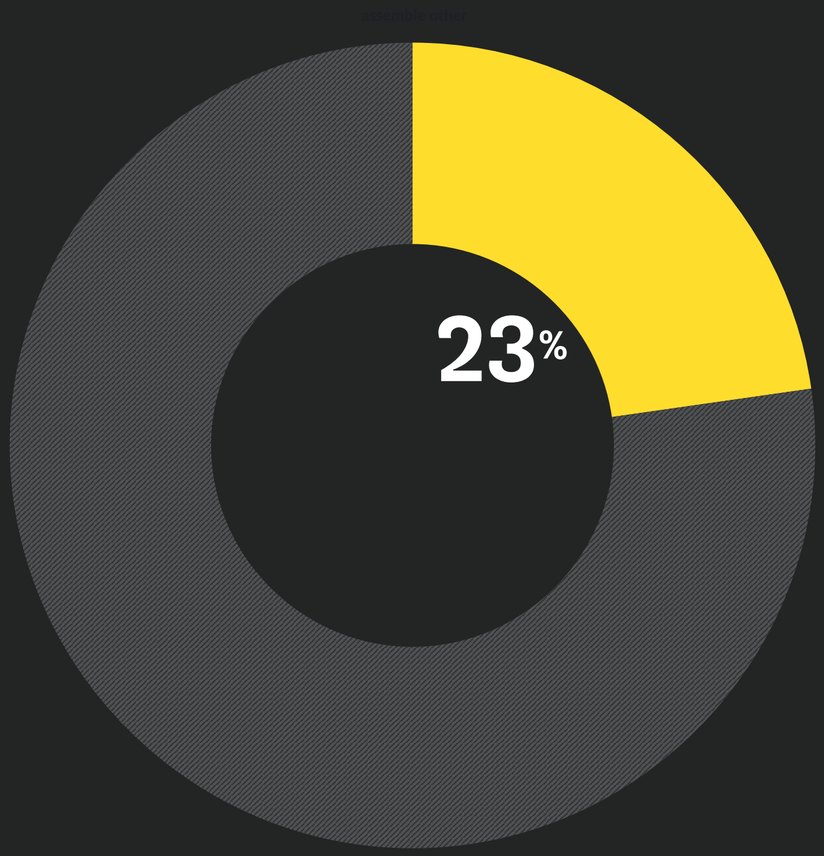
ТИНЬКОФФ

Gradle-сборка. Измеряем самое важное

Юрий Анисимов

Сборка - всего лишь одна из Gradle тасок

Доля сборок в общем количестве



Зачем вообще измерять сборку?

Каков результат улучшений?

Где мы находимся сейчас?

Какая динамика изменений?

Действия команды разработки

Источники замедления сборки повсюду

Следование новомодным трендам

Подключение новых библиотек и фреймворков, использование кодогенерации и т.д.

Свобода настройки в Gradle

Почти любое изменение будет приводить к пересборке всего проекта.

Обновление рабочего окружения

Баги и новые фичи в Gradle, Kotlin и Android Gradle Plugin.

Привнесенные проблемы со стороны

Причины могут не зависеть от вашей команды

- **Антивирусы и файерволы**

  Без комментариев(

- **Пропускная способность сервисов**

  Nexus, Artifactory, Cache node, ...

- **Особенности настроек сети**

  VPN, маршрутизация, ...

Tinkoff Build Metrics Plugin

Сбор метрик

Вычисляет метрики сборки
Андроид приложения
и не только.

Публикация

Отправляет собранную
информацию в хранилище
системы аналитики.

Tinkoff Build Metrics Plugin

- Сбор информации
- Особенности реализации
- Примеры использования данных
- Переиспользование плагина

Инструмент для оценки общей картины

Чтобы измерить результат

- gradle-profiler
- gradle --profile

Сколько длится ваша сборка?

- «Холодная» сборка более 5 минут?

  Стоит задуматься об измерении сборки.

- Сборка идет быстрее?

  Вряд ли будет профит от сбора и анализа метрик

Юрий Анисимов

Разработчик Инфра

Когда-то был C++ разработчиком

y.anisimov@tinkoff.ru

Мобильный банк

4 M
Строк кода

1700+
Модулей

300
Контрибьюторов

4k
Сборок в день

Содержание

История появления

Особенности реализации

Использование данных

Кастомизация и применение

Важно знать: что происходит и, желательно, почему?

Для этих целей мы бы хотели фиксировать параметры каждой сборки (как локальной, так и на CI) для последующей идентификации проблем и поиска узких мест.

Какой вопрос интересует одновременно, и менеджеров, и разработчиков?

Какой вопрос интересует одновременно, и менеджеров, и разработчиков?

Сколько времени разработчики ждут завершения локальной сборки и сборки на мерж реквесте?

Ожидание завершения сборки

Время

Чем дольше ждать получение результата, тем меньше желания запрашивать это результат.

Деньги

Время ожидания результатов легко конвертировать в деньги, которая теряет компания.

Профит от слежения за общим временем сборки

Последствия

Оценить результат внедрения изменений.

Тренды

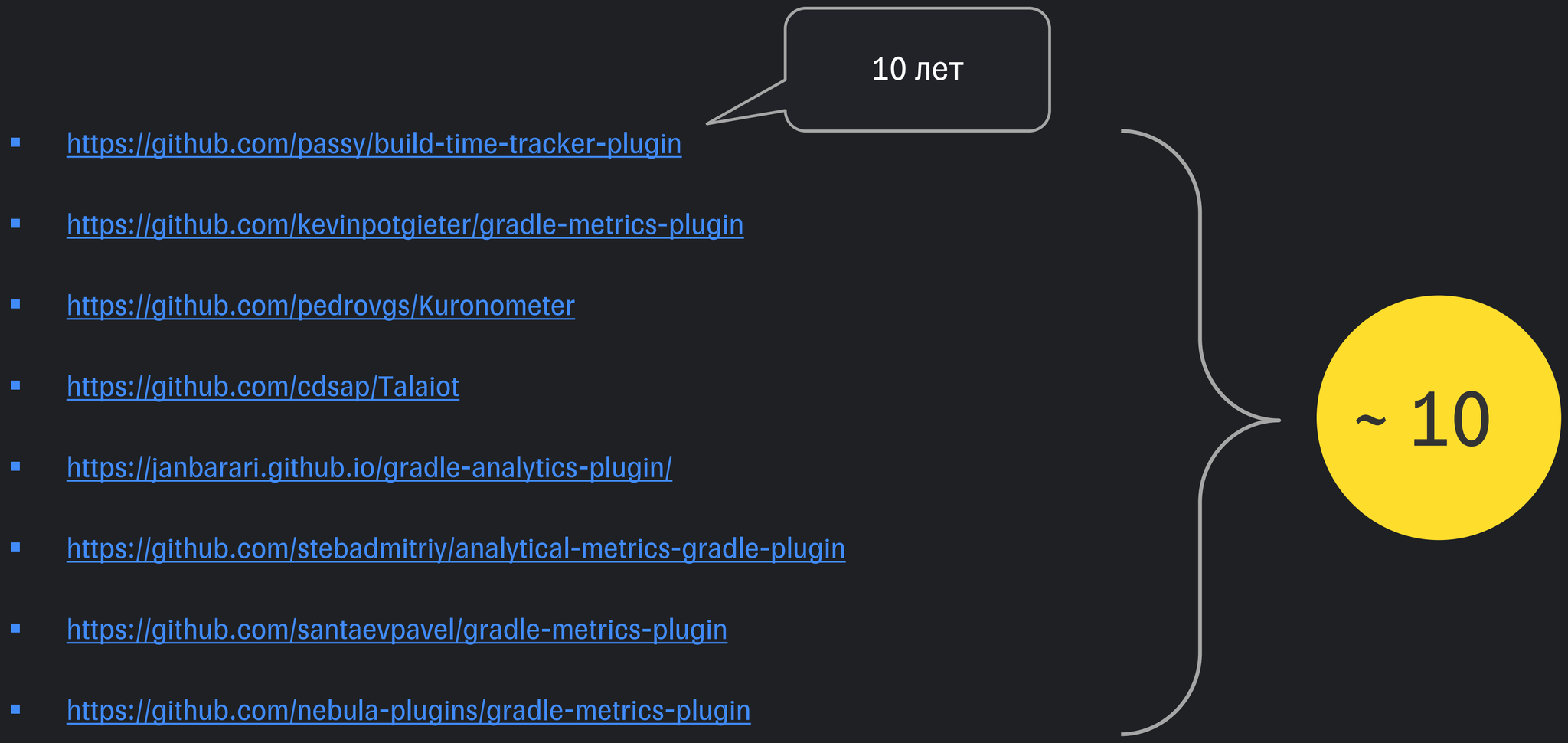
Наблюдать за текущим состоянием и трендами.

Потери и простои

Предоставить показатели продуктивности и затрат для бизнеса.

Длительность сборки — верхушка айсберга

Известные решения



«Живые» плагины

- Develocity (aka Gradle Enterprise)
- Gradle Analytics Plugin
- Talaiot

Talaiot 1.5
Начало

- Бесплатный инструмент
- Много параметров “из коробки”
- Активно развивается

Talaiot 1.5 Завершение истории

Несовместимость с кэшом конфигурации

Искажение информации

При попадании в кэш конфигурации, не собирались данные.

Отсутствие поддержки

Разработчик плагина перешел работать в Gradle и инструмент перестал развиваться.

Несовместимость с Gradle 8.1

С релизом 8.1 Gradle объявил о стабилизации кэша конфигурации.

Допилить Talaiot

или

Написать свой плагин

- История появления
- **Особенности реализации**
- Использование данных
- Кастомизация и применение

Tinkoff Build Metrics Plugin

Необходимый минимум

- **Shared Build Service**

  Хранение общего состояния для переиспользования между тасками

- **Plugin**

  Инкапсуляция логики по настройке и подключению

Создать билд сервис

```
abstract class MyBuildService :
     BuildService<MyBuildService.Params>,
     AutoCloseable,
     OperationCompletionListener
{

    interface Params : BuildServiceParameters { ... }

    override fun onFinish(event: FinishEvent?) { ... }

    override fun close() { ... }
}
```

Создать билд сервис

```
abstract class MyBuildService :
      BuildService<MyBuildService.Params>,
      AutoCloseable,
      OperationCompletionListener
{

    interface Params : BuildServiceParameters { ... }

    override fun onFinish(event: FinishEvent?) { ... }

    override fun close() { ... }
}
```

Создать билд сервис

```
abstract class MyBuildService :
     BuildService<MyBuildService.Params>,
     AutoCloseable,
     OperationCompletionListener
{

    interface Params : BuildServiceParameters { ... }

    override fun onFinish(event: FinishEvent?) { ... }

    override fun close() { ... }
}
```

Создать билд сервис

```
abstract class MyBuildService :
     BuildService<MyBuildService.Params>,
     AutoCloseable,
     OperationCompletionListener
{

    interface Params : BuildServiceParameters { ... }

    override fun onFinish(event: FinishEvent?) { ... }

    override fun close() { ... }
}
```

Создать билд сервис

```
abstract class MyBuildService :
     BuildService<MyBuildService.Params>,
     AutoCloseable,
     OperationCompletionListener
{

    interface Params : BuildServiceParameters { ... }

    override fun onFinish(event: FinishEvent?) { ... }

    override fun close() { ... }
}
```

Создать билд сервис

```
abstract class MyBuildService :
     BuildService<MyBuildService.Params>,
     AutoCloseable,
     OperationCompletionListener
{

    interface Params : BuildServiceParameters { ... }

    override fun onFinish(event: FinishEvent?) { ... }

    override fun close() { ... }
}
```

Между окончанием сборки и завершением задач

Зарегистрировать билд сервис

```
abstract class MyPlugin : Plugin<Project> {

    @get:Inject
    abstract val buildEventListenerRegistry: BuildEventListenerRegistry

    override fun apply(target: Project) {
        val serviceProvider: Provider<MyBuildService> =
                target.gradle.sharedServices.registerIfAbsent(
                    "...",
                    MyBuildService::class.java
                ) { spec -> ... }

        buildEventsListenerRegistry.onTaskCompletion(serviceProvider)
    }
```

Зарегистрировать билд сервис

```
abstract class MyPlugin : Plugin<Project> {

    @get:Inject
    abstract val buildEventListenerRegistry: BuildEventListenerRegistry

    override fun apply(target: Project) {
        val serviceProvider: Provider<MyBuildService> =
                target.gradle.sharedServices.registerIfAbsent(
                    "...",
                    MyBuildService::class.java
                ) { spec -> ... }

        buildEventsListenerRegistry.onTaskCompletion(serviceProvider)
    }
```

Зарегистрировать билд сервис

```
abstract class MyPlugin : Plugin<Project> {

    @get:Inject
    abstract val buildEventListenerRegistry: BuildEventListenerRegistry

    override fun apply(target: Project) {
        val serviceProvider: Provider<MyBuildService> =
                target.gradle.sharedServices.registerIfAbsent(
                    "...",
                    MyBuildService::class.java
                ) { spec -> ... }

        buildEventsListenerRegistry.onTaskCompletion(serviceProvider)
    }
```

Старт по первому требованию

Зарегистрировать билд сервис

```
abstract class MyPlugin : Plugin<Project> {

    @get:Inject
    abstract val buildEventListenerRegistry: BuildEventListenerRegistry

    override fun apply(target: Project) {
        val serviceProvider: Provider<MyBuildService> =
                target.gradle.sharedServices.registerIfAbsent(
                    "...",
                    MyBuildService::class.java
                ) { spec -> ... }

        buildEventsListenerRegistry.onTaskCompletion(serviceProvider)
    }
```

Обработать события и экспортировать данные

- Подключить плагин
- Реализовать обработку событий
- Реализовать паблишер

Обработать события и экспортировать данные

- Подключить плагин
- Реализовать обработку событий
- Реализовать паблишер

Результат реверс-инжиниринга

Типы плагинов Gradle

По области применения

Project Plugin

`build.gradle`

`build.gradle.kts`

Настройка модуля к которому подключен плагин.

Settings Plugin

`settings.gradle`

`settings.gradle.kts`

Применение различных действий ко всему билду.

Init Plugin

`init.gradle`

`init.gradle.kts`

Глобальные настройки Gradle в рамках машины.

Settings Plugin

```
abstract class MyPlugin : Plugin<Settings> {

    ...

}
```

Критерии выбора

- Избежать потери данных
- Доступ ко всем билд операциям
- Сбор метрик для всего билда

Критерии выбора

- Избежать потери данных
- Доступ ко всем билд операциям
- Сбор метрик для всего билда
- Оказалось удачным решением

Из чего складывается время сборки

Конфигурация

- Инициализация
- Загрузка билда
- Конфигурирование
- Построение графа тасок

Фаза выполнения

Выполнение тасок, которые
были подготовлены
на этапе конфигурации.

Из чего складывается время сборки

Зависимости

- Время загрузки
- Количество данных
- Скорость загрузки
- Источник данных

Из чего складывается время сборки

Зависимости

- Время загрузки
- Количество данных
- Скорость загрузки
- Источник данных

Кэш конфигурации

- Время чтения
- Время сохранения
- Объем данных

Из чего складывается время сборки

Зависимости

- Время загрузки
- Количество данных
- Скорость загрузки
- Источник данных

Кэш конфигурации

- Время чтения
- Время сохранения
- Объем данных

Билд кэш

- Время загрузки
- Время сохранения
- Объем информации
- Скорость чтения и записи

Из чего складывается время сборки

Зависимости

- Время загрузки
- Количество данных
- Скорость загрузки
- Источник данных

Кэш конфигурации

- Время чтения
- Время сохранения
- Объем данных

Билд кэш

- Время загрузки
- Время сохранения
- Объем информации
- Скорость чтения и записи

Рабочее окружение

- Используемое железо
- Количество модулей
- Настройки проекта
- Запускаемые таски

Источники информации

Public API

BuildEventsListenerRegistry

Internal API

BuildEventListenerRegistryInternal

Public API

Сделан для удобства потребителей

- Потокобезопасная реализация
- Не блокирует выполнение тасок

Ограниченная информация

Только данные о тасках и общие сведения о конфигурации сборки

Public API

- **BuildPhaseFinishEvent**

  Длительность фаз сборки

- **FileDownloadFinishEvent**

  Источник, размер, время скачивания файла

- **ProjectConfigurationFinishEvent**

  Время конфигурации модулей

- **TaskFinishEvent**

  Длительность выполнения таски

Internal API

```
public final class BuildOperationDescriptor {
    private final OperationIdentifier id;
    private final OperationIdentifier parentId;
    private final String name;
    private final Object details;
    ...
}
```

Известно до выполнения

```
public final class OperationFinishEvent {
    private final long startTime;
    private final long endTime;
    private final Throwable failure;
    private final Object result;
    ...
}
```

Результат выполнения

Internal API

```
public final class BuildOperationDescriptor {
    private final OperationIdentifier id;
    private final OperationIdentifier parentId;
    private final String name;
    private final Object details;
    ...
}

public final class OperationFinishEvent {
    private final long startTime;
    private final long endTime;
    private final Throwable failure;
    private final Object result;
    ...
}
```

Internal API

```
public final class BuildOperationDescriptor {
    private final OperationIdentifier id;
    private final OperationIdentifier parentId;
    private final String name;
    private final Object details;
    ...
}

public final class OperationFinishEvent {
    private final long startTime;
    private final long endTime;
    private final Throwable failure;
    private final Object result;
    ...
}
```

Информация о билд операции

```java
/**
 * A type token for a type of a rich build operation that provides
 * structured metadata.
 *
 * @param <D> the type of details object for the operation
 * @param <R> the type of result object for the operation
 * @since 4.0
 */
@SuppressWarnings("unused")
public interface BuildOperationType<D, R> {
}
```

Контракт данных

- **Поддержка обратной совместимости**

  Для типов details и result

- **Минимальное количество зависимостей**

  Отсутствие внутренних типов Gradle

- **Использование типов Java**

  Примитивы и другие типы из поставки JDK

- **Поддержка сериализации**

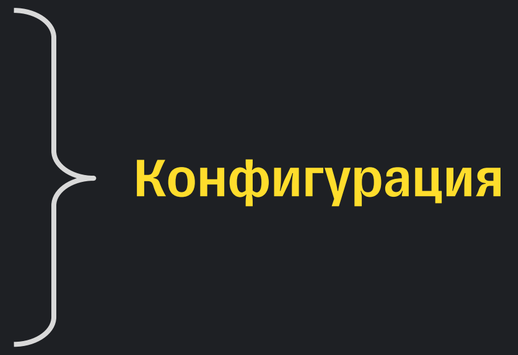
  Сторонние (non Java) форматы: JSON и т.п.

Иерархия билд операций

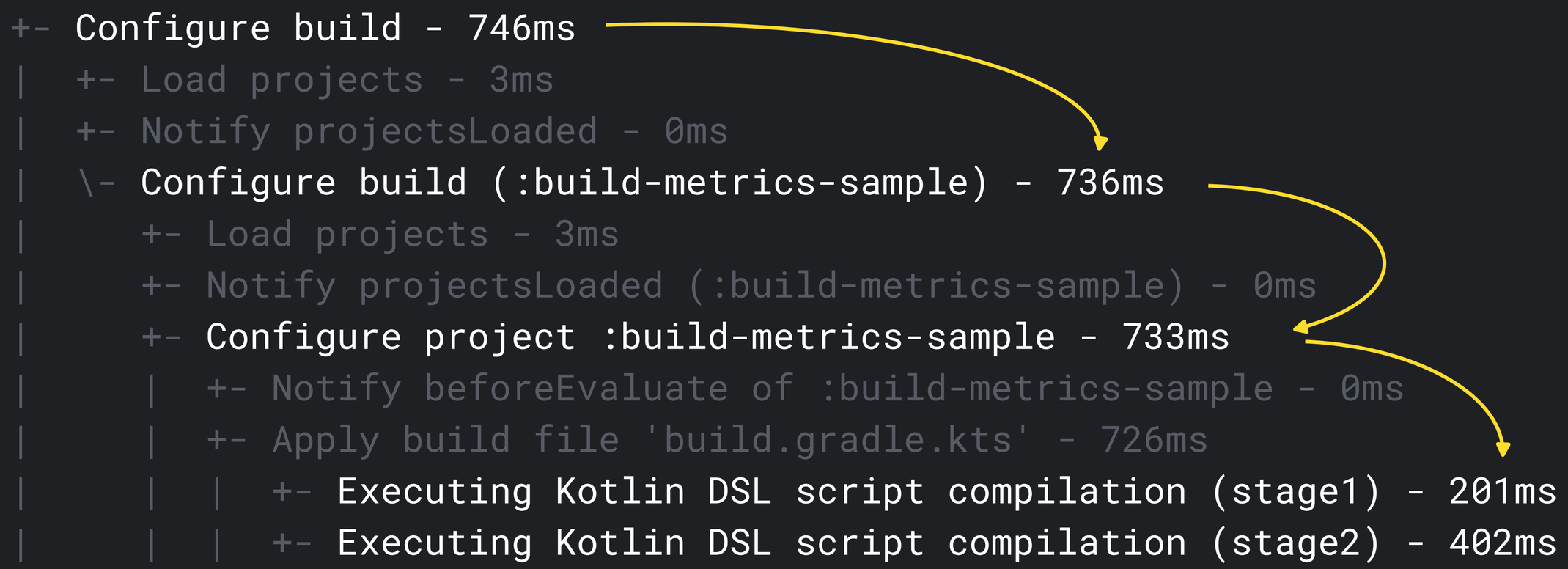
```
public final class BuildOperationDescriptor {
    private final OperationIdentifier id;
    private final OperationIdentifier parentId;
    private final String name;
    private final Object details;
    ...
}
```

Фазы сборки

```
..2809ms +--- Load build - 119595ms
122404ms +--- Configure build - 746ms
123150ms +--- Calculate build tree task graph - 1769ms
124920ms +--- Run main tasks - 76106ms
```



Самые длительные операции



Порядок вызова колбеков

```
+--- Configure build - 746ms
|    +--- Load projects - 3ms
|    +--- Notify projectsLoaded - 0ms
|    \--- Configure build (:build-metrics-sample) - 736ms
|         +--- Notify projectsLoaded (:build-metrics-sample) - 0ms
|         +--- Configure project :build-metrics-sample - 733ms
|         |    +--- Notify beforeEvaluate of :build-metrics-sample - 0ms
|         |    \--- Notify afterEvaluate of :build-metrics-sample - 1ms
|         \--- Notify projectsEvaluated (:build-metrics-sample) - 0ms
+--- Calculate build tree task graph - 1769ms
|    +--- Calculate task graph - 811ms
|    |    +--- Configure project : - 784ms
|    |    |    +--- Notify beforeEvaluate of : - 0ms
|    |    |    \--- Notify afterEvaluate of : - 0ms
|    +--- Notify task graph whenReady (:build-metrics-sample) - 0ms
```

1

Порядок вызова колбеков

```
+--- Configure build - 746ms
|    +--- Load projects - 3ms
|    +--- Notify projectsLoaded - 0ms
|    \--- Configure build (:build-metrics-sample) - 736ms
|         +--- Notify projectsLoaded (:build-metrics-sample) - 0ms
|         +--- Configure project :build-metrics-sample - 733ms
|         |    +--- Notify beforeEvaluate of :build-metrics-sample - 0ms
|         |    \--- Notify afterEvaluate of :build-metrics-sample - 1ms
|         \--- Notify projectsEvaluated (:build-metrics-sample) - 0ms
+--- Calculate build tree task graph - 1769ms
|    +--- Calculate task graph - 811ms
|    |    +--- Configure project : - 784ms
|    |    |    +--- Notify beforeEvaluate of : - 0ms
|    |    |    \--- Notify afterEvaluate of : - 0ms
|    +--- Notify task graph whenReady (:build-metrics-sample) - 0ms
```

2

Иерархия ошибок

```
Could not determine the dependencies of task ':bank:mergeDevDebugNativeLibs'
  > Could not resolve all task dependencies for configuration
    ':bank:devDebugRuntimeClasspath'.
      > Could not resolve ru.tinkoff.core.components.log:log:2.0.1
        > Cannot find a version of 'ru.tinkoff.core.components.log:log'
```

Иерархия ошибок

```
Could not determine the dependencies of task ':bank:mergeDevDebugNativeLibs'
  > Could not resolve all task dependencies for configuration
    ':bank:devDebugRuntimeClasspath'.
      > Could not resolve ru.tinkoff.core.components.log:log:2.0.1
        > Cannot find a version of 'ru.tinkoff.core.components.log:log'
```

Иерархия ошибок

```
Could not determine the dependencies of task ':bank:mergeDevDebugNativeLibs'
  > Could not resolve all task dependencies for configuration
    ':bank:devDebugRuntimeClasspath'.
      > Could not resolve ru.tinkoff.core.components.log:log:2.0.1
        > Cannot find a version of 'ru.tinkoff.core.components.log:log'
```

Сколько билд операций при сборке Мобильного банка?

50 000

500 000

5 000 000

Сколько билд операций при сборке Мобильного банка?

50 000

500 000

5 000 000

Internal API

- Жизненный цикл объектов неопределен
- Избегать удерживания ссылок
- Заботиться о потокобезопасности

Internal API

- Жизненный цикл объектов неопределен
- Избегать удерживания ссылок
- Заботиться о потокобезопасности
- +4 Гб к потреблению памяти

Кэш конфигурации

Основные этапы

Версия 6.5

Первые упоминания

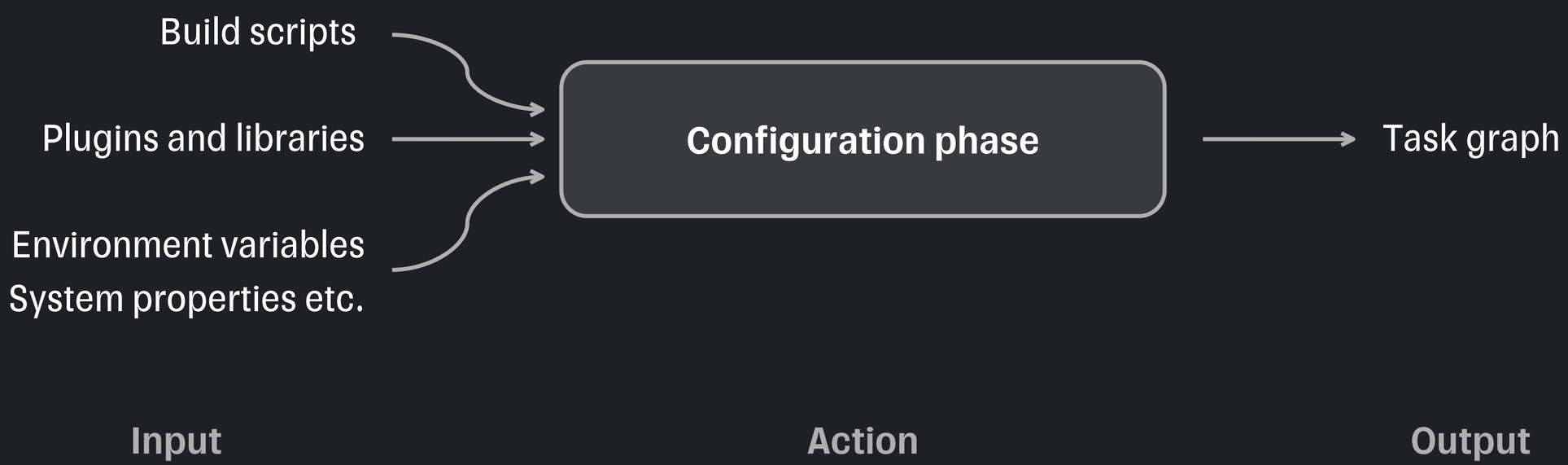
Версия 8.1

Объявлен стабильным. Рекомендуется к использованию.

Версия 9.0

Планируется включить кэш конфигурации по умолчанию

Принцип реализации



Входные данные

- **Билд скрипты и их зависимости**

  Билд скрипты, плагины и библиотеки

- **Переменные окружения**

  Environment variables, System properties etc.

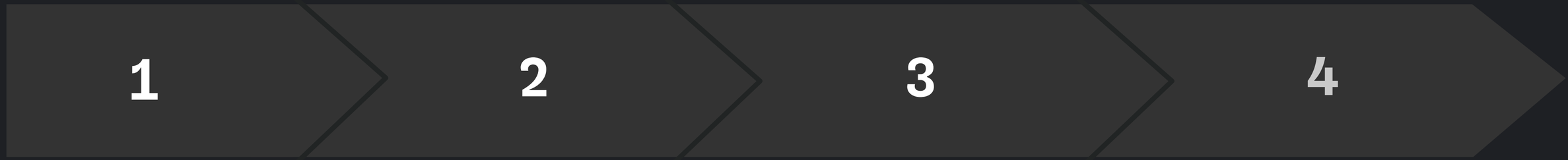
- **Чтение файлов и данных о файловой системе**

  Наличие файлов, структура директорий и т.д.

- **Обращение к внешним процессам**

  Использование сторонних утилит

Порядок работы



Конфигурация

Выполнение фазы конфигурации в обычном режиме

Сохранение

Сохранение снапшота графа тасок для переиспользования

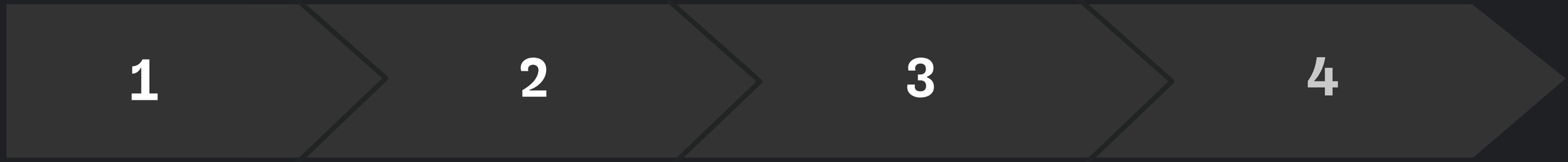
Загрузка из кэша

Чтение графа тасок из кэша для применения оптимизаций

Фаза выполнения

Выполнение тасок в обычном режиме

Порядок работы



Конфигурация

Выполнение фазы конфигурации в обычном режиме

Сохранение

Сохранение снапшота графа тасок для переиспользования

Загрузка из кэша

Чтение графа тасок из кэша для применения оптимизаций

Фаза выполнения

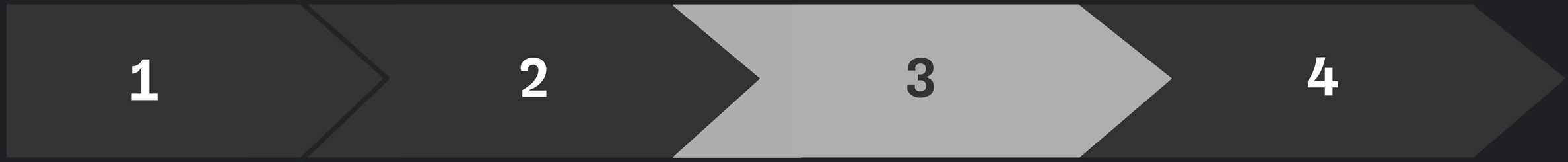
Выполнение тасок в обычном режиме

Требует времени!

Оптимизации

- Параллельное выполнение тасок
- Кэширование разрешения зависимостей
- Освобождение памяти

Повторный запуск



Проверка стейта

Отсутствуют изменения входных данных

Загрузка из кэша

Чтение сохраненного графа тасок из кэша

Конфигурация

Фаза конфигурации полностью пропускается

Фаза выполнения

Выполнение тасок в обычном режиме

Сериализация из коробки

- Произвольный граф объектов
- Зависимости с простым стейтом

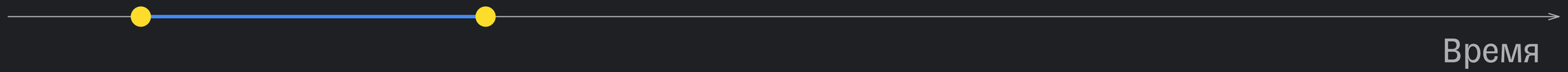
Собственный механизм сериализации с упором на перфоманс

Сериализация из коробки

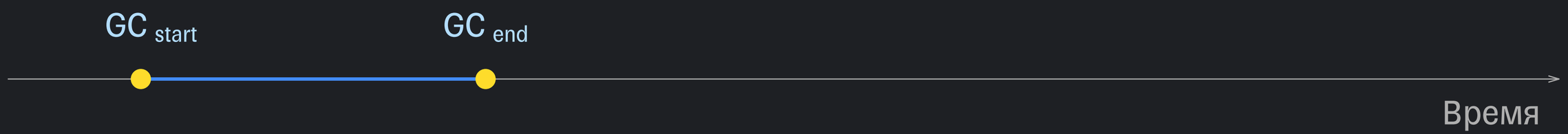
- Произвольный граф объектов
- Зависимости с простым стейтом
- Сериализует все, что попадет в билд

Собственный механизм сериализации с упором на перфоманс

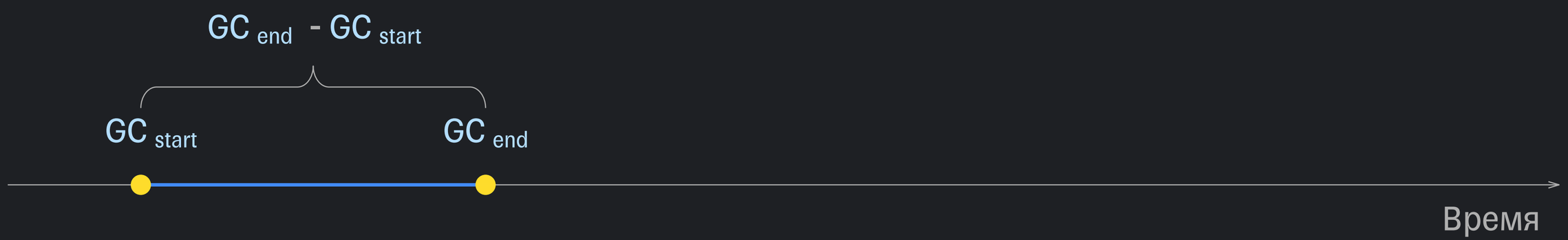
Вычисление времени работы GC



Вычисление времени работы GC



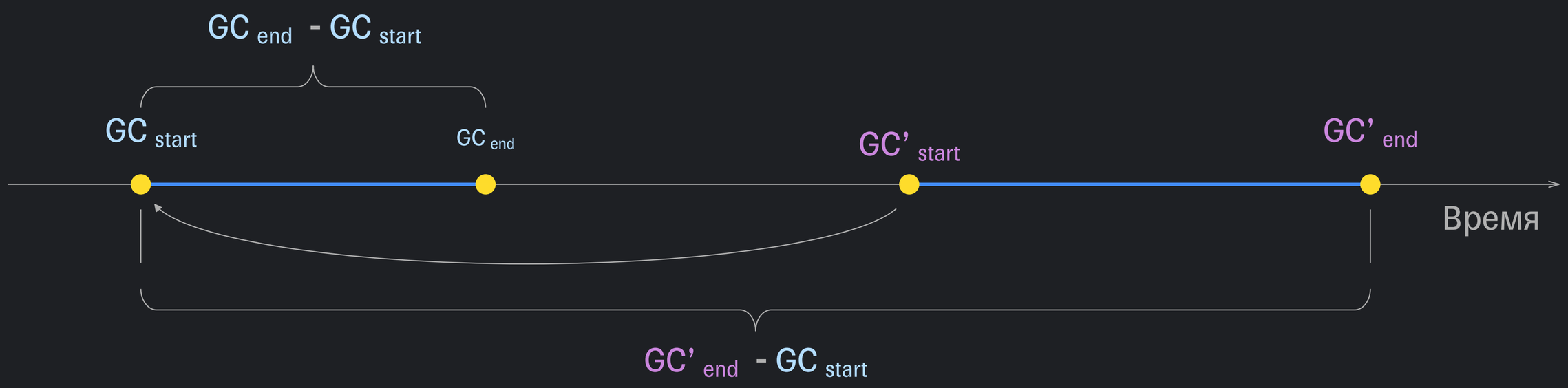
Вычисление времени работы GC



Вычисление времени работы GC

Вычисление времени работы GC

Вычисление времени работы GC



Требуется обойти кэширование

- Исключить сериализацию промежуточных результатов
- Придумать как получить новое начальное значение

Требуется обойти кэширование

- Исключить сериализацию промежуточных результатов
- Придумать как получить новое начальное значение
- Отсутствует механизм определения: был ли переиспользован кэш

Settings Plugin + Internal API

Первый запуск

```
override fun apply(target: Settings) {
    val extension = target.extensions.create<BuildMetricsExtension>(...)

    target.gradle.sharedServices.registerIfAbsent(...) {
        parameters.publishers.set(extension.publishers.list)
    }.also { serviceProvider ->
        ...

        target.gradle.settingsEvaluated {
            serviceProvider.get().setup(
                    publishers = extension.publishers.list
                )
        }
    }
}
```

Первый запуск

```
override fun apply(target: Settings) {
    val extension = target.extensions.create<BuildMetricsExtension>(...)

    target.gradle.sharedServices.registerIfAbsent(...) {
        parameters.publishers.set(extension.publishers.list)
    }.also { serviceProvider ->
        ...

        target.gradle.settingsEvaluated {
            serviceProvider.get().setup(
                    publishers = extension.publishers.list
                )
        }
    }
}
```

Первый запуск

```
override fun apply(target: Settings) {
    val extension = target.extensions.create<BuildMetricsExtension>(...)

    target.gradle.sharedServices.registerIfAbsent(...) {
        parameters.publishers.set(extension.publishers.list)
    }.also { serviceProvider ->
        ...

        target.gradle.settingsEvaluated {
            serviceProvider.get().setup(
                    publishers = extension.publishers.list
                )
        }
    }
}
```

Данные отсутствуют

Первый запуск

```
override fun apply(target: Settings) {
    val extension = target.extensions.create<BuildMetricsExtension>(...)

    target.gradle.sharedServices.registerIfAbsent(...) {
        parameters.publishers.set(extension.publishers.list)
    }.also { serviceProvider ->
        ...

        target.gradle.settingsEvaluated {
            serviceProvider.get().setup(
                    publishers = extension.publishers.list
                )
        }
    }
}
```

При попадании в кэш

```
override fun apply(target: Settings) {
    val extension = target.extensions.create<BuildMetricsExtension>(...)

    target.gradle.sharedServices.registerIfAbsent(...) {
        parameters.publishers.set(extension.publishers.list)
    }.also { serviceProvider ->
        ...

        target.gradle.settingsEvaluated {
            serviceProvider.get().setup(
                    publishers = extension.publishers.list
                )
        }
    }
}
```

Возьмет данные из кэша

При попадании в кэш

```
override fun apply(target: Settings) {
    val extension = target.extensions.create<BuildMetricsExtension>(...)

    target.gradle.sharedServices.registerIfAbsent(...) {
        parameters.publishers.set(extension.publishers.list)
    }.also { serviceProvider ->
        ...

        target.gradle.settingsEvaluated {
            serviceProvider.get().setup(
                    publishers = extension.publishers.list
                )
        }
    }
}
```

Не будет вызван

При попадании в кэш

```
abstract class BuildMetricsService :
    BuildService<BuildMetricsService.Params>,
    BuildOperationListener,
    AutoCloseable {

    init {
        ...
        reinitialize()
    }

    ...
}
```

«Обнулить» состояние

Incubating API

До версии 8.4

Получение признака “использование кэша конфигурации”

`Gradle.isConfigurationCacheRequested()`

Начиная с 8.5

Новый интерфейс

`BuildFeatures.getConfigurationCache()`

Начиная с 8.7

Получить размер данных сохраненных в кэш

Похоже на задачи Android разработчика

- Подключить плагин
- Реализовать обработку событий
- Реализовать паблишер

- История появления
- Особенности реализации
- **Использование данных**
- Кастомизация и применение

Собрать метрики – это еще половина дела

Извлечение полезной информации из данных – такое же увлекательное занятие.

Размеры артефактов

Размеры APK и AAB файлов

Размер файла в байтах

Размер APK сгенерированного из App Bundle

Рассчетные минимальный и максимальный размеры для скачивания

Количество методов в APK

Суммарное количество ссылок на методы в DEX файлах

Частота сбора данных

- Мерж реквесты
- Сборки релиз-кандидатов

Для отладочной и релизной сборок

Размеры APK файлов (мерж реквесты)

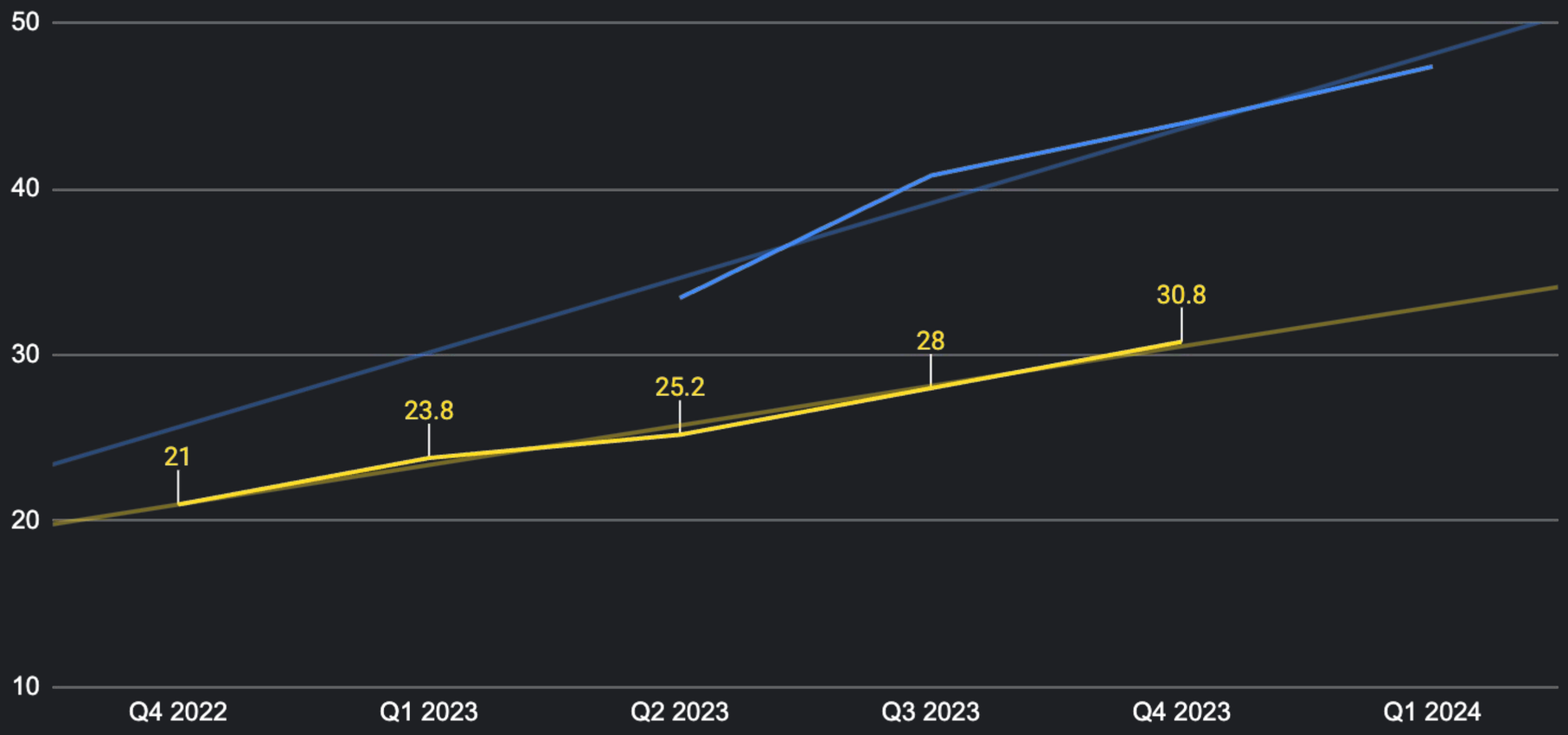
Количество ссылок на методы в DEX файлах

01 Отладочная сборка **12% (18%)**

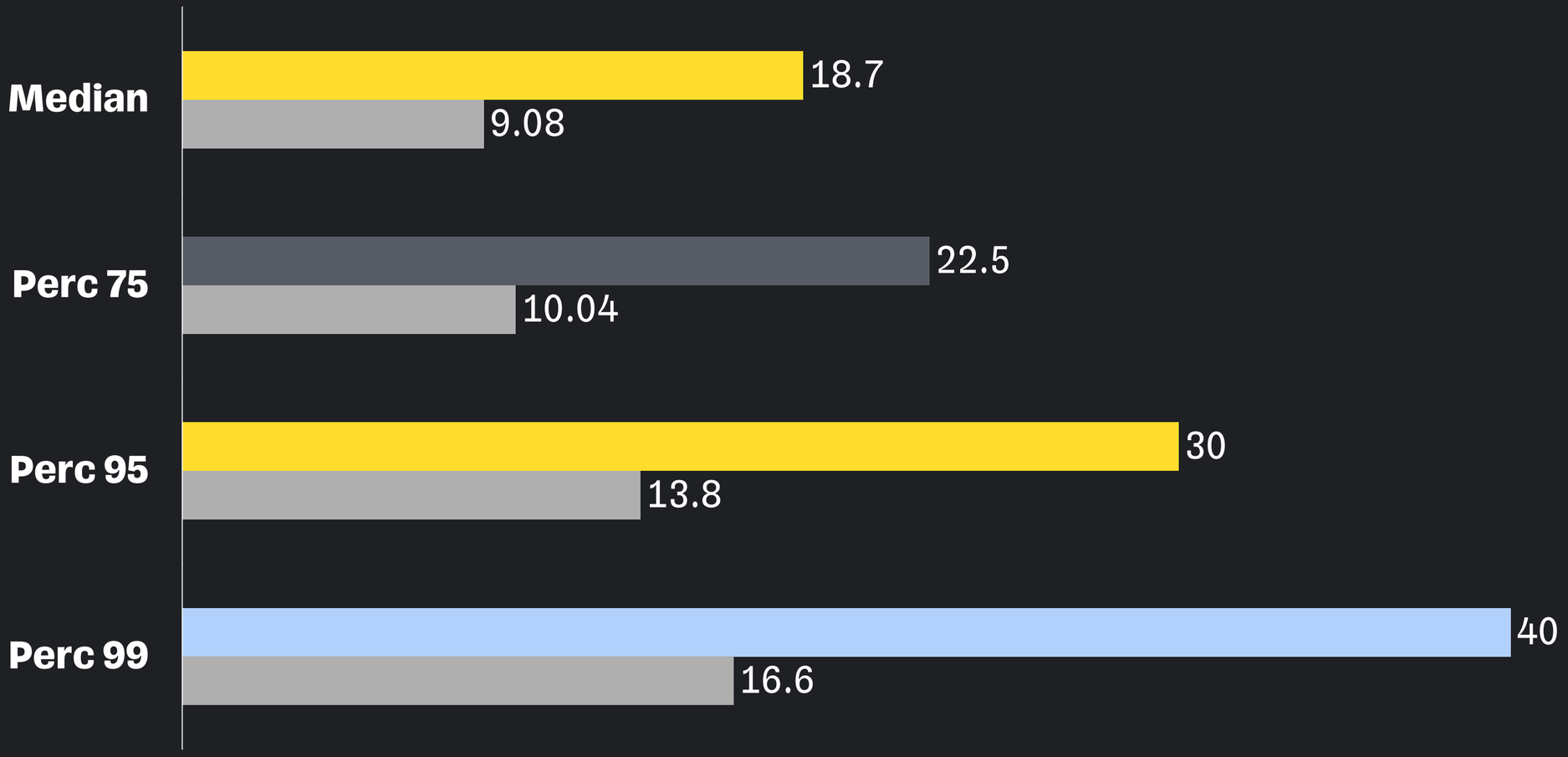
02 Релизная сборка **17% (22%)**

03 Соотношение размеров **50%**

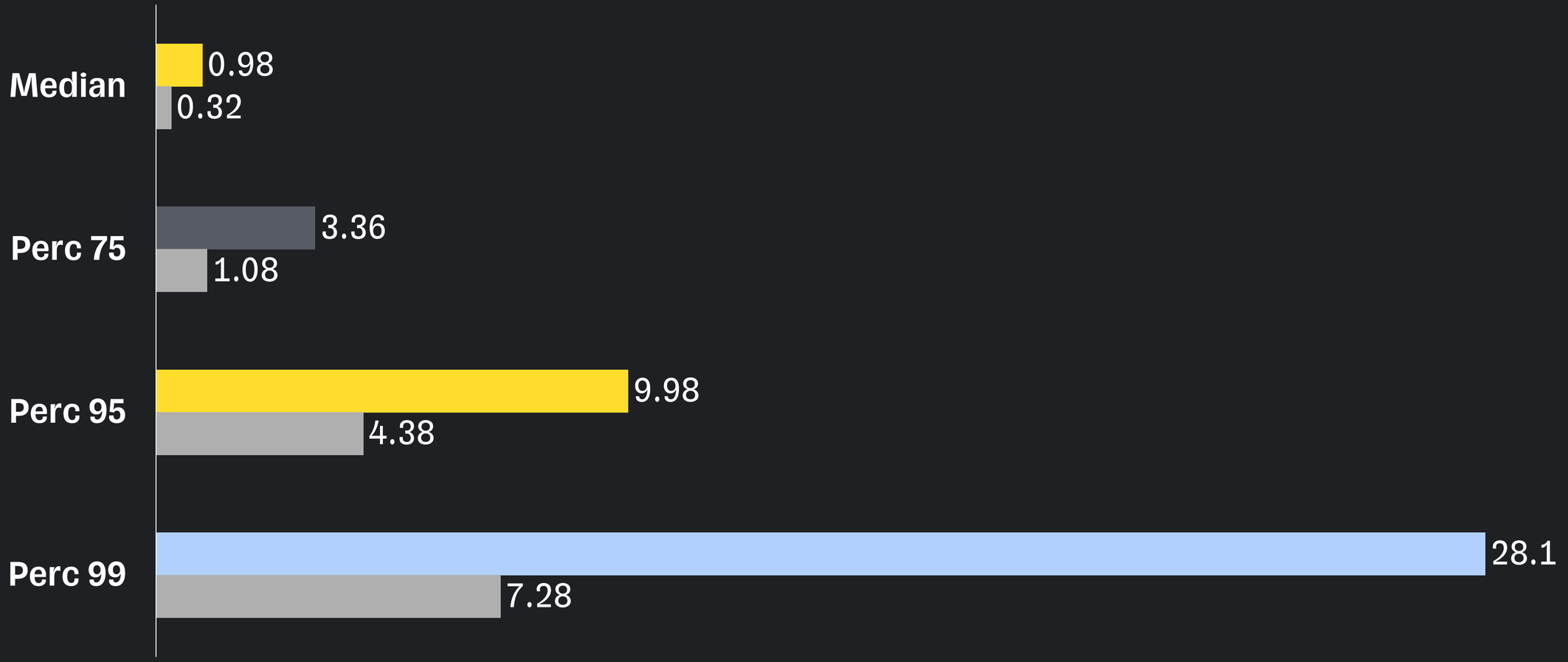
Размер RAM и количество ссылок в DEX



Время сборки и конфигурации Мобильного банка на CI * (в минутах)



Время сборки и конфигурации на машинах разработчиков (в минутах)



Различные условия запуска

Сборка на CI

Одинаковые задачи в одинаковом окружении

Локальные запуски

Зависят от тонны различных параметров

Время сборки локально по неделям

All tasks duration

Metric	0	diff	-1 week	diff	-2 week	diff	-3 week
Median	59.0 s	-16.6 s	1.26 min	28.6 s	47.1 s	-693 ms	47.8 s
Percentile 75	3.37 min	-16.0 s	3.64 min	40.3 s	2.96 min	-10.8 s	3.14 min
Percentile 90	6.76 min	-20.7 s	7.10 min	31.2 s	6.58 min	-13.9 s	6.82 min
Percentile 95	9.98 min	-39.5 s	10.6 min	12.4 s	10.4 min	-18.7 s	10.7 min
Percentile 99	28.1 min	-3.89 min	32.0 min	20.4 s	31.7 min	-10.9 s	31.8 min

Время сборки локально по неделям

All tasks duration

Metric	0	diff	-1 week	diff	-2 week	diff	-3 week
Median	59.0 s	-16.6 s	1.26 min	28.6 s	47.1 s	-693 ms	47.8 s
Percentile 75	3.37 min	-16.0 s	3.64 min	40.3 s	2.96 min	-10.8 s	3.14 min
Percentile 90	6.76 min	-20.7 s	7.10 min	31.2 s	6.58 min	-13.9 s	6.82 min
Percentile 95	9.98 min	-39.5 s	10.6 min	12.4 s	10.4 min	-18.7 s	10.7 min
Percentile 99	28.1 min	-3.89 min	32.0 min	20.4 s	31.7 min	-10.9 s	31.8 min

Время сборки локально по неделям

All tasks duration

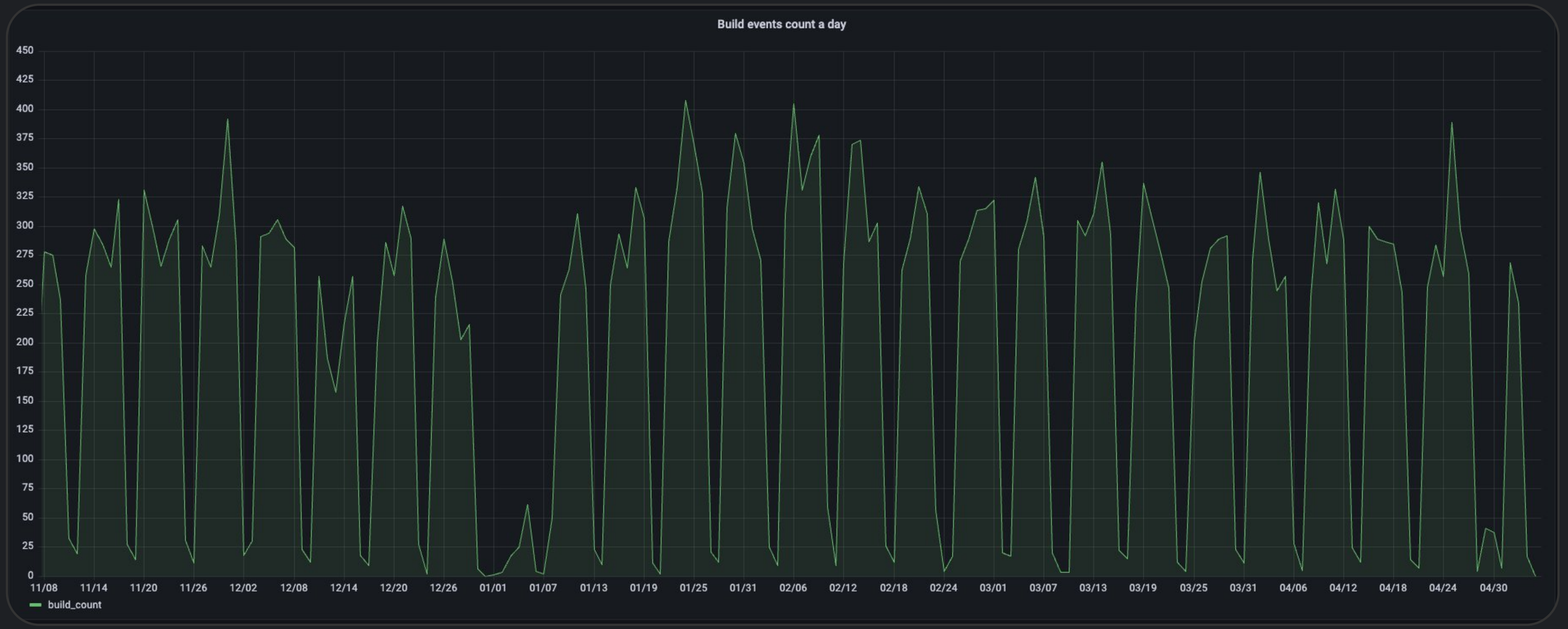
Metric	0	diff	-1 week	diff	-2 week	diff	-3 week
Median	59.0 s	-16.6 s	1.26 min	28.6 s	47.1 s	-693 ms	47.8 s
Percentile 75	3.37 min	-16.0 s	3.64 min	40.3 s	2.96 min	-10.8 s	3.14 min
Percentile 90	6.76 min	-20.7 s	7.10 min	31.2 s	6.58 min	-13.9 s	6.82 min
Percentile 95	9.98 min	-39.5 s	10.6 min	12.4 s	10.4 min	-18.7 s	10.7 min
Percentile 99	28.1 min	-3.89 min	32.0 min	20.4 s	31.7 min	-10.9 s	31.8 min

Время сборки локально по неделям

All tasks duration

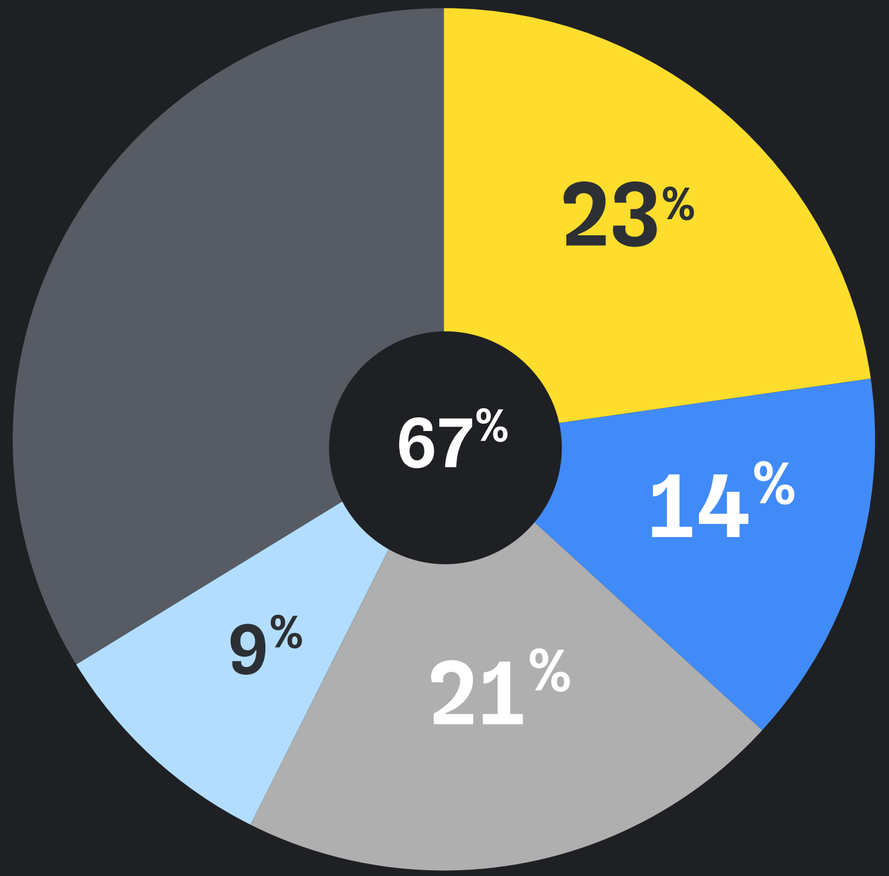
Metric	0	diff	-1 week	diff	-2 week	diff	-3 week
Median	59.0 s	-16.6 s	1.26 min	28.6 s	47.1 s	-693 ms	47.8 s
Percentile 75	3.37 min	-16.0 s	3.64 min	40.3 s	2.96 min	-10.8 s	3.14 min
Percentile 90	6.76 min	-20.7 s	7.10 min	31.2 s	6.58 min	-13.9 s	6.82 min
Percentile 95	9.98 min	-39.5 s	10.6 min	12.4 s	10.4 min	-18.7 s	10.7 min
Percentile 99	28.1 min	-3.89 min	32.0 min	20.4 s	31.7 min	-10.9 s	31.8 min

Количество сборок на CI по дням



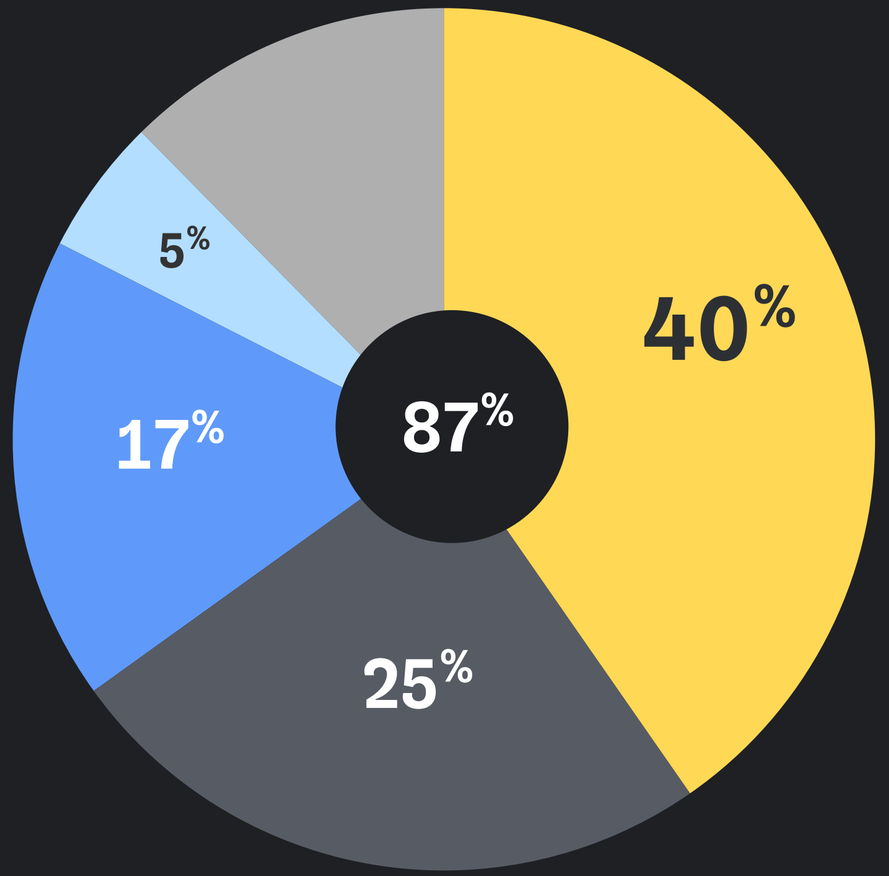
Доля запусков Gradle тасок в общем количестве

- Assemble 23%
- Sync with IDE 14%
- Unit test 21%
- Detekt * 9%



Доля Gradle тасок в общей длительности

- Assemble 40%
- Sync with IDE 25%
- UI test 17%
- Unit test 5%



Длительность всех тасок против синка (в минутах)

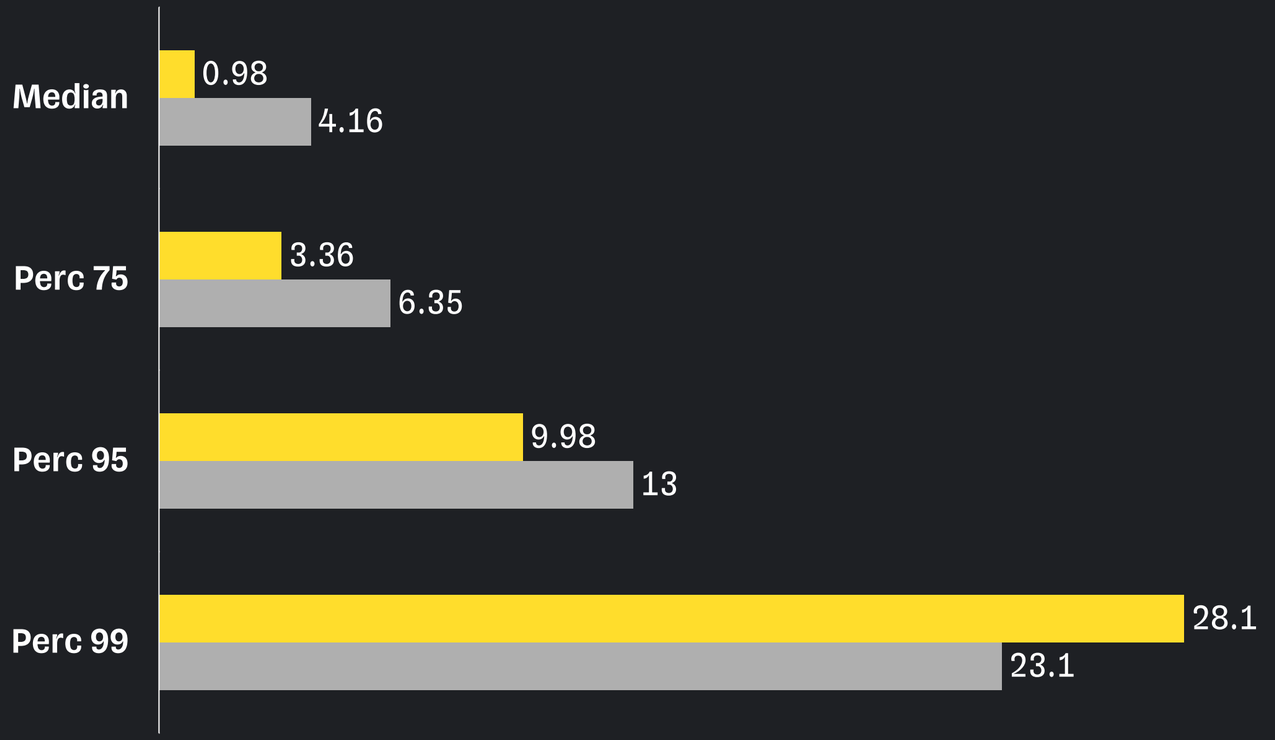
Время синхронизации с IDE

Выделено серым

01 Зависит от количества модулей

02 Не поддерживает кэш конфигурации

03 Влияет скачивание зависимостей (~20%)



Сокращение времени синка с IDE

- Ждать ~~милости~~ оптимизаций от вендоров

Сокращение времени синка с IDE

- Ждать ~~милости~~ оптимизаций от вендоров
- Импортировать часть репозитория
- Отключить часть модулей
- Индексировать только интерфейсы

Меньше кода – быстрее синк

Инструменты уменьшения скоупа

Уменьшить количество модулей

Указать «корневой» модуль. И подключить к проекту только данный модуль и все, от которых он зависит. Учитывая явные и транзитивные зависимости.

- Focus
- Tinkoff Include Subprojects Plugin

Эффективность зависит от связности модулей

Препуш проверки: Detekt * время сборки и конфигурации (в секундах)

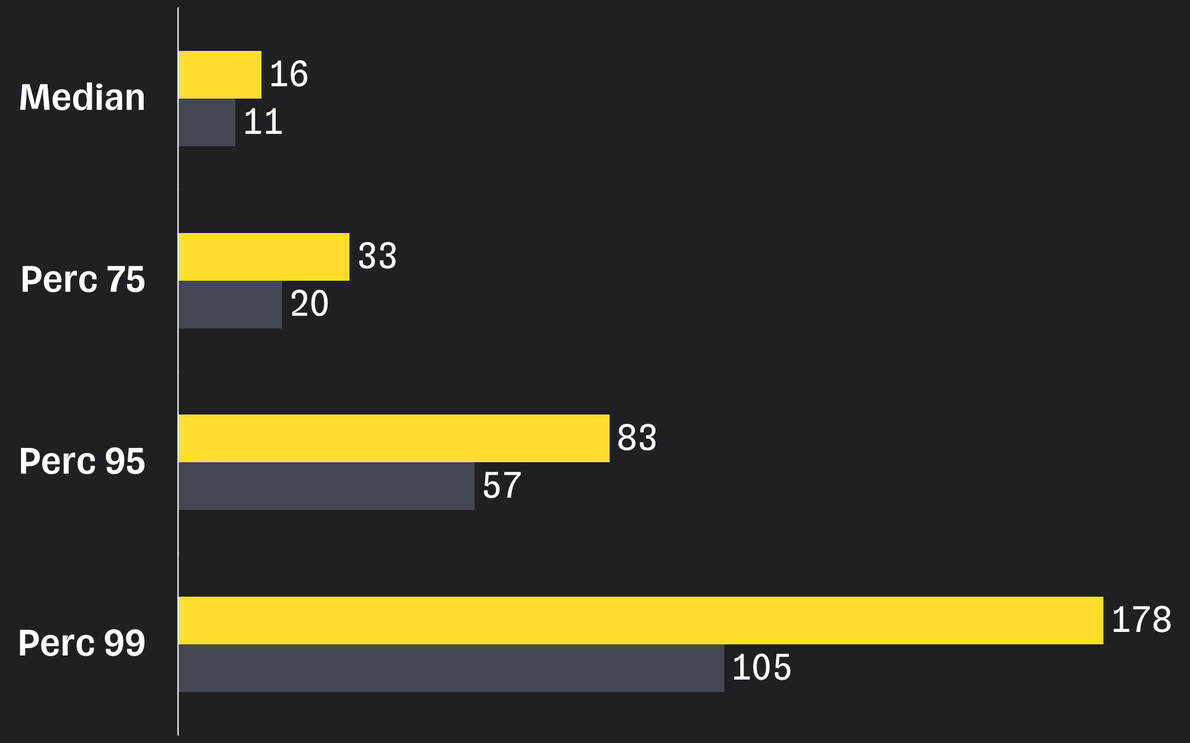
Запуск тасок detekt на измененных модулях

Из локального Git хука pre-push

01 Получение изменений из Git

02 Билд кэш

03 Configuration On Demand



В чем разница между запуском тасок?

help

:help

В чем разница между запуском тасок?

help

1 минута 25 секунд

:help

12 секунд

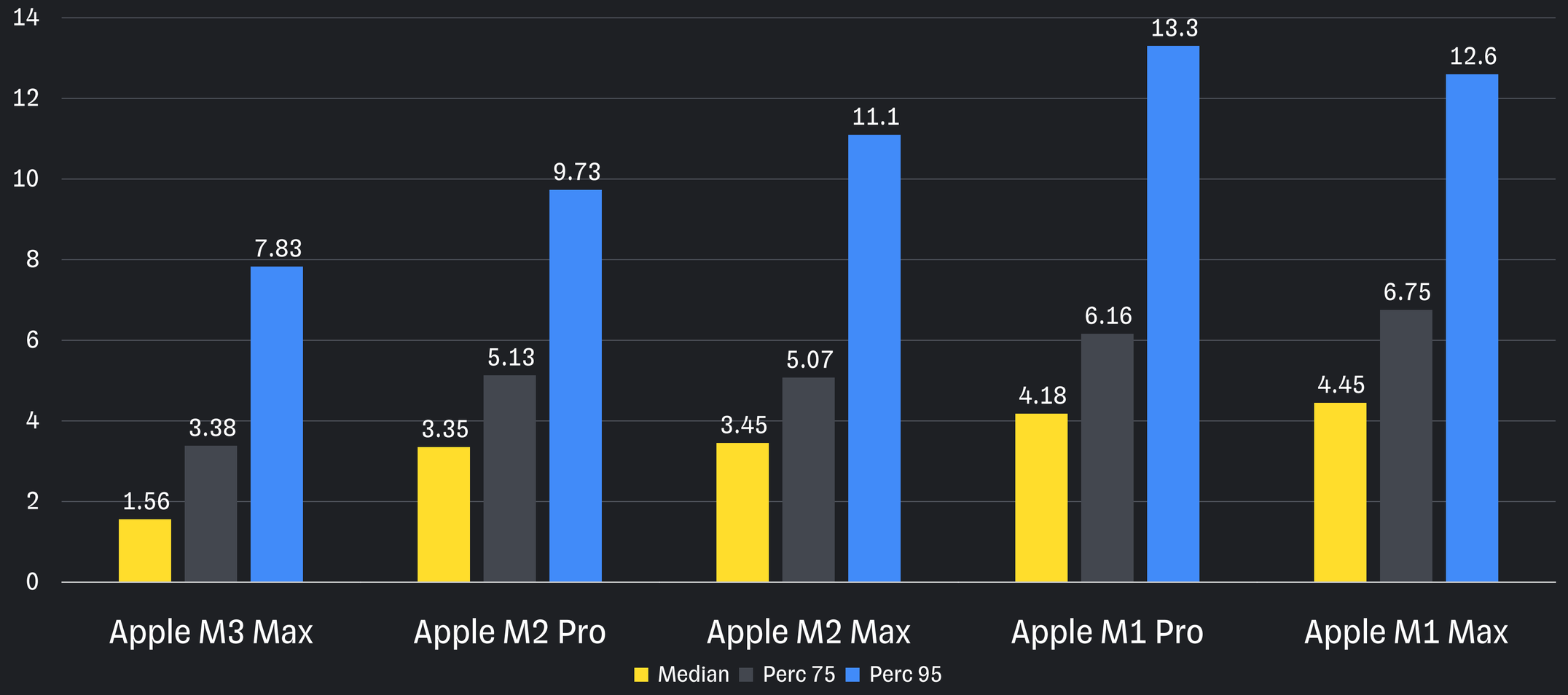
Configuration On Demand

- Конфигрирует только нужные модули
- Учитывает связи между модулями
- Поддерживает зависимости между тасками

Configuration On Demand

- Ограничивает получение списка тасок
- Фича в статусе Incubating

Синхронизация с IDE на разном «железе»



Синхронизация с IDE на разном «железе»

IDE sync duration by CPU

CPU	Users count	Median	Perc 75	Perc 95	Perc 99
Apple M2 Pro	8	3.35 min	5.13 min	9.73 min	16.8 min
Apple M2 Max	82	3.46 min	5.09 min	11.1 min	18.2 min
Apple M1 Max	88	4.45 min	6.75 min	12.6 min	20.6 min
Apple M1 Pro	131	4.18 min	6.16 min	13.3 min	22.3 min
Apple M2	2	10.7 min	15.6 min	28.2 min	32.2 min
AMD Ryzen 5 5500U with Radeon Graphics	1	32.3 min	35.3 min	1.14 hour	1.14 hour

1 - 6 of 22 rows

Тонны разных параметров

На время сборки влияет множество параметров

Оказывают влияние на всех

Частный кейс отдельного сотрудника

- История появления
- Особенности реализации
- Использование данных
- Кастомизация и применение

Как применить решение в вашем проекте

Две изолированные задачи

Сбор данных

Gradle плагин собирает данные. При завершении сборки отправляет в аналитическую систему.

Анализ информации

Построение отчетов. Выдвижение гипотез. Поиск ответов, опровержения или подтверждения.

Группы параметров

Gradle

Информация, которую можно получить из Gradle.
Без привязки к платформе.

Android

Специфичные параметры для Android

System

Данные о “железе”, на котором запустили сборку.

Gitlab

Дополнительные атрибуты из Гитлаб
для построения отчетов

Кастомизация собираемой информации

Добавить собственные параметры

В виде расширения к плагину
или непосредственно в настройках проекта.

Отключить лишнее

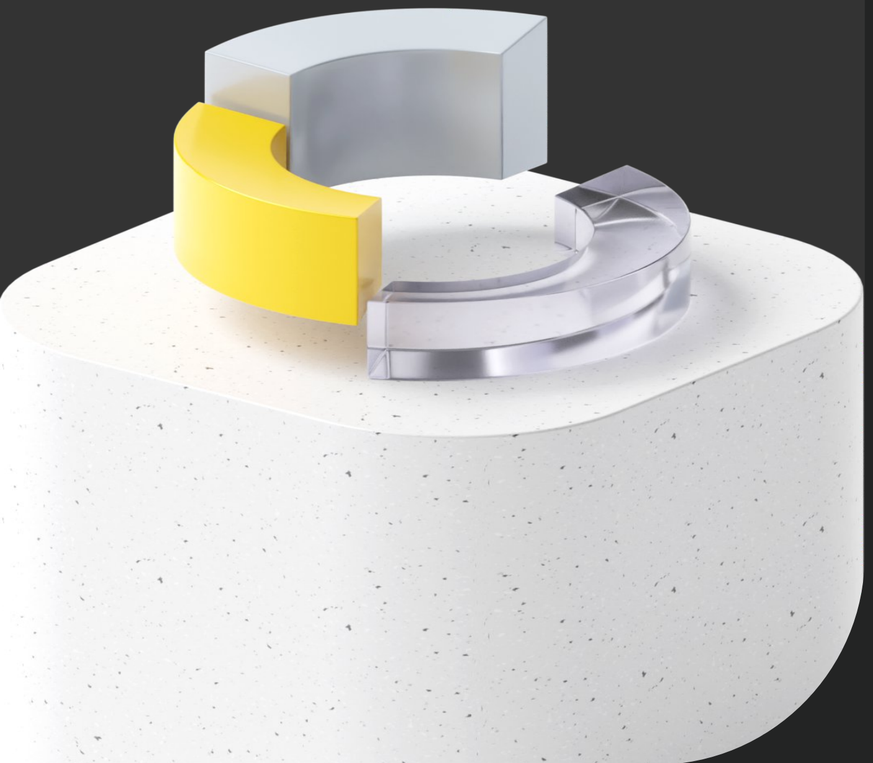
Оставить только нужное. Например,
избавиться от параметров Gitlab.

Экспорт и работа с данными

- Собственный экспортер данных
- Работа с данными зависит от инструмента

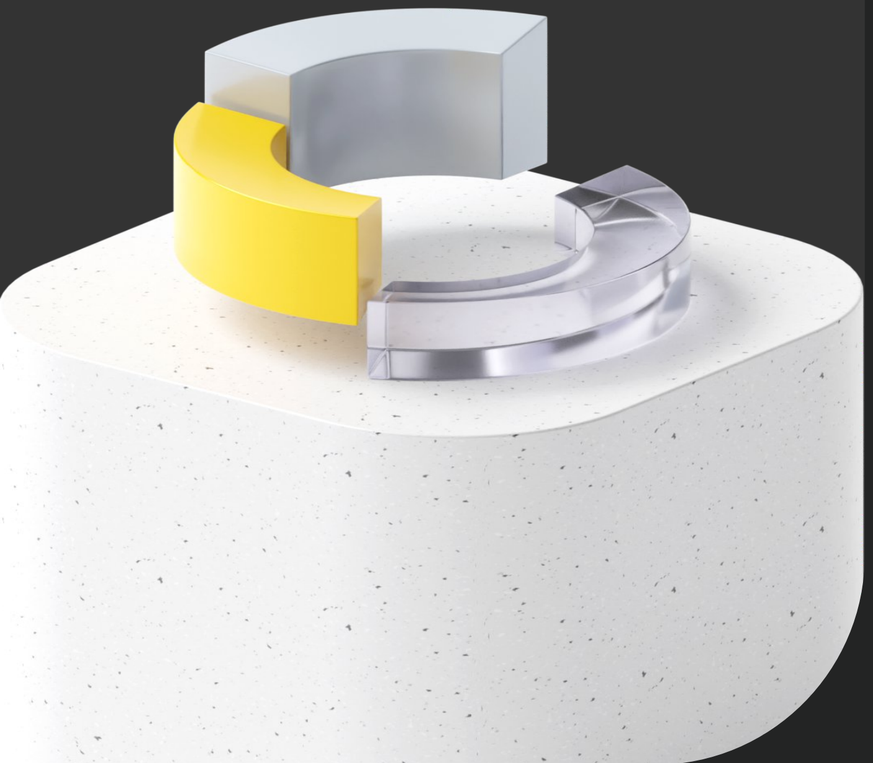
Частная история. Зависит от используемого инструмента.

Инструменты для анализа



- ClickHouse
- Elasticsearch
- СУБД: MySQL, Postgres, ...

Инструменты для анализа



- ClickHouse
- Elasticsearch
- СУБД: MySQL, Postgres, ...

Возможности зависят от инструмента

Исходники и документация

- **Self-hosted Gitlab**

  https://opensource.tinkoff.ru/

- **Tinkoff Build Metrics Plugin**

  https://opensource.tinkoff.ru/tinkoff-mobile-tech/build-metrics-plugin

- **Tinkoff Include Subprojects Plugin**

  https://opensource.tinkoff.ru/tinkoff-mobile-tech/include-subprojects-plugin

Заключение

- Причины появления инструмента
- Особенности реализации
- Есть метрики за рамками сборки
- Собрать данные – половина дела

Мы сделали инструмент для себя и рады поделиться с сообществом

ТИНЬКОФФ

Спасибо!