

Лабораторная работа 3
**Параллельные алгоритмы сортировки в OpenMP
и их практическое применение**

Цель работы. *Изучение различных алгоритмов сортировки и их параллельных вариантов реализации. Знакомство с медианным фильтром для обработки изображений.*

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Ознакомиться с теоретическими сведениями.
2. Ознакомиться с алгоритмами сортировки.
3. Выполнить задания 1–2.
4. Подготовить отчет по лабораторной работе.
5. Защитить лабораторную работу перед преподавателем

ЗАДАНИЯ

Задание 3.1. Разработайте консольное приложение, реализующее сортировку одномерного массива разными алгоритмами (в последовательной и параллельной реализации). При помощи этого приложения выполните экспериментальное исследование на четырех наборах (20 000–50 000 эл.)¹ по обработке разных типов данных (**int**, **double**) с разными реализациями (не менее 30 запусков для каждой реализации). В приложении необходимо реализовать следующие методы сортировки:

- пузырька (классический алгоритм), только последовательный вариант;
- пузырька (алгоритм Чет-нечетной перестановки)²;
- Шелла;
- быстрой сортировки (qsort).

Для каждой реализации функций сортировки массива выполните расчет времени и сформируйте сводные таблицы. При этом сортировку производить для одного и того же набора данных, т. е. исходные значения массива копируются в массив, предназначенный для сортировки. Заполнение массива осуществлять с учетом некоторой закономерности (придумать функцию генерации данных в зависимости от индекса). Результаты эксперимента оформите в виде табл. 3.1.

¹ В зависимости от оборудования (на котором будут выполняться эксперименты) и согласованию с преподавателем возможно изменения наборов данных в сторону уменьшения.

² Пример вариантов последовательной реализации сортировки «Чет-нечетной перестановки» доступен на сервере Дистанционного образования СибГУ им. М.Ф. Решетнева

Таблица 3.1

Функция сортировки	Поток	Время	$S_p(n)$	$E_p(n)$	Время	$S_p(n)$	$E_p(n)$
Тип Int		НД1 20 000			НД		
Классический алгоритм пузырька							
Чет-нечетной перестановки (посл.)							
Чет-нечет перестановки (парал.)	2						
	3						
	4						
Шелла (посл.)							
Шелла (парал.)	2						
	3						
	4						
Быстрой сортировки (посл.)							
Быстрой сортировки (парал.)	2						
	3						
	4						
Тип Double		НД1 20 000			НД		
Классический алгоритм пузырька							
Чет-нечетной перестановки (посл.)							
Чет-нечет перестановки (парал.)	2						
	3						
	4						
Другие алгоритмы и варианты реализации аналогично типу данных int							

Задание 3.2. Разработайте консольное приложение, реализующее выполнение медианной фильтрации изображения. В приложение должен загружаться набор из четырех файлов `input_X.bmp` (где X – номер файла) и формироваться выходные файлы `output_X_alg.bmp`, где `alg` означает имя используемого алгоритма обработки (или его номер). При реализации алгоритмов медианной фильтрации необходимо использовать не менее двух алгоритмов сортировки (на выбор студента взять из задания 3.1) и не менее двух вариантов распараллеливания (например, на основе распараллеливания циклов или создания параллельных секций).

При помощи разработанного приложения проведите экспериментальные исследования по обработке данных с разными реализациями (не менее **20 запусков** для каждой реализации). При применении каждого алгоритма получить показатели для разных размеров окрестности $K_{size} \times K_{size}$, для четырех изображений с разрешением от 1280×720 до 3840×2160 ³. В рамках лабораторной работы $K_{size} = (RH \times 2 + 1) = (RW \times 2 + 1)$. Результаты экспериментального исследования оформить в виде табл. 3.2.

³ В зависимости от оборудования (на котором будут выполняться эксперименты) и согласованию с преподавателем возможно изменения наборов данных в сторону уменьшения.

Таблица 3.2

Функция (реализация)	Поток	K_{size}	НД 1		НД 2		НД 3		НД 4	
			Время	$S_p(n)$	Время	$S_p(n)$	Время	$S_p(n)$	Время	$S_p(n)$
Вариант реализации 1 (посл.)		7								
		11								
		15								
Вариант реализации 1 (парал.)	2	7								
		11								
		15								
	3	7								
		11								
		15								
	4	...								
Вариант реализации 2 (посл.)		7								
		11								
		15								
Вариант реализации 2 (парал.)	2	7								
		11								
		15								
	3	...								
	4	...								
Иные варианты.										

ОТЧЕТ

Отчет сдается преподавателю в электронном виде и должен содержать:

- титульный лист;
- цель работы;
- постановку задачи;
- исходный код программы задания 3.1;
- конфигурация компьютера и параметры операционной системы, на которой производится выполнение задания 3.1;
- результаты экспериментального исследования программы задания 3.1 в табличной форме;
- выводы по результатам экспериментального исследования;
- текст программы задания 3.2, с комментариями;
- конфигурация компьютера и параметры операционной системы, на которой производится выполнение задания 3.2;
- примеры исходных и обработанных данных (картинки);
- результаты выполнения задания 3.2 (экспериментального исследования) в табличной форме;
- выводы по результатам экспериментального исследования;
- выводы по лабораторной работе в целом.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Охарактеризуйте алгоритм сортировки «Пузырьком».
2. Приведите алгоритм сортировки «Чет-нечетной перестановки».
3. Раскройте сущность сортировки алгоритмом «Шелла».
4. Охарактеризуйте сортировку алгоритмом qsort.
5. Перечислите виды параллелизма для алгоритмов сортировки.
6. Поясните известные вам способы распараллеливания сортировки «Чет-нечетной перестановки».
7. Приведите параллельные алгоритмы сортировки «Шелла».
8. Приведите параллельные алгоритмы быстрой сортировки.
9. Опишите подходы к параллельной обработке изображений.
10. Раскройте суть медианной фильтрации изображения.
11. Опишите известные вам алгоритмы медианной фильтрации изображения.
12. Расскажите о методах распараллеливания алгоритмов медианной фильтрации.
13. Назовите известные вам возможные проблемы при реализации параллельной обработки изображений.
14. Охарактеризуйте подходы к обработке изображений на основе параллельных секций.
15. Охарактеризуйте способы представления данных изображения для повышения быстродействия алгоритмов обработки.

ИНФОРМАЦИЯ ПО МЕДИАННОМУ ФИЛЬТРУ

Медианный фильтр представляет собой скользящую по полю изображения нечетную апертуру фильтра (окно по окрестности анализируемого пиксела). В выходное изображение записывается медиана всех элементов (пикселей) изображения, попавших в апертуру фильтра на оригинальном изображении.

Медианой дискретной последовательности $X = \{x_1, x_2, \dots, x_L\}$ для нечетного L называют такой ее элемент, для которого существуют $(L - 1)/2$ элементов, меньших или равных ему по величине, и $(L - 1)/2$ элементов, больших или равных ему по величине. Другими словами, медианой является средний по порядку член ряда, получающегося при упорядочении исходной последовательности. Например, $\text{med}(21, 2, 10, 7, 7) = 7$.

Суть медианной фильтрации изображения заключается в следующем: в одномерный массив заносятся значения пикселей из окрестности, после этого выполняется сортировка и в качестве результирующего значения заносится значение, содержащееся в центральном элементе массива.

Существует также быстрый алгоритм медианной фильтрации, о котором можно почитать в статье: Бардин Б. В. Быстрый алгоритм медиан-

ной фильтрации / Научное приборостроение, 2011, том 21, No 3, с. 135–139
доступной по адресу: <http://iairas.ru/mag/2011/full3/Art16.pdf>.

Пример возможной реализации классического медианного фильтра для обработки одного канала показан с использованием псевдокода.

Используемые обозначения:

IMAGE[] []	– массив элементов входного изображения в формате адресации [Y][X]
IMAGE2[] []	– массив элементов выходного изображения в формате адресации [Y][X]
Height, Width	– размеры изображения по высоте и ширине
RH, RW	– размеры рангов скользящего окна
size	– площадь окна фильтрации $size = M \times N = (RH \times 2 + 1) \times (RW \times 2 + 1)$
MEDMAS[size]	– массив элементов медианы с размером size
Masind	– переменная для индексации элементов массива MEDMAS при заполнении

Медианный фильтр

Начало

Цикл 1 по Y изображению (Y **от** 0 **до** Height)

Цикл 2 по X изображению (X **от** 0 **до** Width)

Начало 2

Заполнение массива MEDMAS элементами изображения

IMAGE (X, Y, RH, RW)

Сортировка массива любым алгоритмом

Запись в выходное изображение IMAGE2[Y][X] значения центрального элемента массива **MEDMAS** (**MEDMAS[size/2]**)

Конец 2

Конец

Заполнение массива MEDMAS элементами изображения

IMAGE (X, Y, RH, RW)

Начало

Masind=0;

Цикл A1 (DY **от** -RH **до** RH **включительно**)

Начало A1

KY = Y + DY

Если KY < 0 **то** KY=0

Если KY > Height - 1 **то** KY = Height - 1

Цикл A2 (DX **от** -RW **до** RW **включительно**)

Начало A2

```

KX = X + DX
Если KX < 0 то KX=0
Если KX > Width - 1 то KX= Width-1
MEDMAS[Masind] = IMAGE[KY][KX]
Masind = Masind+1;
Конец A2
Конец A1
Конец

```

При программной реализации заполнения массива MEDMAS элементами изображения можно не выделять в отдельную функцию.

В случае использования трех цветовых каналов (для цветного RGB изображения) потребуется заполнять и сортировать три массива: MEDMAS_R, MEDMAS_G и MEDMAS_B.