

# OM FEST 2025

ФЕСТИВАЛЬ ЦИФРОВЫХ ТЕХНОЛОГИЙ

## На чем лучше всего писать код под “Эльбрус”

Станислав Емец  
НИЦ ЦТ, руководитель разработки



# КТО Я?

В ИТ с 1999 г., с отечественными процессорами работаю с 2017 года. Начинал как инженер, сейчас как руководитель двух команд.

В меру свободного времени пытаюсь продвигать отечественные разработки.

1. 2021 г. Облачная платформа для Эльбрусов.
2. 2025 г. Операционная система для Эльбрусов.

# ЦЕЛЬ ТЕСТИРОВАНИЯ

1. При разработке операционной системы много компонент написано на скриптовых языках, хотелось понять, что нам ожидать в части производительности.
2. Мы знаем, что код на C/C++ самый быстрый, ну по крайней мере в теории, так ли это на самом деле?
3. Понять какие языки для разработки советовать, партнерам и коллегам.



# МЕТОДИКА ТЕСТИРОВАНИЯ

В качестве набора тестов мы использовали ***Programming language benchmark***.

Набор тестов основан на решении задач идентичными алгоритмами на различных языках программирования. Это позволяет оценивать эффективность генерации кода различными компиляторами (или интерпретаторами) для выбранных архитектур. Тесты используют одно ядро процессора, что позволяет оценить потенциал оптимизации.

1. Bedcov
2. Matmul
3. Nqueen
4. Sudoku

# ОПИСАНИЕ ТЕСТОВ: BEDCOV

*Bedcov (поиск перекрытий)* - этот тест измеряет скорость поиска перекрывающихся интервалов между двумя массивами размером 1 000 000 элементов. Алгоритм использует структуры данных, подобные неявным интервальным деревьям, и выполняет частые обращения к массиву с помощью метода, похожего на бинарный поиск. Таким образом, тест оценивает эффективность работы с большими массивами данных и алгоритмами поиска.

# ОПИСАНИЕ ТЕСТОВ: MATMUL

*Matmul* (умножение матриц) - данный тест измеряет производительность умножения двух квадратных матриц размером 1500x1500. Этот тест фокусируется на вычислительной мощности и эффективности работы с математическими операциями, типичными для научных вычислений и задач линейной алгебры.

# ОПИСАНИЕ ТЕСТОВ: NQUEEN

*Nqueen* (задача о  $N$  ферзях) - этот тест оценивает скорость решения классической задачи о  $N$  ферзях для  $N=15$ . Алгоритм решения включает вложенные циклы и целочисленные битовые операции. Тест проверяет эффективность работы с рекурсивными алгоритмами, обработки битовых данных и общую производительность в задачах комбинаторного поиска.

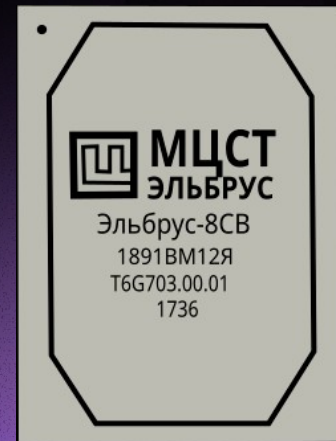
# ОПИСАНИЕ ТЕСТОВ: SUDOKU

*Sudoku* (решение судоку) - этот тест измеряет скорость решения 4000 сложных головоломок судоку (20 уникальных головоломок, каждая решается 200 раз). Используемый алгоритм активно задействует небольшие массивы и сложные логические проверки. Тест предназначен для оценки производительности в задачах с интенсивным использованием ветвлений, рекурсии и обработки небольших структур данных.



# ПРЕДМЕТ ТЕСТИРОВАНИЯ

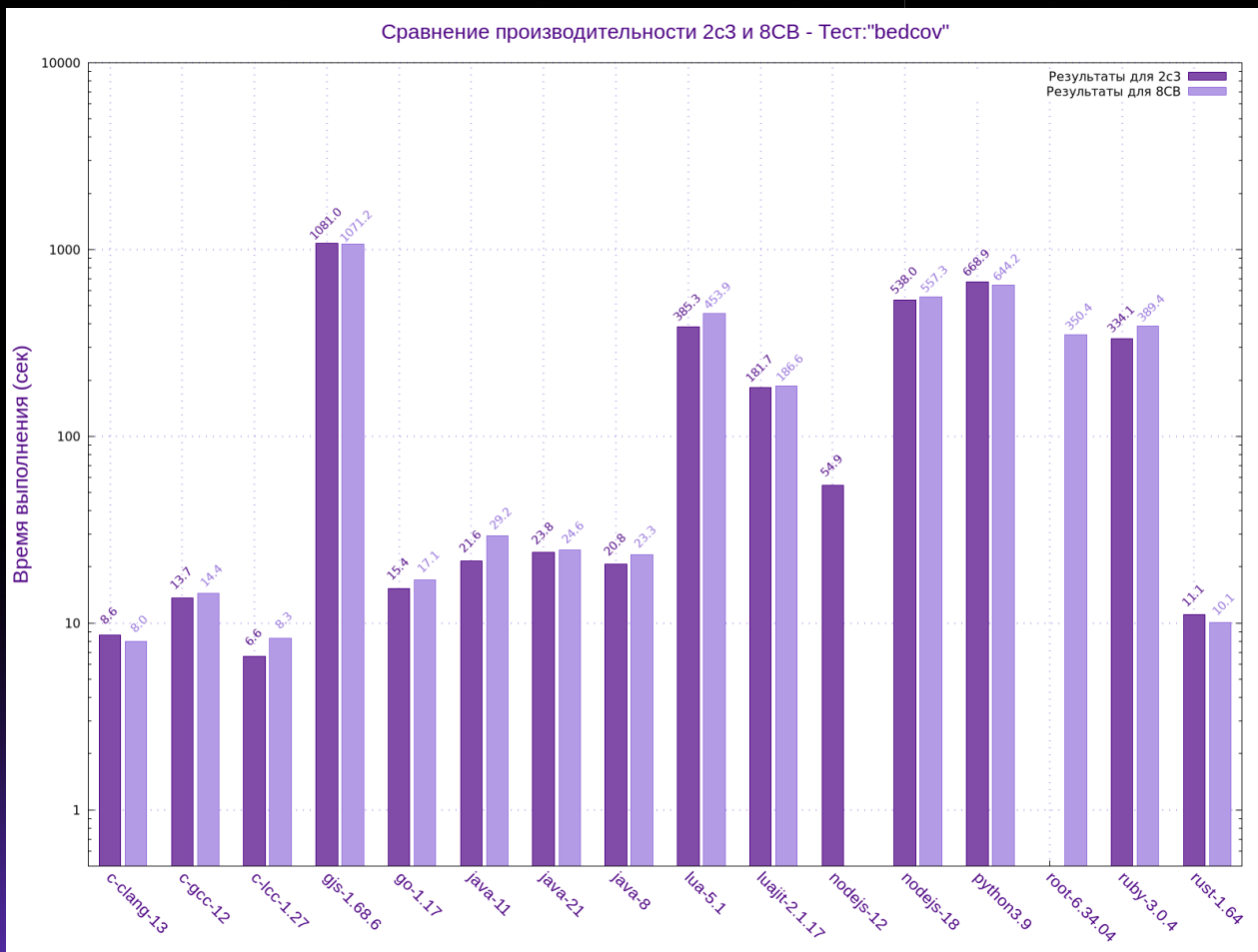
	Эльбрус 8СВ (8с2)	Эльбрус 2с3
Архитектура	VLIW v. 5	VLIW v. 6
Количество ядер	2	8x2
Максимальная частота	1500	2000
Кэш L1d	1 МБ (16 x 64 КБ)	128 КБ (2 x 64 КБ)
Кэш L1i	2 МБ (16 x 128 КБ)	256 КБ (2 x 128 КБ)
Кэш L2	8 МБ (16 x 512 КБ)	4 МБ (2 x 2 МБ)
Кэш L3	32 МБ (2 x 16 МБ)	-
Оперативная память	256 Гб	16 Гб



# BEDCOV

## Топ 5:

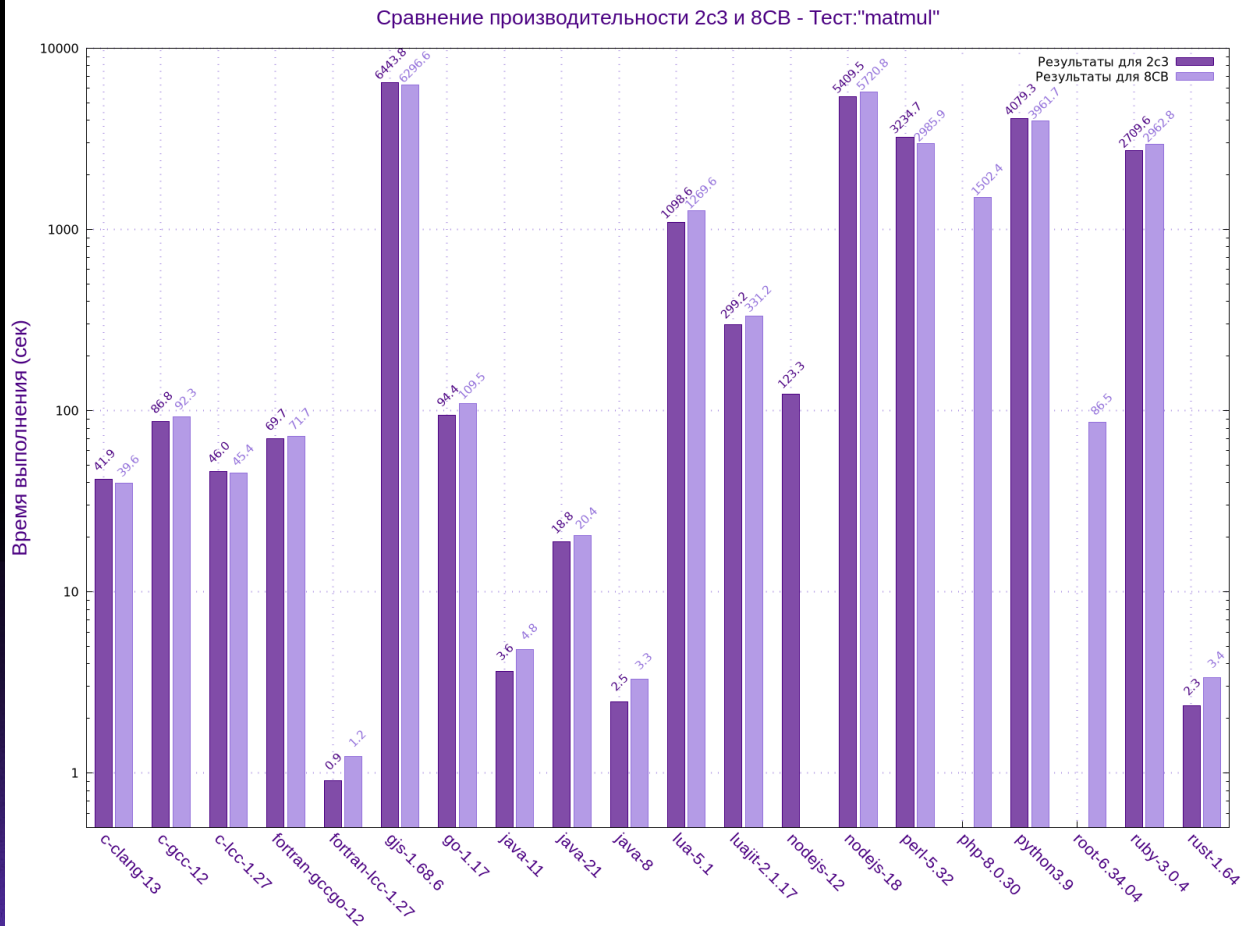
LCC  
Clang  
Rust  
GCC  
Go



Меньше лучше, шкала экспоненциальная

# MATMUL

**Топ 5:**  
Fortran  
Rust  
Java  
Clang  
LCC

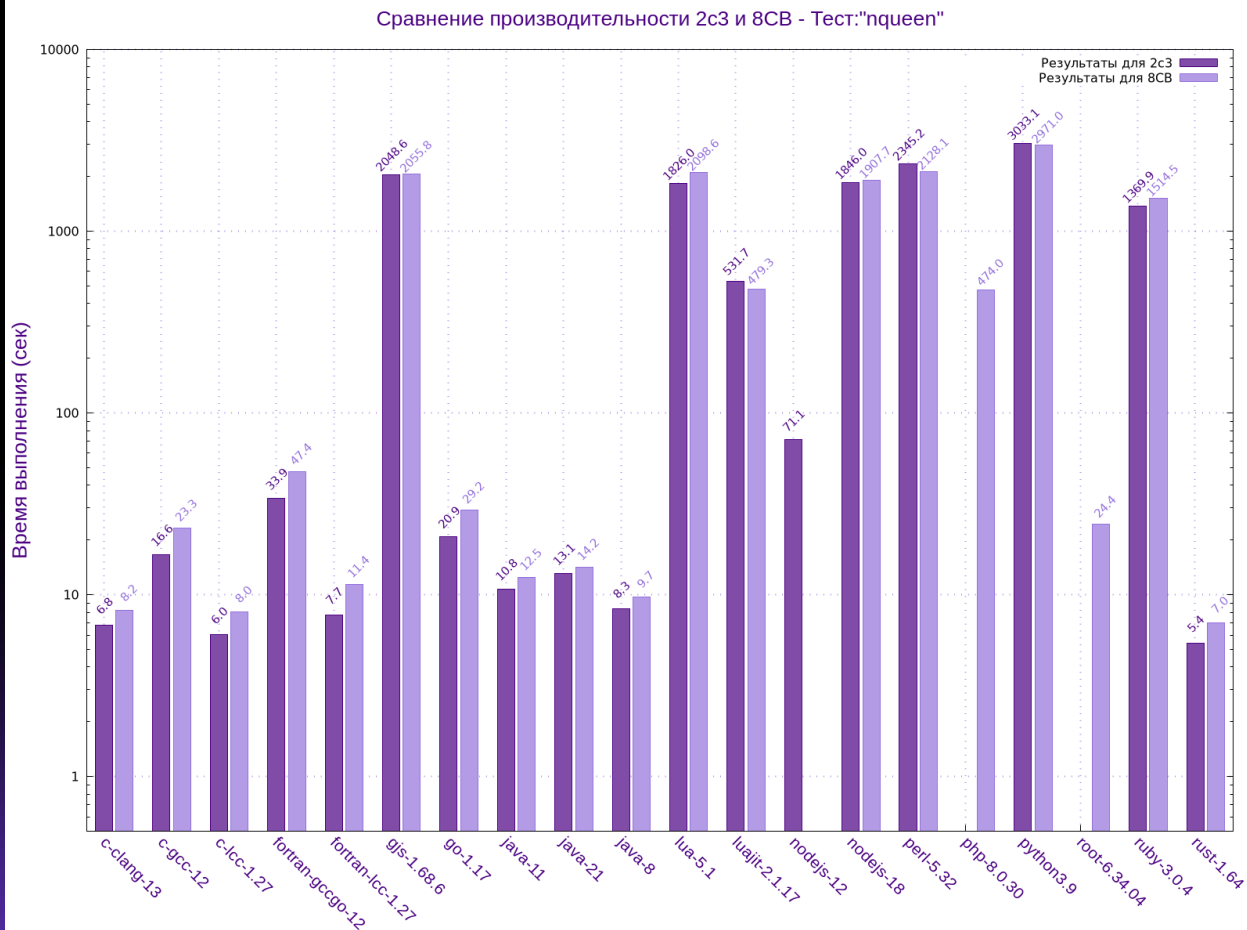


Меньше лучше, шкала экспоненциальная

# NQUEEN

## Топ 5:

Rust  
LCC  
Clang  
Fortran  
Java



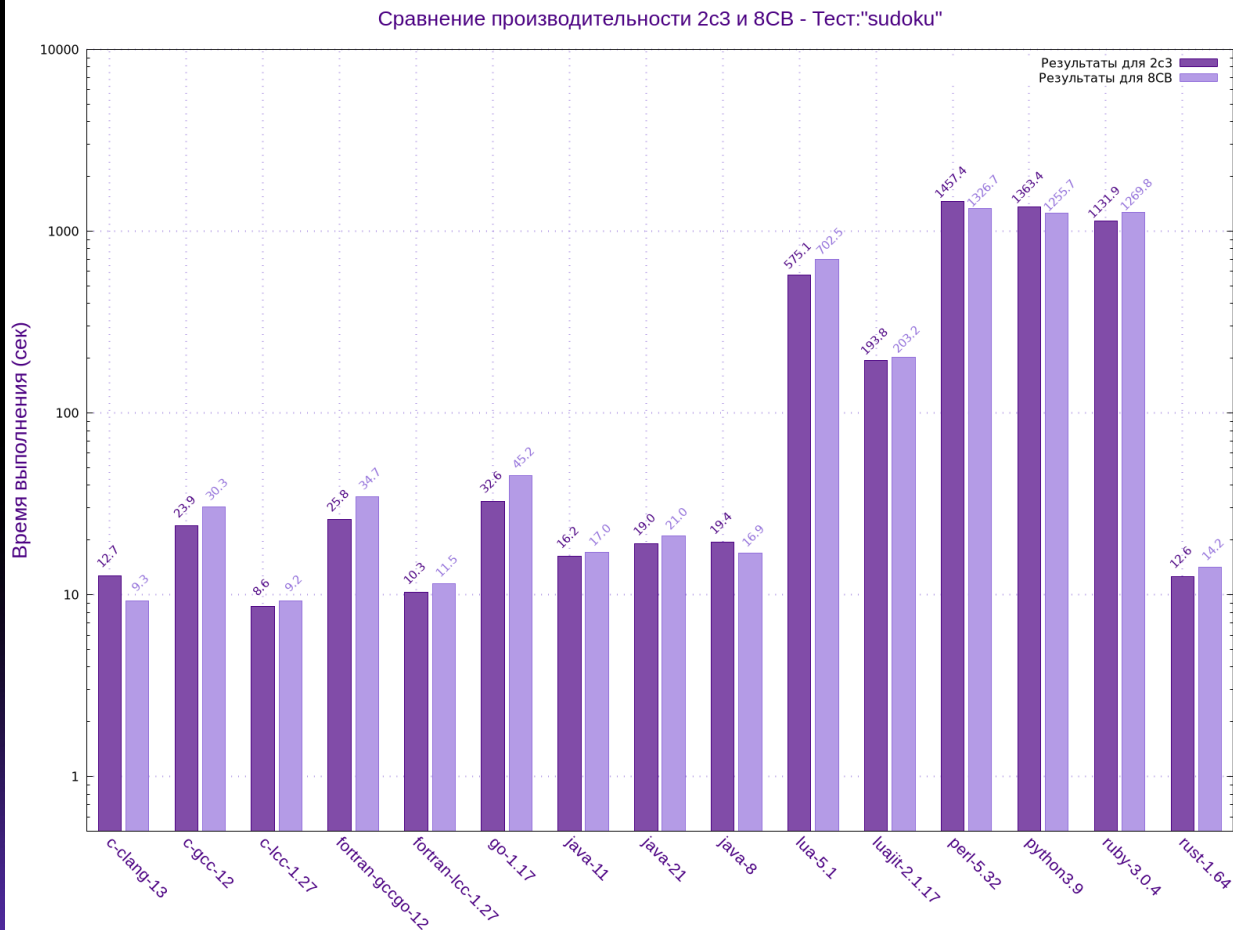
Меньше лучше



# SUDOKU

## Топ 5:

LCC  
Fortran  
Rust  
Clang  
Java

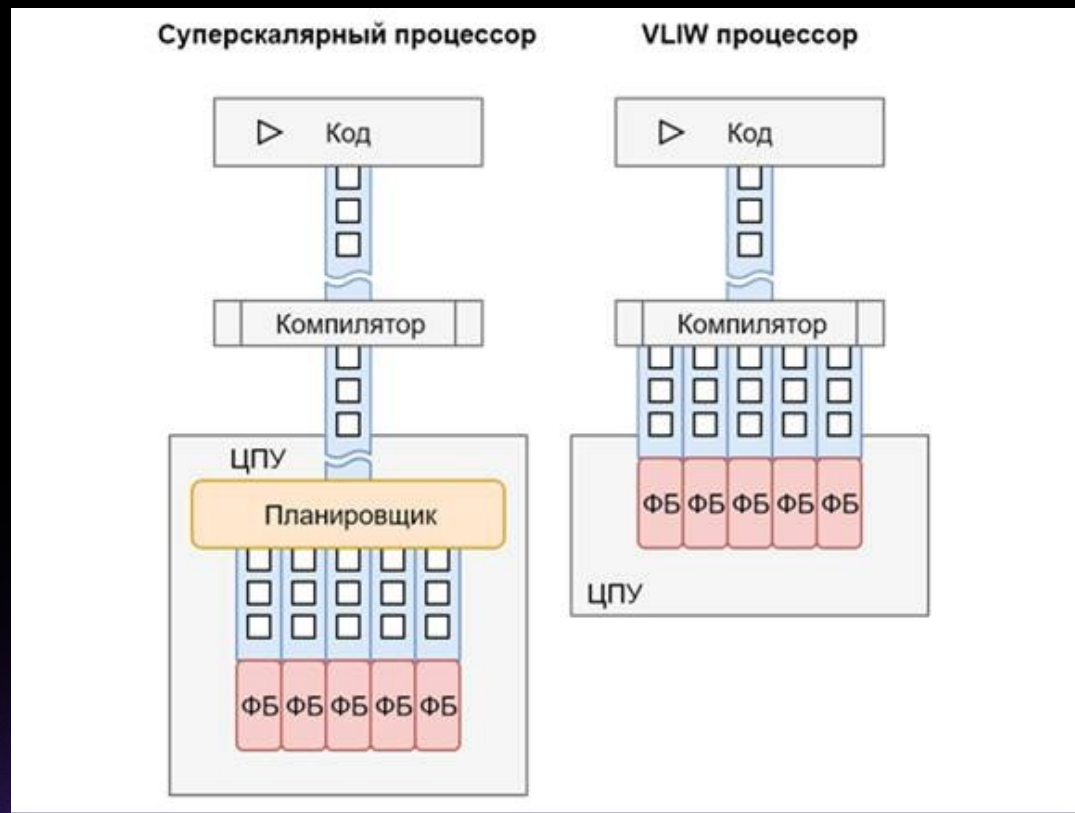


Меньше лучше, шкала экспоненциальная

# ПОЧЕМУ ТАК?

VLIW архитектура критична к заполнению широкой команды. Если у компилятора есть время на анализ и заполнение широкой команды, то у интерпретаторов нет.

Частично эту проблему решает JIT компилятор (Java, V8).



# ССЫЛКИ

1. Статья на Habr
2. Исходный код тестов с нашими доработками
3. Пазл 8и королев (тест nqueen)
4. Судоку (тест судоку)
5. Поиск перекрытий (тест bedcov)
6. Результаты тестов

# КОНТАКТЫ

[s.emets@nicct.ru](mailto:s.emets@nicct.ru)

<https://t.me/azhale>

<https://blog.emets.su>

<https://nicct.ru>