

Задание по алгоритму имитации отжига

Шибает Павел

421 группа, кафедра АСВК, факультет ВМК МГУ им. М.В. Ломоносова

Содержание

1	Прикладная задача	1
2	Формальная постановка задачи	2
3	Экспериментальное исследование последовательного алгоритма	3
4	Экспериментальное исследование параллельного алгоритма	4

1 Прикладная задача

Дано N независимых работ, для каждой работы задано время выполнения. Требуется построить расписание выполнения работ без прерываний на M процессорах. На расписании должно достигаться минимальное значение критерия.

Критерии

- Критерий 1: длительность расписания (т.е. время завершения последней работы).
- Критерий 2: суммарное время ожидания (т.е. сумма, по всем работам, времён завершения работ).
- **Критерий 3:** разбалансированность расписания (т.е. значение разности $T_{max} - T_{min}$, где T_{max} — наибольшая, по всем процессорам, длительность расписания на процессоре; T_{min} — аналогично, наименьшая длительность).

Пояснение к критериям

0: АААББ

1: ВВВВВВВВВВ

- Значение критерия 1: 11 (последним освобождается процессор 1).

- Значение критерия 2: $19 = 3 + 5 + 11$ (сумма времен завершения работ А, Б, В; эти времена считаются не от стартов работ, а от начала расписания).
- Значение критерия 3: 6 (разность между временами освобождения процессоров 1 и 0).

2 Формальная постановка задачи

Дано

Есть множество работ $P = \{p_i\}_{i=1}^N$, где N — число работ. Работы непрерываемые и независимы. Каждая работа имеет время выполнения: работа p_i выполняется время $time_i$. Есть множество процессоров $PU = \{PU_i\}_{i=1}^K$, где K — число процессоров. Каждый процессор может взять на выполнение только одну работу.

Необходимо создать такое расписание HP , чтобы оно уменьшало заданный критерий $F(HP)$.

Расписание HP определено двумя компонентами: HP_B и HP_L . Если заданы HP_B и HP_L , тогда это расписание. Здесь, $HP_B : P \rightarrow PU$ — это отображение задач на процессоры, а HP_L определяет очередность выполнения задач на каждом процессоре.

Чтобы HP было правильным, необходимо удовлетворять следующим условиям:

1. Все задачи должны быть назначены на процессоры.
2. Каждая задача обрабатывается только одним процессором.
3. Процессор может брать на выполнение только одну задачу, выполнить её и только затем приступить к выполнению следующей

Критерий для минимизации

Основной целью является уменьшение разбалансированности расписания (берется разность между наибольшей и наименьшей длительностями выполнения на процессоре). Эти длительности можно узнать из временной диаграммы расписания.

Для создания этой диаграммы для каждого процессора задачи упорядочены в соответствии с их отображением и порядком G_{HP} . Время начала первой задачи p_i на процессоре CPU_k задано как $t_{k,i}^{start} = 0$. Для всех последующих задач на этом процессоре, $t_{k,i}^{start}$ равно $t_{k,j}^{finish}$, где p_j это задача перед p_i в порядке G . Время завершения задачи p_i определено как $t_{k,i}^{finish} = t_{k,i}^{start} + time_i$.

Таким образом, критерий для минимизации определен следующим образом:

$$F(HP) = \max_{j=1}^{j=K} \left(\sum_{i=1}^N t_{j,i}^{finish} \right) - \min_{j=1}^{j=K} \left(\sum_{i=1}^N t_{j,i}^{finish} \right)$$

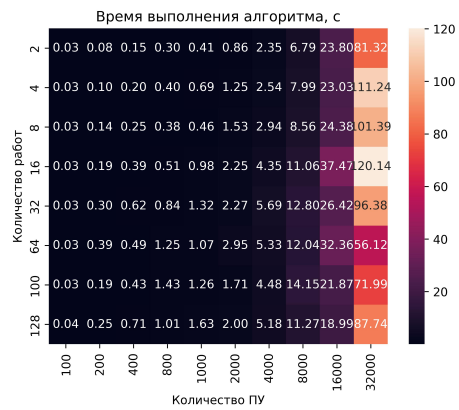


Рис. 1: Исследование времени работы алгоритма

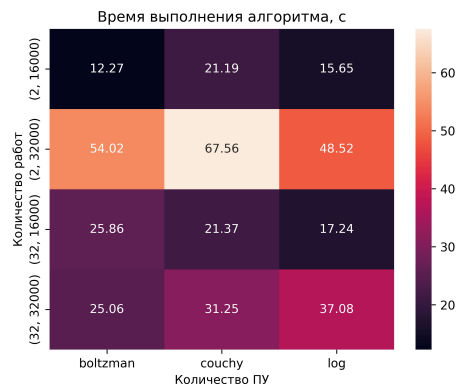


Рис. 2: Enter Caption

3 Экспериментальное исследование последовательного алгоритма

При 16000 работах и 8 процессорах алгоритм работает примерно 101 секунду (выбран закон Коши). Ниже приведены данные по времени работы алгоритма с этим законом.

Наиболее долгое время выполнения алгоритма на "тяжелых" данных наблюдается при запуске с законом Коши. Экспериментальные данные запусков приведены ниже.



Рис. 3: Зависимость времени выполнения от количества потоков



Рис. 4: Зависимость критерия от количества потоков

4 Экспериментальное исследование параллельного алгоритма

В дальнейших исследованиях применялся закон Коши, параллельные запуски проходили для испытаний с 32000 работами и 8 процессорами. Ниже представлена зависимость времени выполнения параллельной программы от количества выделяемых потоков. Локальный минимум наблюдается при 7 потоках, что можно объяснить особенностями работы чипов Apple M2 Pro 2023. Наконец, интересен вопрос об эффективности параллельного алгоритма. Видно, что с увеличением количества потоков качество растёт. Наилучшее качество — при максимальном количестве потоков. Ответы на поставленные вопросы.

1. Оптимальное значение количества потоков — от 4 до 7.
2. Далее, после 7 потоков, имеем даже ухудшение скорости работы.