)
gpc.join(read_thread)

[10]: if buf_out == buf_in:
    print("Echo test result: True")
else:
    print("Echo test result: False")

Echo test result: True

[11]: del(gpc) #Освободить ресурс
```

Рисунок 1 – Результат запуска программы lab4

Пример 5. Демонстрация применения Jupyter ноутбуков и языка python для визуализации графов

Пример демонстрирует варианты анализа графов знаний и их визуализацию. Реализованы пять алгоритмов визуализации:

- Визуализация inbox на основе модулярности Ньюмана.
- Визуализация на основе силового алгоритма Фрухтерамана-Рейнгольда.
- Спиральная визуализация на основе центральности.
- Спиральная матричная визуализация на основе центральности.
- Визуализацию графа-решетки на основе центральности.

На рисунке 2 показана визуализация на основе модулярности Ньюмана.

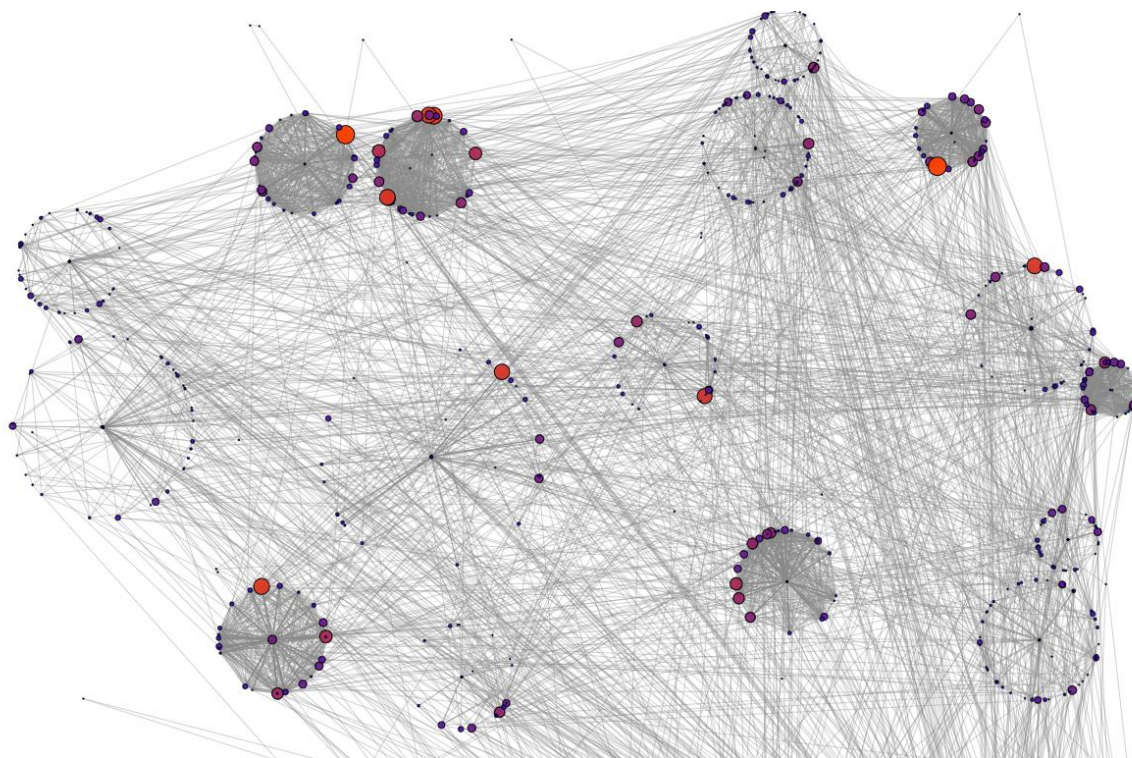


Рисунок 2 – Результат запуска программы lab5

Пример 6. Демонстрация использования микропроцессора Леонард Эйлер для анализа графов знаний.

Пример демонстрирует визуализацию графа гармоний музыкального произведения. Для формирования графа знаний используется запись музыкального произведения в формате midi. По последовательности аккордов строится граф ДеБрюйна с размером окна L , задаваемого параметрически в программе.

В качестве исходного midi файла был выбран 2. February - Carnival. Пусть $L = 3$. Результаты запуска примера представлены на рисунках 3.

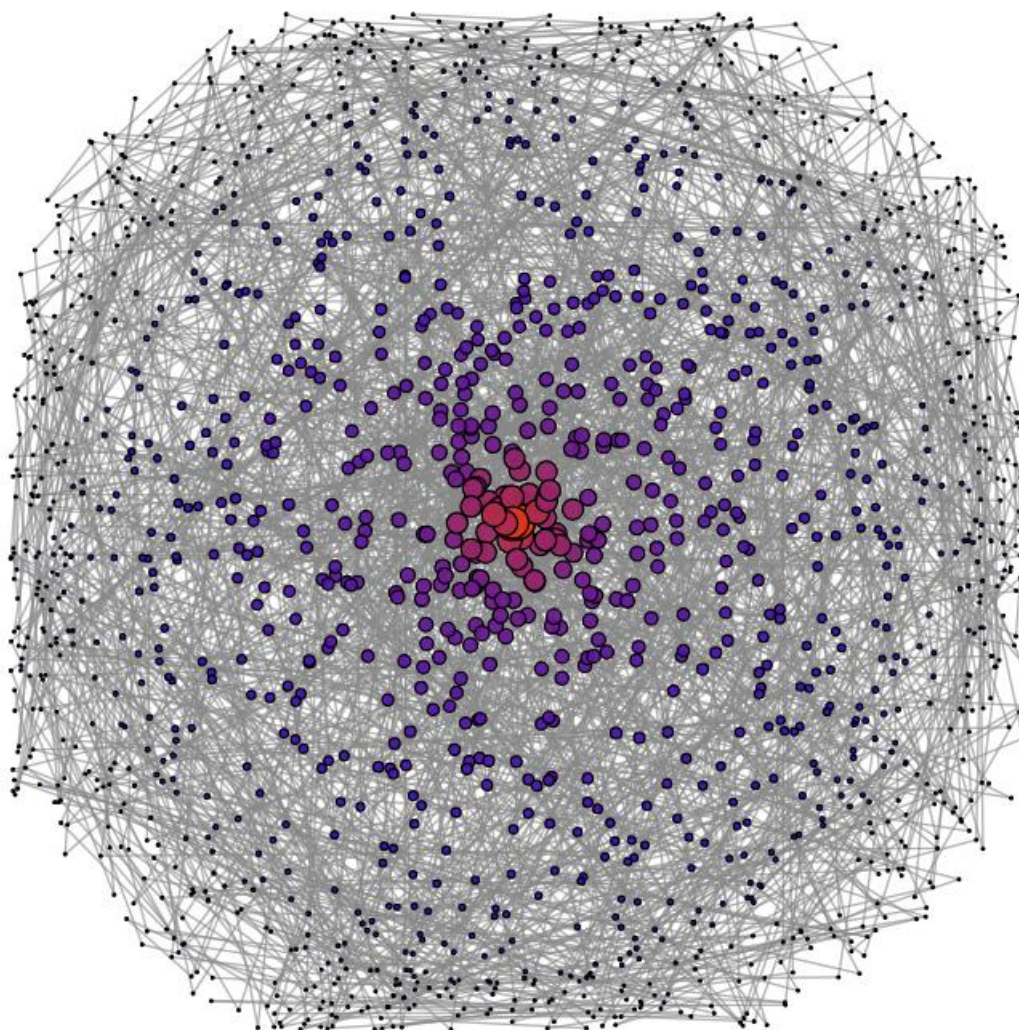


Рисунок 3 – Визуализация на основе модулярности Ньюмана

С помощью приложения AI *Artbreeder* можно получать изображения, объединенные с графами.

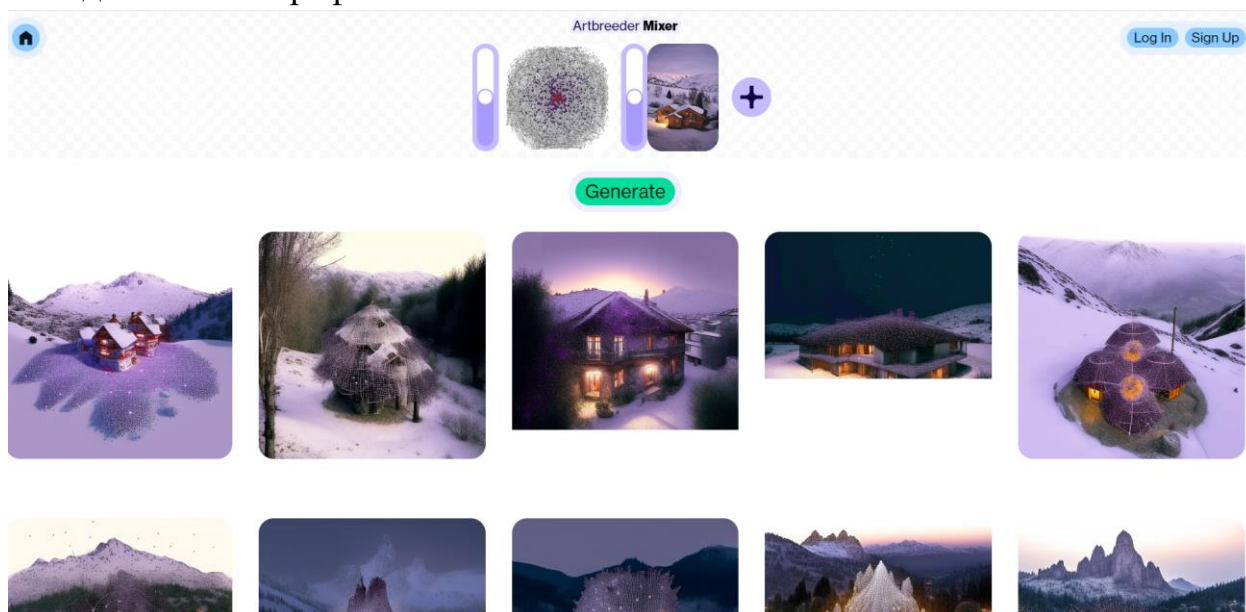


Рисунок 5 – Полученные изображения в Artbreeder

Вывод: В результате работы изучены принципы работы вычислительного комплекса Тераграф, алгоритмы на графах и получены практические навыки решения задач обработки и визуализации графов.