**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение**

**высшего образования**

**«Нижегородский государственный университет им. Н.И. Лобачевского»**

**Национальный исследовательский университет**

**Институт информационных технологий, математики и механики**

**ЛАБОРАТОРНАЯ РАБОТА**

**«Сортировка Шелла со слиянием “Разделяй и властвуй”»**

Выполнил: студент группы 381506-1

Дворянчиков Евгений Алексеевич

Нижний Новгород

2018

**Содержание**

1. Постановка задачи………………………….……………………2

2. Метод решения……………..……………………….……………3

3. Вспомогательные элементы……………………………………4

3.1. Checker……………………………………………………4

3.2. Generator………………………………………………….4

4. Схема распараллеливания………..…………..…….……….….5

5. Описание программной реализации…………………………..6

6. Результаты экспериментов…………….………………….……8

7 Вывод……………………………………………………………...11

**Постановка задачи**

Необходимо разработать программу, которая производит сортировку исходного массива данных.

Исходные данные: размер массива данных, значения содержащиеся в массиве .

Программа должна содержать OpenMP, TBB и последовательную реализацию, а также сравнение времени работы этих реализаций для одних и тех же исходных данных.

**Метод решения**

Пусть задан исходный массив данных. Необходимо отсортировать его элементы в порядке возрастания.

1. Разбиваем исходный массив на n равных (с точность до n) блоков.

2. Сортируем каждый блок методом сортировки Шелла.

3.Производим слияние отсортированных блоков методом “Разделяй и властвуй”

Сортировка Шелла

Сортировка Шелла — это надстройка над сортировкой вставками; вместо одного прохода мы делаем несколько, причем на i-ом проходе мы сортируем подмассивы из элементов, стоящих друг от друга на расстоянии di.

При этом на последнем проходе di должно быть равно 1, это гарантирует корректность алгоритма.

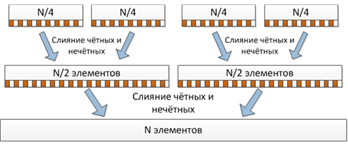
Эффективность сортировки Шелла в определенных случаях обеспечивается тем, что элементы «быстрее» встают на свои места (в простых методах сортировки каждая перестановка двух элементов уменьшает количество инверсий в списке максимум на 1, а при сортировке Шелла это число может быть больше).

Сложность алгоритма зависит от выбора количества проходов и значений di, и может сильно варьироваться.

Воспользуемся эмпирически полученной последовательностью Марцина Циура.

Слияние “Разделяй и властвуй”

Идея слияния по алгоритму «Разделяй и властвуй» заключается в разбиении массивов на участки, которые можно слить независимо . В первом массиве выбирается центральный элемент x (он разбивает массив на две равные половины), а во втором массиве с помощью бинарного поиска находится позиция наибольшего элемента, меньшего x (позиция этого элемента разбивает второй массив на две части). После такого разбиения первые и вторые половины массивов могут сливать независимо, т.к. в первых половинах находятся элементы меньшие элемента x, а во второй – большие. Для слияния двух массивов несколькими потоками можно в первом массиве выбрать несколько ведущих элементов, разделив его на равные порции, а во втором массиве найти соответствующие подмассивы. Каждый поток получит свои порции на обработку.  
Эффективность такого слияние во многом зависит от того, насколько равномерно произошло «разделение» второго массива.



**Вспомогательные элементы**

Checker

Проверяет корректность результата, сравнивая массив полученный в результате работы программы и массив отсортированный встроенными методами.Записывает вердикт(AC = Accepted = Решение выдает корректный результат на данном тесте; WA = Wrong Answer = Решение выдает некорректный результат на данном тесте; NO = No verdict = Вердикт отсутствует), комментарий вида: корректен результат или нет.

Generator

Генерирует случайный массив данных размером N.

**Схема распараллеливания**

Для параллельной реализации разбиваем исходный массив на равные части в зависимости от числа процессов и параллельно выполняем сортировку каждой части.

Отсортированные блоки сливаем. Параллелизм на этом этапе заключается как в независимом слиянии блоков, так и в независимом слиянии участков в блоках, в соответствии с схемой “Разделяй и властвуй”.

**Описание программной реализации**

• Lab1\_v1 – реализация последовательного алгоритма

• Generator\_v1 – генератор входных данных

• Checker\_v2 – программа, проверяющая корректность результата

• Typer\_v1 - программа, конвертирующая текстовый файл в бинарный

• Viewer – программа, конвертирующая бинарный файл в текстовый

• Lab2\_v1 – OpenMP версия программы

• Lab3\_v1 – TBB версия программы

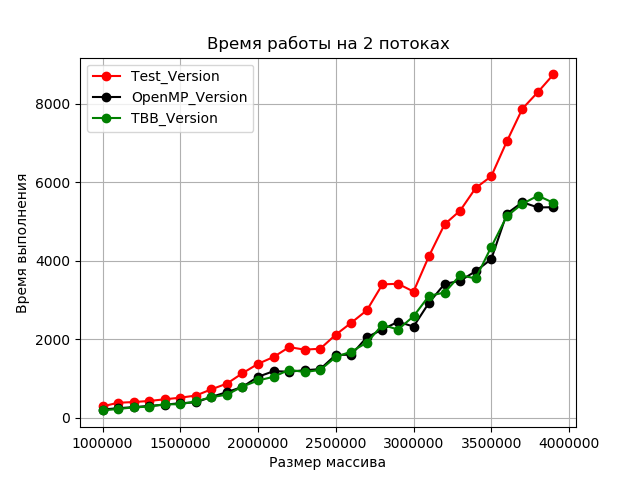
**Результаты экспериментов по оценке масштабируемости**

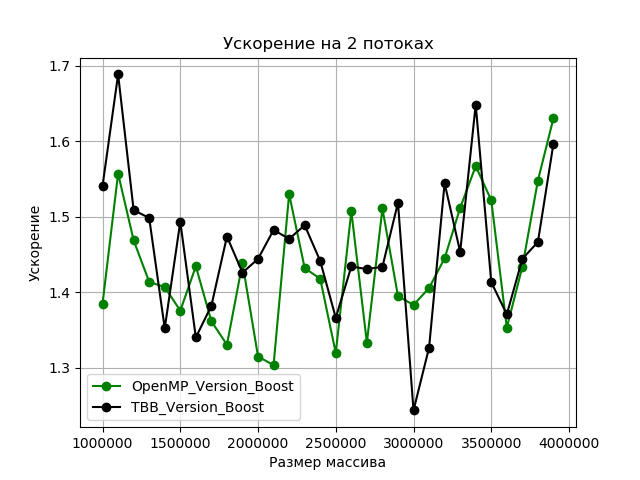
Эксперименты проводились со следующими данными:

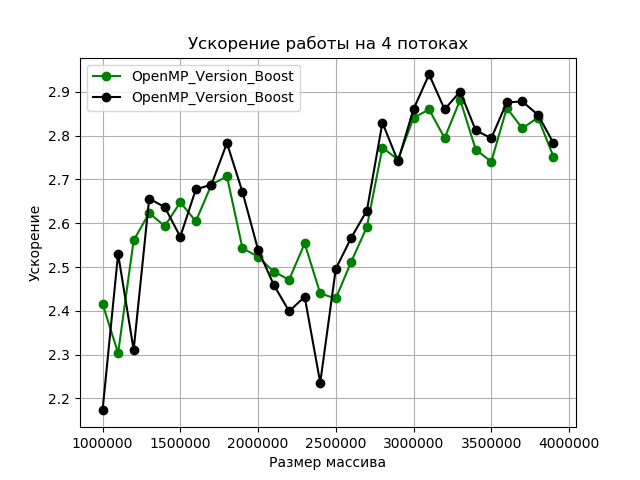
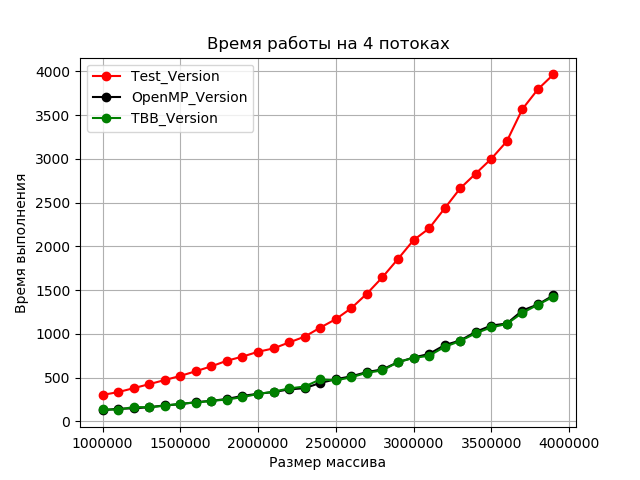
Изначальный размер массива 1000000

Шаг 100000

Количество шагов 30





**Вывод**

Ускорение полученное с помощью OpenMP и TBB версий при равных входных данных не сильно отличаются. При использование 2 потоков ускорение в среднем было в 1,4 раза.При использование 4 потоков ускорение в среднем было в 2,7 раза.