



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»

Лабораторная работа № 5

по дисциплине «Анализ Алгоритмов»

Тема Организация параллельных вычислений по конвейерному принципу

Студент Пермякова Е. Д.

Группа ИУ7-52Б

Преподаватели Строганов Д. В., Волкова Л. Л

Москва, 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Входные и выходные данные	5
2 Преобразование входных данных в выходные	6
3 Примеры работы программы	7
4 Тестирование	8
5 Описание исследования	9
ЗАКЛЮЧЕНИЕ	11
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	12

ВВЕДЕНИЕ

Целью работы является разработка алгоритма, который будет раскладывать числа в группы так, чтобы разница максимальной и минимальной среди сумм чисел каждой группы была наименьшей, с использованием конвейерного метода.

Задачи:

- 1) разбить поставленную задачу на 3 подзадачи;
- 2) реализовать решение каждой из подзадач;
- 3) организовать выполнение подзадач по конвейерному принципу с использованием потока-генератора и трех дополнительных потоков;

1 Входные и выходные данные

Входными данными программы является txt-файл, состоящий из строк вид <строковый ключ из символов a-zA-Z0-9>: <число>.

Выходными данными программы являются N файлов с именами group_<i>.<имя исходного файла>.txt, где i – номер группы, N – количество логических ядер машины.

2 Преобразование входных данных в выходные

Программа выполняет раскладку чисел в группы (размера N , N — по умолчанию количество логических ядер машины) так, чтобы разница максимальной и минимальной среди сумм чисел каждой группы была наименьшей, и записывает в N файлов с именами `group_<i>.<имя исходного файла>.txt`, где i — номер группы, каждую группу в формате

<сумма чисел группы>

<ключ 1>: <значение 1>

...

<ключ N >: <значение N >

Ключи отсортированы в лексикографическом порядке по возрастанию

3 Примеры работы программы

На рисунке 3.1 представлены пример содержимого лог-файла, где $tstart$, $tend$ – начало и конец обслуживания заявки с номером $request$ выполняемой на потоке $thread$.

tstart=249	tend=270	request=0	thread=1
tstart=271	tend=273	request=0	thread=2
tstart=274	tend=275	request=1	thread=1
tstart=274	tend=424	request=0	thread=3
tstart=275	tend=276	request=2	thread=1
tstart=276	tend=276	request=3	thread=1
tstart=276	tend=276	request=4	thread=1
tstart=277	tend=277	request=5	thread=1
tstart=277	tend=277	request=6	thread=1

Рисунок 3.1 — Пример содержимого лог-файла

4 Тестирование

В таблице 4.1 представлены функциональные тесты разработанного алгоритма распределения чисел по группам так, чтобы разница максимальной и минимальной среди сумм чисел каждой группы была наименьшей. Для упрощения определения правильности работы алгоритма было взято N равное 3.

Все тесты пройдены успешно.

Таблица 4.1 — Функциональные тесты

№	Входные данные	Выходные данные	Ожидаемые выходные данные
1	0 5 9 9 8 3 2 1 0 2	Группа 1: 0 9 2 2 0 (сумма 13) Группа 2: 5 8 (сумма 13) Группа 3: 9 3 1 (сумма 13)	Группа 1: 0 9 2 2 0 (сумма 13) Группа 2: 5 8 (сумма 13) Группа 3: 9 3 1 (сумма 13)
2	-	Группа 1: (сумма 0) Группа 2: (сумма 0) Группа 3: (сумма 0)	Группа 1: (сумма 0) Группа 2: (сумма 0) Группа 3: (сумма 0)
3	1 1 1	Группа 1: 1 (сумма 1) Группа 2: 1 (сумма 1) Группа 3: 1 (сумма 1)	Группа 1: 1 (сумма 1) Группа 2: 1 (сумма 1) Группа 3: 1 (сумма 1)

5 Описание исследования

Был выбран язык программирования с++, так как на linux с его помощью можно создавать нативные потоки [1]. Проводились замеры реального времени работы программы, для этого использовалась функция gettimeofday [2]. Замеры проводились на виртуальной машине Linux с 8 логическими ядрами, и замерялось время для 3 дополнительных рабочих потоков и 1 главного потока–диспетчера.

Было проведено исследование параллельности обработки задач при конвейерном методе. В результате был сформирован лог обработки задач 5.1 и рассчитаны средние временные характеристики 5.2.

Таблица 5.1 — Лог

№ заявки	№ потока	Время начала обслуживания заявки, мс	Время конца обслуживания заявки, мс
0	1	238	250
0	2	251	254
0	3	254	373
1	1	256	256
2	1	257	257
3	1	257	257
4	1	258	258
5	1	258	258
6	1	259	259
7	1	259	259
8	1	260	260
9	1	260	260
10	1	261	280
1	2	265	266
2	2	266	267
...
496	3	36204	36270
497	3	36271	36339

Продолжение таблицы 5.1

№ заявки	№ потока	Время начала обслуживания заявки, мс	Время конца обслуживания заявки, мс
498	3	36340	36404
499	3	36404	36471

Таблица 5.2 — Средние временные характеристики

Средние временные характеристики	Время в мс
Среднее время существования задачи	18519.166
Среднее время ожидания задачи в очереди 2	321.256
Среднее время ожидания задачи в очереди 3	18124.676
Среднее время обработки задачи на стадии 1	0.354
Среднее время обработки задачи на стадии 2	1.216
Среднее время обработки задачи на стадии 3	71.664

Из проведённых замеров был сделан вывод о том, что решение задач происходит параллельно, так как временные отметки начала и конца обработки задач пересекаются.

ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы была выполнена поставленная цель, которая заключалась в разработке программного обеспечения, которое будет раскладывать числа в группы так, чтобы разница максимальной и минимальной среди сумм чисел каждой группы была наименьшей, с использованием конвейерного метода.

Были выполнены следующие задачи:

- 1) разбить поставленную задачу на 3 подзадачи;
- 2) реализовать решение каждой из подзадач;
- 3) организовать выполнение подзадач по конвейерному принципу с использованием потока-генератора и трех дополнительных потоков;

Основываясь на проведенном исследовании был сделан вывод о том, что решение задач происходит параллельно, так как временные отметки начала и конца обработки задач пересекаются.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. pthread_create(3) — Linux manual page [Электронный ресурс]. Режим доступа: https://man7.org/linux/man-pages/man3/pthread_create.3.html. (дата обращения: 20.10.2024).
2. gettimeofday(2) — Linux manual page [Электронный ресурс]. Режим доступа: <https://man7.org/linux/man-pages/man2/gettimeofday.2.html>. (дата обращения: 20.10.2024).