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 **ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04 Программная инженерия**

О Т Ч Е Т

по лабораторной работе № 3

Название: Организация памяти суперскалярных ЭВМ

Дисциплина: Архитектура электронно-вычислительных машин

Студент	ИУ7-52Б		Короткая В. М.
	(Группа)	(Подпись, дата)	(И.О. Фамилия)
Преподаватель			А.Ю.Попов
		(Подпись, дата)	(И.О. Фамилия)

Москва, 2021

Эксперимент 1. Исследования расслоения динамической памяти

Параметры:

1. Максимальное расстояния между читаемыми блоками,
2. Шаг увеличения расстояния между читаемыми 4-х байтовыми ячейками,
3. Размер массива.

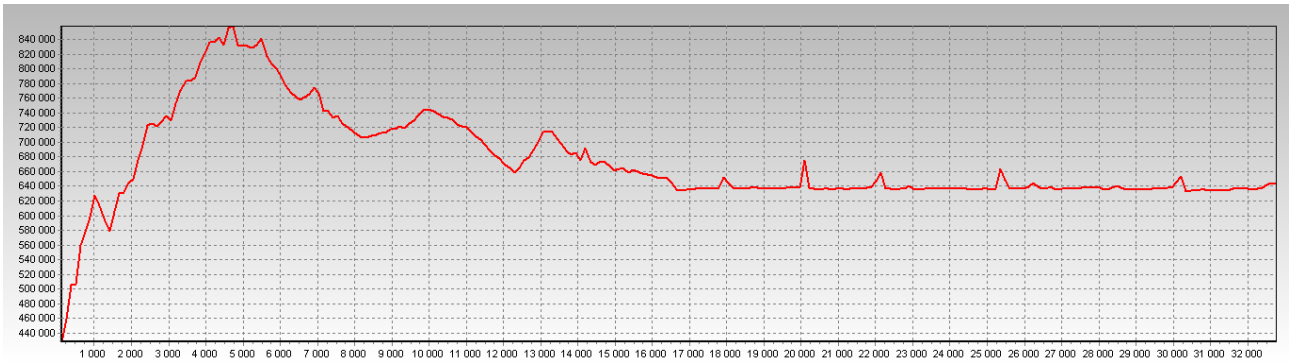

Рисунок 1: Исследование расслоения динамической памяти. Параметры: 38, 128, 1.

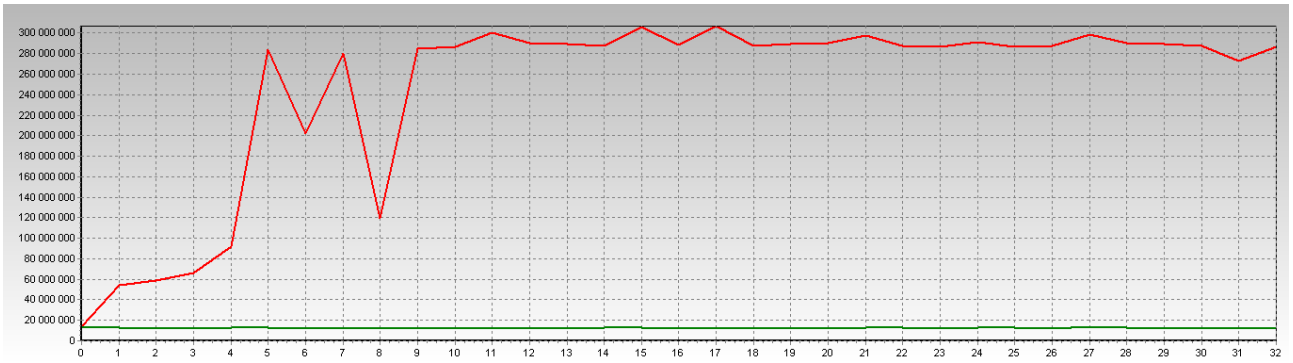

Рисунок 2: Исследование расслоения динамической памяти. Параметры: 32, 64,1.

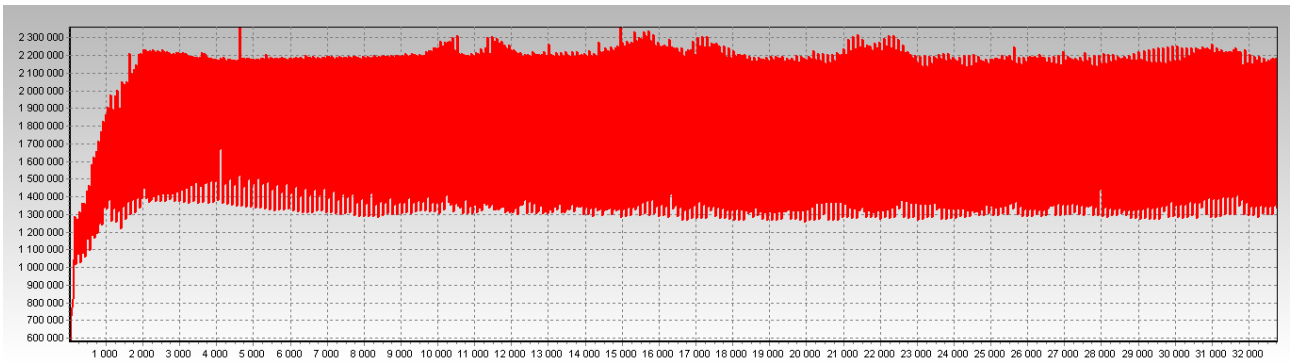

Рисунок 3: Исследование расслоения динамической памяти. Параметры: 32, 32, 1

Вывод:

Память не однородная, она расслоена, это необходимо учитывать при обращении к ней и непосредственной обработки. Чем больше адресное расстояние, тем больше время доступа. Данные для непосредственной обработки лучше вместе разместить.

Эксперимент 2. Сравнение эффективности ссылочных и векторных структур

Параметры:

1. Количество элементов в списке,
2. Максимальная фрагментации списка,
3. Шаг увеличения фрагментации.

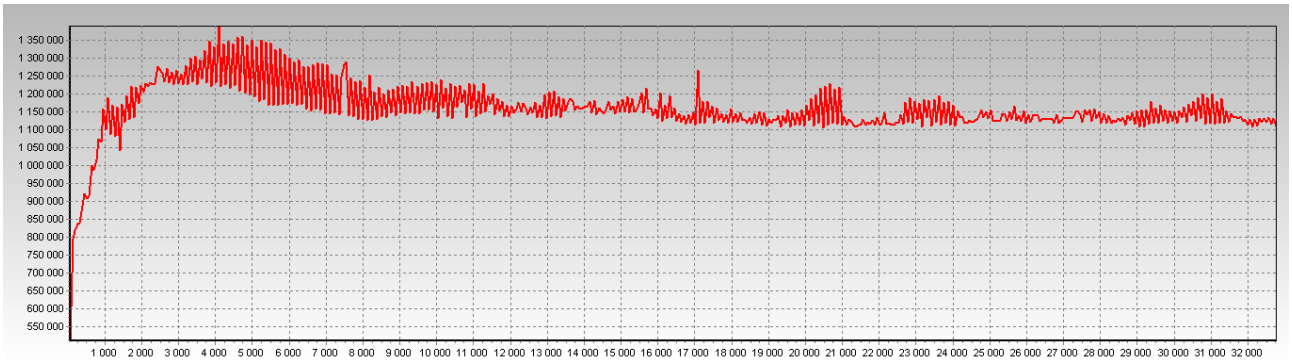

Рисунок 4: Сравнение эффективности ссылочных и векторных структур. Параметры: 1,32,1.

Вывод:

Список обрабатывается в 19.8 раз хуже.
существует семантический разрыв между тем что от машины хочет человек и тем что машина может.
Из полученного графика видна процесса семантического разрыва. Использовать типы данных надо с учетом особенностей работы машины. Предпочтительнее использовать массив, т.к. это быстрее обрабатывается машиной.

Эксперимент 3. Исследование эффективности программной предвыборки

Параметры:

1. Шаг увеличения расстояния между читаемыми данными
2. Размер массива

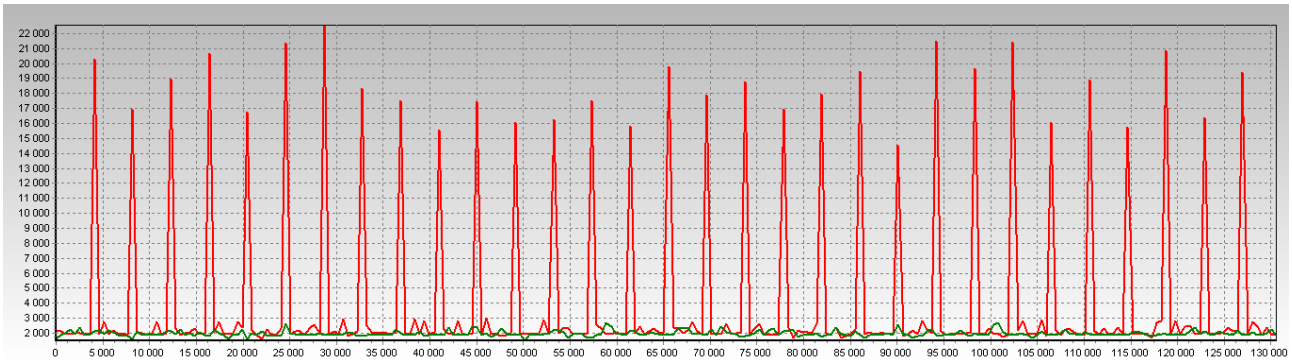

Рисунок 5: Исследование эффективности программной предвыборки. Параметры 512, 128.

Скачки (каждые 4096) - это преобразование в физический адрес. Процессор обращается к таблице стрниц, потом к TLB, потом снова к ОП за данными.
В зеленом графике заранее прочли по байтику с каждых 4096 (предвыборка). TLB содержит нужные адреса, нет необходимости обращаться 2 раза.
Предвыборка лучше в 2 раза.

Вывод:

Для повышения производительности можно использовать предвыборку. Массив лучше использовать более одного раза (последующие чтения будут быстрее).

Эксперимент 4. Исследование способов эффективного чтения оперативной памяти

Параметры:

1. Размер массива
2. Количество потоков данных.

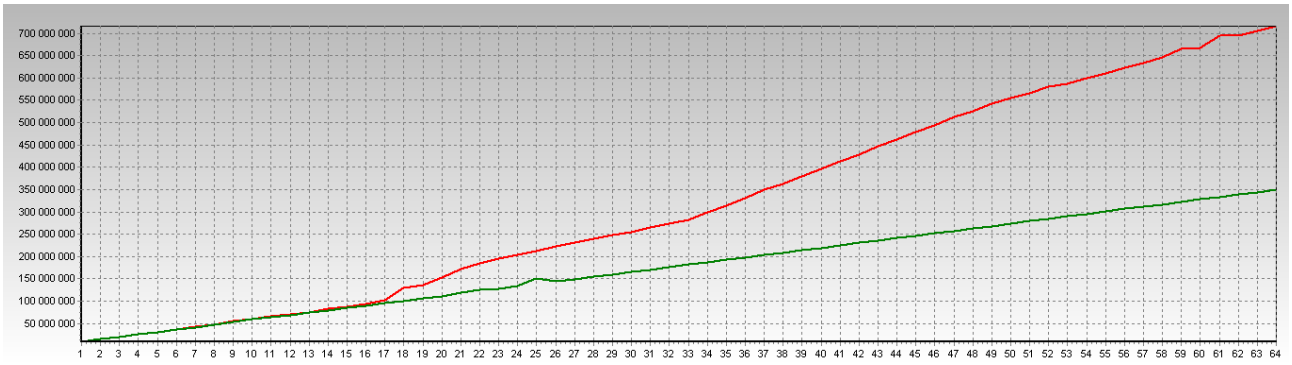

Рисунок 6: Исследование способов эффективного чтения оперативной памяти. Параметры 2,64

Вывод

Правильное упорядочивание данных ускоряет алгоритм. Упорядочиваем не по логике, а по удобству последующих вычислений.

Эксперимент 5. Исследование конфликтов в кэш-памяти

Параметры:

1. Размер банка кэш-памяти
2. Размер линейки кэш-памяти
3. Количество читаемых линеек

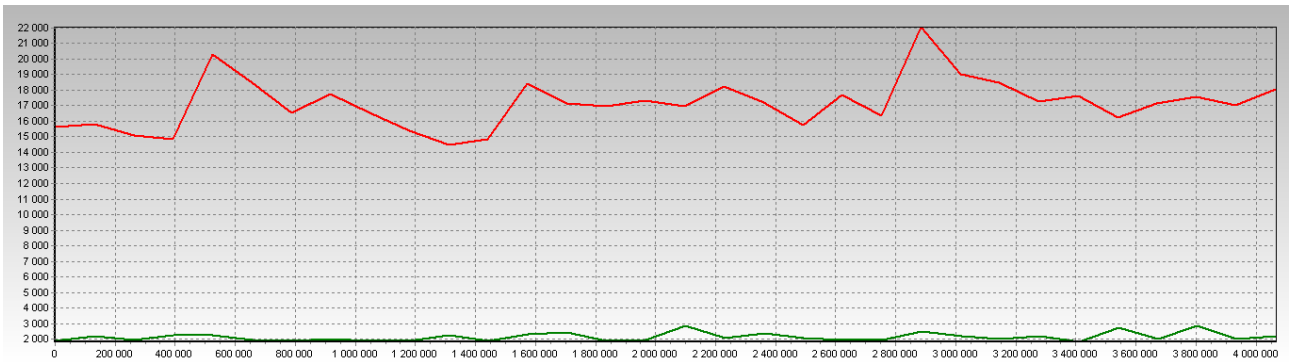

Рисунок 7: Исследование конфликтов в кеш-памяти. Параметры: 128, 128, 32.

В эксперименте сравнивается скорость работы процессора с кеш-памятью и без. (делается так, чтобы кеш-память работала максимально не эффективно)

Шаг 128K - размер банка, банков 8, следовательно, всего кеш памяти 1M.

Зел. лучше красного в 8 раз

Вывод

Кеш память ускоряет работу процессу в 8 раз.

Эксперимент 6. Сравнение алгоритмов сортировки

Параметры:

1. Количество 64-х разрядных элементов массивов
2. Шаг увеличения размера массива

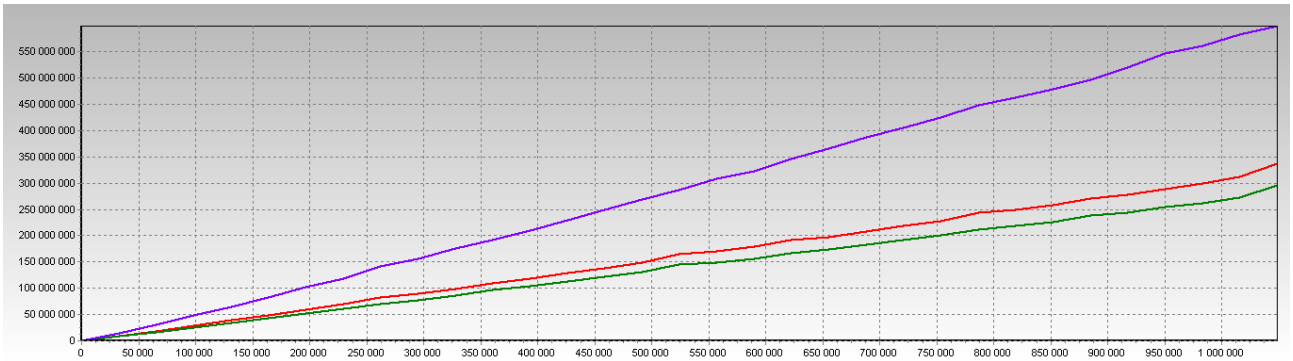

Рисунок 8: Сравнение алгоритмов сортировки. Параметры: 1, 32.

Сложность приведенного алгоритма сортировки O(n/log(n)).

Вывод

Существует поразрядный алгоритм сортировки с менее чем линейной сложностью

Radix-counting sort.