

Постановка замерного эксперимента.

1. Обычный кусочно-линейный график.

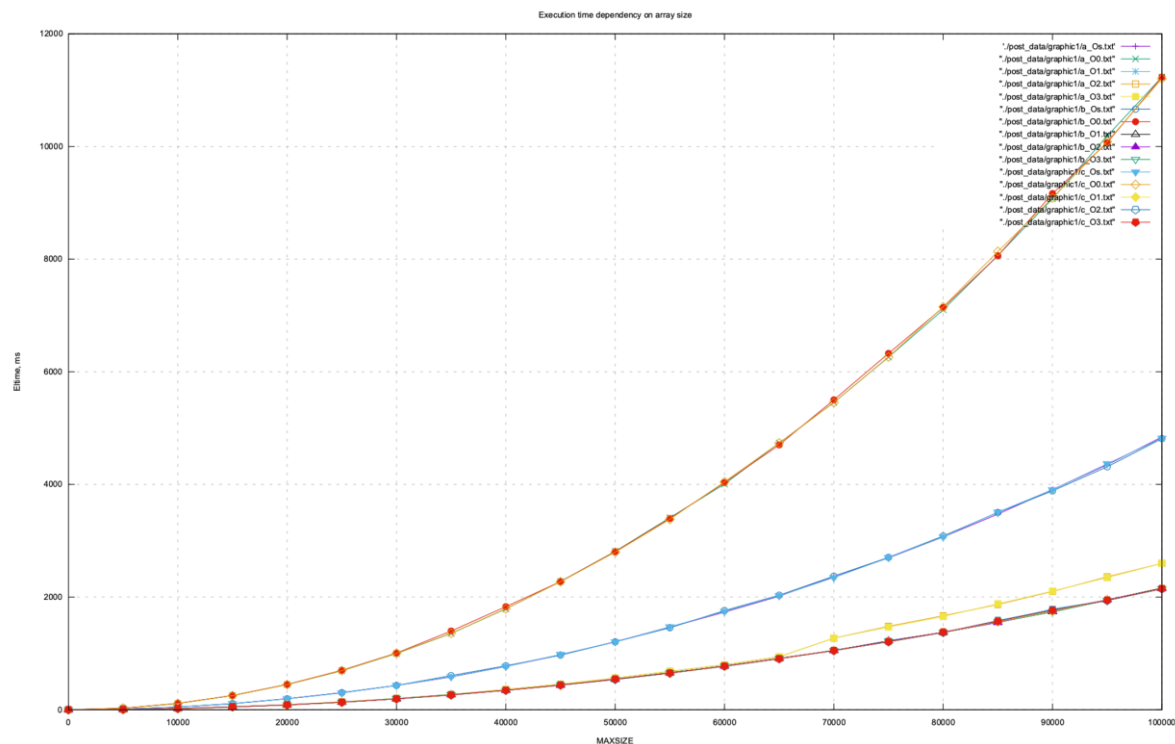


График представляет зависимость времени выполнения в миллисекундах от числа элементов массива для всех 15 вариантов программы в зависимости от уровня оптимизации “O0, O0-3”.

На графике видно, что программы с _O1-3, b_O1-3 выполняются быстрее чем остальные программы. Связано это с уровнем оптимизации и различием в реализации функции. При работе с массивом формальная замена операции индексации на выражение и использование указателей для работы с массивом, осуществляются быстрее.

“O1” это наиболее простой уровень оптимизации. Компилятор попытается сгенерировать быстрый, занимающий меньше объема код, без затрачивая наибольшего времени компиляции. Он достаточно простой, но должен всегда выполнять свою работу.

“O2” активирует несколько дополнительных флагов вдобавок к флагам, активированных –“O1”. С параметром -O2, компилятор попытается увеличить производительность кода без нарушения размера, и без затрачивая большого количества времени компиляции.

После того как размер массива превысил отметку в 65.000, время работы программ “a_02” и “a_03”, где при работе с массивом используется операции индексации вида “array_name[i]”.

Для программ с оптимизацией “Os”, время работы при изменении количества элементов массива одинаково. На этом уровне код будет оптимизирован по объему. Он может быть полезным на компьютерах, которые обладают чрезвычайно ограниченным пространством жесткого диска и/или процессоры с небольшим размером кэша.

Наибольшее время наблюдается у программ , с уровнем оптимизации “O0”. Этот уровень отключает оптимизацию полностью и является уровнем по умолчанию.

Это сокращает время компиляции и может улучшить данные для отладки, но некоторые приложения не будут работать должным образом без оптимизации. Эта опция не рекомендуется, за исключением использования в целях отладки.

Таблицы, с данными, которые используются для построения графиков.

a_O0	
Размер массива	Среднее арифметическое
1	0.000
5000	28.200
10000	111.100
15000	252.200
*****	*****
95000	10187.400
100000	11229.800

a_O1	
Размер массива	Среднее арифметическое
1	0.000
5000	5.600
10000	21.800
15000	50.000
*****	*****
95000	1944.900
100000	2153.200

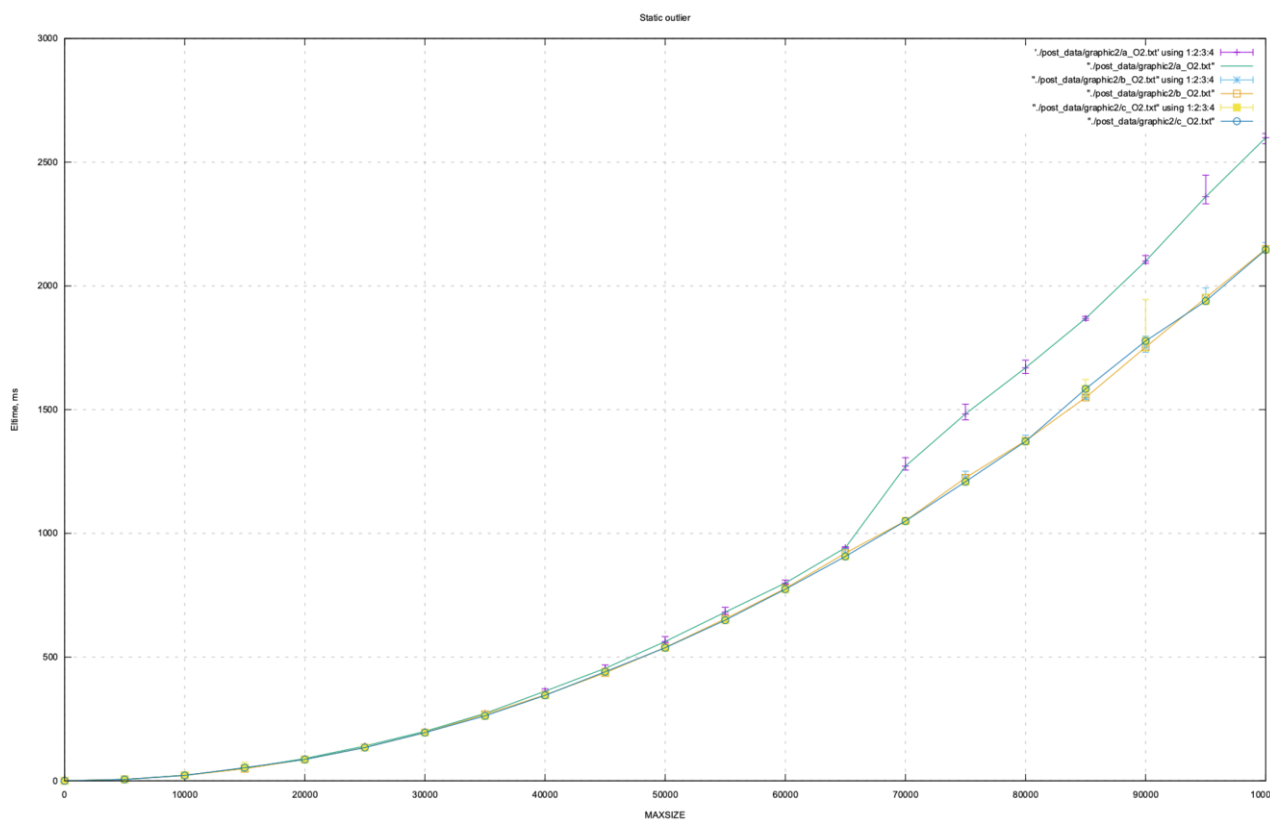
a_O2	
Размер массива	Среднее арифметическое
1	0.000
5000	6.000
10000	22.700
15000	50.600
*****	*****
95000	2360.900
100000	2599.200

a_O3	
Размер массива	Среднее арифметическое
1	0.000
5000	5.800
10000	22.600
15000	50.400
*****	*****
95000	2346.300
100000	2600.000

a_Os	
Размер массива	Среднее арифметическое
1	0.000
5000	13.100
10000	48.400
15000	107.500
*****	*****
95000	4350.300
100000	4841.500

В таблицах наглядно видны различия значений для каждого уровня оптимизации.

2. Кусочно-линейный график с ошибкой



На графике показаны статистические выбросы для всех вариантов обработки массива при уровне оптимизации “O2”.

При размерах массива 70.000 – 80.000 и 95.000 видны статистические выбросы в программе “a”. Для программ “b” и “c” статистические выбросы видны при размерах 85.000 – 95.000.

Для построения графика используются среднее арифметическое, минимальное и максимальные значения.

Пример таблицы, по которой строятся графики.

Размер массива	Среднее арифмитическое	Максимум	Минимум
1	0.000	0.000	0.000
5000	6.000	6.000	6.000
10000	22.700	23.000	22.000
*****	*****	****	****
100000	2599.200	2616.000	2574.000

3. График с усами

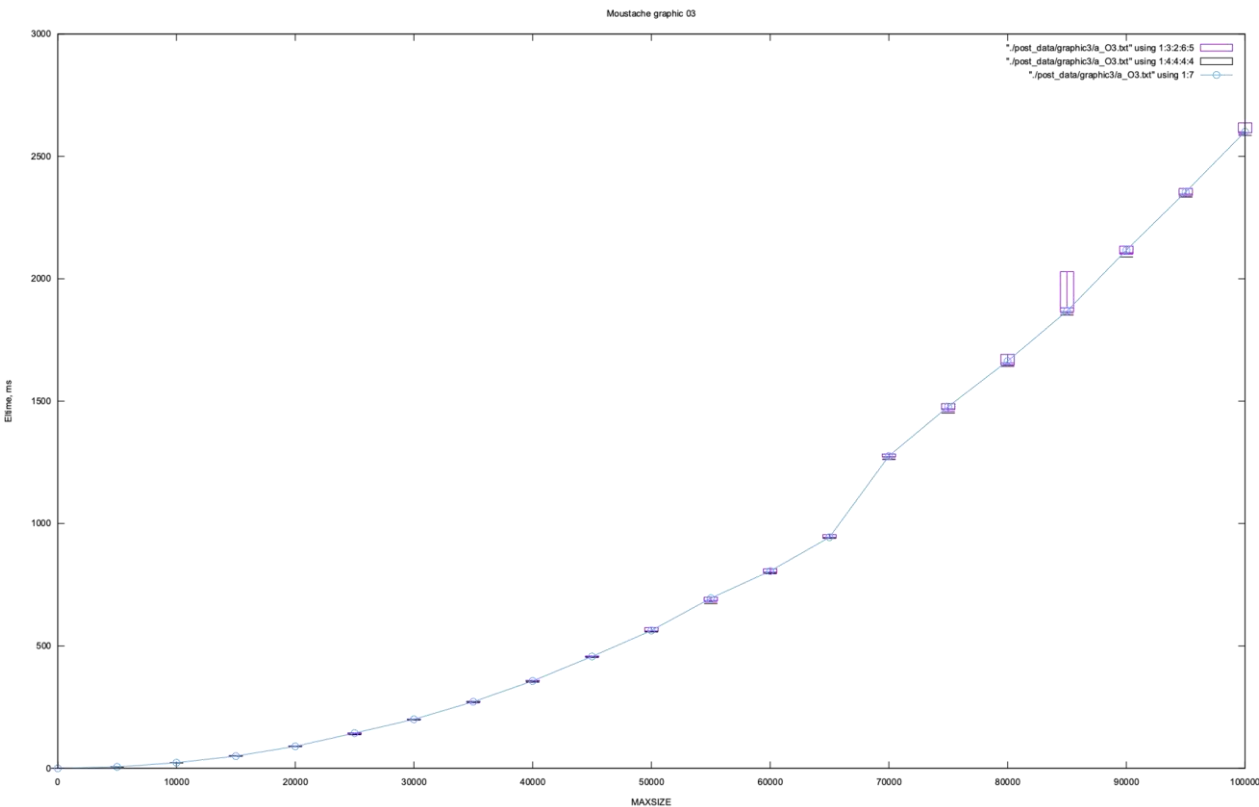


График построен для программы, в которой обработка элементов массива осуществляется использованием операции индексации $a[i]$ при уровне оптимизации “О3”. На графике показан разброс при изменении количества элементов массива. Больше всего заметно при 85.000 элементов. Также небольшой разброс имеется при размере массива больше 75.000.

Таблица данных, для а_О3						
Размер массива	Среднее арифметическое	Медиана	Минимум	Максимум	Нижняя квартиль	Верхняя квартиль
1	0.000	0.000	0.000	0.000	0.000	0.000
5000	5.800	6.000	5.000	6.000	6.000	6.000
10000	22.600	23.000	22.000	23.00	22.000	23.000
*****	*****	*****	*****	*****	*****	*****
95000	2346.300	2341.000	2334.000	2369.000	2335.000	2355.000
100000	2600.000	2597.000	2586.000	2637.000	2591.000	2601.000

Таблица для результатов обработки с использованием операции индексации с уровнем оптимизации O2

n, длина массива	t, время	$(\ln(t_{i+1}) - \ln(t_i)) / (\ln(n_{i+1}) - \ln(n_i))$
1	0	-
5000	6.000000	1,91965789
10000	22.700000	1,97695593
15000	50.600000	2,009433
20000	90.200000	1,98925557
25000	140.600000	1,92460654
30000	199.700000	2,00920846
35000	272.200000	2,13926815
40000	362.200000	1,92915189
45000	454.600000	2,02980894
50000	563.000000	2,00415762
55000	681.500000	1,83096719
60000	799.200000	2,05250284
65000	941.900000	4,04780773
70000	1271.400000	2,22941202
75000	1482.800000	1,83289685
80000	1669.000000	1,86158365
85000	1868.400000	2,04607022
90000	2100.200000	2,16416382
95000	2360.900000	1,87472464
100000	2599.200000	-

1. Из реализованных способов работы обработки, быстрее работают указатели, ведь в отличие от работы с индексами "a[i]" мы имеем доступ к памяти напрямую. В случае с массивами индекс обрабатывается и вычисляются эффективные адреса.
2. Из-за того, что мы не знаем каков был порядок, то нельзя определить где был статистический выброс, поэтому вырезать его мы не можем.
3. Мы не можем вырезать данные со статистическим выбросом из датасета, поскольку эксперимент был поставлен вчера и настройки машины сбились, из-за чего будет отличаться время работы. Поэтому следует восстановить данные из резервной копии либо же провести эксперимент заново.
4. Заменить одним экспериментом нельзя, так как мы не будем учитывать некоторые значения, из-за чего наши данные будут не совсем корректными.

5. Из-за того что мы используем функцию `rand()` , у которой время выполнения непостоянное, то если мы будем учитывать эти значения, то данные будут некорректными.