

# **Отчет по практикуму №1**

Шпилевой Владислав Дмитриевич, 621гр., 2 курс  
магистратуры

Вариант 28

## Формулировка задания

Номер варианта: 28.

Задача — исследовать зависимость производительности распределенной реализации алгоритма BFS от размера и типа графа, от количества потоков. Исследование предполагает использование только одного процесса.

Реализация BFS для исследования предоставлена курсом «Суперкомпьютерного моделирования и технологий программирования» и далее по тексту так и называется — «предоставленная реализация».

Целевая архитектура: 8-ядерный процессор Intel, без GPU, общая память 12Гб по полтора гигабайта на ядро. Сборка производится компилятором Intel версии 15.0.0.

## Выполнение задания

Для исследования выбраны:

- графы размеров  $2^{17}$ ,  $2^{18}$ ,  $2^{19}$ ,  $2^{20}$ ,  $2^{21}$ ;
- количество потоков 1, 2, 4, 8, 16;
- типы графов — RМAT и SSCA2.

Множество тестовых конфигураций получено как декартово произведение трех множеств выше. То есть перебраны все возможные варианты сочетаний размеров графов, количества потоков и типов графов: 50 вариантов.

Размеры графов выбраны именно такими, так как их размер в памяти иначе становится больше полутора гигабайт, выделенных на узел. А графы меньшего размера представляют меньший интерес.

## Генерация графов

Для генерации графов использован генератор, взятый из предоставленной реализации. Генератором было создано 5 файлов для RMAT графов и 5 файлов для SSCA2 графов указанных выше размеров.

Ниже представлена команда для генерации графов:

```
./bin/generator -s <size> -file bin/graph_directed_<type>_<size>.bin -type <type>
```

Параметр <size> использован со значениями 17, 18, 19, 20, 21. Параметр <type> принимал значения RMAT и SSCA2.

## Реализация BFS

Реализацию можно скачать по ссылке:

<https://gist.github.com/Gerold103/9d4d662997a549fa2592b35720d5a709>

Код по указанной ссылке реализует три вещи:

1. Загрузка графа из файла в формате списка ребер;
2. Выполнение 10 запусков алгоритма BFS на графе по указанному файлу с различными стартовыми вершинами, замеры производительности;
3. Сравнение предоставленной реализации с реализацией из библиотеки boost.

В отдельном файле реализован запуск реализации boost: <https://gist.github.com/Gerold103/4c77386244e94a7f54b2a0c7507050ae> для сверки с предоставленной реализацией.

Сборка производилась с использованием Makefile.lomonosov, включенного в предоставленную реализацию:

```
make -f Makefile.lomonosov bfs
```

В make-файл добавлена инструкция для сборки graph\_checker.cpp с реализацией boost.

```
graph_checker.o: graph_checker.cpp
```

```
$(CXX) $(CXXFLAGS) $(Include_Path) -I $(BOOST_DIR) -c graph_checker.cpp -o
object_files/graph_checker.o
```

Для перебора параметров, выбранных для исследования, реализация предоставляет 2 аргумента командной строки: имя файла, содержащего граф, и количество потоков, на котором надо запустить BFS. Тип графа не указывается явно, так как это не влияет на формат хранения в файле. То есть для SCCA2 и RМАТ графов сгенерировано по 5 разных файлов.

Пример запуска: `./bfs graph_directed_ssca_17.bin 1`

Третий аргумент командной строки позволяет вместо 10 запусков предоставленного BFS сделать один его запуск, один запуск boost BFS на том же графе и с той же случайной вершины, и сравнить результаты.

Пример запуска: `./bfs graph_directed_ssca_17.bin 1 -need_check`

## Запуск BFS

Каждый из 50 вариантов конфигурации запускался отдельной задачей в очереди test суперкомпьютера Ломоносов по шаблону:

```
sbatch -n 1 -p test -D ~/_scratch/big-graphs/sample/bin/ run -o
<type>_directed_<size>_<threads>.txt ./bfs graph_directed_<type>_<size>.bin
<threads>
```

Здесь запускается BFS для графа типа `<type>` размера  $2^{<size>}$  с количеством потоков `<threads>`, а результат (замеры TEPS) сохраняется в файл, имя которого уникально для каждой конфигурации: `<type>_directed_<size>_<threads>.txt`.

## Полученные результаты

По результатам запусков собрана статистика производительности на двух типах графов, по 25 запусков на каждый.

RMAT: Bfs algorithm MTEPS dependency on graph scale and thread count

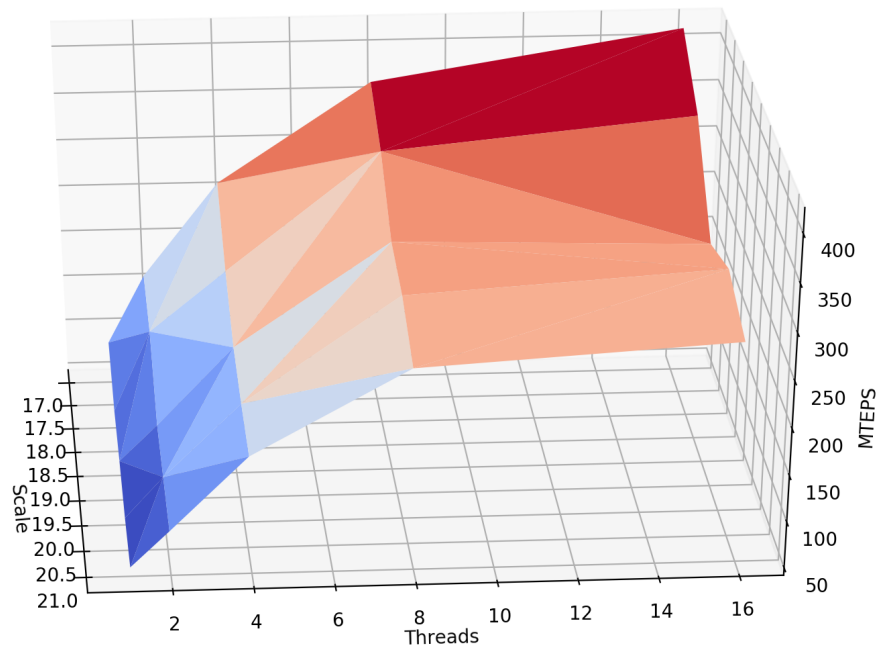


Рисунок 1. Производительность на RMAT графе.

SSCA: Bfs algorithm MTEPS dependency on graph scale and thread count

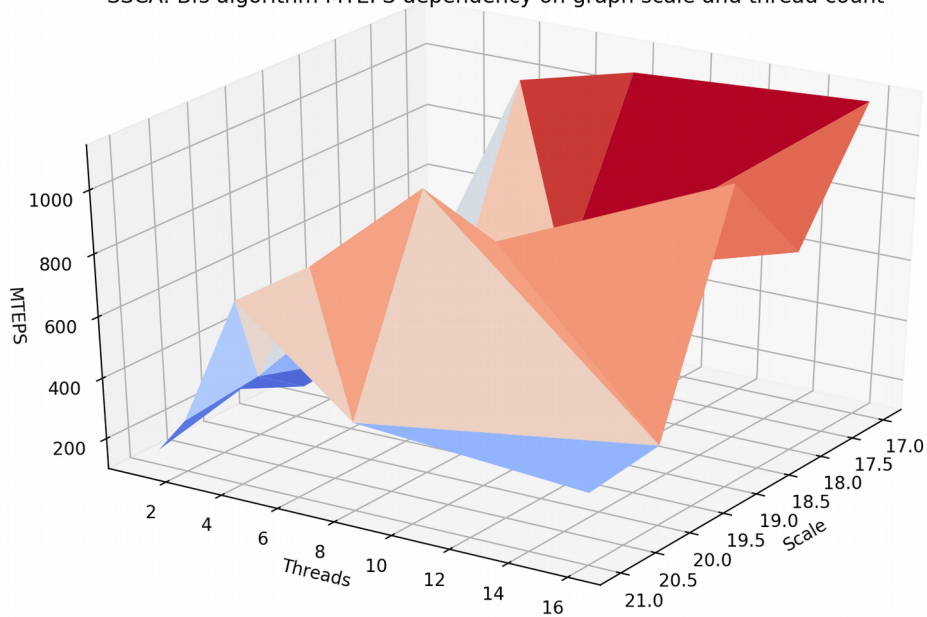


Рисунок 1. Производительность на SSCA2 графе.

На обоих графиках на рисунках 1 и 2 представлена зависимость производительности BFS от размера графа и количества потоков.

Поскольку на каждой из 25 конфигураций для каждого типа графа произведено 10 запусков начиная с разных вершин, в график было решено включить медианы запусков каждой конфигурации. Среднее арифметическое не подходит из-за слишком большой дисперсии — такие результаты не отражали бы ни самые редкие случаи, ни наиболее частые.

Как видно из графиков, наилучшая производительность на графах RMat получена при максимальном количестве потоков на графе наименьшего размера:  $2^{17}$  вершин и 16 потоков дало 418 MTEPS.

Наилучшая производительность на графах типа SSCA2 получена на такой же конфигурации — граф  $2^{17}$  вершин и 16 потоков дал 1117 MTEPS.

Таблицы значений, использованных для построения графиков, находится в приложении.

## Проверка корректности результатов

Проверка правильности реализации выполнена при помощи сравнения результатов предоставленной реализации BFS и реализации из библиотеки boost. Реализация boost для вывода результата использует не явный возврат массива флагов посещенности вершин, как в предоставленной реализации, а вместо этого предоставляет интерфейс для слежения за каждым шагом алгоритма — `default_bfs_visitor`. Этот класс содержит метод `discover_vertex`, который вызывается на каждую добавляемую вершину, и который был переопределен в наследном классе для того, чтобы заполнять такой же массив посещенности вершин. Далее проверка свелась к сравнению двух массивов — от boost и от предоставленной реализации.

Запуск проверки корректности может быть запущен с любым файлом графа и с любым количеством потоков предоставленного BFS с указанием опции `-need_check`.

Пример: `./bfs graph_directed_rmat_18.bin 1 -need_check`

Проверка произведена и не выявила расхождений в результатах. Листинг проверки для всех конфигураций не имеет смысла, так как во всех случаях выведено одно и то же:

```
Visited by boost: <count>, visited by custom: <count>
```

```
The result is the same in boost and in custom bfs
```

Здесь `<count>` варьируется от запуска к запуску в зависимости от стартовой вершины.

## Выводы

Из анализа результатов производительности видно, что ее зависимость от количества потоков не линейна, хотя она и возрастает до определенного момента (это явно видно на RMAТ графе).

В случае SSCA2 графов этот самый момент — это увеличение количества потоков с 8 до 16. Оно дало на многих запусках явное уменьшение производительности. Тот же шаг для RMAТ графов в одном случае снизил производительность, а в других почти не изменил ее.

Это можно связать с тем, что количество выделенных ядер для узла равно 8, и после превышения этого лимита разместить один поток на одном ядре уже не получается, из-за чего существенно возрастают расходы на переключения между потоками.

Нелинейность роста производительности при количестве потоков  $< 8$  можно объяснить тем, что распараллеливается не весь алгоритм целиком, а только его часть, из-за чем увеличение количества потоков в  $N$  раз может ускорить в  $N$  раз только эту часть алгоритма, а остальное останется неизменным.

## Приложение

Таблица МТЕPS на RМАТ графе.

|              | Количество потоков |              |              |              |              |
|--------------|--------------------|--------------|--------------|--------------|--------------|
| Размер графа | 1                  | 2            | 4            | 8            | 16           |
| $2^{17}$     | 47.45 МТЕPS        | 121.89 МТЕPS | 188.29 МТЕPS | 262.76 МТЕPS | 144.3 МТЕPS  |
| $2^{18}$     | 49.16 МТЕPS        | 98.31 МТЕPS  | 183.35 МТЕPS | 235.4 МТЕPS  | 161.39 МТЕPS |
| $2^{19}$     | 43.94 МТЕPS        | 82.84 МТЕPS  | 128.13 МТЕPS | 198.5 МТЕPS  | 207.07 МТЕPS |
| $2^{20}$     | 34.29 МТЕPS        | 71.9 МТЕPS   | 126.68 МТЕPS | 201.06 МТЕPS | 187.22 МТЕPS |
| $2^{21}$     | 32.5 МТЕPS         | 65.76 МТЕPS  | 118.19 МТЕPS | 161.15 МТЕPS | 155.84 МТЕPS |

Таблица МТЕPS на SCCA2 графе.

|              | Количество потоков |              |              |               |               |
|--------------|--------------------|--------------|--------------|---------------|---------------|
| Размер графа | 1                  | 2            | 4            | 8             | 16            |
| $2^{17}$     | 209.06 МТЕPS       | 420.35 МТЕPS | 945.67 МТЕPS | 793.85 МТЕPS  | 794.62 МТЕPS  |
| $2^{18}$     | 248.29 МТЕPS       | 391.7 МТЕPS  | 732.06 МТЕPS | 1121.01 МТЕPS | 598.94 МТЕPS  |
| $2^{19}$     | 242.01 МТЕPS       | 503.23 МТЕPS | 878.9 МТЕPS  | 764.03 МТЕPS  | 1137.6 МТЕPS  |
| $2^{20}$     | 277.84 МТЕPS       | 395.83 МТЕPS | 855.04 МТЕPS | 1135.28 МТЕPS | 1159.44 МТЕPS |
| $2^{21}$     | 204.1 МТЕPS        | 417.38 МТЕPS | 743.18 МТЕPS | 1081.5 МТЕPS  | 960.521 МТЕPS |