

# Экспериментальное исследование разработанного алгоритма имитации отжига в рамках второго задания по курсу «Практикум на ЭВМ.»

Никифоров Никита Игоревич, 421 группа

## Введение

Необходимо было разработать алгоритм имитации отжига в применении к задаче построения распределённого расписания. В моём варианте необходимо было реализовать критерий K3, а именно 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). Также необходимо было реализовать многопоточный алгоритм имитации отжига с синхронизацией через разделяемую память.

## Методика экспериментального исследования

Все тестовые данные используемые в данном экспериментальном исследовании доступны в папке `tests` проекта. Так как алгоритм является недетерминированным бралась средняя оценка по критерию среди 10 прогонов. Время выполнения фиксировалось утилитой `c` с помощью библиотеки `std :: chrono`. Исследование проводилось без запущенных посторонних программ.

## Тестовый стенд

- Процессор — AMD Ryzen 2700X 8/16 (ядер/потоков) 4.2 Ghz на одно ядро, 4.0 Ghz на все ядра.
- Память — 32Gb DDR4 3200Mhz (Пропускная способность 44GB/s).

## Экспериментальное исследование

### Поиск опорной точки

Необходимо найти опорную точку экспериментального исследования, а именно вычислить количество задач и процессоров, такое, что время выполнения на одном потоке будет больше 1 минуты при использовании какого либо закона понижения температуры.

Таким образом на графике 1 мы видим, что алгоритм ведёт себя одинаково в рамках погрешности при изменении закона понижения температуры. Также мы видим, что для всех законов время выполнения заходит за минуту только на 40000 задач. Соответственно в дальнейшем я буду использовать именно это количество.

### Зависимость изменений в поведении алгоритма от количества потоков

Во второй части экспериментального исследования необходимо было исследовать изменения поведения алгоритма при изменении количество потоков. В данной части будет использован тестовый файл `test/procs/40000_10.xml`, и закон понижения температуры Коши. Как можно видеть по графикам 2 и 3, при увеличении количества процессов до 4-х время работы алгоритма не увеличивается, А точность алгоритма улучшается в переходе от одного процесса на два на 20%. Дальнейшее улучшение производительности не было значительным, а прирост времени исполнения сильным.

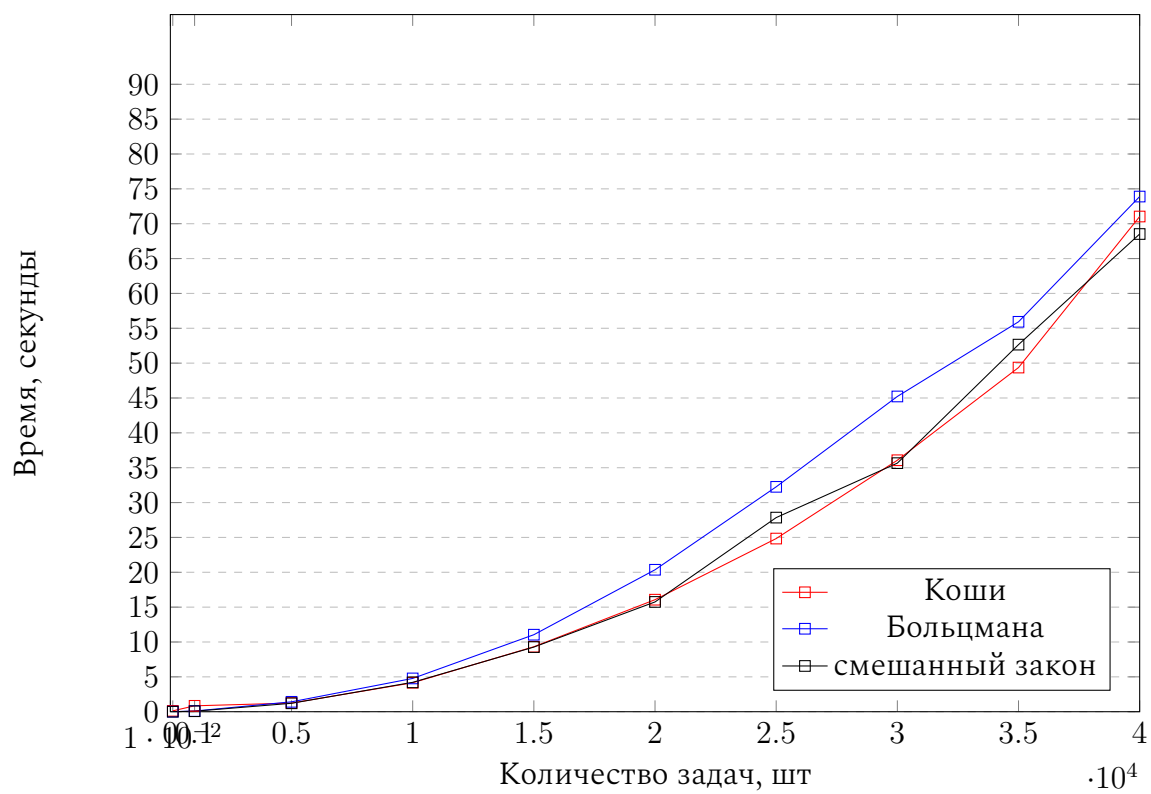


Рис. 1: Зависимость времени исполнения от количества задач

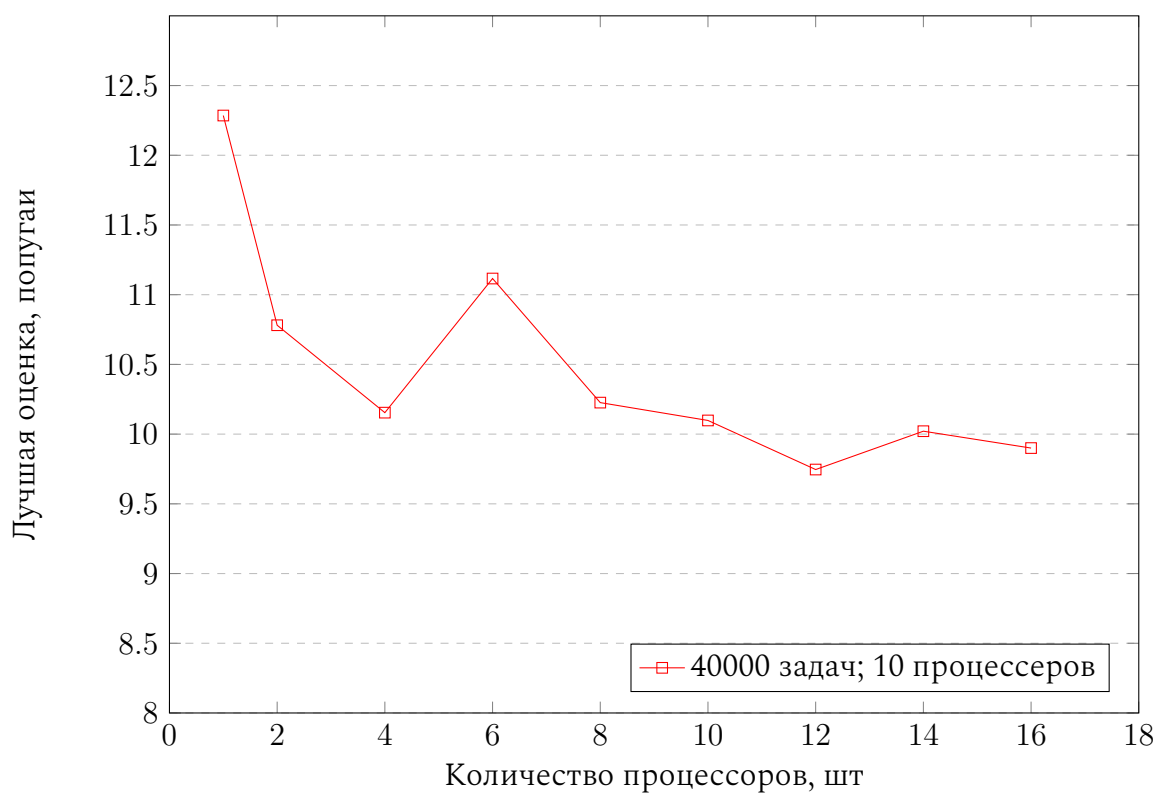


Рис. 2: Зависимость оценки лучшего решения от количества процессоров.

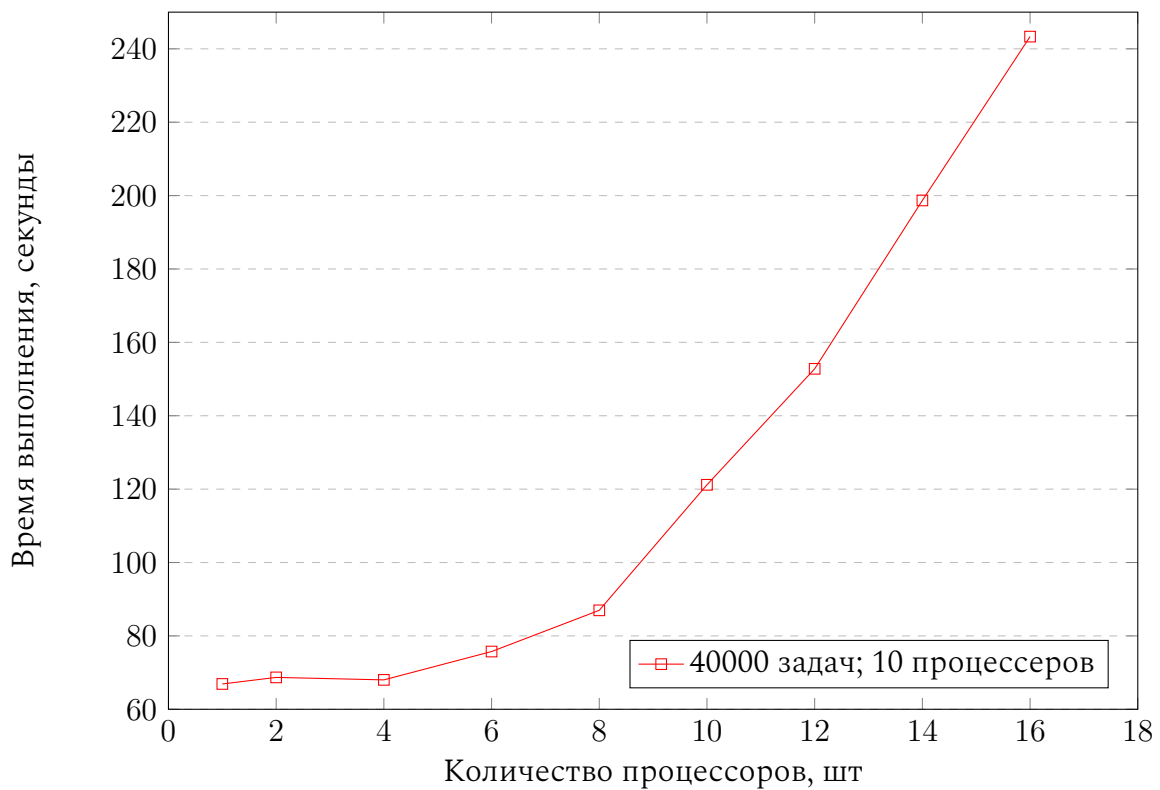


Рис. 3: Зависимость времени выполнения от количества процессов

## Послесловие

В папке *tests/root\_point* находится файл *50000\_100.xml*, если запустить программу вместе с сидом равным 1604022559, то при использовании оптимизированного начального решения, алгоритм не сходится. А именно после завершения Параллельного  $Nproc = 1$  алгоритма имитации отжига, оценка лучшего найденного решения равна оценке исходного 3196.39. При этом алгоритм работает меньше секунды. Если же задать прямое начальное решение, а именно, когда все задачи находятся на одном процессоре алгоритм имитации отжига работает более 2-х минут, и находит оптимальное решение оценка которого равна 51.69.

В данном случае мы наблюдаем стандартный вариант нахождения локального минимума.