

Нисуев Нису ИУ7-22Б

Задание №4.2 по тестированию, отладке и профилированию ПО

Сложение квадратных матриц

Цель: узнать самый быстрый способ работы с двумерными массивами

Количество запусков: 2100

Способы работы с матрицами:

- **snrestrict** – матрица передается в функцию без использования **restrict** и со стандартным обходом элементов
- **cnrestrict** – матрица передается в функцию без использования **restrict** и с обходом элементов по столбцам
- **restrict** – матрица передается в функцию с использованием **restrict** и со стандартным обходом элементов

Рабочая среда:

apps – каталог с собранными исполняемыми файлами. Размеры создаваемых массивов передаются в качестве аргумента

measures – каталог, который содержит собранные измерения. Он организован следующим образом: есть каталоги, содержащие различные способы обработки и передачи матриц. В этих каталогах хранятся каталоги с различными уровнями оптимизации. И внутри этих каталогов находятся файлы, названия которых указывают на длины массивов и содержат замеры времени выполнения. Также в каталоге есть файл **runs.txt**, который хранит информацию о количестве проведенных измерений.

progs – каталог с исходными кодами программ

data_work – каталог работы с данными

data_work/data – каталог с измеряемыми данными. Каталог хранит в себе: файл **optimizations** с уровнями оптимизаций и файл **matrix_sizes**, который хранит в себе измеряемые размеры матриц

data_work/calced_data – каталог с измеренными данными. Содержит в себе текстовые файлы, которые хранят в себе размеры измеряемых матриц и среднее время. Данные распределены в текстовых файлах по виду передачи и обработки матриц, уровню оптимизации

data_work/build_apps.sh – скрипт собирает исполняемые файлы и помещает их в каталог **./apps**. Скрипт собирает исполняемые файлы по разным видам передачи и обработки матрицы, уровням оптимизации. Также скрипт определяет макрос максимальной длины массива при компиляции, которая передается в скрипт как аргумент (при отсутствии аргумента стандартное значение максимальной длины массива равно 1000)

data_work/update_data.sh – скрипт добавление новых замеров в каталог **./measures**. Скрипт создает нужные каталоги при их отсутствии и записывает в них определенное количество замеров, которые передаются в скрипт как аргумент(при отсутствии аргумента стандартное количество замеров равно 100)

data_work/clean_data.sh – скрипт удаления измерений из каталога **./measures**

data_work/make_preproc.py – программа считает среднее значение измерений времени для каждого измеряемого размера матрицы, уровня оптимизации и способа передачи и обработки матрицы. Программа записывает полученные измерения в файлы расположенные в каталоге **./data_work/calced_data**

data_work/make_postproc.sh – скрипт отрисовывает графики по скриптам описанным в каталоге **./data_work/graph_draw**. Графики строятся по данным, которые находятся в каталоге **./data_work/calced_data**

data_work/go.sh – скрипт запускает всю систему работы с данными

data_work/graph_draw – каталог хранит в себе все скрипты для заданных графиков

* **data_work/table_create.py** – программа чертит таблицы с относительной стандартной ошибкой среднего для каждой длины массива. Таблицы чертятся для каждого вида массива, уровня оптимизации и способа обращения к элементам массива.

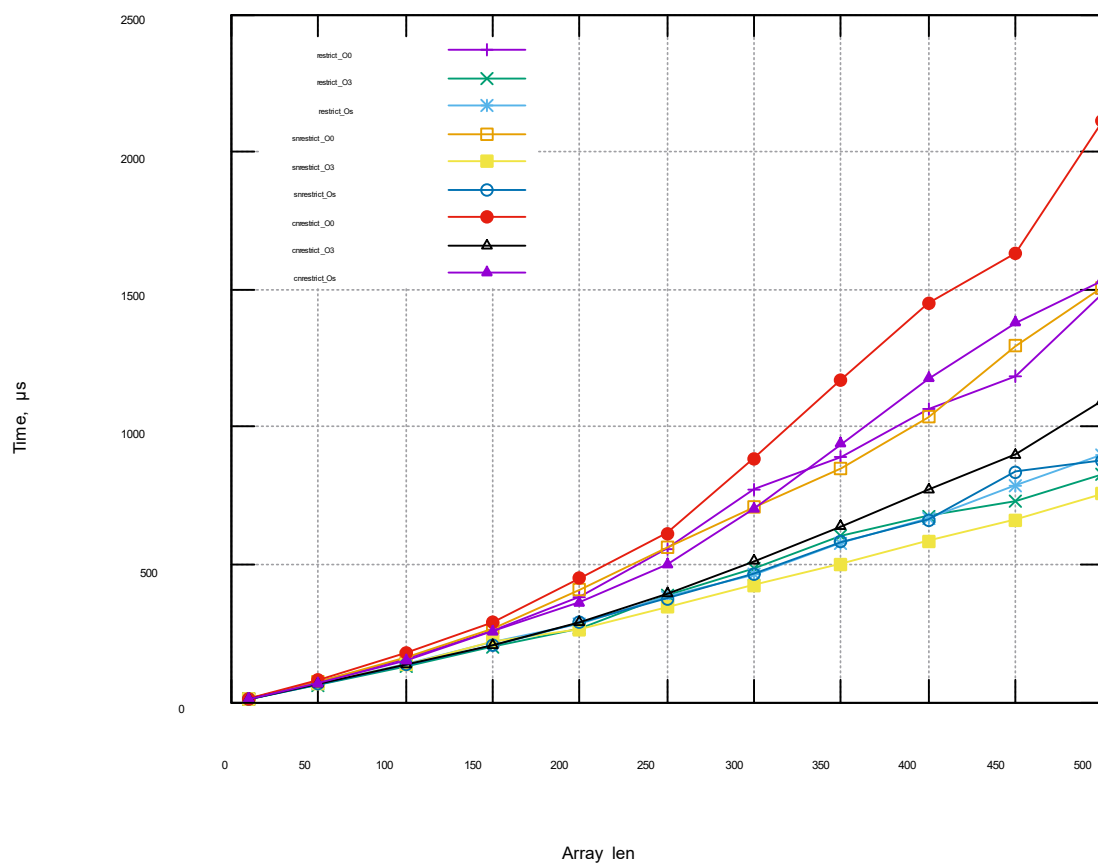
Таблицы:

	restrict_O3			snrestrict_O3			cnrestrict_O3		
length	t_i , мкс	RSE, %	$\frac{\ln(t_{i+1}) - \ln(t_i)}{\ln(n_{i+1}) - \ln(n_i)}$	t_i , мкс	RSE, %	$\frac{\ln(t_{i+1}) - \ln(t_i)}{\ln(n_{i+1}) - \ln(n_i)}$	t_i , мкс	RSE, %	$\frac{\ln(t_{i+1}) - \ln(t_i)}{\ln(n_{i+1}) - \ln(n_i)}$
10	11,12	2,01	1,08	10,95	1,98	1,14	10,64	0,68	1,14
50	62,91	0,62	1,04	68,36	0,65	0,99	66,49	0,52	1,03
100	129,25	0,44	1,10	135,42	0,58	1,20	135,79	0,51	1,03
150	201,55	1,71	0,98	220,21	0,65	0,65	206,36	0,39	1,19
200	267,52	0,37	1,67	265,61	0,69	1,17	290,62	0,64	1,35
250	387,91	0,58	1,23	344,98	0,36	1,16	393,05	0,37	1,44
300	485,67	0,65	1,42	426,57	0,47	1,08	511,02	0,67	1,44
350	604,65	1,72	0,86	503,70	0,29	1,15	637,78	0,47	1,43
400	677,79	0,98	0,63	587,15	0,81	1,05	772,13	0,42	1,30
450	730,01	0,43	1,20	664,20	0,25	1,24	900,37	0,41	1,87
500	828,27	0,47		756,66	1,16		1096,24	0,42	

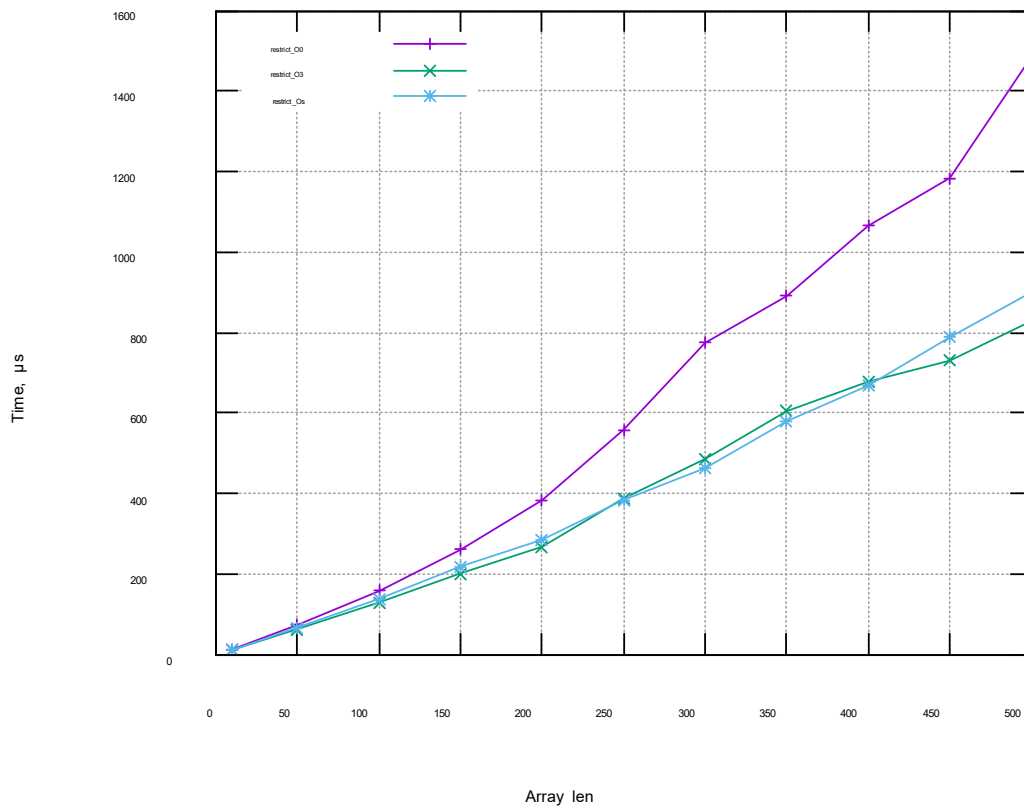
Графики:

Графики зависимости времени работы от размеров квадратных матриц

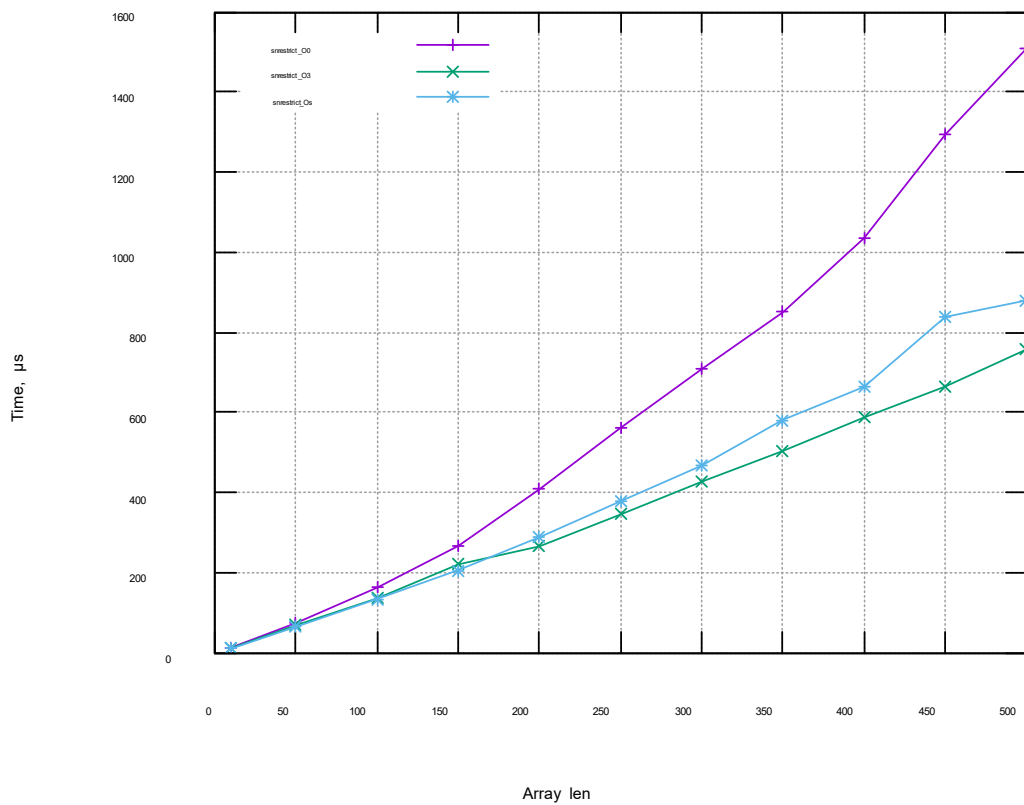
- **all.svg** – кусочно-линейные графики всех уровней оптимизации и способов обработки передачи и обработки матрицы



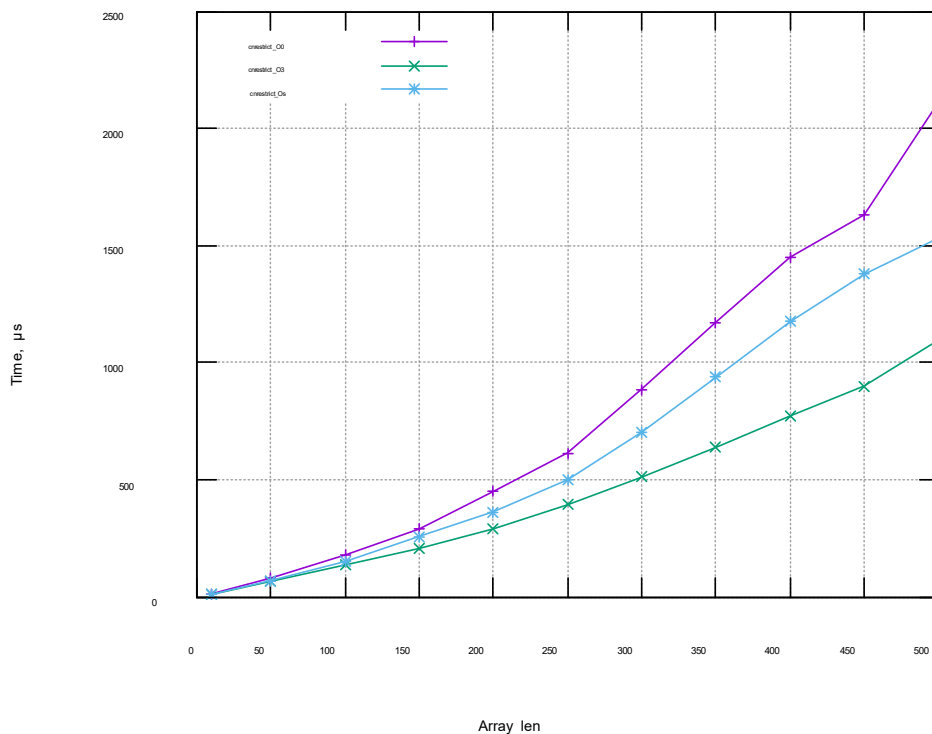
- **restr.svg** — кусочно-линейные графики всех уровней оптимизации при использовании **restrict**



- **snrestr.svg** – кусочно-линейные графики всех уровней оптимизации без использования **restrict** и со стандартным обходом элементов



- **cnrestr.svg** – кусочно-линейные графики всех уровней оптимизации без использования **restrict** и обходом элементов по столбцам



Вывод: Графики показывают, что использование **restrict** ускоряет работу программы. **restrict** ускоряет код, т.к. он сообщает компилятору, что нет никаких зависимостей между памятью, на которую указывают различные его указатели.