

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО
ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский
Нижегородский государственный университет им. Н.И. Лобачевского»
(ННГУ)

Институт информационных технологий, математики и механики

Направление подготовки: «Фундаментальная информатика и
информационные технологии»

Отчёт по лабораторной работе

**Поразрядная сортировка для целых чисел со
слиянием "Разделяй и властвуй".**

Выполнил:
Студент ИИТММ гр. 381606-1

Дубовской А.В.

Проверила:
доцент каф. МОСТ, ИИТММ,
кандидат технических наук

Сысоев А.В.

Нижний Новгород
2019 г.

Оглавление

Постановка задачи	3
Описание алгоритма	4
Описание схемы распараллеливания	5
Описание OpenMP-версии	5
Описание TVB-версии	6
Результаты экспериментов	7
Выводы из результатов	8
Заключение	9
Литература	10

Постановка задачи

Необходимо реализовать программу, которая делит произвольный массив целых чисел на число кусков, соответствующее заданному числу потоков, сортирует их по возрастанию, используя алгоритм поразрядной сортировки и собирает их в отсортированный массив, используя слияние «Разделяй и властвуй».

Цель данной работы:

1. Реализовать последовательную и параллельную версию программы;
2. Оценить эффективность и масштабируемость данной программы на кластере (или многопоточный запуск на персональном компьютере).
3. Проанализировать полученные результаты и исходя из анализа данных сделать заключение

Описание алгоритма

Алгоритм работы последовательной версии программы делится на следующие части:

1. Генерация данных.
2. Сортировка и слияние.
3. Проверка корректности и вывод результатов.

На этапе 1 генерируются входные данные программы в виде массива целых чисел.

На втором этапе происходит сортировка входного массива данных с помощью алгоритма поразрядной LSD сортировки. При LSD сортировке (сортировке с выравниванием по младшему разряду, направо, к единицам) получается порядок, уместный для чисел. Например: 1, 2, 9, 10, 21, 100, 200, 201, 202, 210. То есть, здесь значения сначала сортируются по единицам, затем сортируются по десяткам, сохраняя отсортированность по единицам внутри десятков, затем по сотням, сохраняя отсортированность по десяткам и единицам внутри сотен, и т. п.

Пример:

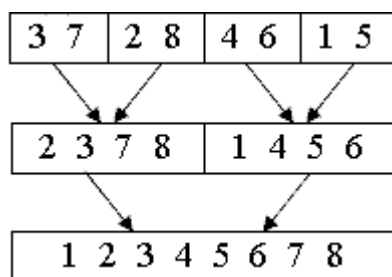
исходная k=3 последовательность	первый проход по 3му разряду	второй проход по 2му разряду	третий проход по 1му разряду
523	523	523	088
153	153	235	153
088	554	153	235
554	235	554	523
235	088	088	554
	↑	↑	↑
Offset(номер позиции)	3	2	1

Этап 3 работы программы является заключительным. В консоль выводится результат корректности сортировки, время работы последовательной и параллельной версии программы.

Описание схемы распараллеливания

На этапе 2 параллельной версии программы массив делится на отрезки, кол-во которых равно количеству указанных потоков, остаток добавляется к последнему отрезку. Далее, с помощью поразрядной сортировки каждым потоком сортируются полученные отрезки.

Отсортированные отрезки сливаются методом «Разделяй и властвуй». Суть метода состоит в разбиении массивов на участки, которые можно слить независимо. В первом массиве выбирается последний элемент - x , 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 первый массив и первую часть второго массива можно сливать независимо, вторая половина второго массива добавляется целиком в конец полученного массива. Пример:



Корректность результата проверяется проверкой возрастания элементов в полученном массиве.

Описание OpenMP версии

В OpenMP-версии, как говорилось ранее, все вычисления распределяются между потоками. Деление массива на куски происходит непосредственно внутри функции, с запоминанием размера каждого куска и размером остатка. Далее каждый кусок сортируется поразрядной LSD сортировкой. После того как все куски массива будут отсортированы произойдёт слияние с помощью вспомогательных массивов, хранящих в себе адреса левых и правых концов каждого куска. По завершению алгоритма функция возвращает отсортированный массив.

Для распараллеливания данного алгоритма использовались директивы `parallel for`, предназначенные для распараллеливания циклов, с параметром `schedule(dynamic, 1)`, который делит итерации на блоки размером 1 и динамически разделит их между потоками.

Описание ТВВ версии

Для распараллеливания алгоритма сортировки в ТВВ-версии используется шаблонная функция `tbb::parallel_for` с использованием одномерного итерационного пространства с диапазоном в виде полуинтервала `[begin, end)`. Алгоритм деления массива на куски и последующего слияния аналогичен алгоритму из `openMP`-версии.

Результаты экспериментов

Ускорением параллельного алгоритма называют отношение времени выполнения лучшего последовательного алгоритма к времени выполнения параллельного алгоритма:

$$S = \frac{T_l}{T_p}$$

Параллельный алгоритм может давать большое ускорение, но использовать для этого множество процессов неэффективно. Для оценки масштабируемости параллельного алгоритма используется понятие эффективности:

$$E = \frac{S}{p}$$

Эксперименты проводились на персональном компьютере с использованием многопоточного запуска.

Характеристики персонального компьютера:

1. Процессор Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz (4 ядра)
2. Оперативная память 4 Гб 1600 МГц DDR3

Выводы из результатов

Из таблицы эффективностей OpenMP и TBB видно, что максимальная эффективность достигается на 2 потоках. Это связано с тем, что при увеличении числа потоков возрастают накладные расходы при делении массива перед сортировкой, и слияние массивов после сортировки. При этом из таблицы можно видеть, что с увеличением числа потоков вдвое, эффективность так же падает примерно вдвое.

Заключение

Была реализована последовательная и параллельная реализация поразрядной сортировки вещественных чисел со слиянием «Разделяй и властвуй», замерено время их работы, а также вычислена эффективность на разных объемах данных и количествах потоков.

Схожесть результатов OpenMP и TBB версий можно объяснить тем, что при разработке программ использовались идентичные способы распараллеливания для уменьшения переработки существующего кода.

Литература

1. Гергель, В.П. Параллельное программирование с использованием OpenMP // Нижегородский государственный университет им. Н.И. Лобачевского / Национальный исследовательский университет. – 2014. №1.
2. Сысоев, А.В. Параллельное программирование // Параллельное программирование с использованием OpenMP // Нижегородский государственный университет им. Н.И. Лобачевского / Национальный исследовательский университет. – 2016. №2.
3. Мееров И. Б. Инструменты параллельного программирования для систем с общей памятью. Библиотека Intel Threading Building Blocks – краткое описание / Сысоев А.В., Сиднев А.А. // Нижегородский государственный университет им. Н.И. Лобачевского. – 2009. – с. 172.
4. Седжвик Р. Алгоритмы на C++ // «Национальный Открытый Университет «ИНТУИТ».
5. Сиднев А.А., Сысоев А.В., Мееров И.Б. Параллельные численные методы: «Лабораторная работа: Сортировки».