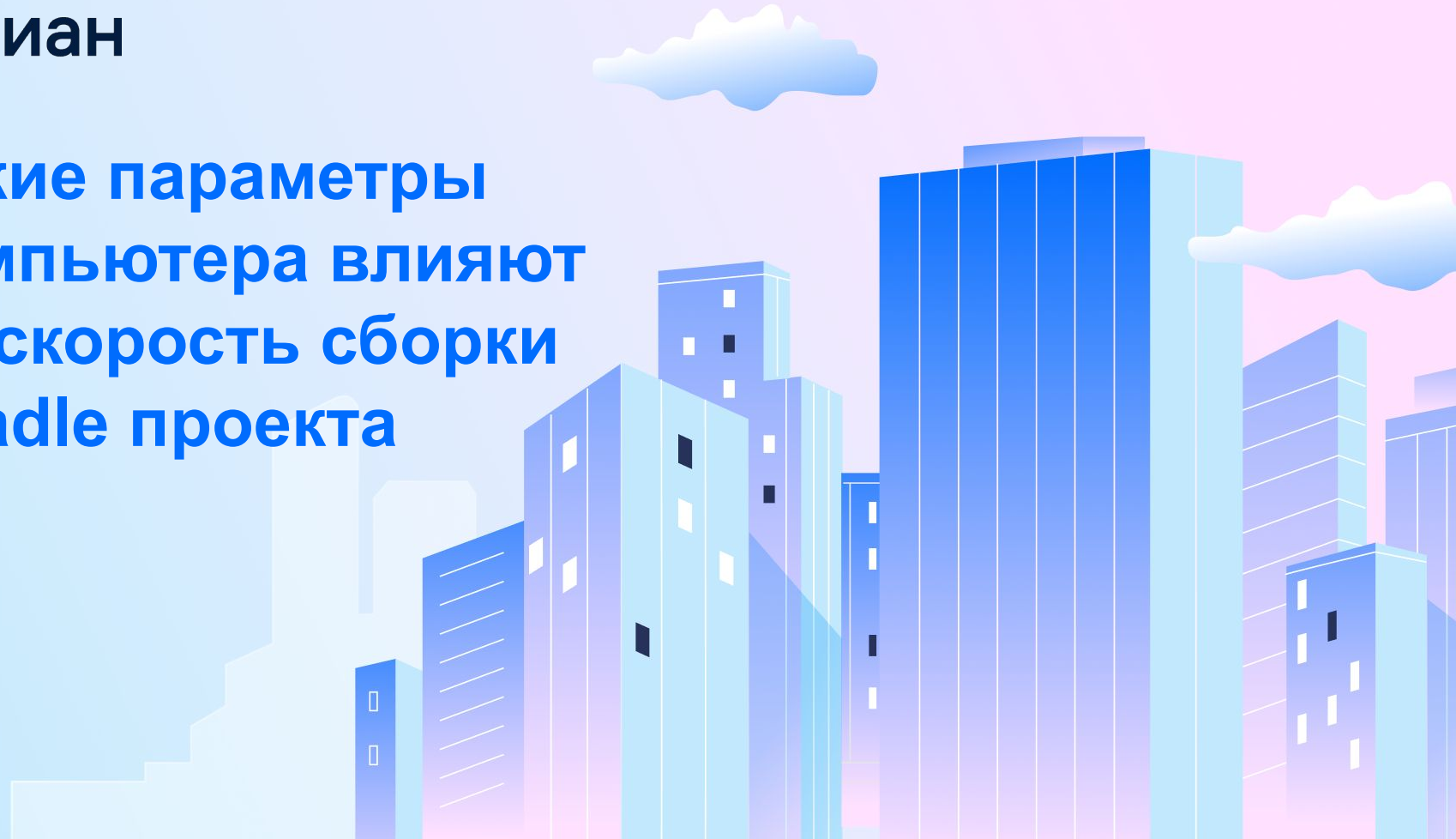
циан

Какие параметры компьютера влияют на скорость сборки Gradle проекта



Давайте знакомиться

Данил Перевалов
Android Developer

циан

Основная проблема

Какое железо брать?

циан

Цели

1. Понять, какие параметры и как влияют на скорость сборки
2. Понять, что лучше купить
3. Понять, как сэкономить
4. Увидеть разницу между одно- и много-модульной сборкой

Тесты


ДЛЯ БЫСТРОГО И ТОЧНОГО ОПРЕДЕЛЕНИЯ
EVITEST
PLUS
ДВОЙНОЙ ТЕСТ ДЛЯ ПОДТВЕРЖДЕНИЯ РЕЗУЛЬТАТОВ,
ТЕСТ-ПОЛОСКА
Надежный: точность 99%*
Быстрый: результат через 1 минуту**
SANAVITA
SELF CARE
2 ТЕСТА
В УПАКОВКЕ

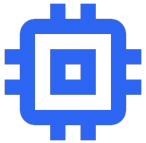
Методология

циан

Компьютер

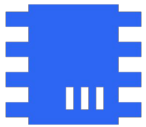
циан

Компьютер



CPU

Ryzen 5 3600
6 ядер 12 потоков



RAM

Crucial 2 x 8 Гб
Crucial 2 x 16 Гб

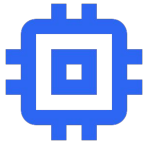
ROM

HDD 2.5”
HDD 3.5”
Sata SSD
NVMe SSD

Windows 11

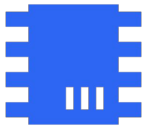
циан

По умолчанию



CPU

Режим Auto с поднятием частот вплоть до 4200 МГц



RAM

2 x 16 Гб

ROM

NVMe SSD

циан

Проект

циан

циан

PropTech

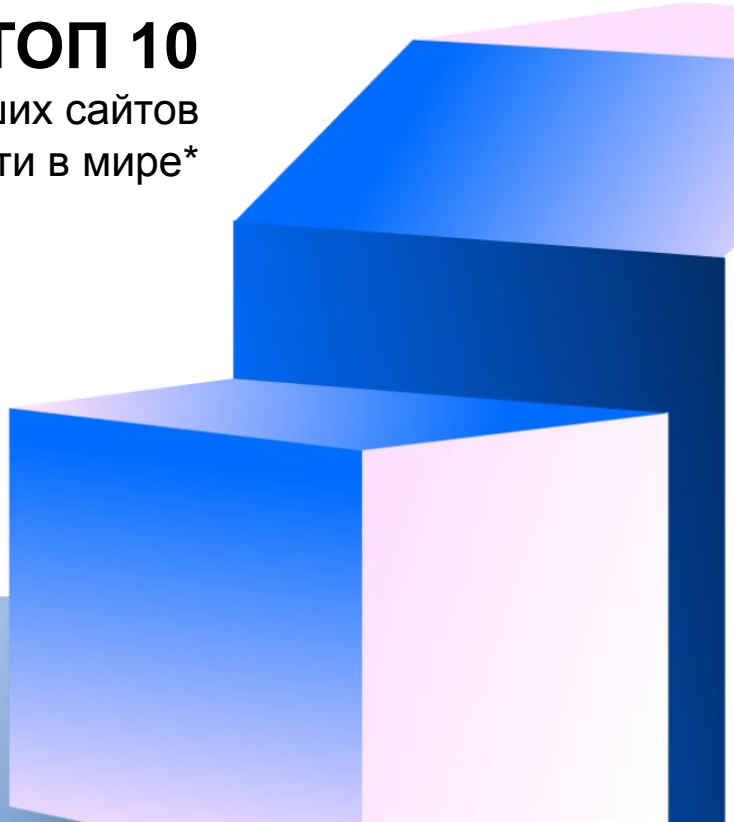
900+
сотрудников

18,2 млн

уникальных пользователей ежемесячно

ТОП 10
крупнейших сайтов по недвижимости в мире*

12
продуктовых направлений



Циан – высокотехнологичный сервис по покупке, продаже и аренде коммерческой и жилой недвижимости с набором качественных сопутствующих услуг, таких как Циан.Ипотека и Циан.Сделка.

*По количеству пользователей в рейтинге SimilarGroup на 1 сентября 2021 г.

Приложение

20

Android-разработчиков

1.7+ млн.

MAO

270+ тыс.

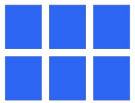
DAO

3.4+ млн

Активных устройств

циан

Проект



Gradle

410 модулей

Java

110 тыс. строк

Kotlin

440 тыс. строк

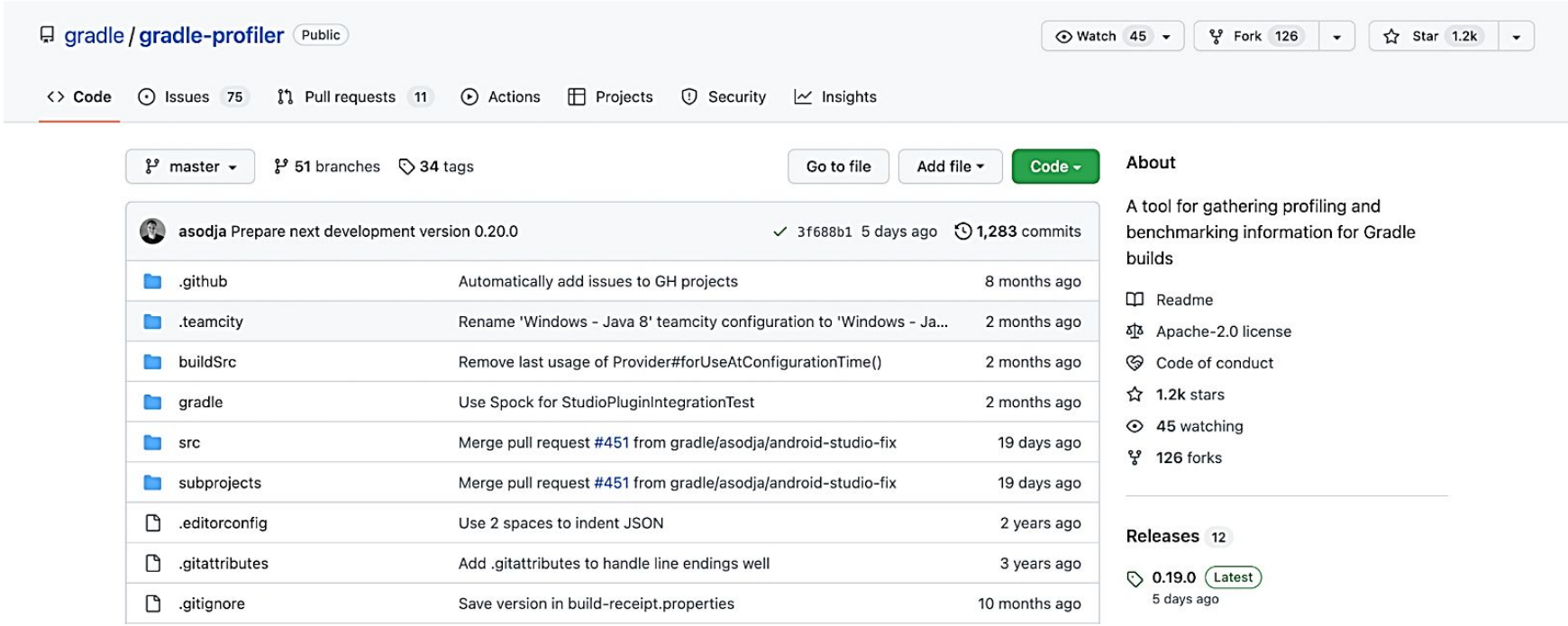
Xml

170 тыс. строк

циан

Инструмент измерения

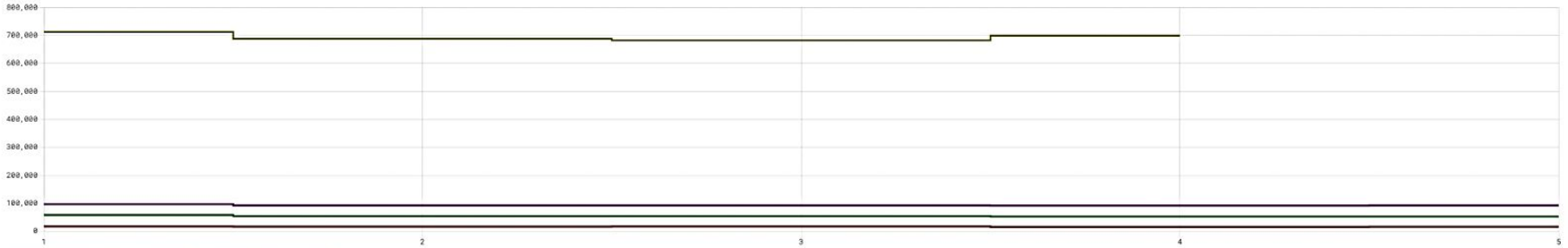
Инструмент измерения



циан

Отчёт

Benchmark results — Gradle Profiler



Historical | Sorted | Select all | Clear all | Show all | Show selected only

Scenario	Baseline	Sample	Mean	Min	P25	Median	P75	Max	Std.dev	Iterations					
+ Clean	☐	☑ total execution time	16 559,03 ms	15 828,93 ms	16 265,44 ms	16 545,30 ms	17 030,44 ms	17 125,05 ms	482,09 ms	18 694,14 ms	17 125,05 ms	16 545,30 ms	17 030,44 ms	15 828,93 ms	16 265,44 ms
		garbage collection time	1 180,80 ms	920,00 ms	1 028,00 ms	1 144,00 ms	1 189,00 ms	1 623,00 ms	240,14 ms	1 257,00 ms	1 144,00 ms	1 189,00 ms	1 623,00 ms	1 028,00 ms	920,00 ms
		task start	16 317,00 ms	15 608,00 ms	16 038,00 ms	16 279,00 ms	16 782,00 ms	16 878,00 ms	471,79 ms	18 428,00 ms	16 878,00 ms	16 279,00 ms	16 782,00 ms	15 608,00 ms	16 038,00 ms
+ Clean AssembleDebug without cache	☐	☑ total execution time	696 318,60 ms	682 818,60 ms	687 466,96 ms	694 555,54 ms	703 407,18 ms	713 344,74 ms	11 615,94 ms	671 610,47 ms	713 344,74 ms	689 016,42 ms	682 818,60 ms	700 094,66 ms	
		garbage collection time	106 840,75 ms	101 637,00 ms	103 111,50 ms	107 102,00 ms	110 831,25 ms	111 522,00 ms	4 289,98 ms	106 784,00 ms	101 637,00 ms	110 601,00 ms	103 603,00 ms	111 522,00 ms	
		task start	70 446,50 ms	64 153,00 ms	68 138,50 ms	71 638,00 ms	73 946,00 ms	74 357,00 ms	4 097,74 ms	70 118,00 ms	64 153,00 ms	69 467,00 ms	73 809,00 ms	74 357,00 ms	
+ AssembleDebug with change kt file	☐	☑ total execution time	54 467,92 ms	52 582,67 ms	52 717,66 ms	54 048,36 ms	54 291,36 ms	58 699,54 ms	2 224,02 ms	310 909,17 ms	58 699,54 ms	54 291,36 ms	54 048,36 ms	52 582,67 ms	52 717,66 ms
		garbage collection time	6 908,60 ms	5 937,00 ms	6 360,00 ms	7 016,00 ms	7 062,00 ms	8 168,00 ms	757,12 ms	41 085,00 ms	8 168,00 ms	5 937,00 ms	7 062,00 ms	6 360,00 ms	7 016,00 ms
		task start	27 328,40 ms	25 643,00 ms	26 474,00 ms	26 922,00 ms	27 791,00 ms	29 812,00 ms	1 422,53 ms	55 652,00 ms	27 791,00 ms	29 812,00 ms	26 922,00 ms	26 474,00 ms	25 643,00 ms
+ Clean mono	☐	total execution time	16 666,01 ms	15 798,36 ms	16 016,91 ms	16 578,77 ms	17 327,90 ms	17 608,10 ms	708,15 ms	19 264,03 ms	17 608,10 ms	17 327,90 ms	16 578,77 ms	15 798,36 ms	16 016,91 ms
		garbage collection time	1 158,80 ms	761,00 ms	911,00 ms	1 040,00 ms	1 230,00 ms	1 852,00 ms	379,30 ms	981,00 ms	911,00 ms	1 852,00 ms	1 230,00 ms	761,00 ms	1 040,00 ms
		task start	16 422,20 ms	15 561,00 ms	15 791,00 ms	16 351,00 ms	17 058,00 ms	17 350,00 ms	694,29 ms	18 931,00 ms	17 350,00 ms	17 058,00 ms	16 351,00 ms	15 561,00 ms	15 791,00 ms
+ Clean AssembleDebug without cache mono	☐	total execution time	2 296 108,84 ms	2 289 265,95 ms	2 291 110,01 ms	2 294 603,88 ms	2 297 228,94 ms	2 308 335,41 ms	6 705,55 ms	2 313 393,49 ms	2 294 603,88 ms	2 297 228,94 ms	2 308 335,41 ms	2 289 265,95 ms	2 291 110,01 ms

циан

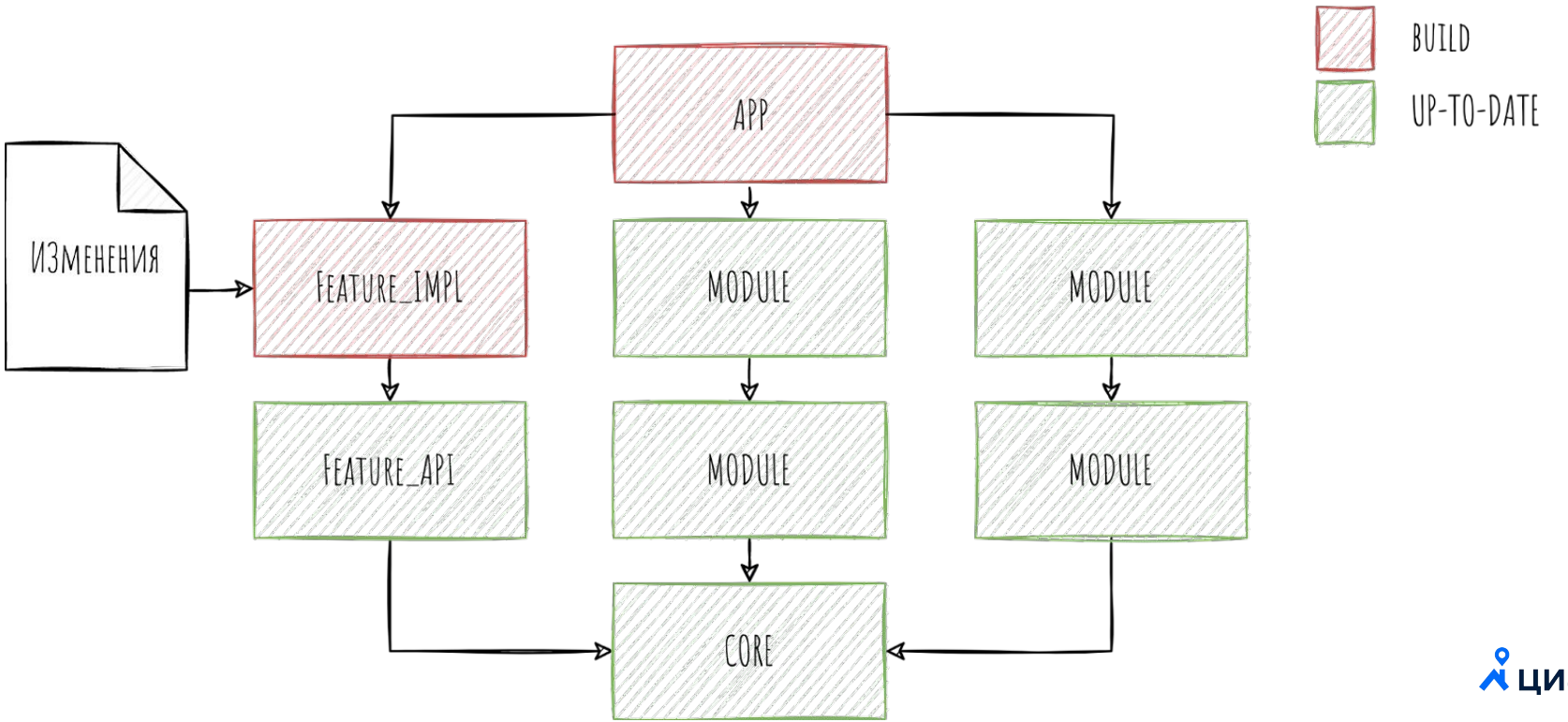
Сценарии

циан

Сценарии

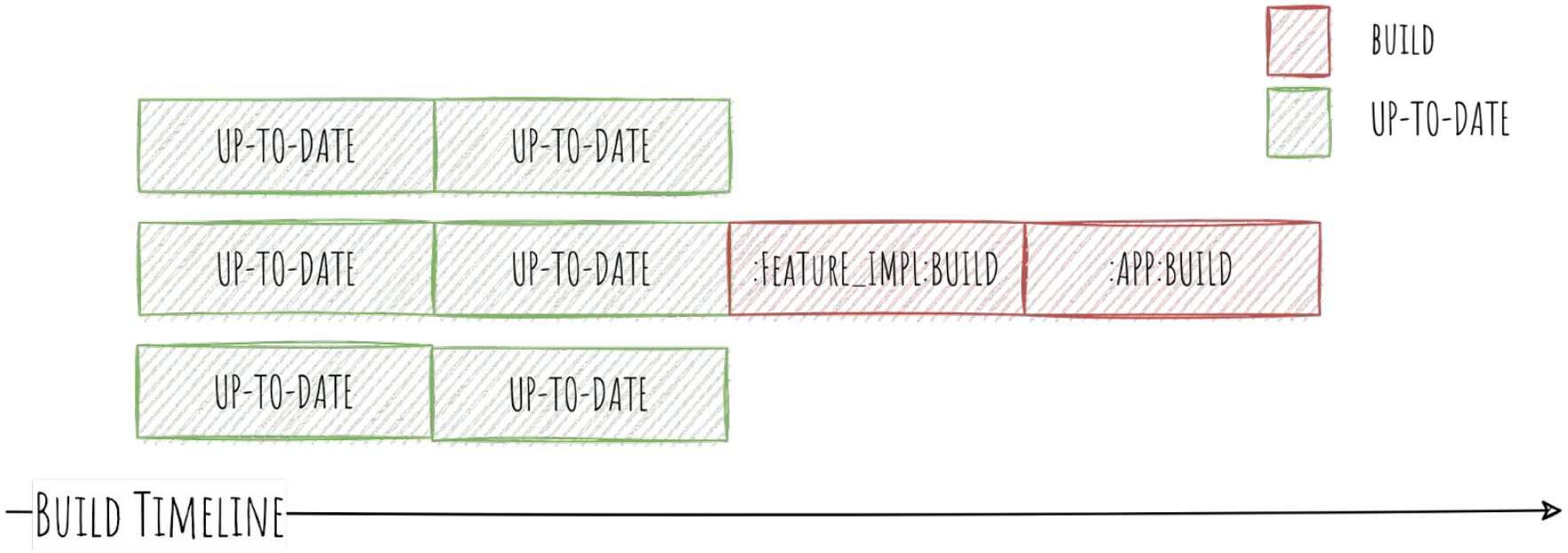
1. Многомодульная горячая
2. Многомодульная холодная
3. Одномодульная холодная

Многомодульная горячая. Структура



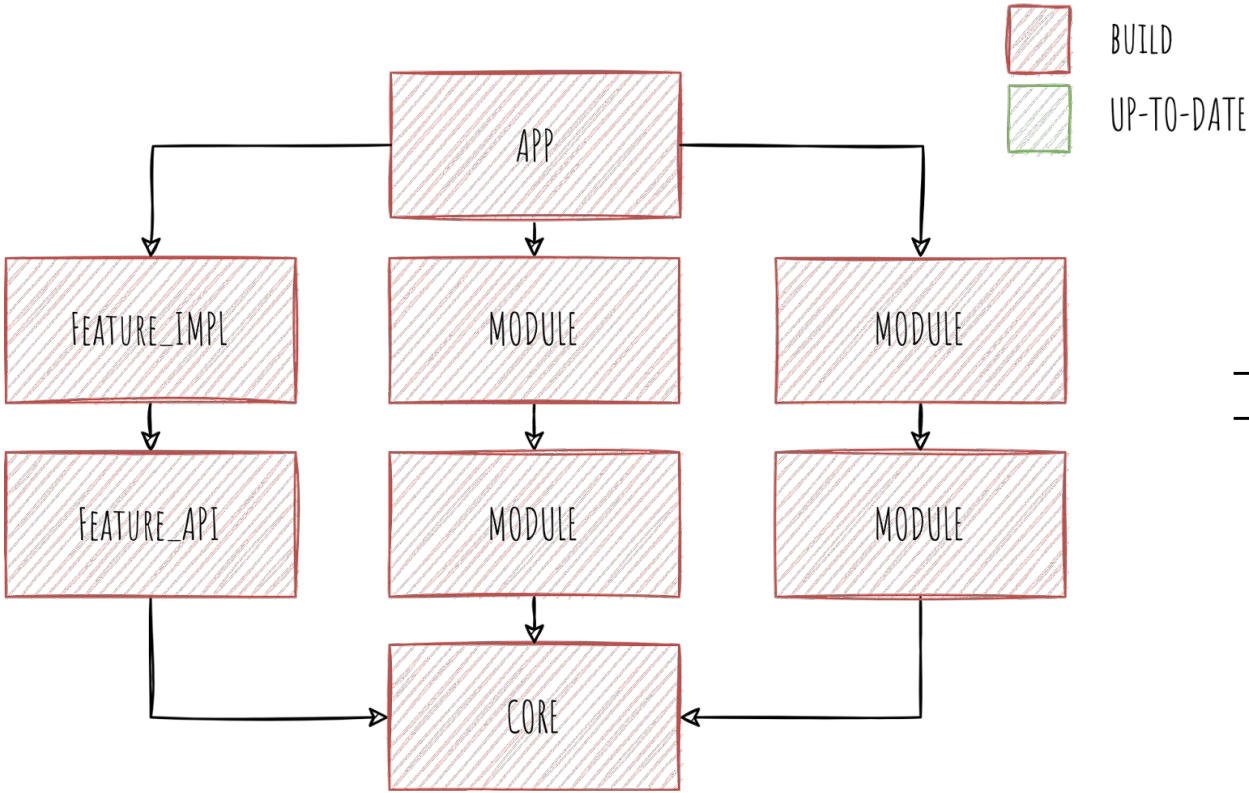
циан

Многомодульная горячая. Timeline



циан

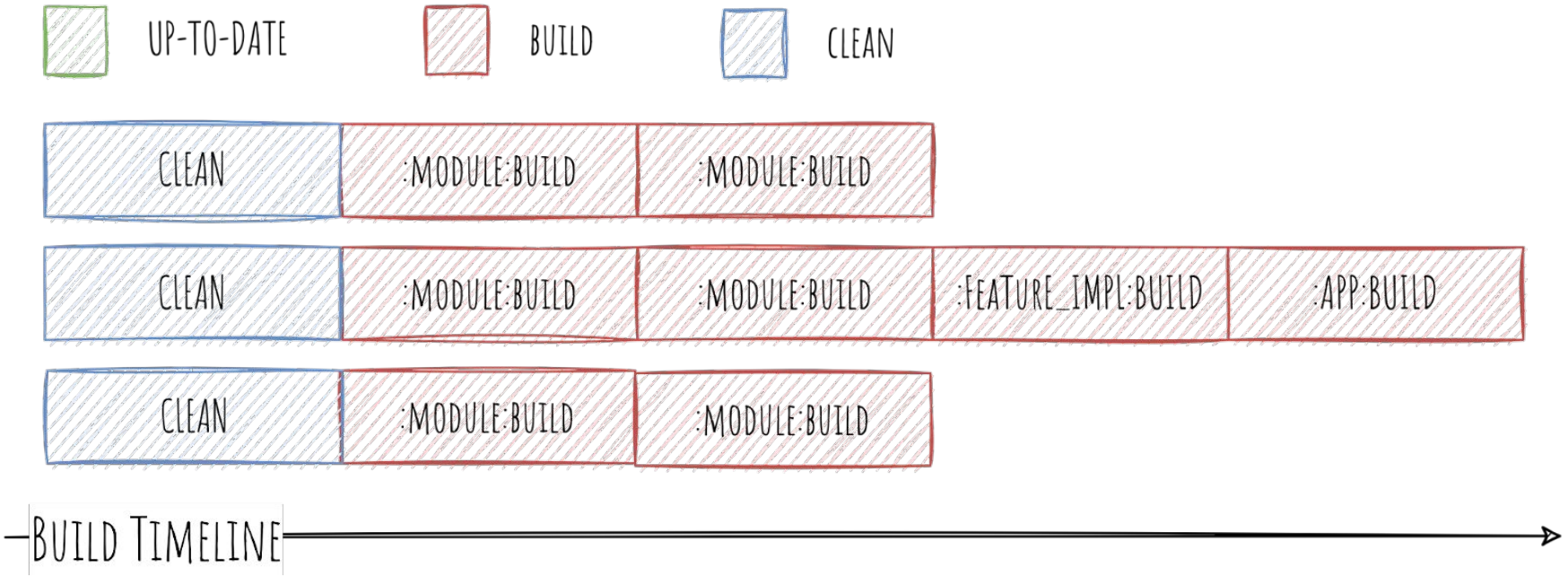
Многомодульная холодная. Структура



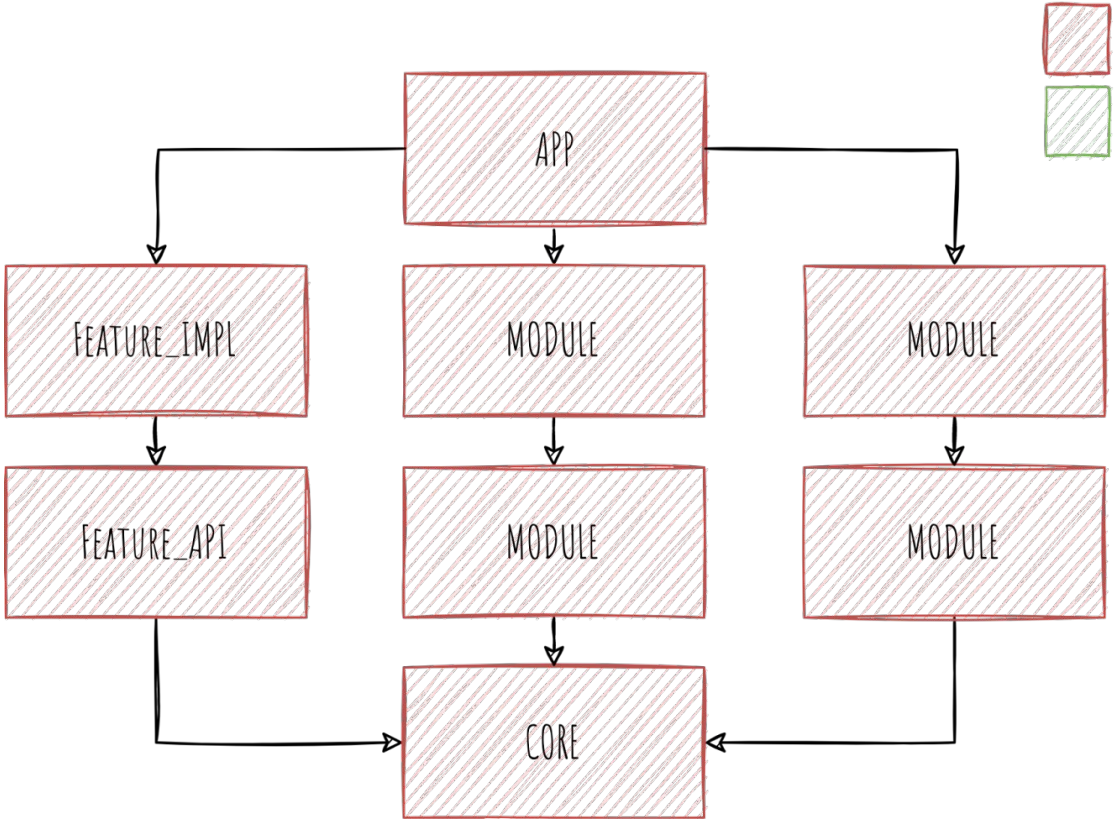
— clean
— no-build-cache

циан

Многомодульная холодная. Timeline



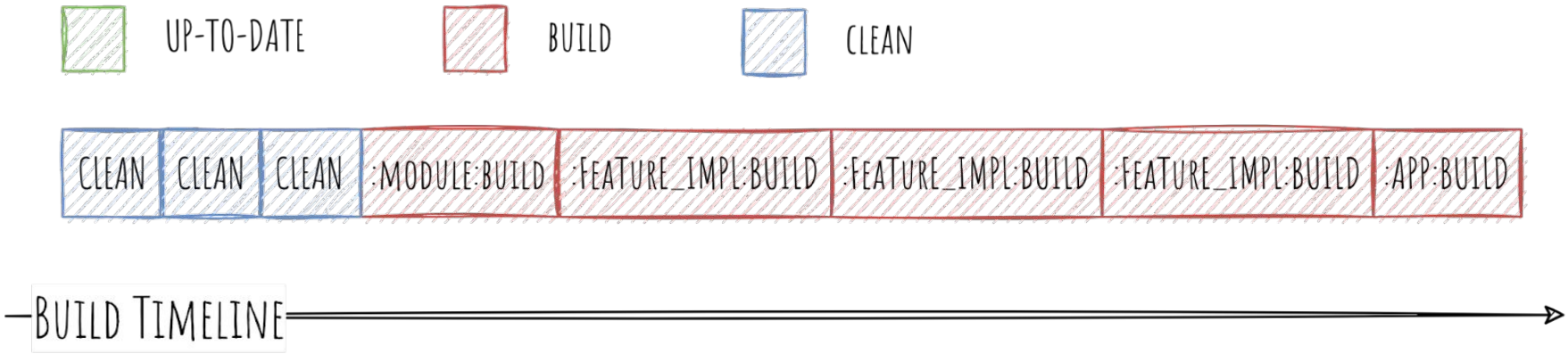
Одномодульная холодная. Структура



— clean
— no-build-cache
— max-workers 1
— no-parallel

циан

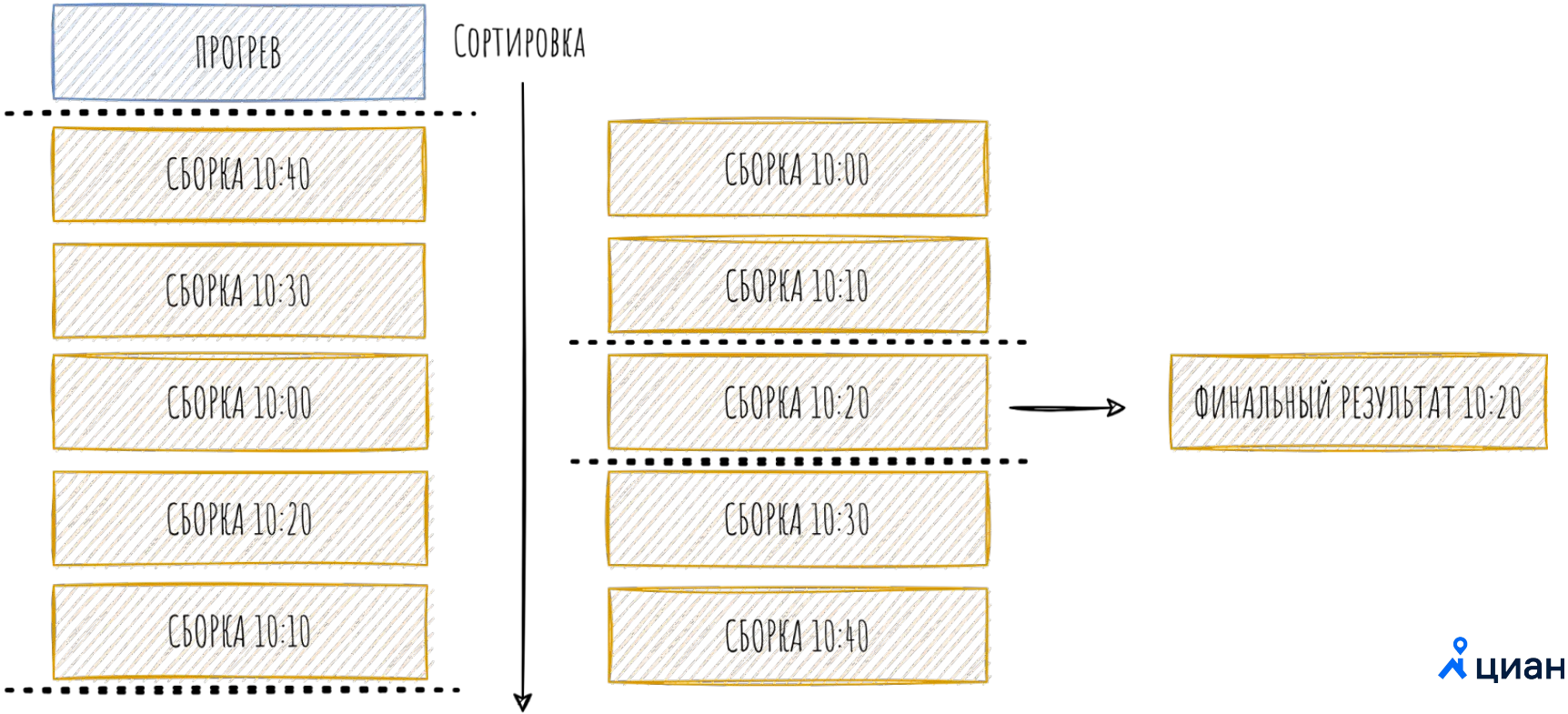
Одномодульная холодная. Timeline



циан

Прогоны и погрешность

Количество прогонов



циан

Погрешность

2%

примерная разница между последовательными прогонами

циан

Gradle Properties

Никаких внешних кешей

Build Cache	Configuration Cache	Gradle Daemon
Выключаем	Выключаем	Выключаем

Сильно влияют на погрешность времени сборки

Постоянные настройки

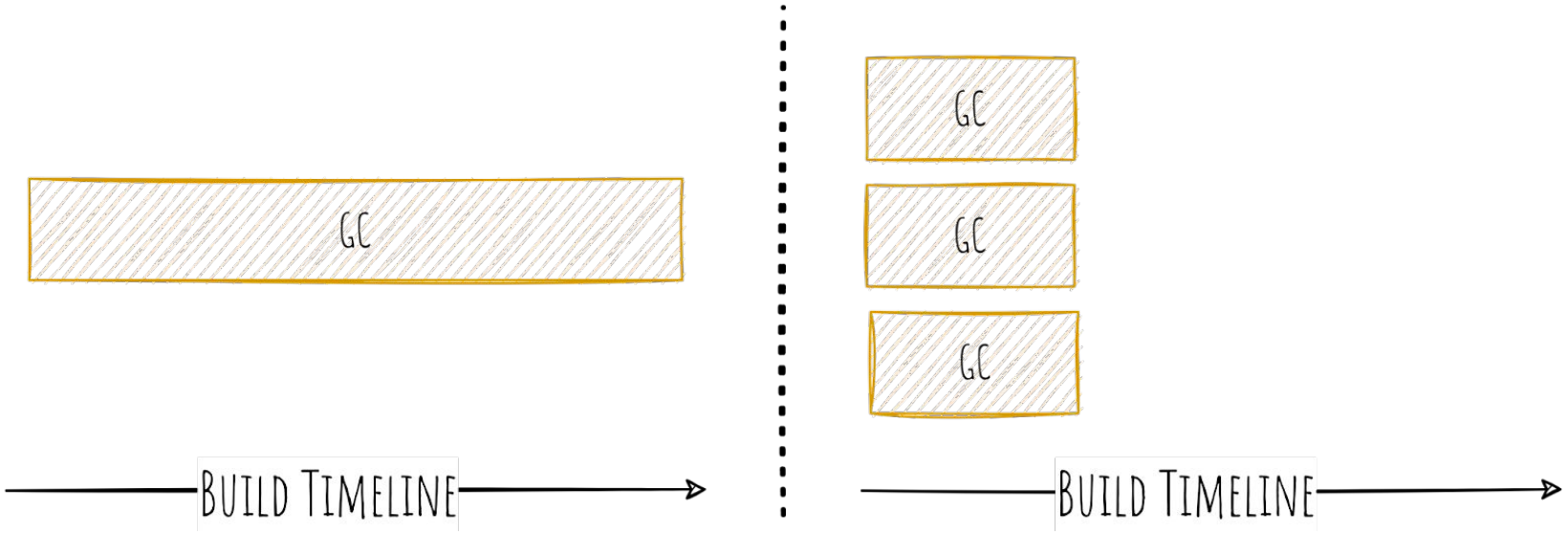
```
org.gradle.daemon=false
org.gradle.caching=false
org.gradle.configureondemand=false

kotlin.daemon.jvm.options=-Xmx4g
kotlin.incremental=true
kapt.use.worker.api=true
kapt.incremental.apt=true
kapt.include.compile.classpath=false
```

Меняющиеся настройки

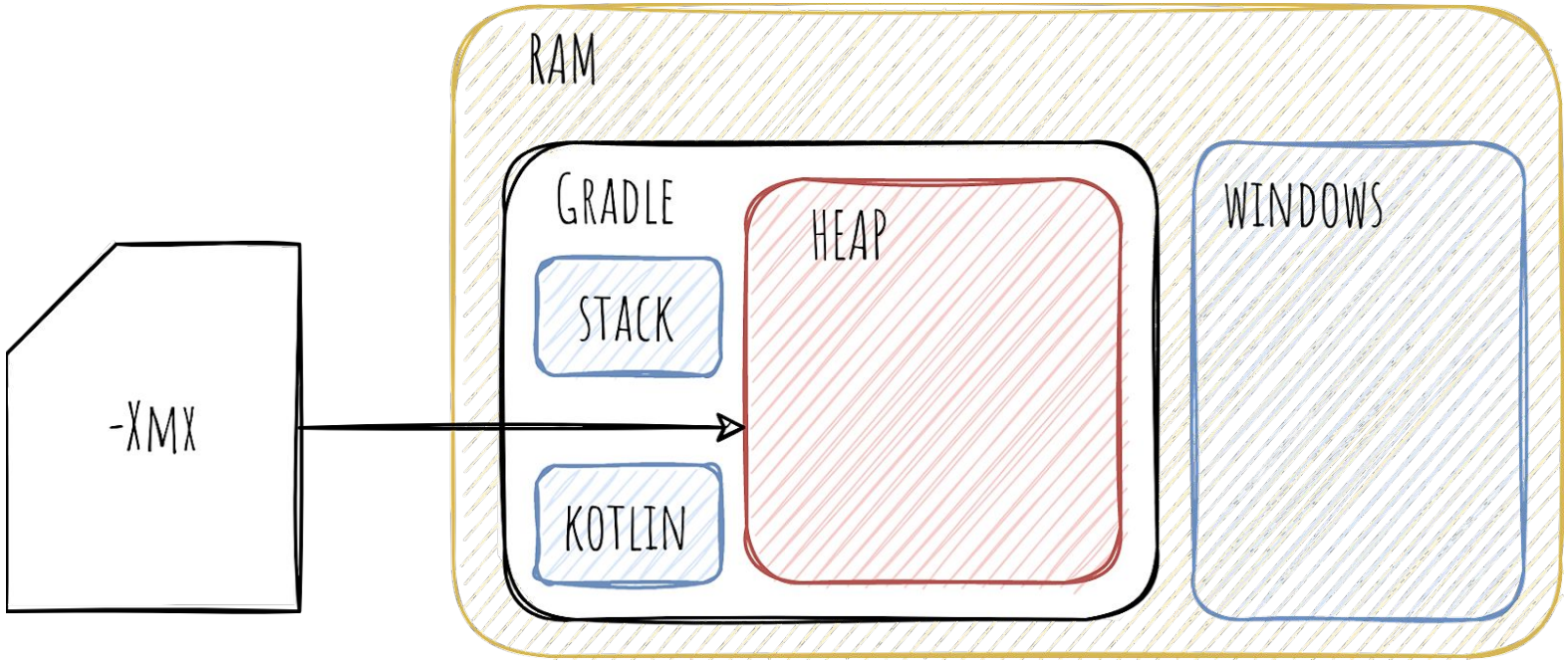

```
org.gradle.jvmargs=-Xmx24g -XX:+UseParallelGC
org.gradle.workers.max=12
```

циан

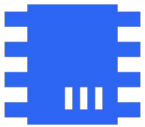
Gradle Properties. -XX:+UseParallelGC



Gradle Properties. -Xmx

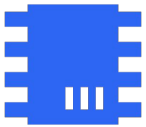


Gradle Properties. -Xmx. Варианты



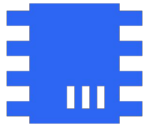
4g

Не собирается



5g

Собирается, но долго

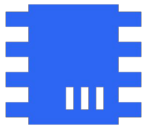


6g

Собирается, норм

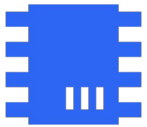
циан

Gradle Properties. -Xmx. Формула



-xmx=8g

Для 16 Гб



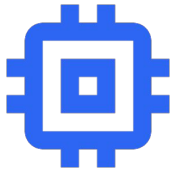
-xmx=24g

Для 32 Гб

Формула: -xmx=<количество RAM> - 8 Гб

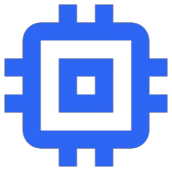
циан

Gradle Properties. org.gradle.workers.max. Формула



8

Для 8 потоков



12

Для 12 потоков

Формула: workers.max=<количество потоков>

циан

Что измеряем?

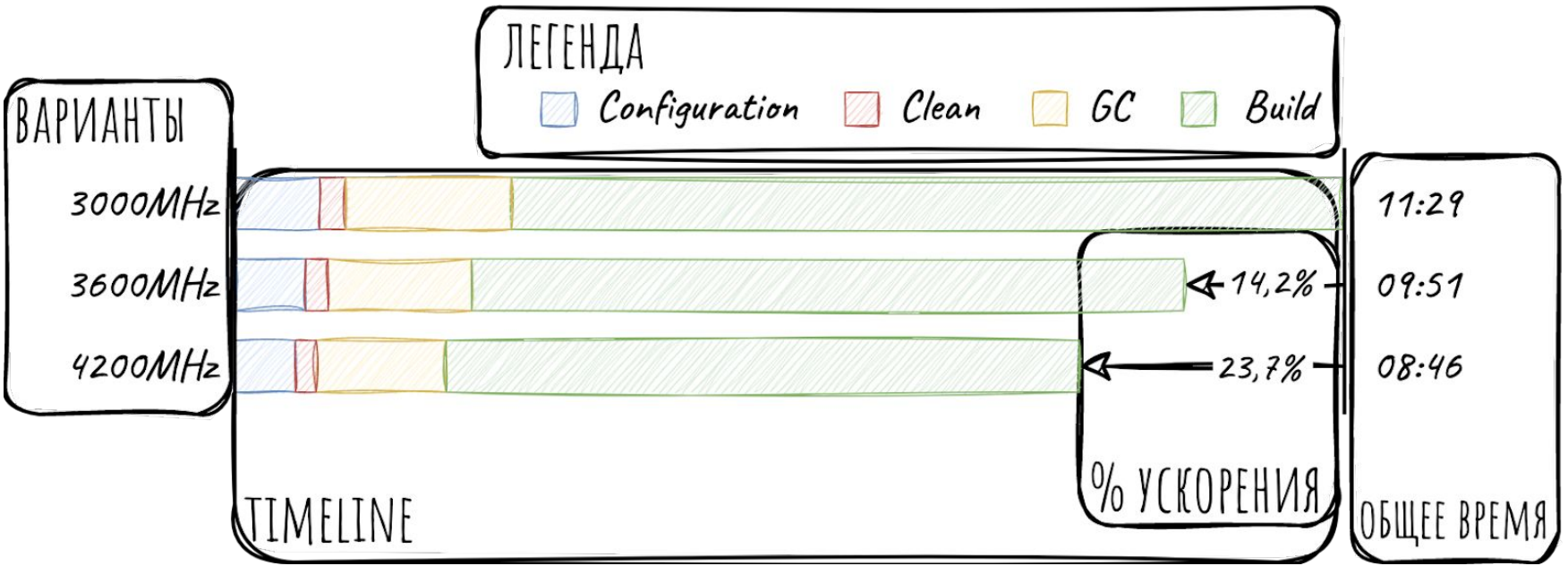
Параметры компьютера

1. Частота ядер CPU
2. Количество ядер/потоков CPU
3. Количество оперативной памяти
4. Частота оперативной памяти
5. Скорость чтения/записи ROM

Составляющие сборки

- Время конфигурации
- Время clean, для холодных сборок
- Время GC
- Время самой сборки

Пример теста



циан

Наконец-то тесты

циан

Частота ядер

циан

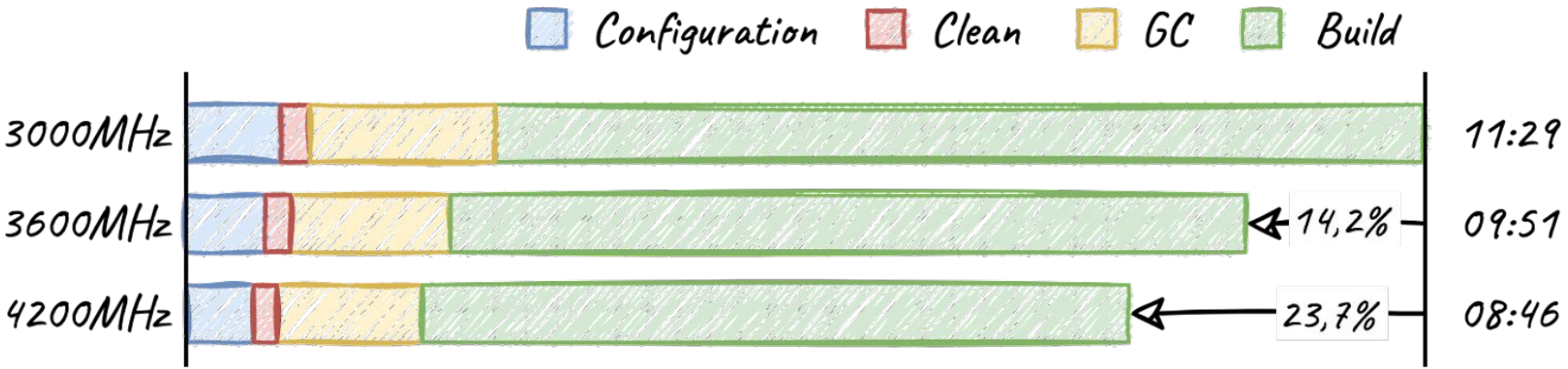
Частота ядер

3000 MHz	3600 MHz	4200 MHz
Минимум	+20%	+40%

Выключаем Turbo Boost

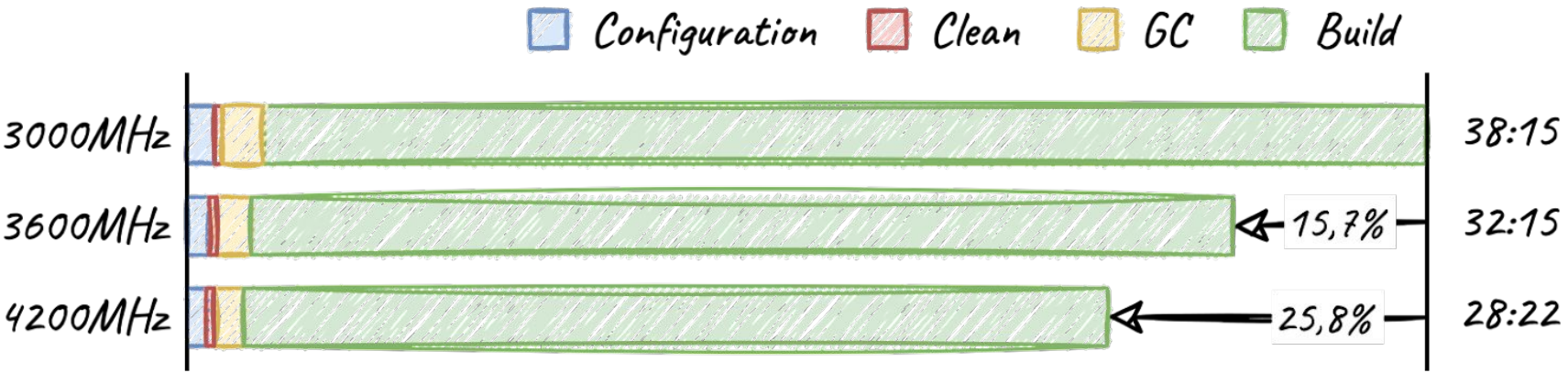
циан

Многомодульная холодная сборка



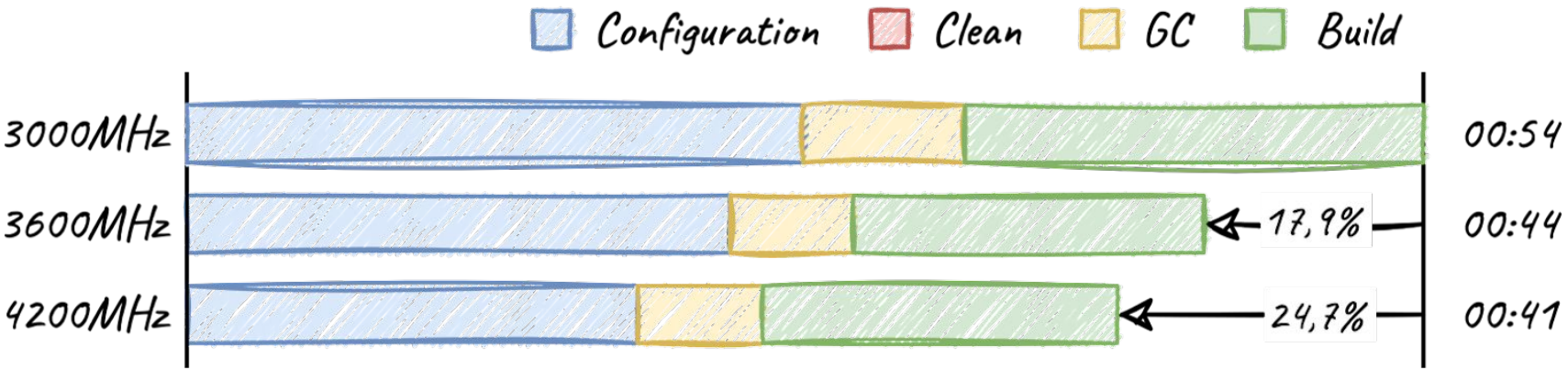
циан

Одномодульная холодная сборка



циан

Многомодульная горячая сборка



циан

Выводы

1. Частота — ключевой фактор
2. Разница между 3Ггц и 4.2 Ггц — 25%,
   при приросте частоты на 40%
3. Холодная многомодульная сборка быстрее не в 12 раз,
   а в 3 раза
4. Горячая сборка очень любит Configuration Cache

Количество ядер

Количество ядер

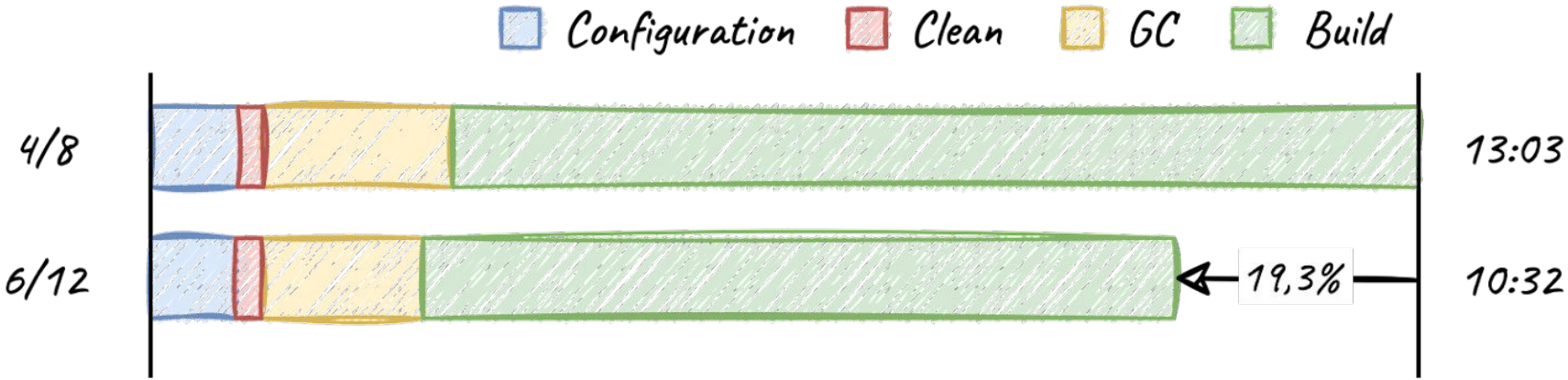
4 ядра 8 потоков	6 ядер 12 потоков
Минимум	+50%

Включаем Turbo Boost

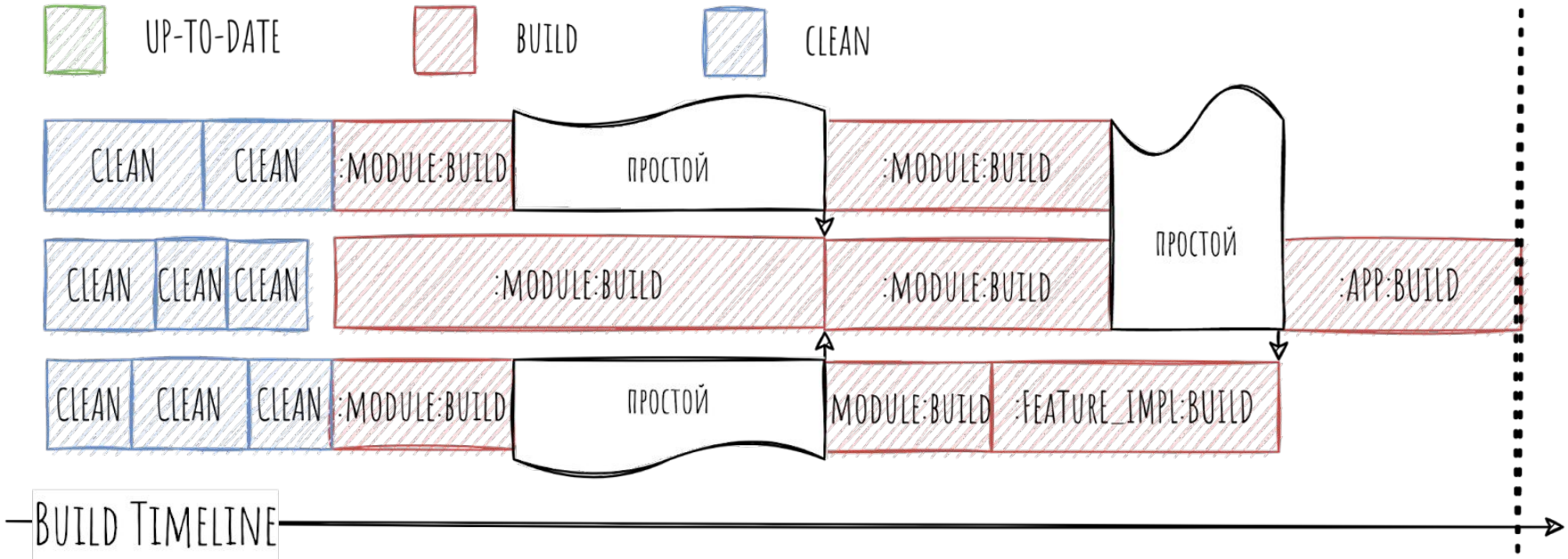
Частоту на Auto

циан

Многомодульная холодная сборка

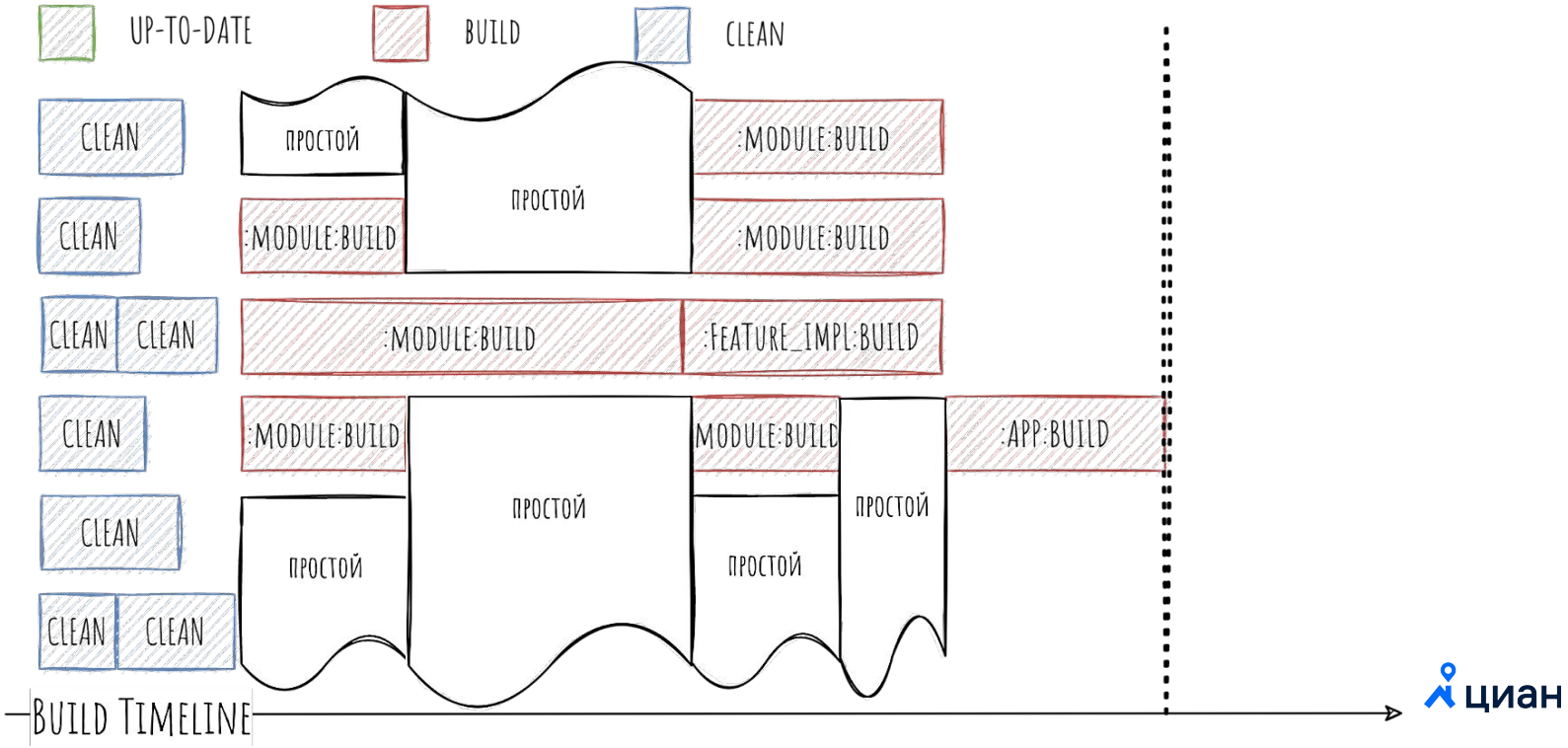


Продуктовые монолиты. 3 Gradle Worker



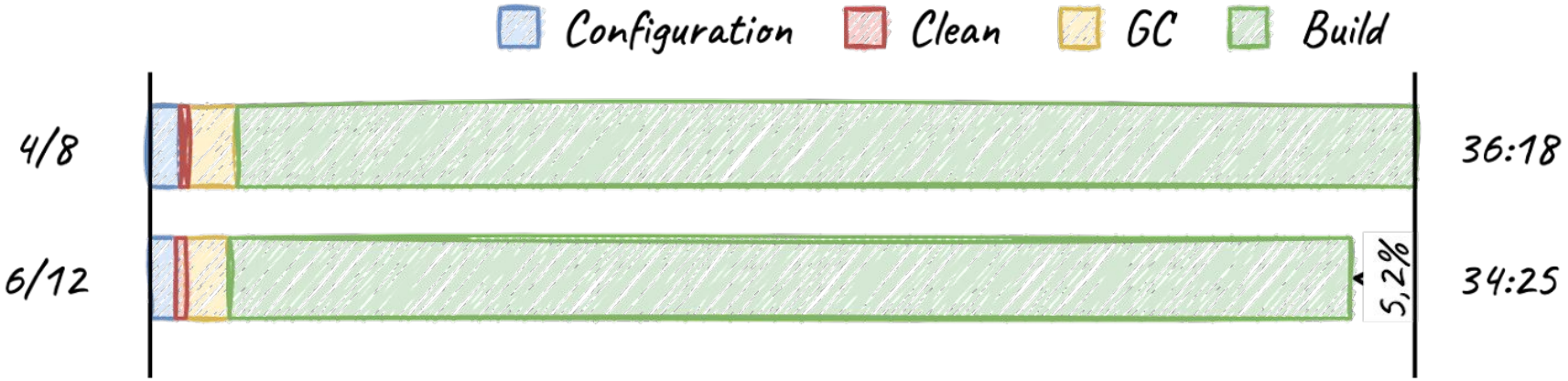
циан

Продуктовые монолиты. 6 Gradle Worker

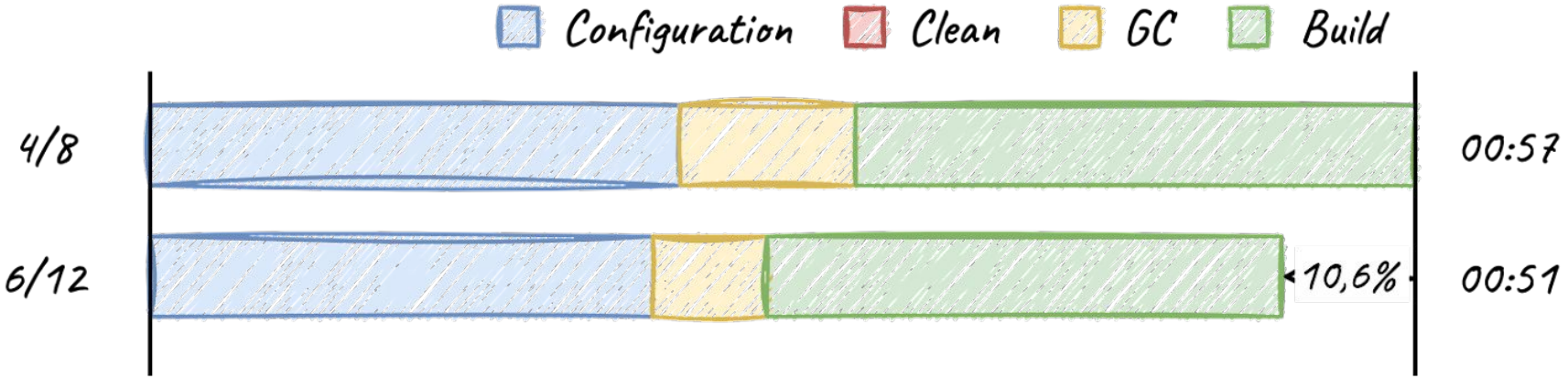


циан

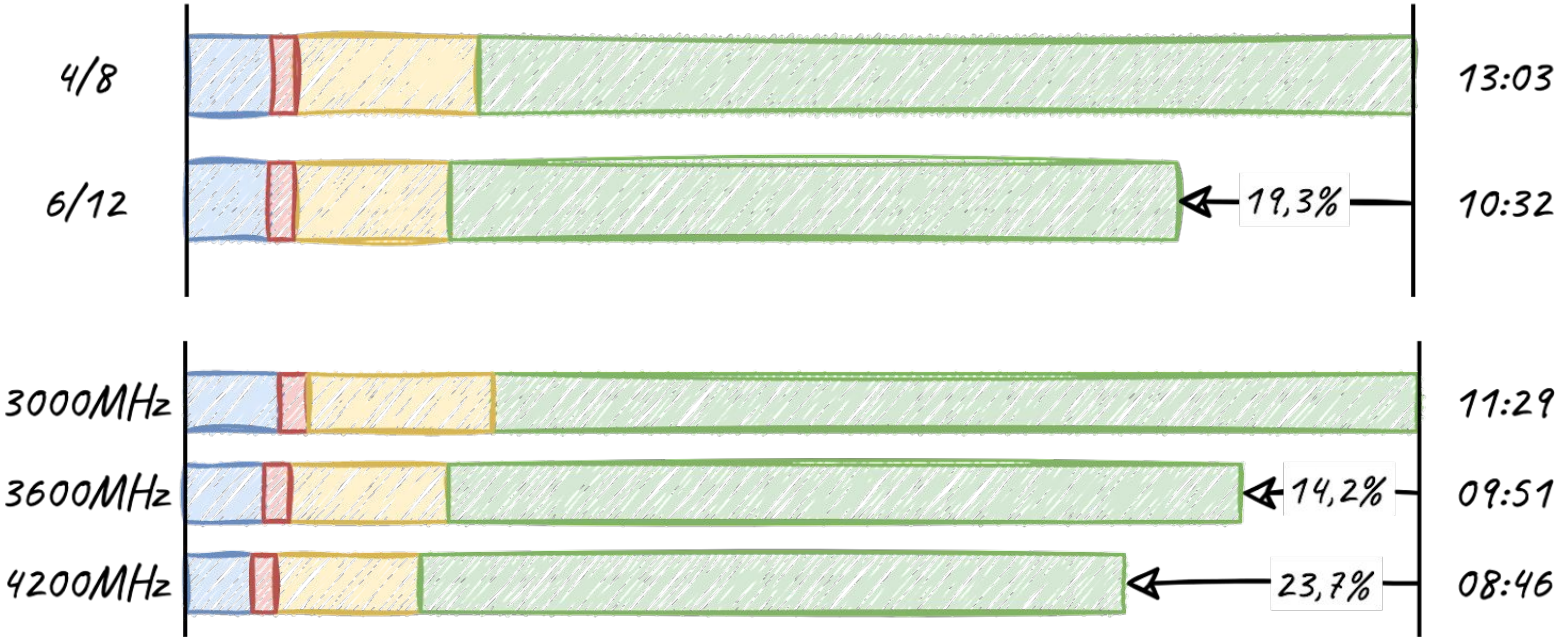
Одномодульная холодная сборка



Многомодульная горячая сборка



Непонятный факт



циан

Непонятный факт. Частоты при сборке

```
  Run: run default tasks
  Cleanup: run tasks clean
  Gradle args: [--no-parallel, --max-workers, 1]
  Build changes: []
  Warm-ups: 1
  Builds: 5
Scenario: Clean AssembleDebug without cache mono using Gradle 7.3.3
  Gradle 7.3.3 (C:\Users\tzars\Downloads\Cian_Android-master\cian_android\gradle-user-home\wrapper\dists\gradle-7.3.3-all\4295vidhdd9hd3gb
jyw1xqxpo\gradle-7.3.3)
  Run using: `gradle` command with --no-daemon
  Run: run tasks assembleDebug
  Cleanup: run tasks clean
  Gradle args: [--no-build-cache, --no-parallel, --max-workers, 1]
  Build changes: []
  Warm-ups: 1
  Builds: 5

* Running scenario Clean using Gradle 7.3.3 (scenario 1/5)

* Stopping daemons

* Running cleanup for warm-up build #1

* Running warm-up build #1
Execution time 16538 ms

* Running cleanup for measured build #1

* Running measured build #1
Execution time 14627 ms

* Running cleanup for measured build #2

* Running measured build #2
Execution time 13648 ms

* Running cleanup for measured build #3

* Running measured build #3
Execution time 13441 ms

* Running cleanup for measured build #4

* Running measured build #4
Execution time 12961 ms

* Running cleanup for measured build #5

* Running measured build #5
Execution time 12926 ms
```

Датчик	актуальный
Система: GIGABYTE B550I AORUS PRO AX	
ЦП [#0]: AMD Ryzen 5 3600	
Core VIDs	1.365 В
Базовые частоты	4,052.6 МГц
Core 0 Частота (perf #5/6)	4,056.7 МГц
Core 1 Частота (perf #4/4)	4,056.7 МГц
Core 2 Частота (perf #3/2)	4,032.1 МГц
Core 3 Частота (perf #1/1)	4,056.7 МГц
Core 4 Частота (perf #2/5)	4,056.7 МГц
Core 5 Частота (perf #1/3)	4,056.7 МГц
частота шины	98.3 МГц
Основные эффективные тактовые частоты	3,122.5 МГц
Средняя эффективная частота	3,122.5 МГц
Основное использование	89.4 %
Core 0 T0 использование	70.7 %
Core 0 T1 использование	80.9 %
Core 1 T0 использование	86.3 %
Core 1 T1 использование	78.9 %
Core 2 T0 использование	89.1 %
Core 2 T1 использование	88.4 %
Core 3 T0 использование	97.9 %
Core 3 T1 использование	99.3 %

циан

Непонятный факт. Частоты в игре

циан

Выводы

1. Количество ядер сильно влияет только в многомодульной сборке
2. Чем лучше распараллелен проект, тем больший прирост
3. Слишком много ядер на локальной машине не нужно

Количество памяти

Количество памяти

16 Гб всего 8 Гб на кучу	32 Гб всего 24 Гб на кучу
Минимум	+100%

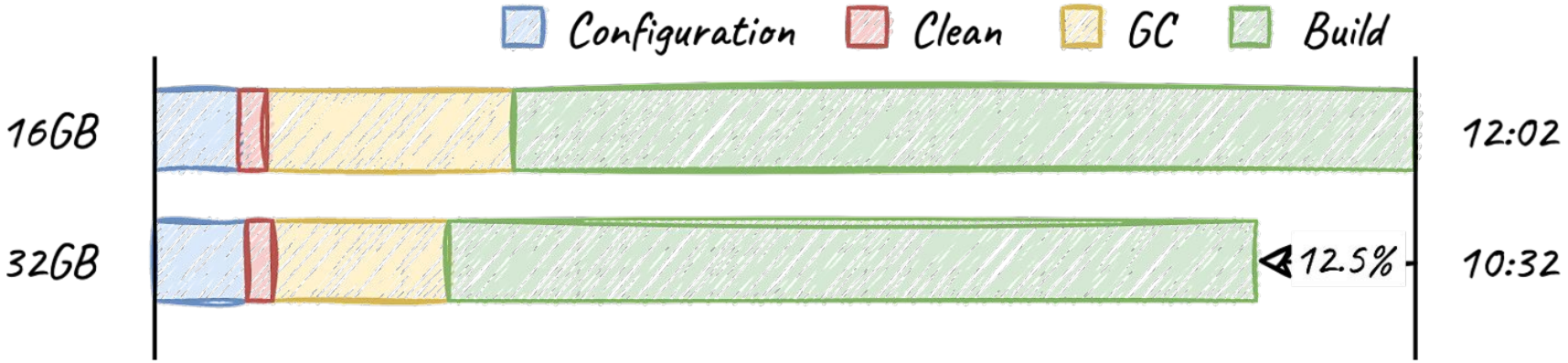
Одинаковые частоты

Одинаковые тайминги

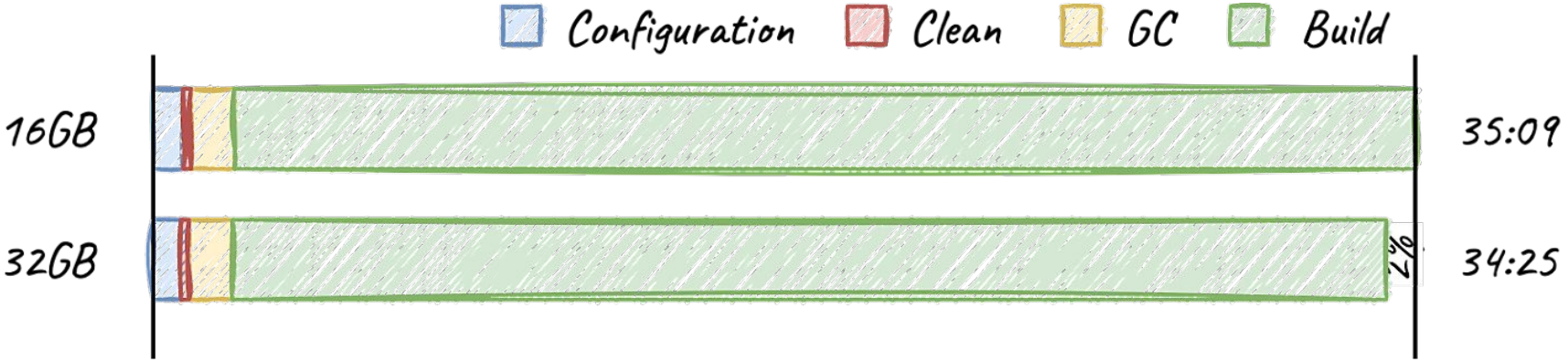
Одинаковое напряжение

циан

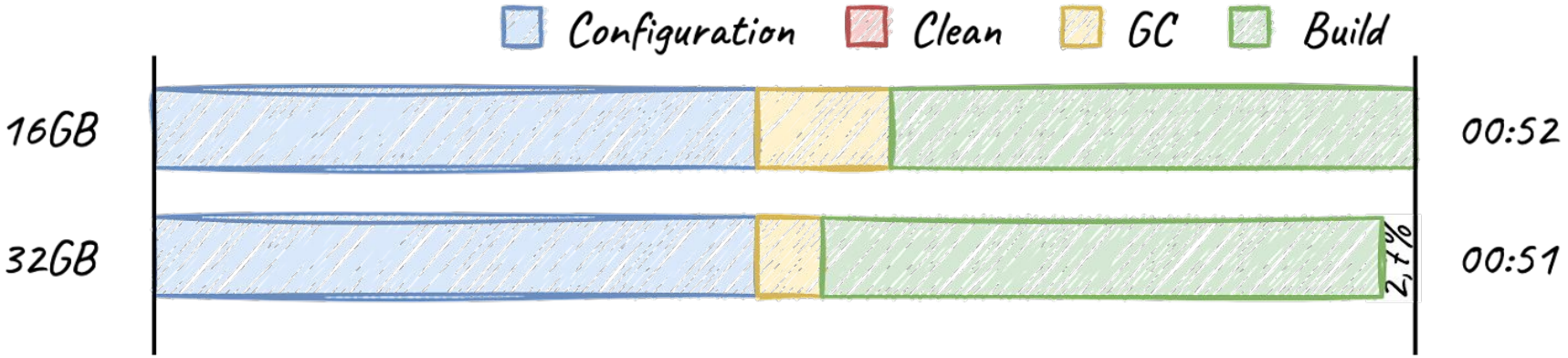
Многомодульная холодная сборка



Одномодульная холодная сборка



Многомодульная горячая сборка



Выводы

1. На CI Gradle можно дать больше памяти
2. На локальной машине 16 Гб пока достаточно
3. Одномодульному проекту вообще плевать

Частота памяти

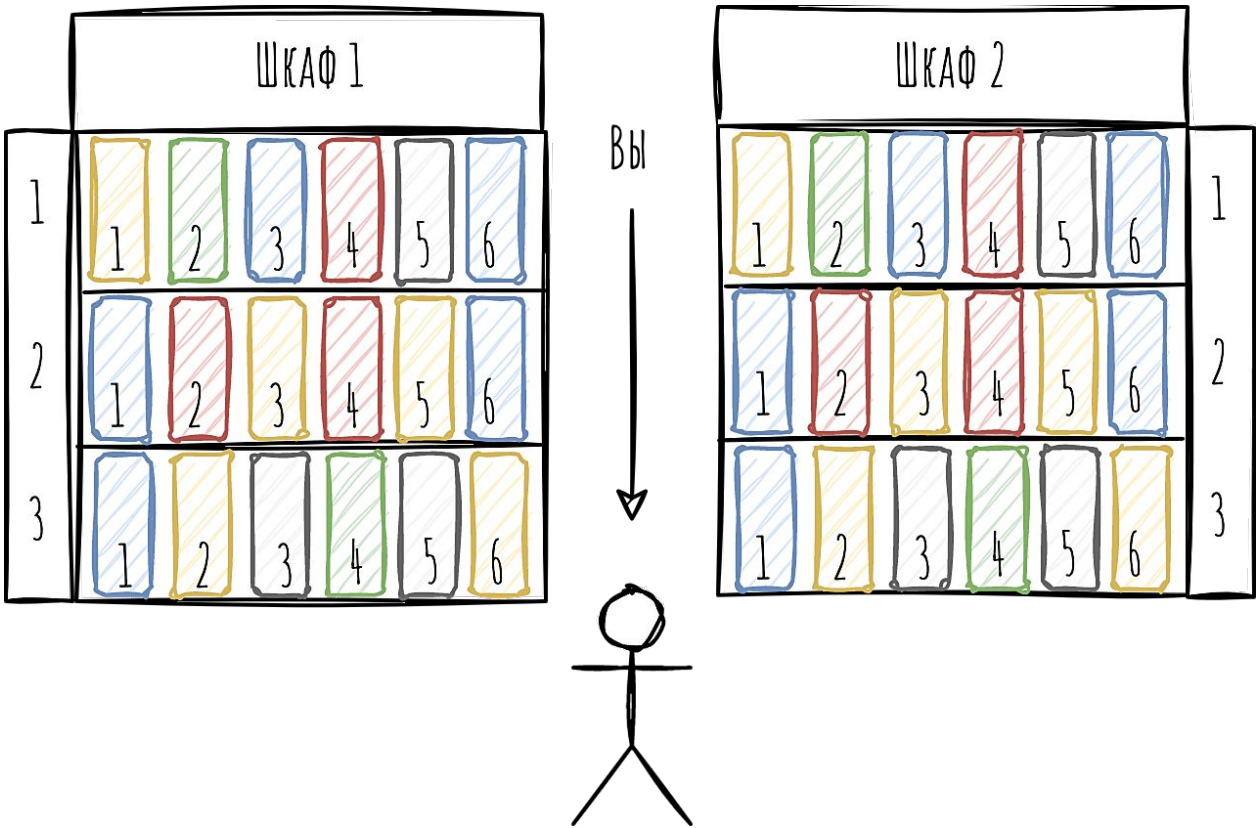
Частота памяти

2400 MHz	3000 MHz	3600 MHz
Минимум	+25%	+50%

Обязательно фиксируем тайминги

циан

Таймингы



Частота

Количество тактов в секунду

15*(1/3000) = 15/3000 = 0.005 с = 5 мс

Тайминг 15
Частота 3000

15*(1/3000) = 15/3000 = 0.005 с = 5 мс

Тайминг 16
Частота 3200

16*(1/3200) = 16/3200 = 0.005 с = 5 мс

Тайминги

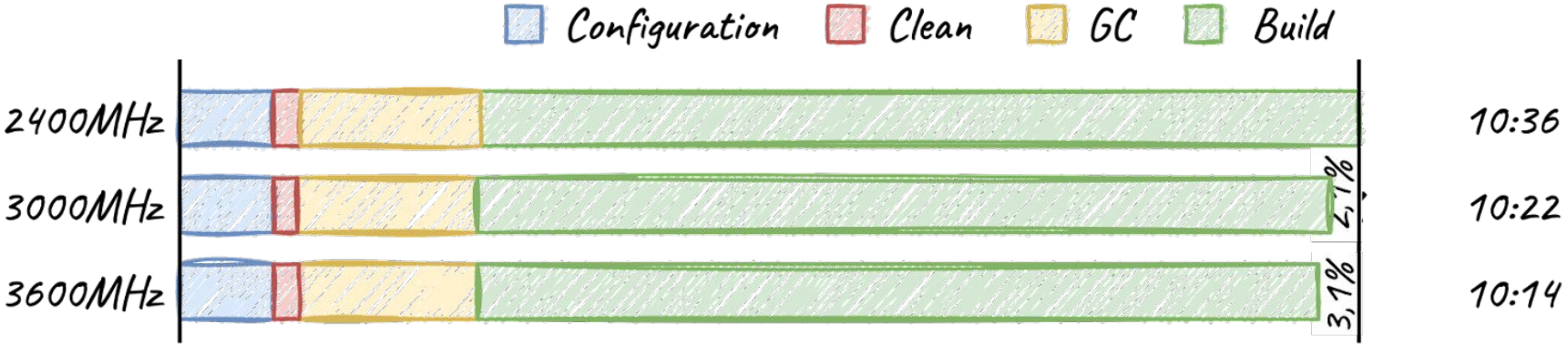
- Фиксируем тайминги на 20-20-20-42
- Сабтайминги фиксируем на те значения, что предлагает материнская плата на частоте 3600 МГц

Скорость

2400 MHz	3000 MHz	3600 MHz
Минимум	+25%	+50%
Чтение	Чтение	Чтение
25 GB/s	36 GB/s	42 GB/s
Запись	Запись	Запись
20 GB/s	25 GB/s	31 GB/s

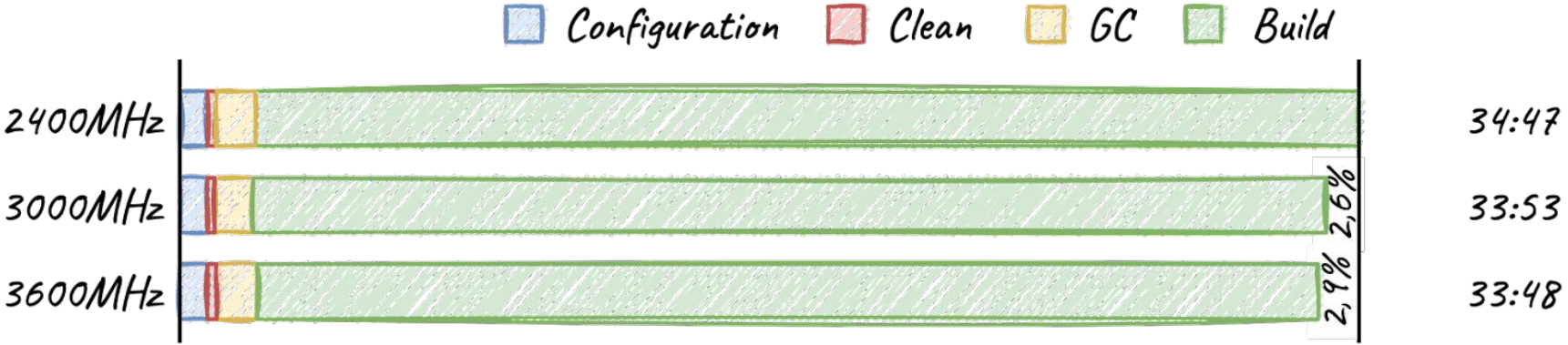
циан

Многомодульная холодная сборка



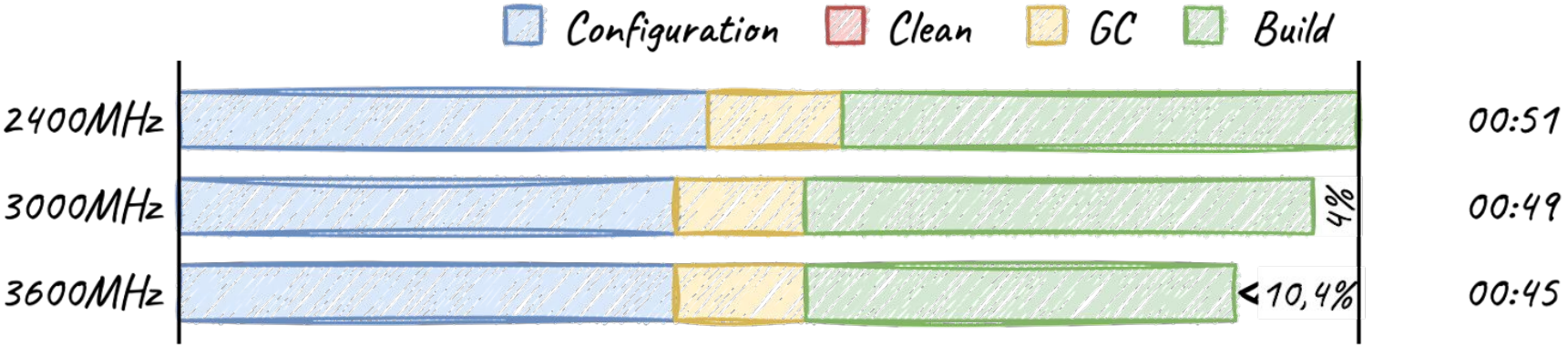
циан

Одномодульная холодная сборка



циан

Многомодульная горячая сборка



циан

Выводы

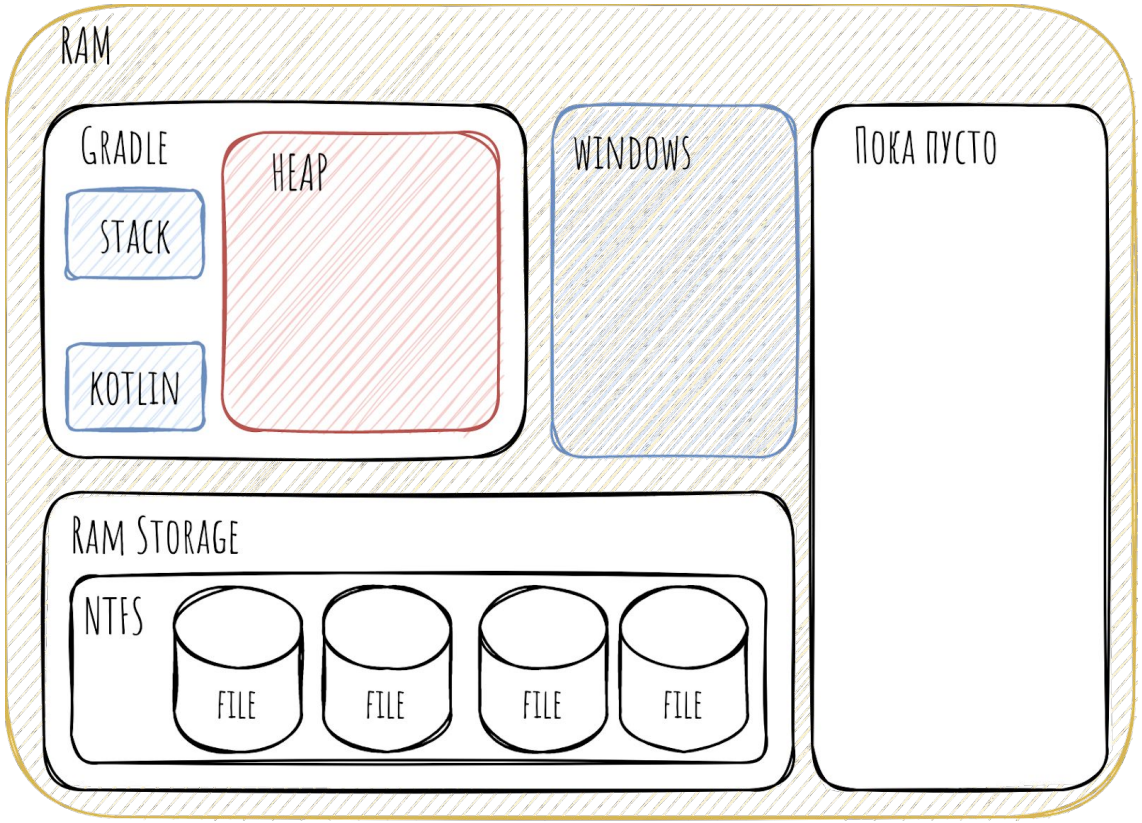
1. Покупать дорогую RAM не стоит
2. Хватит и комплекта с базовой частотой

Скорость накопителя

Накопители

- HDD 2.5”
- HDD 3.5”
- Sata SSD
- NVMe SSD
- RAM Storage

Ram Storage



циан

Скорости

	Sequence Read	Random Read	Sequence Write	Random Write
HDD 2.5”	80.845	0.629	67.491	0.727
HDD 3.5”	155.733	1.614	142.615	1.619
Sata SSD	446.519	163.999	353.164	126.276
NVMe SSD	3567.495	1542.787	2325.665	1577.327
RAM Storage	7841.693	1056.208	12537.931	658.770

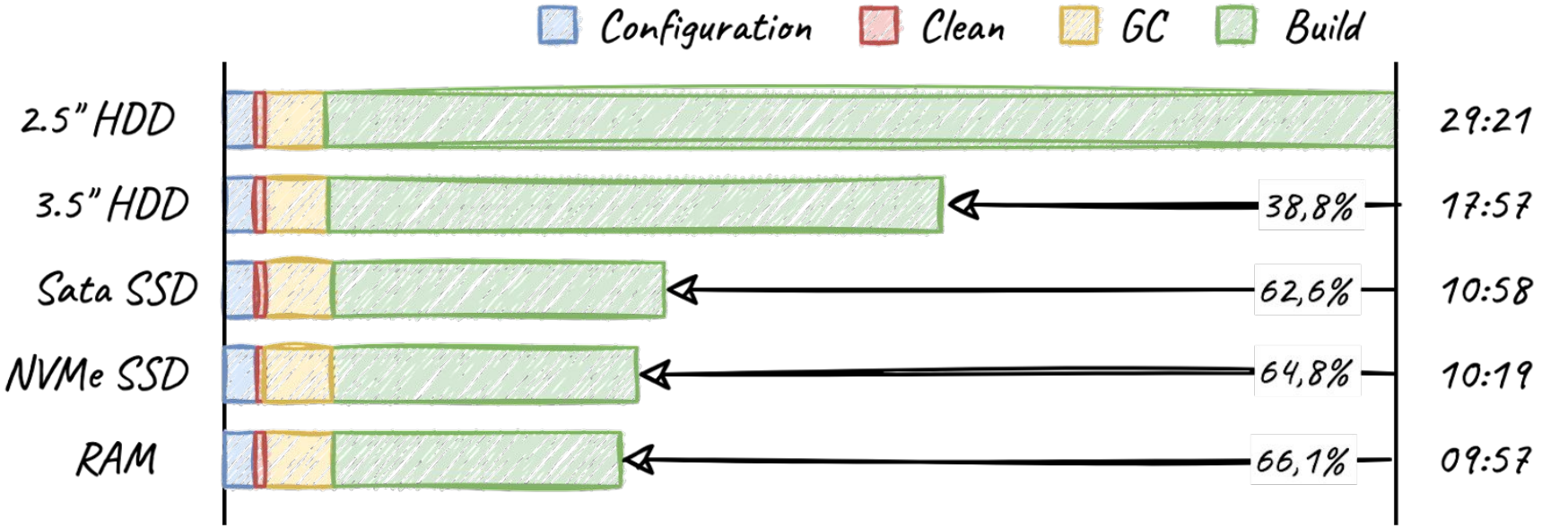
Никаких внешних кешей

Антивирус	Индексатор	ОС
Добавляем проект в исключения	Добавляем проект в исключения	Находится на другом накопителе

Первые два правила рекомендует Google

циан

Многомодульная холодная сборка

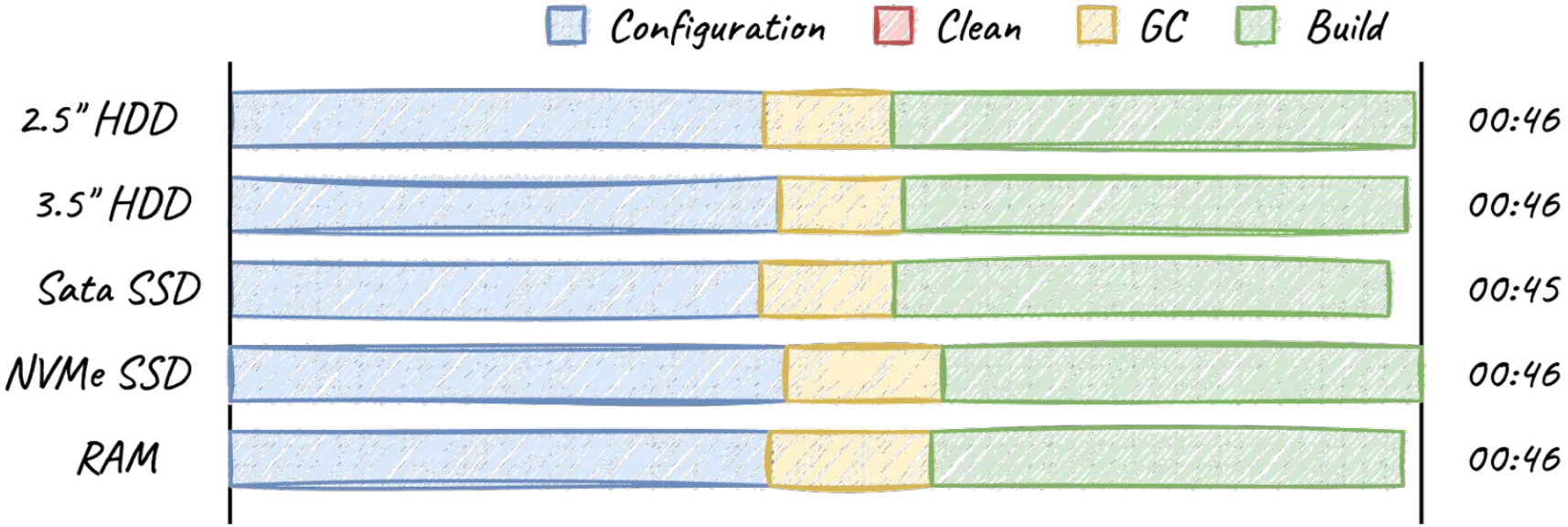


циан

Одномодульная холодная сборка

циан

Многомодульная горячая сборка



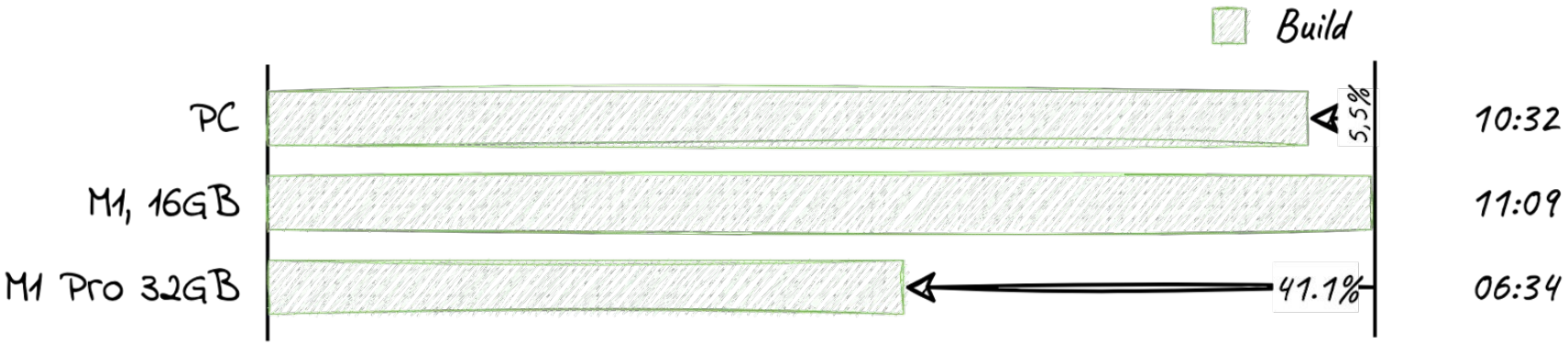
циан

Выводы

1. HDD пока!
2. Достаточно и обычного SSD, но лучше — NVMe
3. RAM Storage — ненужный шик

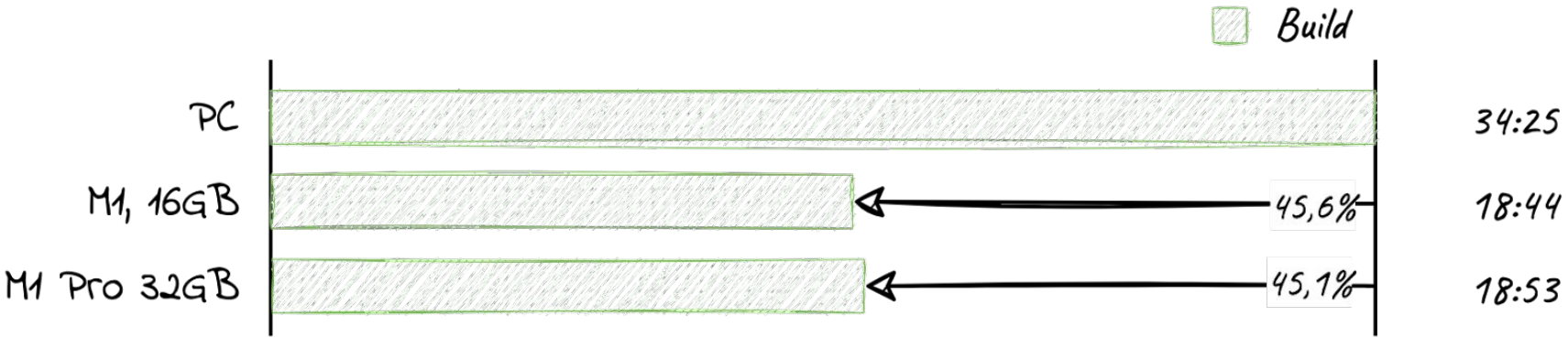
Apple M1

Многомодульная холодная сборка



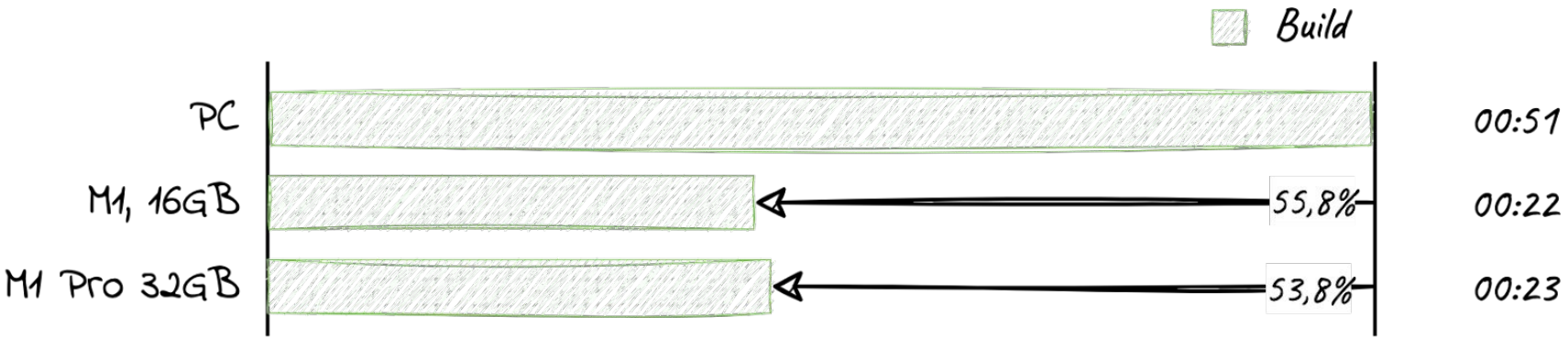
циан

Одномодульная холодная сборка



циан

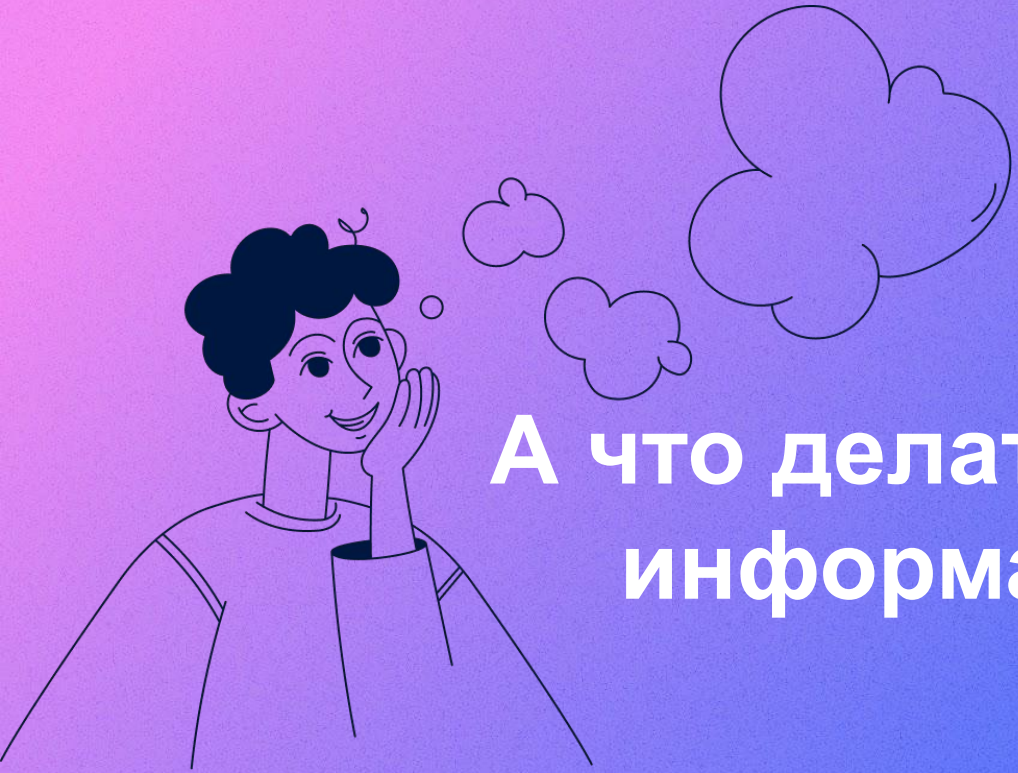
Многомодульная горячая сборка



циан

Выводы

1. M1/M2 хорош в производительности на одно ядро
2. Pro/Max/Ultra версии хороши и в многопотоке

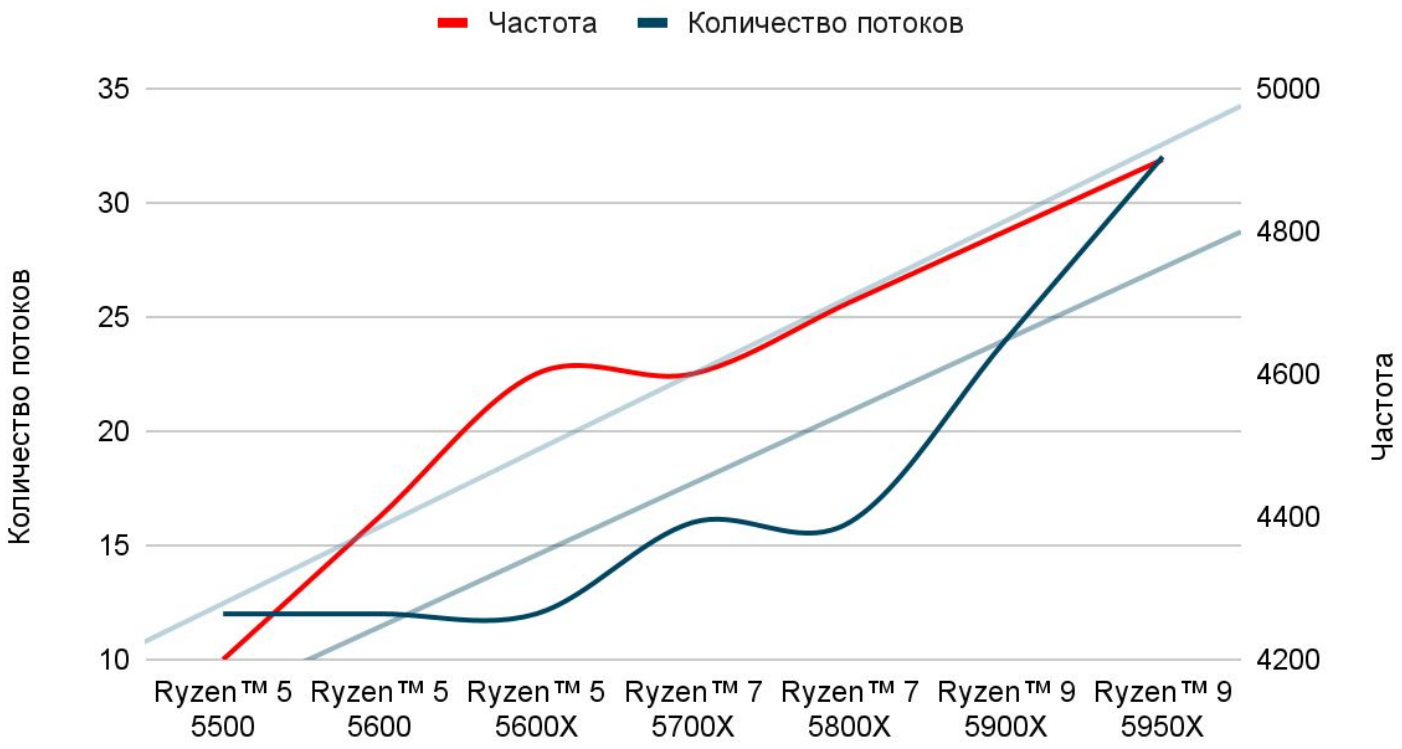


А что делать с этой информацией?

циан

ПК

AMD. Политика

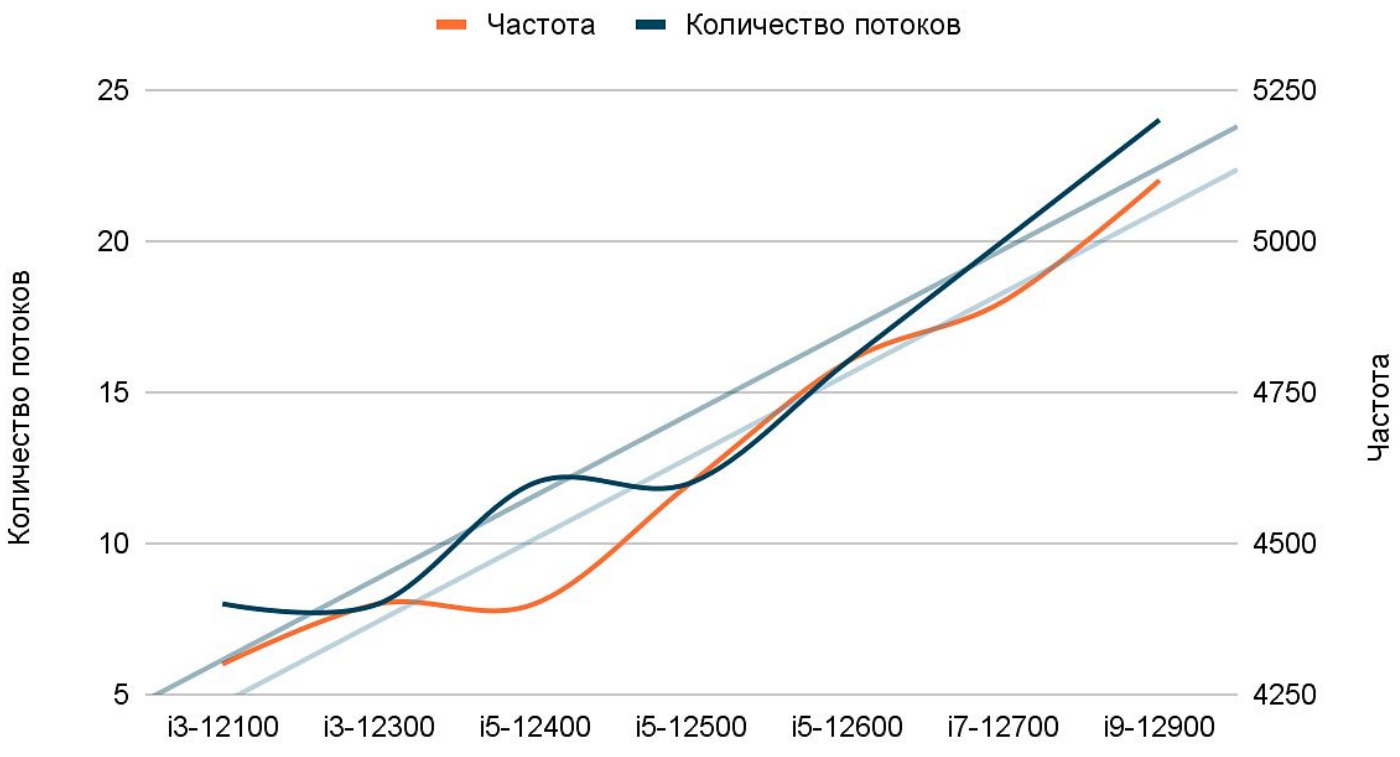


циан

AMD. Выбор

Модель	Количество ядер	Количество потоков	Макс. частота
Ryzen™ 9 5950X	16	32	4.9GHz
Ryzen™ 9 5900X	12	24	4.8GHz
Ryzen™ 7 5800X3D	8	16	4.5GHz
Ryzen™ 7 5800X	**8**	**16**	**4.7GHz**
Ryzen™ 7 5700X	**8**	**16**	**4.6GHz**
Ryzen™ 5 5600X	**6**	**12**	**4.6GHz**
Ryzen™ 5 5600	6	12	4.4GHz
Ryzen™ 5 5500	6	12	4.2GHz

Intel. Политика



циан

Intel. Выбор

Модель	Количество ядер	Количество потоков	Макс. частота
i9-12900	8P8E	24	5,10 GHz
i7-12700	**8P4E**	**20**	**4,90 GHz**
i5-12600	**6P4E**	**16**	**4,80 GHz**
i5-12500	**6P**	**12**	**4,60 GHz**
i5-12400	6P	12	4,40 GHz
i3-12300	4P	4	4,40 GHz
i3-12100	4P	4	4,30 GHz

Оперативная память

16 Гб

Если вы используете Android Studio
и пару-тройку программ

Но для очень тяжёлых проектов всё равно
не хватит

32 Гб

Вдвое дороже

Если вы не любите закрывать
вкладки браузера и программы

циан

Накопитель

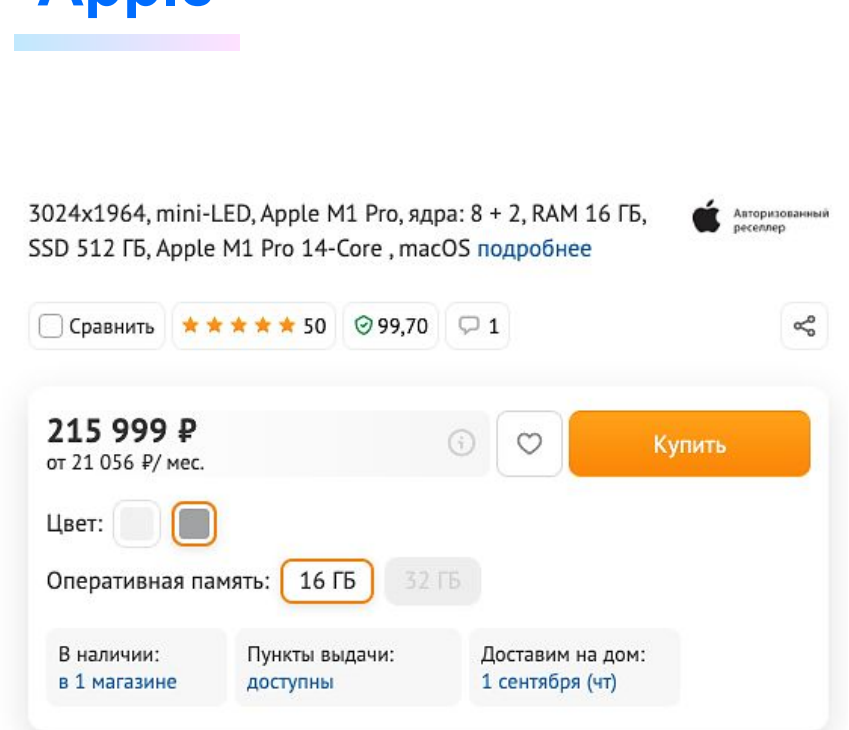
SATA SSD	Только если нет слота под NVMe
NVMe SSD	Стоит также, но быстрее

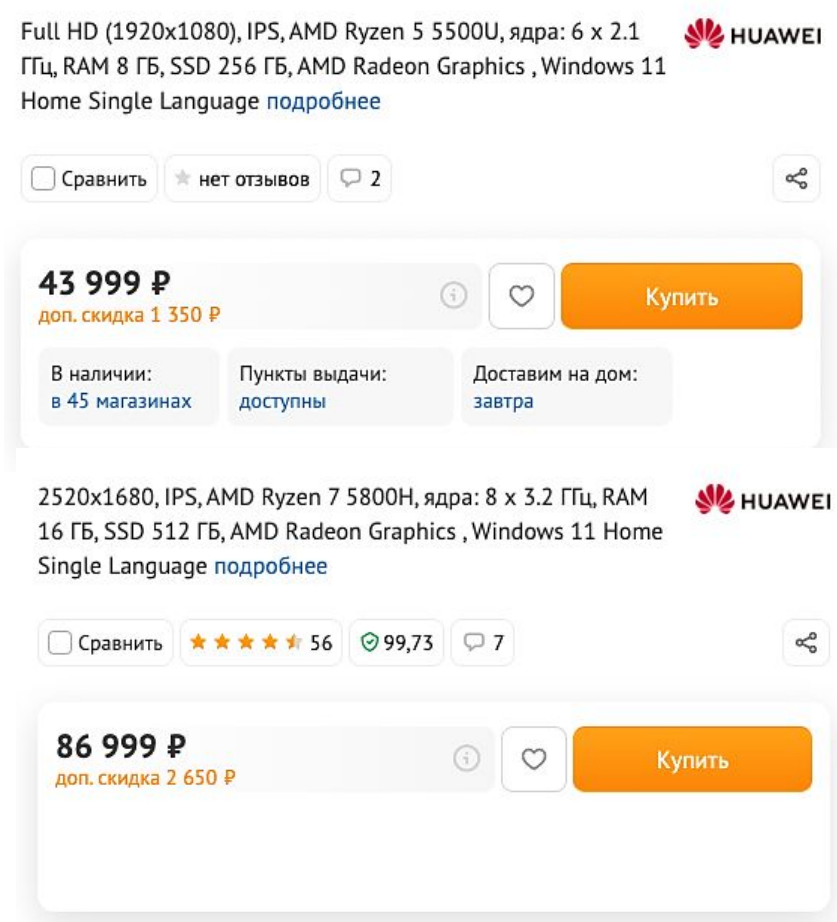
Ноутбук

циан

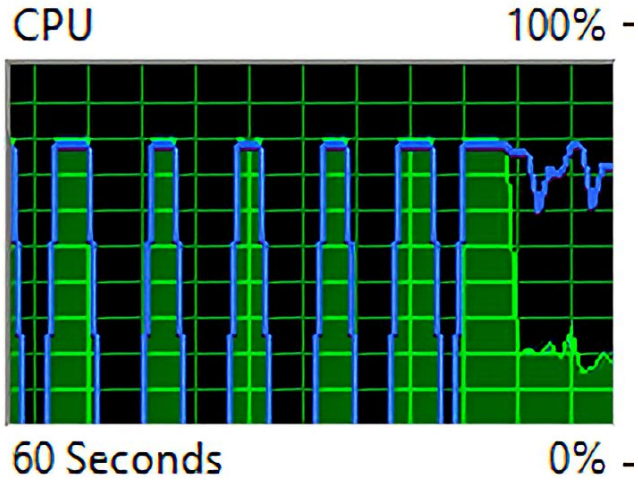
Apple	Тихо, производительно, очень дорого
PC Laptop	Дешевле

Apple


3024x1964, mini-LED, Apple M1 Pro, ядра: 8 + 2, RAM 16 ГБ, SSD 512 ГБ, Apple M1 Pro 14-Core , macOS подробнее
Авторизованный реселлер
Сравнить
50
99,70
1
215 999 ₽
от 21 056 ₽/ мес.
Купить
Цвет:
Оперативная память:
16 ГБ
32 ГБ
В наличии:
в 1 магазине
Пункты выдачи:
доступны
Доставим на дом:
1 сентября (чт)


Full HD (1920x1080), IPS, AMD Ryzen 5 5500U, ядра: 6 x 2.1 ГГц, RAM 8 ГБ, SSD 256 ГБ, AMD Radeon Graphics , Windows 11 Home Single Language подробнее
HUAWEI
Сравнить
нет отзывов
2
43 999 ₽
доп. скидка 1 350 ₽
Купить
В наличии:
в 45 магазинах
Пункты выдачи:
доступны
Доставим на дом:
завтра
2520x1680, IPS, AMD Ryzen 7 5800H, ядра: 8 x 3.2 ГГц, RAM 16 ГБ, SSD 512 ГБ, AMD Radeon Graphics , Windows 11 Home Single Language подробнее
HUAWEI
Сравнить
56
99,73
7
86 999 ₽
доп. скидка 2 650 ₽
Купить

Тротлинг



Что это?

Сбрасывание частот процессора при сильном нагреве

В чем проблема?

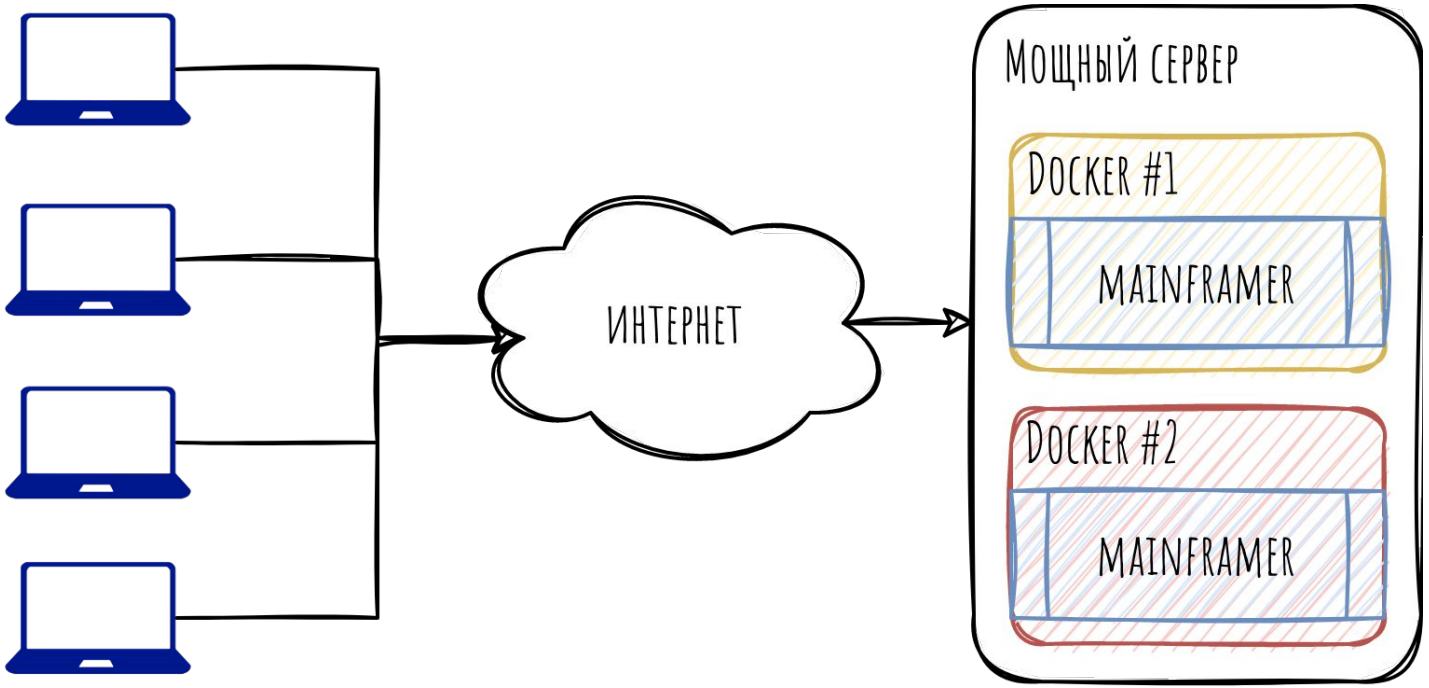
Сборка будет собираться сильно дольше положенного

В целом ниже частоты

На ноутбуках максимальные частоты ниже на 400 — 600 МГц

циан

Mainframer


Мощный сервер
Docker #1
MAINFRAMER
Docker #2
MAINFRAMER
Интернет

CI

HDD, пока!

В целом медленный

Помимо сборок, другие задачи тоже тормозятся

Замедляет сборку

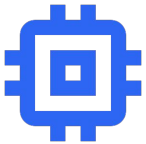
На минуточку, на 25%

Жрёт электричество

Почти в 10 раз больше

Если где-то остались — лучше убрать

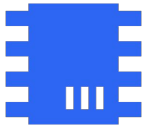
CI. Наш выбор



CPU

от 8 потоков на сборку

лучше выше частоты, чем количество ядер



RAM

в -Xmx выставить на 4 Гб больше чем на личных машинах

на сборку -Xmx + 4 Гб

ROM

Sata SSD

Дилемма

VS

Apple M1, 8x3.2 ГГц, 16 ГБ DDR4, SSD 256 ГБ, macOS
подробнее

Авторизованный реселлер

Сравнить | 29 | 99,29

102 999 ₽
от 10 041 ₽/ мес.

Уведомить

Товара нет в наличии

Apple M1, 8x3.2 ГГц, 8 ГБ DDR4, SSD 256 ГБ, macOS
подробнее

Авторизованный реселлер

Сравнить | 33 | 99,50 | 3

80 999 ₽
от 7 896 ₽/ мес.

Уведомить

Итоговый итог

Итоговый итог

1. Стоит посматривать в сторону Apple
2. Частота CPU важнее количества ядер
3. Частота RAM ничего не решает
4. HDD умерли
5. Хотите сидеть за ноутбуком — используйте mainframer

Что можно сделать сейчас, каков первый шаг. Многомодульный проект

1. Определяем максимально нужное количество ядер
2. Берём самый многоядерный комп, до которого можете дотянуться
3. Устанавливаем в gradle max-workers 4..8..12..16 и запускаем сборку со –scan
4. В результатах scan на timeline смотрим количество пустот. Когда их будет больше 30%, останавливаемся
5. Больше этого количества ядер вам точно не нужно

Что можно сделать сейчас, каков первый шаг. Одномодульный проект

1. Наращиваем частоту
2. Разгон
3. Покупка более частотного процессора

Вопросы

@princeparadoxes

циан