



# Нетипичные подходы к перфу в поисковом движке Ozon

**Пётр Портнов**

Старший разработчик

[pportnov@ozon.ru](mailto:pportnov@ozon.ru)





Привет!

Я Петя  
Портнов —

старший разработчик  
в команде базового поиска



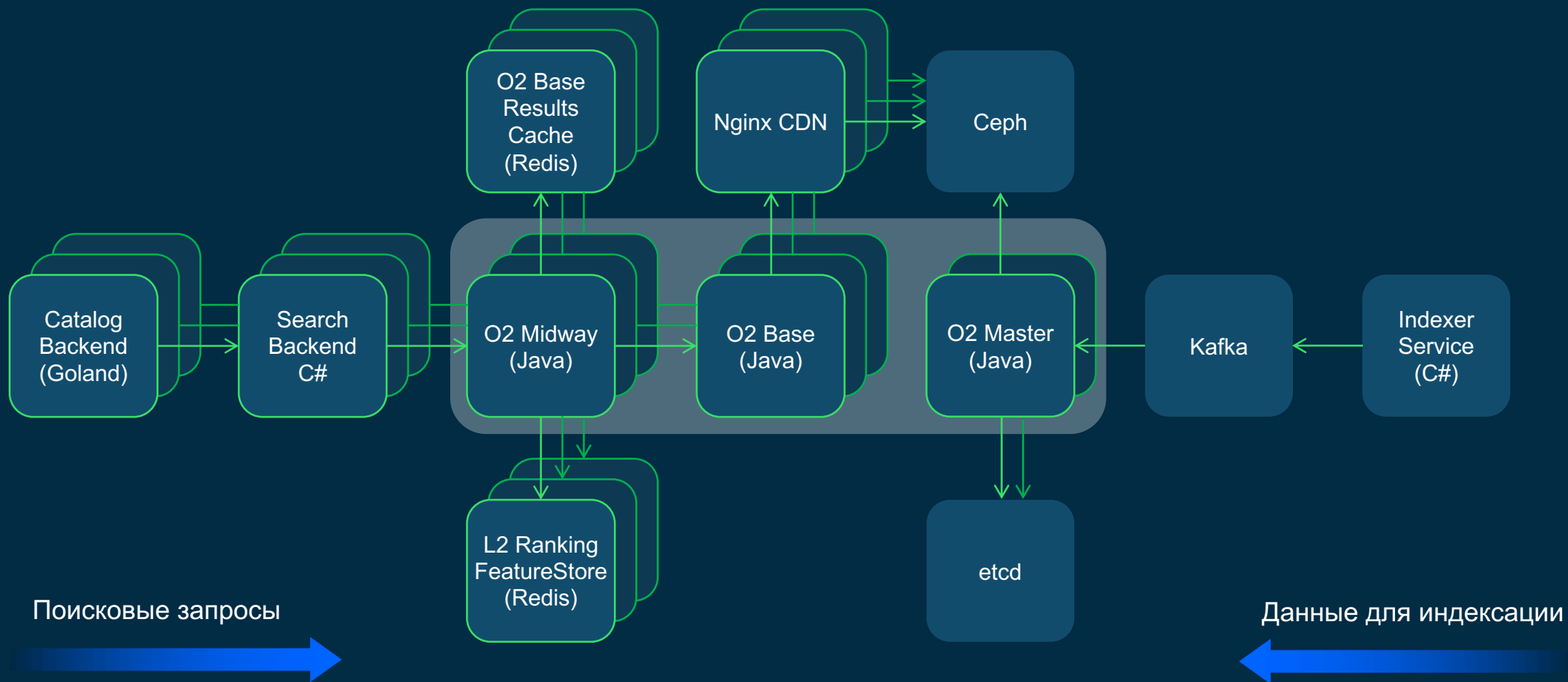
07:07 



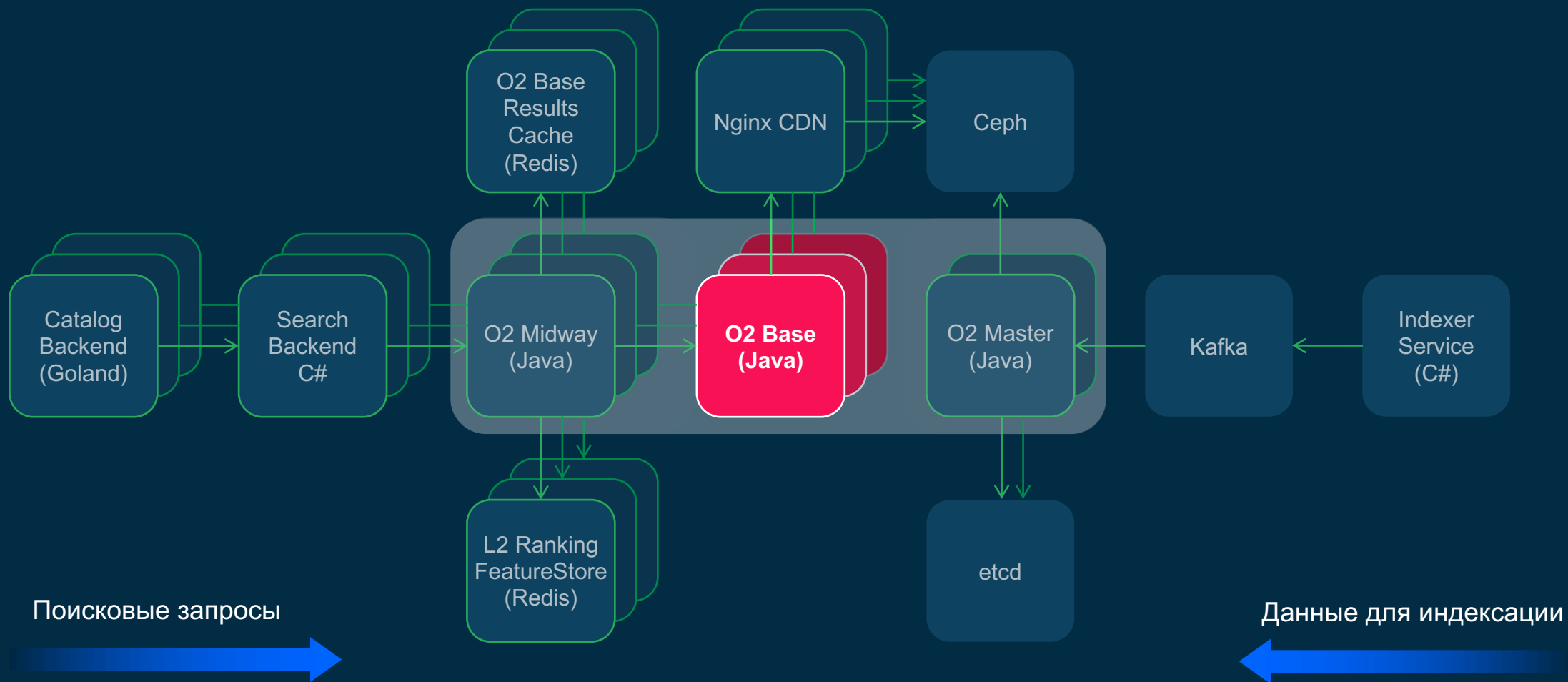
# Введение в O2



# Архитектура поиска в Ozon



# Архитектура поиска в Ozon



Подробнее об  
устройстве поиска —  
в статье на Хабре



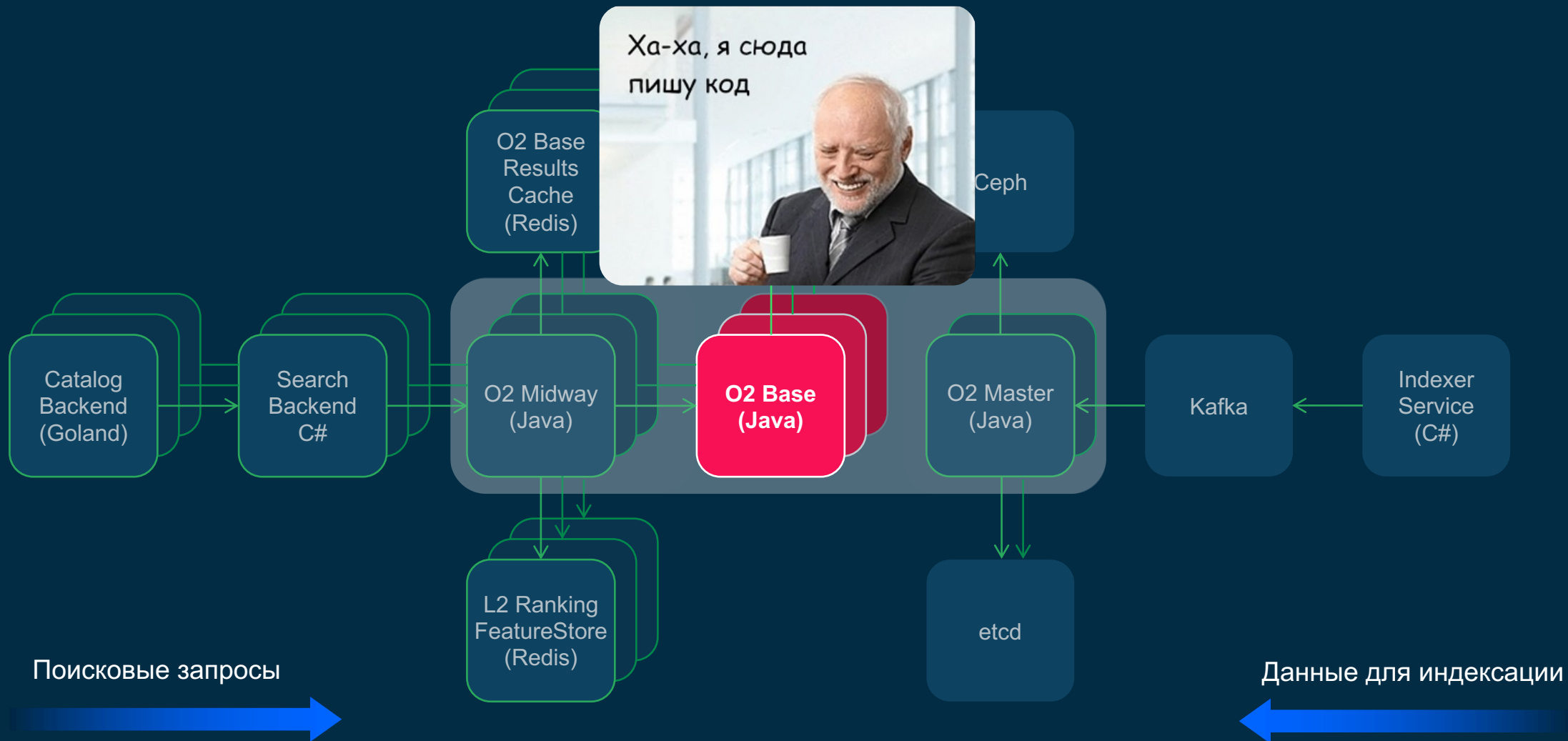
ozon{tech

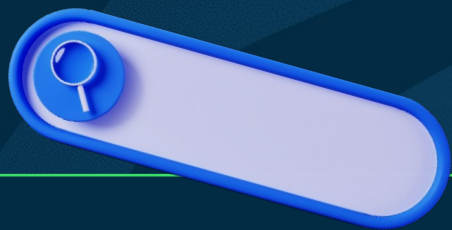


**Сергей Саиян**

Руководитель департамента  
"Поиск, Рекомендации, Реклама"

# Архитектура поиска в Ozon





Базовый поиск  
выполняет  
базовый поиск!



По запросу трава для кота найден 1591 товар

Категория

Дом и сад  
Товары для животных

Популярные

Не упустите распродажу Осталось 13 дней

Распродажа



Сроки доставки

- Неважно
- От 1 часа
- Сегодня
- Завтра
- До 3 дней
- До 7 дней

Бренды

- Деревенские лакомства
- Titbit
- Triol
- Monge
- Гавриш
- Аэлита
- ПОИСК Агрохолдинг

[Посмотреть все](#)

Цена

от 77 до 12 541

- до 500 Р
- 500–2 000 Р
- 2 000 Р и дороже
- Неважно

Товары со скидкой



Высокий рейтинг



Рассрочка 0%



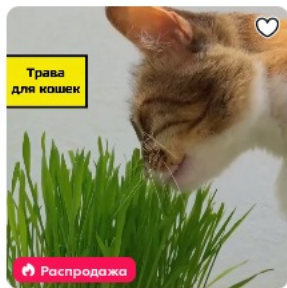
**196 Р** 370 Р -47%  
Осталось 1748 шт  
Трава для кошек в лотке, лакомство, витамины, вкусняшки,...

★ 4.6 ● 2 223 отзыва  
[Завтра](#)



**195 Р** 350 Р -44%  
Осталось 369 шт  
Травка для кошек, набор для посадки, лакомство для кошек и...

★ 4.8 ● 131 отзыв  
[Завтра](#)



**108 Р** 275 Р -61%  
Осталось 2 692 шт  
Трава для кошек, витамины, лакомство, добавка в корм,...

★ 4.7 ● 7161 отзыв  
[Завтра](#)



**196 Р** 290 Р -32%  
Осталось 899 шт  
Травка для кошек в лотке, 3 знака

★ 4.6 ● 4 041 отзыв  
[Завтра](#)



**195 Р** 499 Р -61%  
Triol  
Травка для кошек в лотке Триол, 120 гр. для проращивания, трава...

★ 4.8 ● 13 643 отзыва  
[Завтра](#)



**139 Р** 399 Р -65%  
Осталось 317 шт  
Травка для кошек, набор для проращивания на 3 посадки,...

★ 4.7 ● 3 975 отзывов  
[Завтра](#)



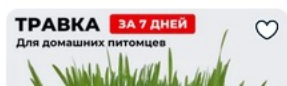
**159 Р** 400 Р -60%  
Осталось 1 633 шт  
Травка для кошек в лотке, 120 гр., 3 знака

★ 4.7 ● 9 293 отзыва  
[Завтра](#)



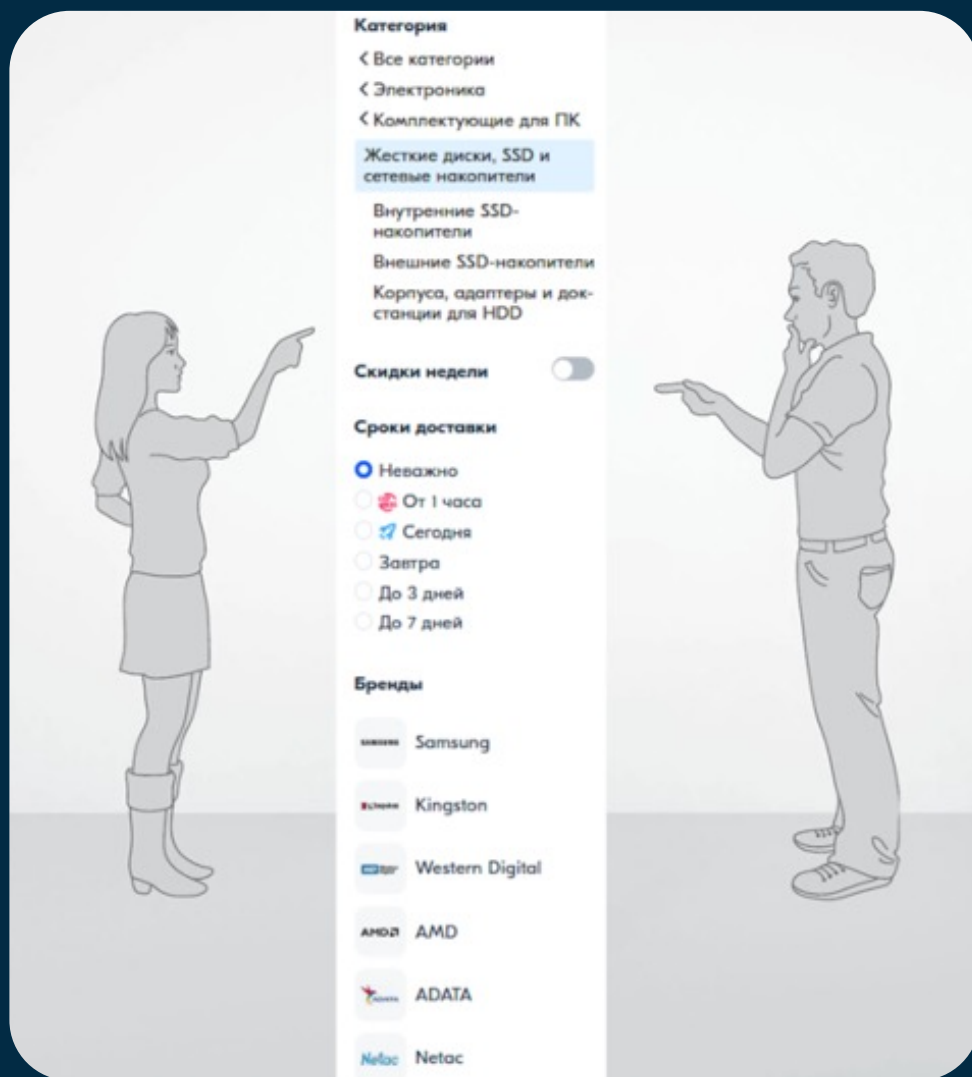
**122 Р** 430 Р -72%  
Осталось 383 шт  
Трава для кошек. Травка в лотке без земли и грязного дома

★ 4.7 ● 3 975 отзывов  
[Завтра](#)



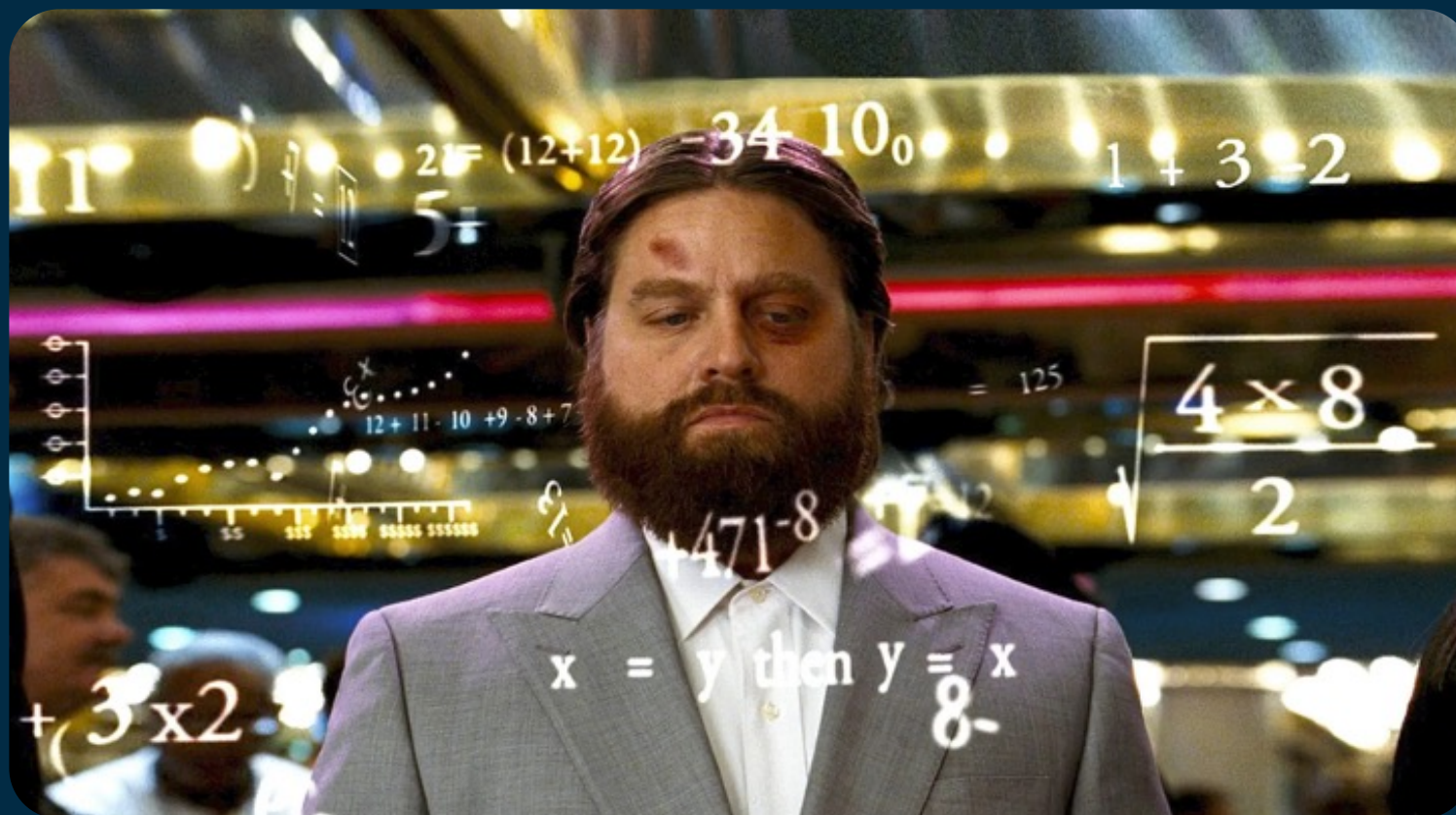
# Базовый поиск

Выдаёт фасеты



# Базовый поиск

Считает количество страничек





Базовый  
поиск выдаёт  
хайлайты

**ВСЁ В ДОМ!**  
распродажа

**13 дней**  
до конца

На страницу акции >

**ТОП-ВЫГОДА >**

<p><b>Трой</b> Коптильня</p> <p>1 ед. в руки</p> <p><del>2 299 Р</del> -70% <b>709 Р</b> 1145 шт.</p>	<p><b>Glider</b> 57.4x98.5x92.8 см</p> <p>1 ед. в руки</p> <p><del>23 024 Р</del> -77% <b>5 420 Р</b> 454 шт.</p>	<p><b>appetite</b> 2 и 3 л</p> <p>1 ед. в руки</p> <p><del>2 299 Р</del> -50% <b>1 159 Р</b> 3012 шт.</p>
---	---	---

**Все товары распродажи**

до 80%

**%**

Смотреть

ПРИМЕНИЛИ СКИДКУ

**-15%**  
ПО OZON КАРТЕ  
НА ОДЕЖДУ И ОБУВЬ

**деньги на карту**  
первый платёж только через месяц

Подробнее

до **300 ТЫС Р**

**Специальные предложения!**

Мощная  
посудомоечная  
машина

Паровый  
эффект  
300°

Керамическая  
и эмалевая  
подложка

**11 843 Р** ~~24 990 Р~~ -53%

Philips Парогенератор GC6740/30, фиолетовый

УЛЬТРАЛЕГКИЙ КРЕМ-ЭССЕНЦИЯ SPF 50+

ЗАЩИТА И УВЛАЖНЕНИЕ НА ВСЁ ДЕНЬ

+ ПОДАРОК

**2 143 Р** ~~4 100 Р~~ -48%

d'Alba Ультралёгкий солнцезащитный крем для...

СУХОЙ КОРМ СО СКОПОМ

PERFECT FIT

**3 882 Р** ~~6 610 Р~~ -41%

Сухой корм Perfect Fit™ для собак средних и крупных...

в ПОДАРОК

**109 990 Р** ~~162 990 Р~~ -28%

Honor Смартфон Magic6 Pro Ростекст (ЕАС) 12/512 Г...

Годовой ЗАПАС

Gillette Fusion ProGlide

сменные кассеты для...

**3 878 Р** ~~6 199 Р~~ -37%

Мощный пар

Напольная гладка SteamBoard

Защита от перегрева

**13 208 Р** ~~21 990 Р~~ -40%

Вертикальный отпариватель Philips...

DOMLEGRAND

**ИМПЕРИЯ**

РУЛОННАЯ ШТОРА

KUPPERSBERG

**СКИДКИ** ~50%  
на бытовую технику

Orbit

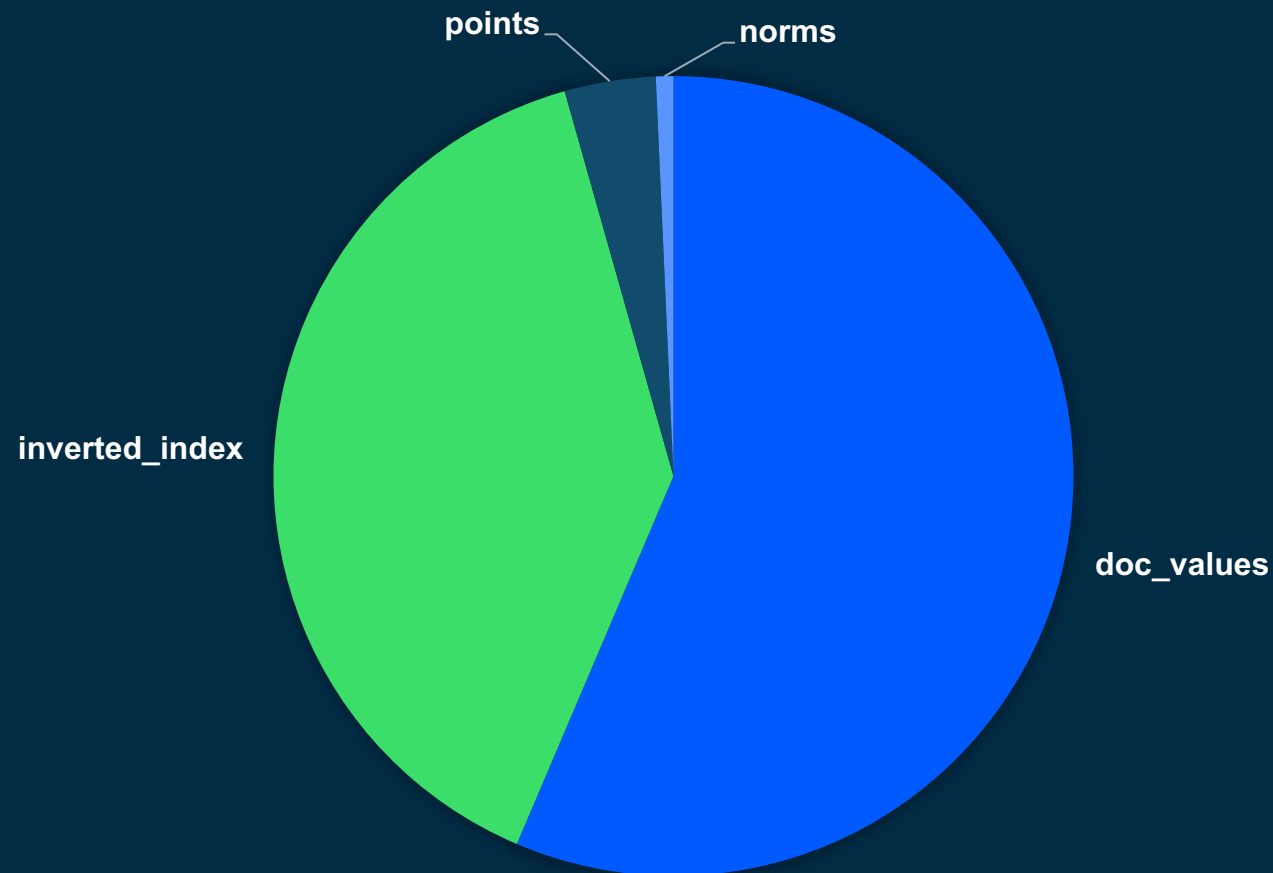
НАКРУТИЛ СЕБЯ?

**Жуй Выдыхай**

# Наши объёмы

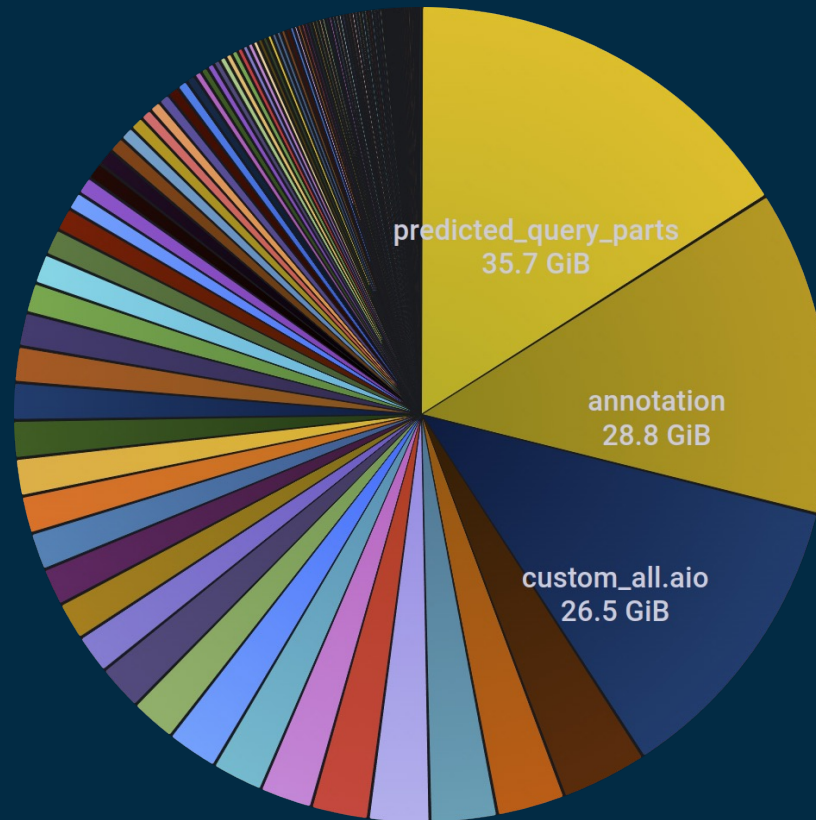
Крупнейший индекс — более 500 Гб

## РАЗМЕР КОМПОНЕНТОВ ИНДЕКСА, ГbB



# Наши объёмы

Порядка 10000 полей в основных индексах



# Наши объёмы

Почти 600 stateful подов, 3 ЦОДа

```
$ kubectl --namespace osrch --context o-prod get pods |\n  grep o2-base-shard |\n  wc -l
```

594



# Наши объёмы

Десятки тысяч RPS

>65 000 RPS

~22 000 RPS



Наши требования  
к производительности

# Регулярные стрельбы

По всему поиску



overloado3 BOT

Завершился тест в коллекции searchteam search

<https://overload.o3.ru/job?id=3117297>

status: Success ●

throughput: 70164

throughput target: 70000

start: 2024-04-06

stop: 2024-04-06

duration: 40m5s

description: Нагрузочный тест Search api



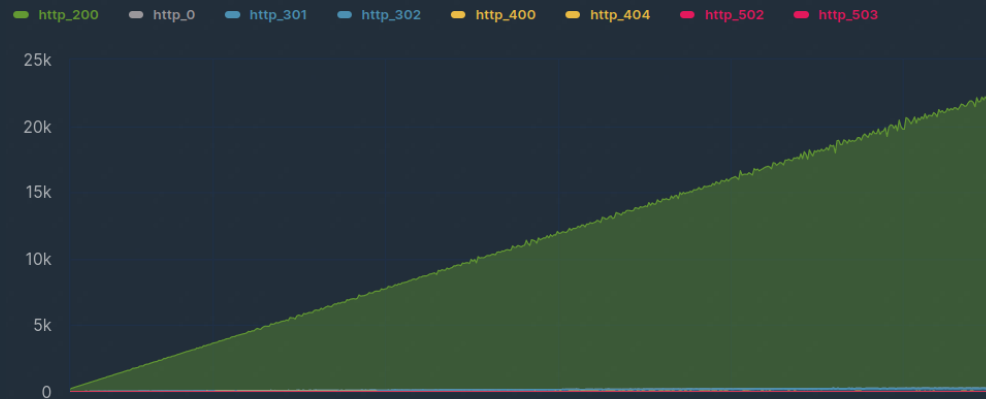
4



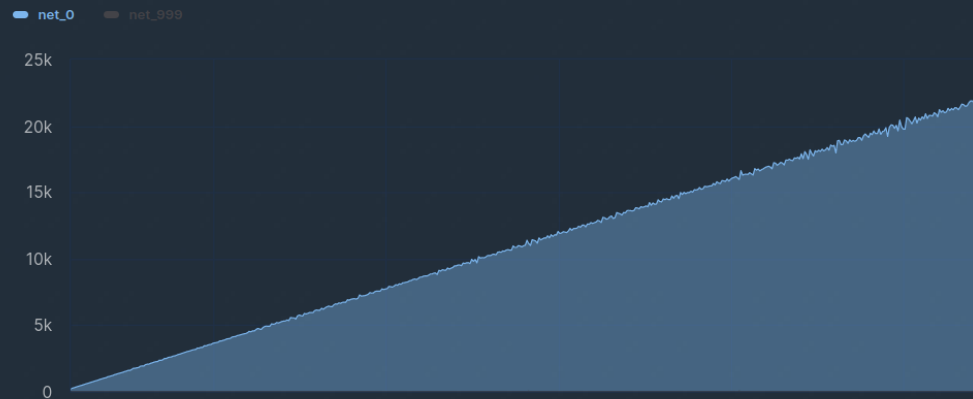
1 reply | Follow

Overall

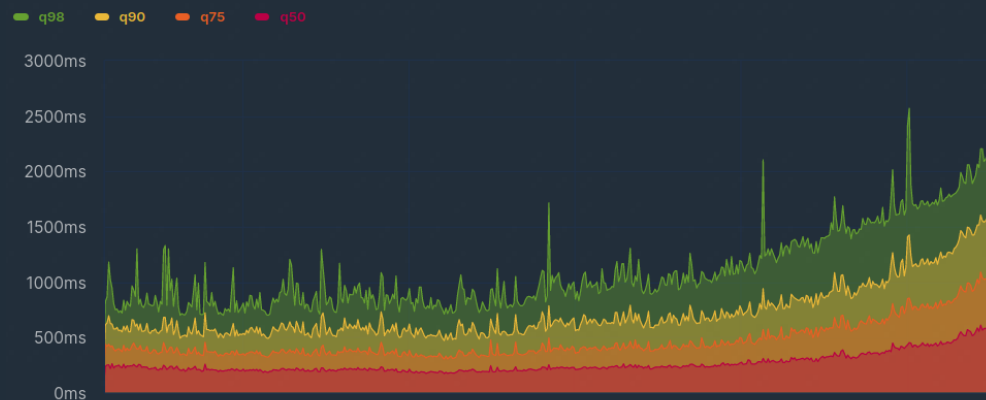
### RPS



### Net codes



### Quantiles



### Threads



# Регулярные стрельбы

По базовому поиску

Добавить конкретный график...

Yesterday



o2\_nightwatch ВОТ 04:42



Стрельбы базового поиска (master)

Доступны свежие результаты тестов:

Название теста	Начало	Окончание	Ссылка на результат
Mixed: 1 to 1000 RPS during 15m	2024-04-10T00:35:04Z	2024-04-10T00:40:18Z	<a href="https://overload.o3.ru/job?id=3136739">https://overload.o3.ru/job?id=3136739</a>
Mixed: 120 RPS during 10m	2024-04-10T00:40:21Z	2024-04-10T00:52:34Z	<a href="https://overload.o3.ru/job?id=3136770">https://overload.o3.ru/job?id=3136770</a>
Cardinality: 100 RPS during 5m	2024-04-10T00:52:39Z	2024-04-10T00:59:14Z	<a href="https://overload.o3.ru/job?id=3136809">https://overload.o3.ru/job?id=3136809</a>
Facets: 50 RPS during 5m	2024-04-10T00:59:18Z	2024-04-10T01:06:09Z	<a href="https://overload.o3.ru/job?id=3136817">https://overload.o3.ru/job?id=3136817</a>
CustomSearch (cardinality (0; 10k]): stepped load up to 100 RPS	2024-04-10T01:06:13Z	2024-04-10T01:19:23Z	<a href="https://overload.o3.ru/job?id=3137953">https://overload.o3.ru/job?id=3137953</a>
CustomSearch (cardinality (10k; 30k]): stepped load up to 100 RPS	2024-04-10T01:19:27Z	2024-04-10T01:32:30Z	<a href="https://overload.o3.ru/job?id=3137988">https://overload.o3.ru/job?id=3137988</a>
CustomSearch (cardinality (30k; 60k]): stepped load up to 50 RPS	2024-04-10T01:32:34Z	2024-04-10T01:41:11Z	<a href="https://overload.o3.ru/job?id=3138165">https://overload.o3.ru/job?id=3138165</a>
CustomSearch (cardinality (60k; 100k]): stepped load up to 50 RPS	2024-04-10T01:41:15Z	2024-04-10T01:42:11Z	<a href="https://overload.o3.ru/job?id=3138228">https://overload.o3.ru/job?id=3138228</a>
CustomSearch (cardinality (100k; Inf]): stepped load up to 50 RPS	2024-04-10T01:42:15Z	2024-04-10T01:42:31Z	<a href="https://overload.o3.ru/job?id=3138232">https://overload.o3.ru/job?id=3138232</a>

## Graphs

Save comparison

Tags

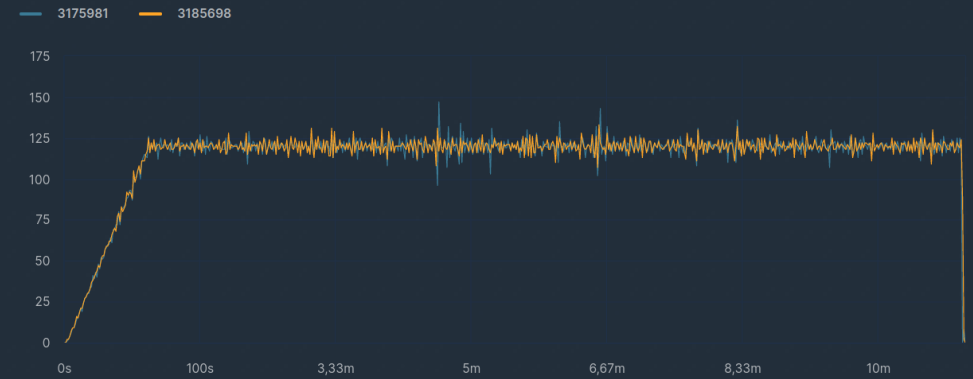
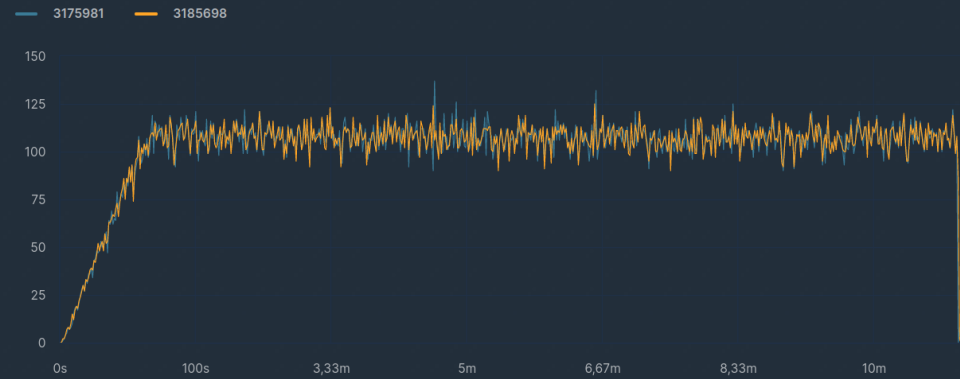
Overall

RPS

http\_200

Net codes

net\_0

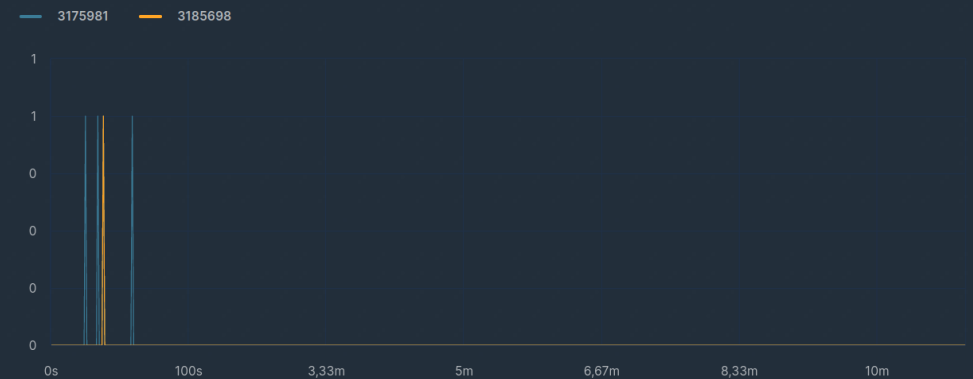
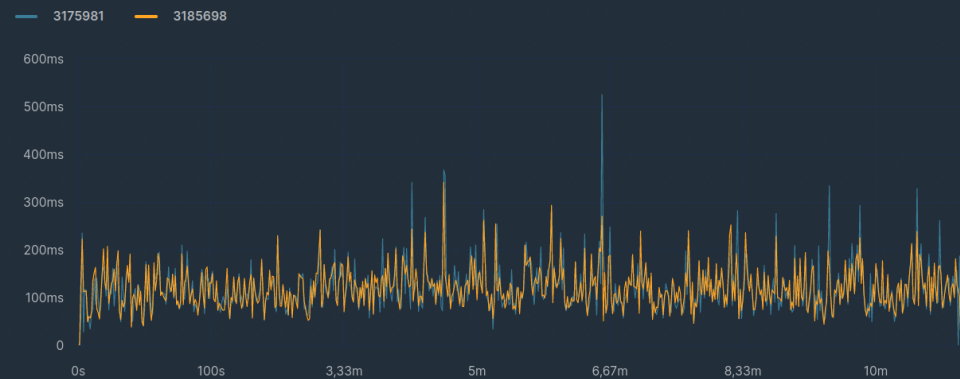


Quantiles

q90

Threads

threads



Узнай больше о  
подходах к улучшению  
производительности  
в базовом поиске



ozon{tech



**Александр Меньшиков**  
Ведущий разработчик команды  
"Базовый поиск"



Нетипичные пути  
к улучшению перфа



01



Виртуальные  
ВЫЗОВЫ

# Query в Lucene

Описывает логику поиска

○○○

```
1 public abstract class Query {
2     public abstract Weight createWeight(
3         IndexSearcher searcher,
4         ScoreMode scoreMode,
5         float boost
6     ) throws IOException;
7
8     public abstract Query rewrite(
9         IndexSearcher indexSearcher
10    ) throws IOException;
11
12    public abstract void visit(
13        QueryVisitor visitor
14    );
15 }
```

# Query в Lucene

Описывает логику поиска

○○○

```
1 public abstract class Query {
2     public abstract Weight createWeight(
3         IndexSearcher searcher,
4         ScoreMode scoreMode,
5         float boost
6     ) throws IOException;
7
8     public abstract Query rewrite(
9         IndexSearcher indexSearcher
10    ) throws IOException;
11
12    public abstract void visit(
13        QueryVisitor visitor
14    );
15 }
```



*а как  
искать*

# Query в Lucene

Описывает логику поиска

○○○

```
1 public abstract class Query {
2     public abstract Weight createWeight(
3         IndexSearcher searcher,
4         ScoreMode scoreMode,
5         float boost
6     ) throws IOException;
7
8     public abstract Query rewrite(
9         IndexSearcher indexSearcher
10    ) throws IOException;
11
12    public abstract void visit(
13        QueryVisitor visitor
14    );
15 }
```



*а как  
оптимизировать*

# Query в Lucene

Описывает логику поиска

○○○

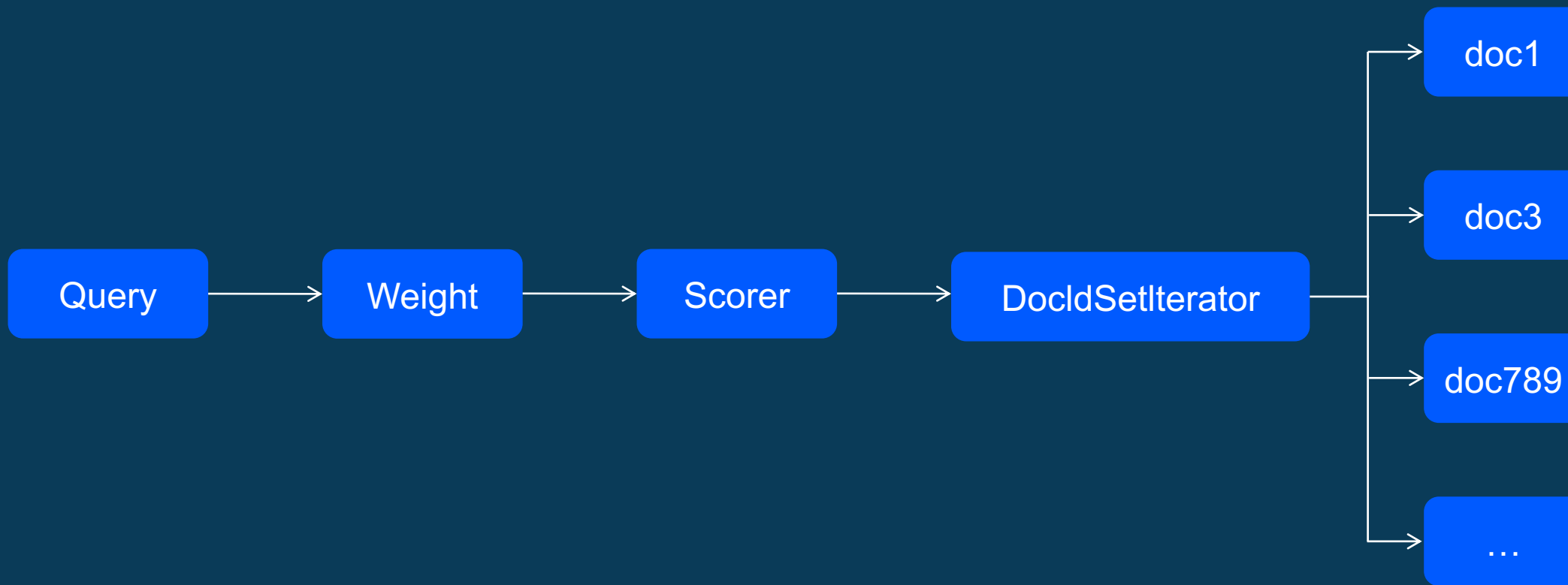
```
1 public abstract class Query {
2     public abstract Weight createWeight(
3         IndexSearcher searcher,
4         ScoreMode scoreMode,
5         float boost
6     ) throws IOException;
7
8     public abstract Query rewrite(
9         IndexSearcher indexSearcher
10    ) throws IOException;
11
12    public abstract void visit(
13        QueryVisitor visitor
14    );
15 }
```



*а как  
рефлексировать*

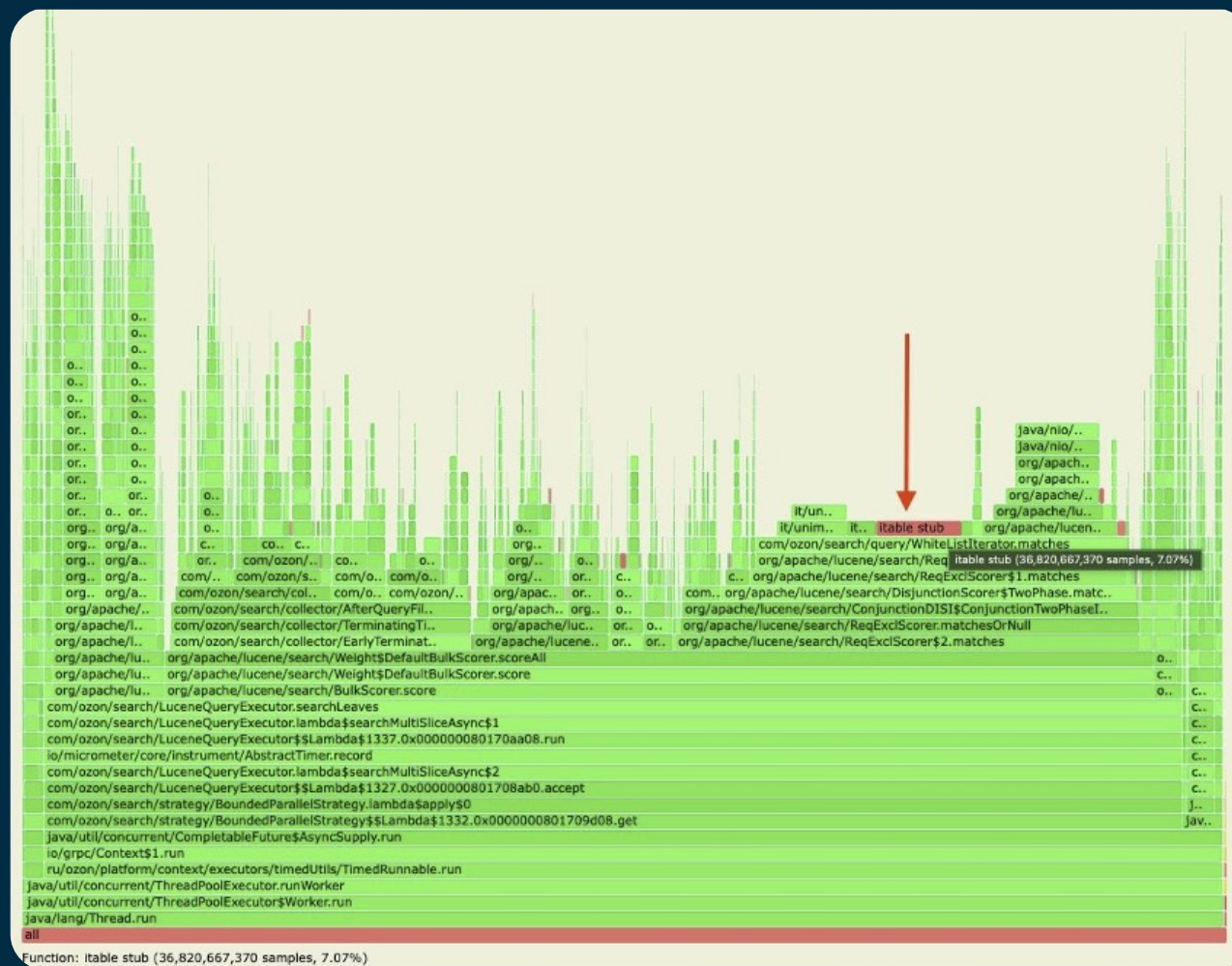
# Матрёшка обработки запроса

Проход по документам идёт на уровне DocIdSetIterator



# Исследование флеймграфов привело нас к istub

«Честному» вызову методов интерфейса



# Виртуальные вызовы метода интерфейса

Немного ASM'a

○○○

```
1 callSite:
2   set #calledInterface, CHECK
3   call #itableStub[itableSlot]
4 ---
5 itableStub[itableSlot]:
6   load (RCVR + #klass), KCLASS_TEM
7   load (KCLASS_TEM + #vtableSize), TEM
8   add (KCLASS_TEM + TEM), SCAN_TEM
9 tryAgain:
10  # this part is repeated zero or more times, usually zero
11  load (SCAN_TEM + #itableEntry.interface), TEM
12  cmp TEM, CHECK
13  jump,eq foundInterface
14  test TEM
15  jump,z noSuchInterface
16  inc #sizeof(itableEntry), SCAN_TEM
17  jump tryAgain
18 foundInterface:
19  load (SCAN_TEM + #itableEntry.offset), TEM
20  load (KCLASS_TEM + TEM + #itableSlot), METHOD
21  load (METHOD + #compiledEntry), TEM
22  jump TEM
23 ---
24 compiledEntry:
25  ...
```





# Виртуальные вызовы метода интерфейса

Нужно вызвать метод интерфейса на каком-то объекте

○ ○ ○

```
1 private static void invokeInterface(  
2     Klass calledInterface, int itableSlot,  
3     JObject object, JObject... params  
4 ) {  
5     var interfaceImpl = object.header.klass.interfaceImpl;  
6     do {  
7         var type = interfaceImpl.klass;  
8         if (type == calledInterface) {  
9             interfaceImpl.vtable[itableSlot].call(object, params);  
10            return;  
11        }  
12        if (type == null) throw new ClassCastException("noSuchInterface");  
13  
14        interfaceImpl = interfaceImpl.nextInterfaceImpl;  
15    } while (true);  
16 }
```

# Виртуальные вызовы метода интерфейса

Получаем начало списка интерфейсов

○ ○ ○

```
1 private static void invokeInterface(  
2     Klass calledInterface, int itableSlot,  
3     JObject object, JObject... params  
4 ) {  
5     var interfaceImpl = object.header.klass.interfaceImpl;  
6     do {  
7         var type = interfaceImpl.klass;  
8         if (type == calledInterface) {  
9             interfaceImpl.vtable[itableSlot].call(object, params);  
10            return;  
11        }  
12        if (type == null) throw new ClassCastException("noSuchInterface");  
13  
14        interfaceImpl = interfaceImpl.nextInterfaceImpl;  
15    } while (true);  
16 }
```

# Виртуальные вызовы метода интерфейса

Проверили, тот ли это интерфейс, который нам нужен

○○○

```
1 private static void invokeInterface(  
2     Klass calledInterface, int itableSlot,  
3     JObject object, JObject... params  
4 ) {  
5     var interfaceImpl = object.header.klass.interfaceImpl;  
6     do {  
7         var type = interfaceImpl.klass;  
8         if (type == calledInterface) {  
9             interfaceImpl.vtable[itableSlot].call(object, params);  
10            return;  
11        }  
12        if (type == null) throw new ClassCastException("noSuchInterface");  
13  
14        interfaceImpl = interfaceImpl.nextInterfaceImpl;  
15    } while (true);  
16 }
```

# Виртуальные вызовы метода интерфейса

Если да, то вызвали этот метод по табличке

○○○

```
1 private static void invokeInterface(  
2     Klass calledInterface, int itableSlot,  
3     JObject object, JObject... params  
4 ) {  
5     var interfaceImpl = object.header.klass.interfaceImpl;  
6     do {  
7         var type = interfaceImpl.klass;  
8         if (type == calledInterface) {  
9             interfaceImpl.vtable[itableSlot].call(object, params);  
10            return;  
11        }  
12        if (type == null) throw new ClassCastException("noSuchInterface");  
13  
14        interfaceImpl = interfaceImpl.nextInterfaceImpl;  
15    } while (true);  
16 }
```

# Виртуальные вызовы метода интерфейса

Если нет, взяли следующий интерфейс в списке и повторили всё для него

○ ○ ○

```
1 private static void invokeInterface(  
2     Klass calledInterface, int itableSlot,  
3     JObject object, JObject... params  
4 ) {  
5     var interfaceImpl = object.header.klass.interfaceImpl;  
6     do {  
7         var type = interfaceImpl.klass;  
8         if (type == calledInterface) {  
9             interfaceImpl.vtable[itableSlot].call(object, params);  
10            return;  
11        }  
12        if (type == null) throw new ClassCastException("noSuchInterface");  
13  
14        interfaceImpl = interfaceImpl.nextInterfaceImpl;  
15    } while (true);  
16 }
```

# Виртуальные вызовы метода интерфейса

Если так и не нашли, значит интерфейс не реализован типом

○○○

```
1 private static void invokeInterface(  
2     Klass calledInterface, int itableSlot,  
3     JObject object, JObject... params  
4 ) {  
5     var interfaceImpl = object.header.klass.interfaceImpl;  
6     do {  
7         var type = interfaceImpl.klass;  
8         if (type == calledInterface) {  
9             interfaceImpl.vtable[itableSlot].call(object, params);  
10            return;  
11        }  
12        if (type == null) throw new ClassCastException("noSuchInterface");  
13  
14        interfaceImpl = interfaceImpl.nextInterfaceImpl;  
15    } while (true);  
16 }
```



**ЭТОТ МАЛЕНЬКИЙ МАНЕВР БУДЕТ СТОИТЬ НА 51 ГОД**

# Источник проблемы

○ ○ ○

```
1 public static final class FooIterator extends TwoPhaseIterator {
2     private final SortedNumericDocValues values;
3     private final LongSet numbers;
4
5     public FooIterator(SortedNumericDocValues values, LongSet numbers) { ... }
6
7     @Override
8     public boolean matches() throws IOException {
9         var values = this.values;
10        var numbers = this.numbers;
11        var count = values.docValueCount();
12        for (var i = 0; i < count; i++) {
13            if (numbers.contains(values.nextValue())) {
14                return true;
15            }
16        }
17
18        return false;
19    }
20
21    ...
22 }
```



# Источник проблемы

○ ○ ○

```
1 public static final class FooIterator extends TwoPhaseIterator {
2     private final SortedNumericDocValues values;
3     private final LongSet numbers;
4
5     public FooIterator(SortedNumericDocValues values, LongSet numbers) { ... }
6
7     @Override
8     public boolean matches() throws IOException {
9         var values = this.values;
10        var numbers = this.numbers;
11        var count = values.docValueCount();
12        for (var i = 0; i < count; i++) {
13            if (numbers.contains(values.nextValue())) {
14                return true;
15            }
16        }
17
18        return false;
19    }
20
21    ...
22 }
```



На мономорфизации  
можно рассчитывать  
только в monomorphic-  
сценарии

# OLD MAN YELLS AT JVM



l'isid g'ira nio'peri  
no'j kf o'uké'lf:  
jer' aud r'ia'eer'j  
m'istj / Ka'f'j  
d' hdy'ivr 'sue j'ul  
sc' - ueme as'ie' of a  
e'el'ly ne. r'asow' a  
is'it'j  
is'it'  
me'j  
i'f' na "0' m)  
la'ad' iz'it'it'  
x'ad' q'ez' r'is

Cid'ner Abraham' é'it'it'  
i'c'ive' v'st' z'it'it' r'is'it' j

# ВИНОВНИК: fastutil

Который был призван улучшать производительность

**Package** `it.unimi.dsi.fastutil.longs`

## Interface LongSet

### All SuperInterfaces:

`Collectionl<Longl>`, `Iterablel<Longl>`, `LongCollection`, `LongIterable`, `Setl<Longl>`

### All Known SubInterfaces:

`LongSortedSet`

### All Known Implementing Classes:

`AbstractLongSet`, `AbstractLongSortedSet`, `LongArraySet`, `LongAVLTreeSet`, `LongLinkedOpenCustomHashSet`, `LongLinkedOpenHashSet`, `LongOpenCustomHashSet`, `LongOpenHashBigSet`, `LongOpenHashSet`, `LongRBTreeSet`, `LongSets.EmptySet`, `LongSets.Singleton`, `LongSets.SynchronizedSet`, `LongSets.UnmodifiableSet`, `LongSortedSets.EmptySet`, `LongSortedSets.Singleton`, `LongSortedSets.SynchronizedSortedSet`, `LongSortedSets.UnmodifiableSortedSet`



# Зачем платить дважды?



```
1 public abstract class DocIdSetIterator {  
2     public abstract int docID();  
3  
4     public abstract int nextDoc()  
5         throws IOException;  
6  
7     public abstract int advance(int target)  
8         throws IOException;  
9 }
```

# Идея: ручные специализации

○○○

```
1 public static final class FooIterator_Hash extends TwoPhaseIterator {
2     private final SortedNumericDocValues values;
3     private final LongHashSet numbers;
4
5     public FooIterator(SortedNumericDocValues values, LongSet numbers) { ... }
6
7     @Override
8     public boolean matches() throws IOException {
9         var values = this.values;
10        var numbers = this.numbers;
11        var count = values.docValueCount();
12        for (var i = 0; i < count; i++) {
13            if (numbers.contains(values.nextValue())) {
14                return true;
15            }
16        }
17
18        return false;
19    }
20
21    ...
22 }
```

# Идея: ручные специализации

○○○

```
1 public static TwoPhaseIterator create(  
2     SortedNumericDocValues values,  
3     LongSet someNumbers  
4 ) {  
5     switch someNumbers {  
6         case LongOpenHashSet numbers  
7             -> new FooIterator_Hash(values, values);  
8         case LongArraySet numbers  
9             -> new FooIterator_Array(values, numbers);  
10        case LongSets.Singleton numbers  
11            -> new FooIterator_Single(values, numbers.getFirst());  
12        case LongSet numbers  
13            -> new FooIterator_Interface(values, numbers);  
14    }  
15 }
```



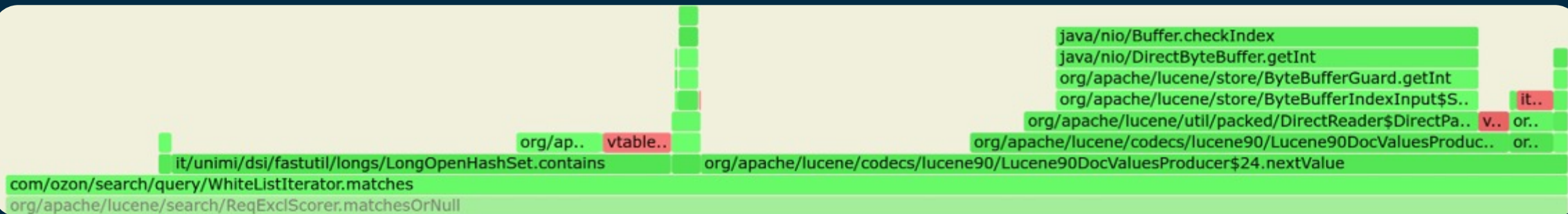
# Идея: ручные специализации

○○○

```
1 public static TwoPhaseIterator create(  
2     SortedNumericDocValues values,  
3     LongSet someNumbers  
4 ) {  
5     switch someNumbers {  
6         case LongOpenHashSet numbers  
7             -> new FooIterator_Hash(values, values);  
8         case LongArraySet numbers  
9             -> new FooIterator_Array(values, numbers);  
10        case LongSets.Singleton numbers  
11            -> new FooIterator_Single(values, numbers.getFirst());  
12        case LongSet numbers  
13            -> new FooIterator_Interface(values, numbers);  
14    }  
15 }
```

# itable ушли

Логика осталась





# А потом пришло API v2

# DocValuesInt64FilterQuery

Фильтр по множеству документов

○○○

```
1 message DocValuesInt64FilterQueryNode {
2   string field = 1;
3   Mode mode = 2;
4   bool match_if_empty = 3;
5   message Mode {
6     oneof kind {
7       ContainsAny contains_any = 1;
8       ContainsNone contains_none = 2;
9       ContainsAnyOther contains_any_other = 3;
10      ContainsNoneOther contains_none_other = 4;
11      ContainsExactly contains_exactly = 5;
12    }
13    message ContainsAny { repeated int64 values = 1; }
14    message ContainsNone { repeated int64 values = 1; }
15    message ContainsAnyOther { repeated int64 values = 1; }
16    message ContainsNoneOther { repeated int64 values = 1; }
17    message ContainsExactly { repeated int64 values = 1; }
18  };
19 }
```

# DocValuesInt64FilterQuery

5 режимов работы

○○○

```
1 message DocValuesInt64FilterQueryNode {
2   string field = 1;
3   Mode mode = 2;
4   bool match_if_empty = 3;
5   message Mode {
6     oneof kind {
7       ContainsAny contains_any = 1;
8       ContainsNone contains_none = 2;
9       ContainsAnyOther contains_any_other = 3;
10      ContainsNoneOther contains_none_other = 4;
11      ContainsExactly contains_exactly = 5;
12    }
13    message ContainsAny { repeated int64 values = 1; }
14    message ContainsNone { repeated int64 values = 1; }
15    message ContainsAnyOther { repeated int64 values = 1; }
16    message ContainsNoneOther { repeated int64 values = 1; }
17    message ContainsExactly { repeated int64 values = 1; }
18  };
19 }
```

# DocValuesInt64FilterQuery

2 способа обработки пустых значений

○○○

```
1 message DocValuesInt64FilterQueryNode {
2   string field = 1;
3   Mode mode = 2;
4   bool match_if_empty = 3;
5   message Mode {
6     oneof kind {
7       ContainsAny contains_any = 1;
8       ContainsNone contains_none = 2;
9       ContainsAnyOther contains_any_other = 3;
10      ContainsNoneOther contains_none_other = 4;
11      ContainsExactly contains_exactly = 5;
12    }
13    message ContainsAny { repeated int64 values = 1; }
14    message ContainsNone { repeated int64 values = 1; }
15    message ContainsAnyOther { repeated int64 values = 1; }
16    message ContainsNoneOther { repeated int64 values = 1; }
17    message ContainsExactly { repeated int64 values = 1; }
18  };
19 }
```



# DocValuesInt64FilterQuery

И всё те же 4 часто встречающиеся fastutil-коллекции

○○○

```
1 message DocValuesInt64FilterQueryNode {
2   string field = 1;
3   Mode mode = 2;
4   bool match_if_empty = 3;
5   message Mode {
6     oneof kind {
7       ContainsAny contains_any = 1;
8       ContainsNone contains_none = 2;
9       ContainsAnyOther contains_any_other = 3;
10      ContainsNoneOther contains_none_other = 4;
11      ContainsExactly contains_exactly = 5;
12    }
13    message ContainsAny { repeated int64 values = 1; }
14    message ContainsNone { repeated int64 values = 1; }
15    message ContainsAnyOther { repeated int64 values = 1; }
16    message ContainsNoneOther { repeated int64 values = 1; }
17    message ContainsExactly { repeated int64 values = 1; }
18  };
19 }
```





Почему мы должны  
писать специализации  
сами?



# Как это делают в других ЯП?

Например, в Rust

```
1 use std::{collections::HashSet, hash::Hash};
2
3 trait Collection<T> {
4     fn has(&self, value: &T) -> bool;
5 }
6
7 impl<T: Eq + Hash> Collection<T> for HashSet<T> {
8     fn has(&self, value: &T) -> bool {
9         self.contains(value)
10    }
11 }
12
13 impl<T: Eq> Collection<T> for Vec<T> {
14     fn has(&self, value: &T) -> bool {
15         self.iter().any(|element| element == value)
16    }
17 }
18
19 fn print_if_contains<T: Eq + Hash, C: Collection<T>>(
20     collection: &C,
21     value: &T,
22 ) {
23     if collection.has(value) {
24         println!("Contains value");
25     }
26 }
27
28 fn main() {
29     let hash_set = HashSet::from([1, 2, 3]);
30     let vec = Vec::from([1, 2, 3, 2, 1]);
31
32     print_if_contains(&hash_set, &2);
33     print_if_contains(&vec, &2);
34 }
35
```

	Execution	ASM
	andb \$1, %al	
	movzbl %al, %eax	
	addq \$24, %rsp	
	retq	
	playground::print_if_contains:	
	subq \$72, %rsp	
	movq %rdi, 56(%rsp)	
	movq %rsi, 64(%rsp)	
	callq <std::collections::hash::set::HashSet<T> as playground::Collection<T>::has	
	testb \$1, %al	
	jne .LBB204_2	
	.LBB204_1:	
	addq \$72, %rsp	
	retq	
	.LBB204_2:	
	leaq 8(%rsp), %rdi	
	leaq .L__unnamed_32(%rip), %rsi	
	movl \$1, %edx	
	callq core::fmt::Arguments::new_const	
	leaq 8(%rsp), %rdi	
	callq *std::io::stdio::_print@GOTPCREL(%rip)	
	jmp .LBB204_1	
	playground::print_if_contains:	
	subq \$72, %rsp	
	movq %rdi, 56(%rsp)	
	movq %rsi, 64(%rsp)	
	callq <alloc::vec::Vec<T> as playground::Collection<T>::has	
	testb \$1, %al	
	jne .LBB205_2	

Все реализации имеют общие черты

# Как это делают в других ЯП?

Например, в Rust

```
1 use std::{collections::HashSet, hash::Hash};
2
3 trait Collection<T> {
4     fn has(&self, value: &T) -> bool;
5 }
6
7 impl<T: Eq + Hash> Collection<T> for HashSet<T> {
8     fn has(&self, value: &T) -> bool {
9         self.contains(value)
10    }
11 }
12
13 impl<T: Eq> Collection<T> for Vec<T> {
14     fn has(&self, value: &T) -> bool {
15         self.iter().any(|element| element == value)
16    }
17 }
18
19 fn print_if_contains<T: Eq + Hash, C: Collection<T>>(
20     collection: &C,
21     value: &T,
22 ) {
23     if collection.has(value) {
24         println!("Contains value");
25     }
26 }
27
28 fn main() {
29     let hash_set = HashSet::from([1, 2, 3]);
30     let vec = Vec::from([1, 2, 3, 2, 1]);
31
32     print_if_contains(&hash_set, &2);
33     print_if_contains(&vec, &2);
34 }
35
```

Execution	ASM
andb \$1, %al movzbl %al, %eax addq \$24, %rsp retq	
playground::print_if_contains: subq \$72, %rsp movq %rdi, 56(%rsp) movq %rsi, 64(%rsp) callq <std::collections::hash::set::HashSet<T> as playground::Collection<T>::has testb \$1, %al jne .LBB204_2	
.LBB204_1: addq \$72, %rsp retq	
.LBB204_2: leaq 8(%rsp), %rdi leaq .L__unnamed_32(%rip), %rsi movl \$1, %edx callq core::fmt::Arguments::new_const leaq 8(%rsp), %rdi callq *std::io::stdio::_print@GOTPCREL(%rip) jmp .LBB204_1	
playground::print_if_contains: subq \$72, %rsp movq %rdi, 56(%rsp) movq %rsi, 64(%rsp) callq <alloc::vec::Vec<T> as playground::Collection<T>::has testb \$1, %al jne .LBB205_2	

Все реализации имеют общие черты

В некоторых языках программирования генерики/шаблоны работают через мономорфизацию

# Как это делают в других ЯП?

Например, в fastutil

```
# We pass each generated Java source through the gccpreprocessor.
```

```
TEST compiles in the test code,
```

```
# whereas ASSERTS compiles in some assertions
```

```
(whose testing, of course, must be enabled in the JVM).
```

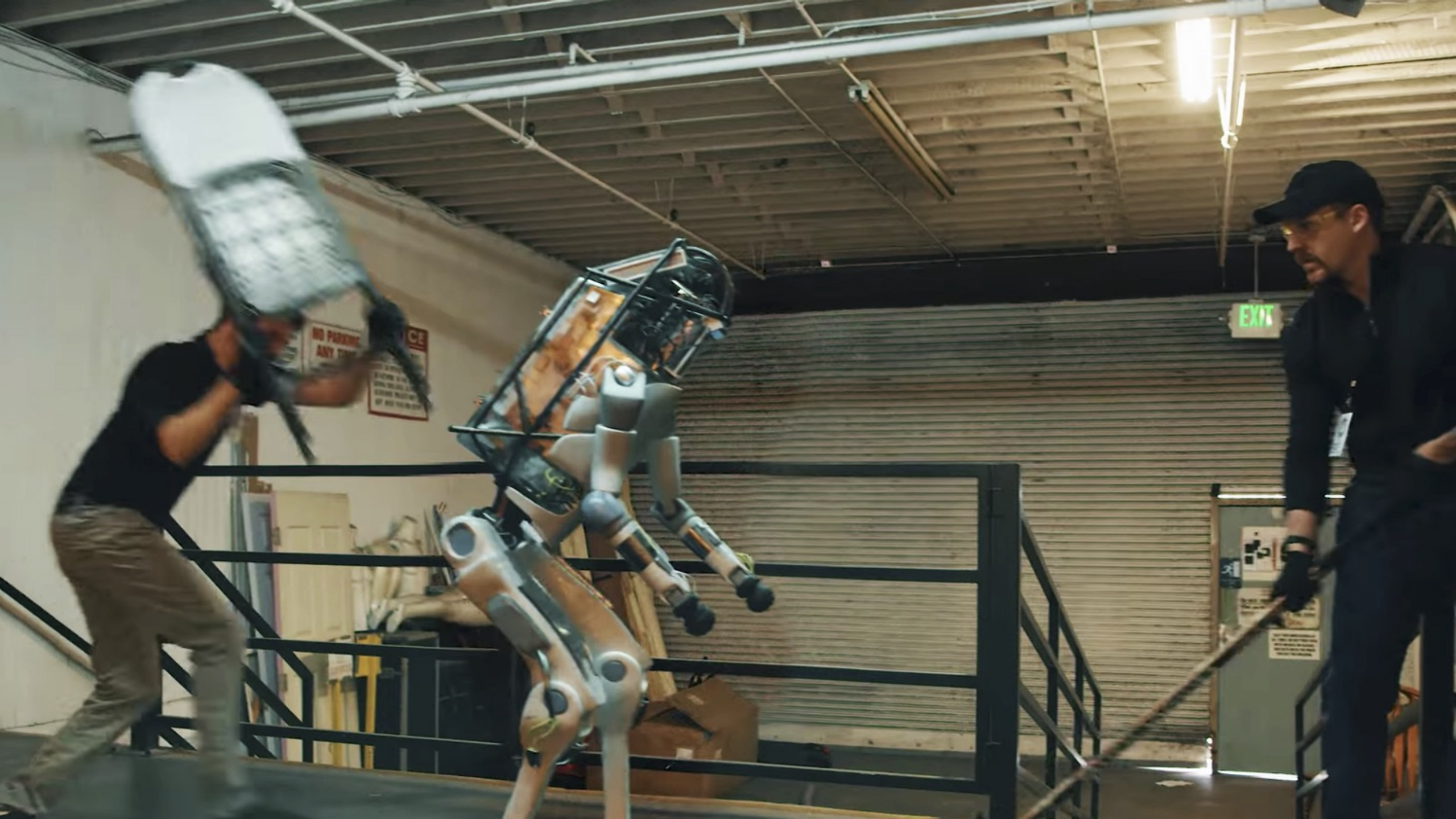
```
$(JSOURCES): %.java: %.c
```

```
$(CC) -w -I. $(if $(TEST),-DTEST,) \
```

```
$(if $(ASSERTS),-DASSERTS_CODE,) \
```

```
-DASSERTS_VALUE=$(if $(ASSERTS),true,false) -E -C -P $< \
```

```
| sed -e '1,/START_OF_JAVA_SOURCE/d' -e 's/^ / /' >$@
```

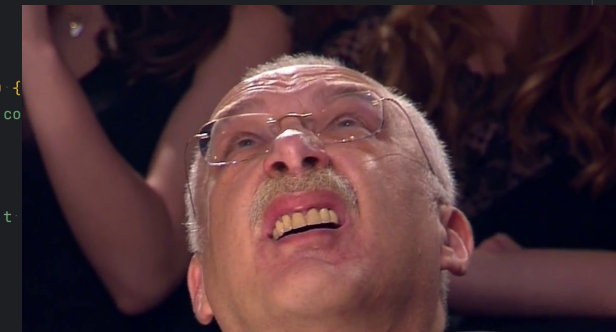


Узрите...

# Шаблон запроса на основе Freemarker

```
<#macro filter matchIfEmptyMode setSize queryMode mode fastMode>
    <#switch mode>
        <#case 'MATCH'>
            var docValues = this.docValues;
            <#if matchIfEmptyMode == 'true'>
                if (!docValues.advanceExact(approximation.docID())) {
                    return ${match(mode, "docValues is empty")};
                }
            </#if>
            <#break/>
        <#case 'EXPLAIN'>
            <#if isFast(fastMode)>
                var docValues = sortedNumericDocValues(context.reader(), storage, persistentIdField);
            <#else>
                var docValues = sortedNumericDocValues(context.reader(), field);
            </#if>
            if (!docValues.advanceExact(doc)) {
                <#if matchIfEmptyMode == 'true'>
                    return ${match(mode, "docValues is empty")};
                <#else>
                    return ${noMatch(mode, "docValues is empty")};
                </#if>
            }
            <#break/>
        <#default><#stop "Unknown mode: ${mode}">
    </#switch>
<!-- Implementation of the very check -->
<#switch setSize>
    <#case '0'>
        return <#switch queryMode>
            <#case 'CONTAINS_ANY'>${noMatch(mode, "document doesn't contain allowed value")}<#break/>
            <#case 'CONTAINS_NONE'>${match(mode, "document doesn't contain prohibited value")}<#break/>
            <#case 'CONTAINS_ANY_OTHER'>${match(mode, "document contains at least one other value")}<#break/>
            <#case 'CONTAINS_NONE_OTHER'>${noMatch(mode, "document contains at least one other value")}<#break/>
            <#case 'CONTAINS_EXACTLY'>${noMatch(mode, "document contains more values or doesn't contain this value")}<#break/>
            <#default><#stop "Unknown primitive type: ${queryMode}">
        </#switch>;
    <#break/>
    <#case '1'>
        <#switch queryMode>
            <#case 'CONTAINS_ANY'>
                var count = docValues.docValueCount();
                <#if mode == 'MATCH'>
                    var value = this.value;
                </#if>
                for (var i = 0; i < count; i++) {
                    if (docValues.nextValue() == value) {
                        return ${match(mode, "document co
                    }
                }
                return ${noMatch(mode, "document doesn't
            <#break/>
            <#case 'CONTAINS_NONE'>
                var count = docValues.docValueCount();
                <#if mode == 'MATCH'>

```







# Про различную логику работы виртуальных ВЫЗОВОВ



**Алексей Шипилёв**  
aleksey@shipilev.net



[https://shipilev.net/blog/2015/  
black-magic-method-dispatch/](https://shipilev.net/blog/2015/black-magic-method-dispatch/)

# Выводы

01 Виртуальные вызовы на интерфейсах — это дорого 





# Выводы

01 Виртуальные вызовы на интерфейсах — это дорого

02 Порой кодогенерация — это выход 



# Выводы

01 Виртуальные вызовы на интерфейсах — это дорого

02 Порой кодогенерация — это выход

03 Понимание того, как работают определённые механизмы JVM, — необходимо



02



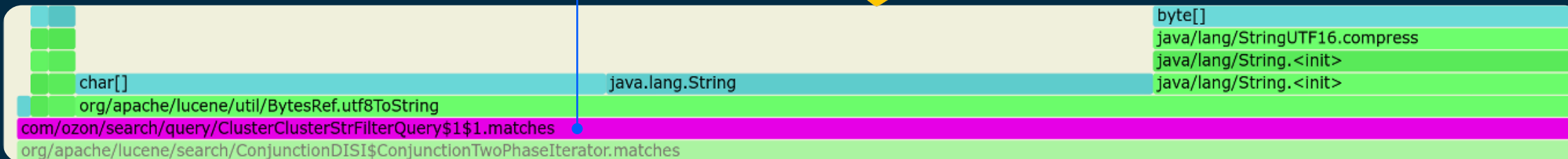
Аллокации

# Query наносят ответный удар

Но теперь с другой стороны



19,43%





# Что делают авторы Lucene?

Переиспользуют  
объекты

○ ○ ○

```
1 public final class BytesRef
2     implements Comparable<BytesRef>, Cloneable
3 {
4     public byte[] bytes;
5
6     public int offset;
7
8     public int length;
9
10    ...
11 }
```



На них перевести  
стрелки **не получится**

# Иногда аллокации нужны

Для удобства работы с бизнес-логикой



```
1 record LongTuple(long from, long to) {
2     public static LongTuple from(BytesRef ref) {
3         var buffer = ByteBuffer.wrap(
4             ref.bytes,
5             ref.offset,
6             ref.length,
7         );
8
9         return new LongTuple(
10            buffer.getLong(), buffer.getLong()
11        );
12    }
13 }
```

# Иногда аллокации нужны

Оборачиваем данные в ByteBuffer

○ ○ ○

```
1 record LongTuple(long from, long to) {
2     public static LongTuple from(BytesRef ref) {
3         var buffer = ByteBuffer.wrap(
4             ref.bytes,
5             ref.offset,
6             ref.length,
7         );
8
9         return new LongTuple(
10            buffer.getLong(), buffer.getLong()
11        );
12    }
13 }
```



# Иногда аллокации нужны

Возвращаем удобный Tuple

○ ○ ○

```
1 record LongTuple(long from, long to) {
2     public static LongTuple from(BytesRef ref) {
3         var buffer = ByteBuffer.wrap(
4             ref.bytes,
5             ref.offset,
6             ref.length,
7         );
8
9         return new LongTuple(
10            buffer.getLong(), buffer.getLong()
11        );
12    }
13 }
```

# Иногда аллокации нужны

Читаем данные

○○○

```
1 long ord;
2 while ((ord = restrictions.nextOrd()) != NO_MORE_ORDS) {
3     var tuple = LongTuple.fromBytesRef(
4         restrictions.lookupOrd(ord)
5     );
6
7     var to = tuple.to();
8     if (to == userCluster) {
9         var from = tuple.from();
10        ...
11    }
12 }
```

# Иногда аллокации нужны

Создаём Tuple

○ ○ ○

```
1 long ord;
2 while ((ord = restrictions.nextOrd()) != NO_MORE_ORDS) {
3     var tuple = LongTuple.fromBytesRef(
4         restrictions.lookupOrd(ord)
5     );
6
7     var to = tuple.to();
8     if (to == userCluster) {
9         var from = tuple.from();
10        ...
11    }
12 }
```

# Иногда аллокации нужны

Лениво работаем с его полями

○○○

```
1 long ord;
2 while ((ord = restrictions.nextOrd()) != NO_MORE_ORDS) {
3     var tuple = LongTuple.fromBytesRef(
4         restrictions.lookupOrd(ord)
5     );
6
7     var to = tuple.to();
8     if (to == userCluster) {
9         var from = tuple.from();
10        ...
11    }
12 }
```

# Иногда аллокации нужны

Вот тут

○○○

```
1 long ord;
2 while ((ord = restrictions.nextOrd()) != NO_MORE_ORDS) {
3     var tuple = LongTuple.fromBytesRef(
4         restrictions.lookupOrd(ord)
5     );
6
7     var to = tuple.to();
8     if (to == userCluster) {
9         var from = tuple.from();
10        ...
11    }
12 }
```



# ЧТО МЫ ХОТИМ?

Читать без аллокаций



```
1 long ord;
2 while ((ord = restrictions.nextOrd()) != NO_MORE_ORDS) {
3     var bytes = restrictions.lookupOrd(ord);
4
5     var to = LongTuple.readTo(bytes);
6     if (to == userCluster) {
7         var from = LongTuple.readFrom(bytes);
8         ...
9     }
10 }
```

# ЧТО МЫ ХОТИМ?

Лениво вычитывать примитивы без промежуточного объекта



```
1 long ord;
2 while ((ord = restrictions.nextOrd()) != NO_MORE_ORDS) {
3     var bytes = restrictions.lookupOrd(ord);
4
5     var to = LongTuple.readTo(bytes);
6     if (to == userCluster) {
7         var from = LongTuple.readFrom(bytes);
8         ...
9     }
10 }
```



Но есть же  
**ГОТОВЫЕ форматы**



# Protobuf

Слишком жирно

→ Заточен на поддержку обратной совместимости при изменениях

→ Перед полями пишутся теги

→ Жирное Java-API

○ ○ ○

```
1 message LongTuple {  
2     int64 from = 1;  
3     int64 to = 1;  
4 }
```



# А также

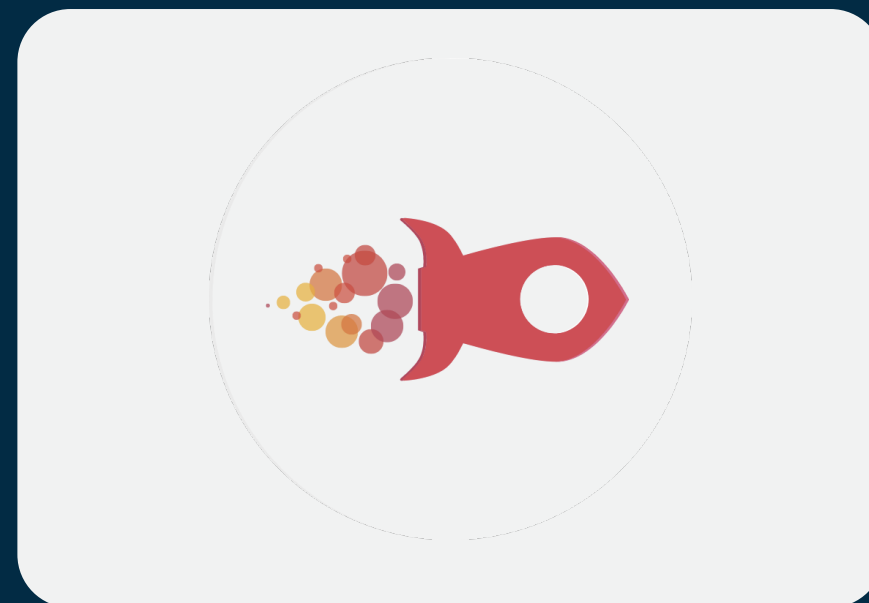
Специфический вывод

→ Требуется создание промежуточного объекта

→ Архаичный Java-кодогенератор

○ ○ ○

```
1 struct LongTuple {  
2     from:long;  
3     to:long;  
4 }
```



# Альтернативы

Любые компактные бинарные форматы

→ SBE

→ Chronicle Wire

→ CBOR

→ NBT (для особо извращённых)

→ И многое другое

Но нам-то нужна  
эффективная работа  
с BytesRef из Lucene



# Ещё один кодогенератор

Простое описание структуры данных



```
1 # structures.yaml
2
3 structures:
4   long_tuple:
5     name: LongTuple
6     representation: plain
7     fields:
8       - name: from
9         type: int64
10      - name: to
11        type: int64
```

# Ещё один кодогенератор

Тривиальное подключение к проекту



```
1 // package-info.java
2
3 @GeneratePlainbuffers("structures.yaml")
4 package ru.ozon.search.o2.structure;
```

# Ещё один кодогенератор

Нужный нам кодген

○ ○ ○

```
1 // LongTuple.java
2
3 public static final class Read {
4     public static long from(ByteBuffer buffer) {
5         return buffer.getLong(0);
6     }
7     public static long to(ByteBuffer buffer) {
8         return buffer.getLong(8);
9     }
10 }
11 public static final class Write {
12     public static void from(ByteBuffer buffer, long value) {
13         buffer.putLong(0, value);
14     }
15     public static void to(ByteBuffer buffer, long value) {
16         buffer.putLong(8, value);
17     }
18 }
```

# Ещё один кодогенератор

Методы для записи

○○○

```
1 // LongTuple.java
2
3 public static final class Read {
4     public static long from(ByteBuffer buffer) {
5         return buffer.getLong(0);
6     }
7     public static long to(ByteBuffer buffer) {
8         return buffer.getLong(8);
9     }
10 }
11 public static final class Write {
12     public static void from(ByteBuffer buffer, long value) {
13         buffer.putLong(0, value);
14     }
15     public static void to(ByteBuffer buffer, long value) {
16         buffer.putLong(8, value);
17     }
18 }
```

# Ещё один кодогенератор

Методы для чтения

○ ○ ○

```
1 // LongTuple.java
2
3 public static final class Read {
4     public static long from(ByteBuffer buffer) {
5         return buffer.getLong(0);
6     }
7     public static long to(ByteBuffer buffer) {
8         return buffer.getLong(8);
9     }
10 }
11 public static final class Write {
12     public static void from(ByteBuffer buffer, long value) {
13         buffer.putLong(0, value);
14     }
15     public static void to(ByteBuffer buffer, long value) {
16         buffer.putLong(8, value);
17     }
18 }
```



# Ещё один кодогенератор

А также методы для вычитки в record

○○○

```
1 public record LongTuple(long from, long to) {
2     public static final class Read { ... }
3
4     public static final class Write { ... }
5
6     public static LongTuple read(ByteBuffer buffer) {
7         return new LongTuple(
8             LongTuple.Read.from(buffer),
9             LongTuple.Read.to(buffer)
10        );
11    }
12
13    public static void write(ByteBuffer buffer, long from, long to) {
14        LongTuple.Write.from(buffer, from);
15        LongTuple.Write.to(buffer, to);
16    }
17 }
```

# Больше не аллоцируем

И удобно читаем данные

○○○

```
1 long ord;
2 while ((ord = restrictions.nextOrd()) != NO_MORE_ORDS) {
3     var bytes = BytesRefs.toByteBuffer(
4         restrictions.lookupOrd(ord)
5     );
6     var to = LongTuple.Read.to(bytes);
7     if (to == userCluster) {
8         var from = LongTuple.Read.from(bytes);
9         ...
10    }
11 }
```



# Больше не аллоцируем

Ведь так?

○○○

```
1 long ord;
2 while ((ord = restrictions.nextOrd()) != NO_MORE_ORDS) {
3     var bytes = BytesRefs.toByteBuffer(
4         restrictions.lookupOrd(ord)
5     );
6     var to = LongTuple.Read.to(bytes);
7     if (to == userCluster) {
8         var from = LongTuple.Read.from(bytes);
9         ...
10    }
11 }
```





Но чем лучше читать  
байтики в примитивах?

# Как будем читать байтики?

ByteBuffer

○○○

```
1 public static class Read {
2     public static long from(ByteBuffer buffer) {
3         return buffer.getLong(0);
4     }
5     public static long to(ByteBuffer buffer) {
6         return buffer.getLong(8);
7     }
8 }
9
10 public static class Write {
11     public static void from(ByteBuffer buffer, long value) {
12         buffer.putLong(0, value);
13     }
14     public static void to(ByteBuffer buffer, long value) {
15         buffer.putLong(8, value);
16     }
17 }
```

# Как будем читать байтики?

VarHandle

○○○

```
1 private static final VarHandle LONG_VIEW = MethodHandles
2   .byteBufferViewVarHandle(long[].class, ByteOrder.LITTLE_ENDIAN);
3
4 public static class Read {
5     public static long from(BytesRef ref) {
6         return (long) LONG_VIEW.get(buffer.bytes, ref.offset + 0 * Long.BYTES);
7     }
8     public static long to(BytesRef buffer) {
9         return (long) LONG_VIEW.get(buffer.bytes, ref.offset + 1 * Long.BYTES);
10    }
11 }
12
13 public static class Write {
14     public static void from(BytesRef buffer, long value) {
15         LONG_VIEW.set(buffer.bytes, ref.offset + 0 * Long.BYTES, value);
16     }
17     public static void to(BytesRef buffer, long value) {
18         LONG_VIEW.set(buffer.bytes, ref.offset + 1 * Long.BYTES, value);
19     }
20 }
```

# Выводы

01 Аллокации тоже могут становиться проблемой



# Выводы

01 Аллокации тоже могут становиться проблемой

02 Кодген — это всё ещё полезно





# Выводы

01 Аллокации тоже могут становиться проблемой

02 Кодген — это всё ещё полезно

03 В современных джавах™ есть прекрасное API VarHandle 



03



Уменьшение  
индирекций



# Сеньорские пути запутывания кода во имя наносекунд

# Сокращение чтений полей

## Гипотеза

Если значение **поля** используется **более одного раза**, то стоит **один раз** прочитать его **в локальную переменную**



# Сокращение чтений полей

До рефакторинга



```
1 @Override
2 public boolean matches() throws IOException {
3     var count = this.values.docValueCount();
4     for (var i = 0; i < count; i++) {
5         if (this.numbers.contains(this.values.nextValue())) {
6             return true;
7         }
8     }
9     return false;
10 }
```

# Сокращение чтений полей

До рефакторинга



```
1 @Override
2 public boolean matches() throws IOException {
3     var count = this.values.docValueCount();
4     for (var i = 0; i < count; i++) {
5         if (this.numbers.contains(this.values.nextValue())) {
6             return true;
7         }
8     }
9     return false;
10 }
```

# Сокращение чтений полей

После рефакторинга

○ ○ ○

```
1 @Override
2 public boolean matches() throws IOException {
3     var values = this.values;
4     var numbers = this.numbers;
5
6     var count = values.docValueCount();
7     for (var i = 0; i < count; i++) {
8         if (numbers.contains(values.nextValue())) {
9             return true;
10        }
11    }
12    return false;
13 }
```

# Сокращение чтений полей

## Реальность

- В некоторых случаях можно увидеть сокращение расходов на виртуальные вызовы
- Зачастую в коде становится проще ориентироваться с точки зрения его эффектов





# Использование record

## Гипотеза

Если есть возможность использовать record, то стоит так делать, потому что его компоненты волшебны для JVM



# Использование record

До рефакторинга



```
1 public class AtomicHitCounter
2     implements HitCounter
3 {
4     private final LongAdder count = new LongAdder();
5
6     ...
7 }
```

# Использование record

После рефакторинга



```
1 public record AtomicHitCounter(LongAdder count)
2     implements HitCounter
3 {
4     public AtomicHitCounter() {
5         this(new LongAdder());
6     }
7
8     ...
9 }
```



# record действительно волшебные

И вызывают доверие

```
219  ✓ static bool trust_final_non_static_fields(ciInstanceClass* holder) {
220      if (holder == nullptr)
221          return false;
222      if (holder->name() == ciSymbols::java_lang_System())
223          // Never trust strangely unstable finals: System.out, etc.
224          return false;
225      // Even if general trusting is disabled, trust system-built closures in these packages.
226      if (holder->is_in_package("java/lang/invoke") || holder->is_in_package("sun/invoke") ||
227          holder->is_in_package("java/lang/reflect") || holder->is_in_package("jdk/internal/reflect") ||
228          holder->is_in_package("jdk/internal/foreign/layout") || holder->is_in_package("jdk/internal/foreign") ||
229          holder->is_in_package("jdk/internal/vm/vector") || holder->is_in_package("jdk/incubator/vector") ||
230          holder->is_in_package("java/lang"))
231          return true;
232      // Trust hidden classes. They are created via Lookup.defineHiddenClass and
233      // can't be serialized, so there is no hacking of finals going on with them.
234      if (holder->is_hidden())
235          return true;
236      // Trust final fields in all boxed classes
237      if (holder->is_box_klass())
238          return true;
239      // Trust final fields in records
240      if (holder->is_record())
241          return true;
242      // Trust final fields in String
243      if (holder->name() == ciSymbols::java_lang_String())
244          return true;
245      // Trust Atomic*FieldUpdaters: they are very important for performance, and make up one
246      // more reason not to use Unsafe, if their final fields are trusted. See more in JDK-8140483.
247      if (holder->name() == ciSymbols::java_util_concurrent_atomic_AtomicIntegerFieldUpdater_Impl() ||
248          holder->name() == ciSymbols::java_util_concurrent_atomic_AtomicLongFieldUpdater_CASUpdater() ||
249          holder->name() == ciSymbols::java_util_concurrent_atomic_AtomicLongFieldUpdater_LockedUpdater() ||
250          holder->name() == ciSymbols::java_util_concurrent_atomic_AtomicReferenceFieldUpdater_Impl()) {
251          return true;
252      }
253      return TrustFinalNonStaticFields;
254  }
```



# Подробнее про (не)доверие к final- полям



**Алексей Шипилёв**  
aleksey@shipilev.net



[https://shipilev.net/jvm/  
anatomy-quarks/  
17-trust-nonstatic-final-fields/](https://shipilev.net/jvm/anatomy-quarks/17-trust-nonstatic-final-fields/)

# Использование record

## Реальность

- Обычно использование record'ов может упростить код
- Для record'ов действительно доступно больше оптимизаций
- Переход на record'ы может ломать инкапсуляцию



# Выводы

01 К изменениям стоит подходить осмысленно



# Выводы

01 К изменениям стоит подходить осмысленно

02 Наносекунды рантайма могут не стоить времени, потраченного на попытку разобраться в переусложнённом коде





# Выводы

01 К изменениям стоит подходить осмысленно

02 Наносекунды рантайма могут не стоить времени, потраченного на попытку разобраться в переусложнённом коде

03 Новые языковые фичи могут помогать в достижении перфа 



04

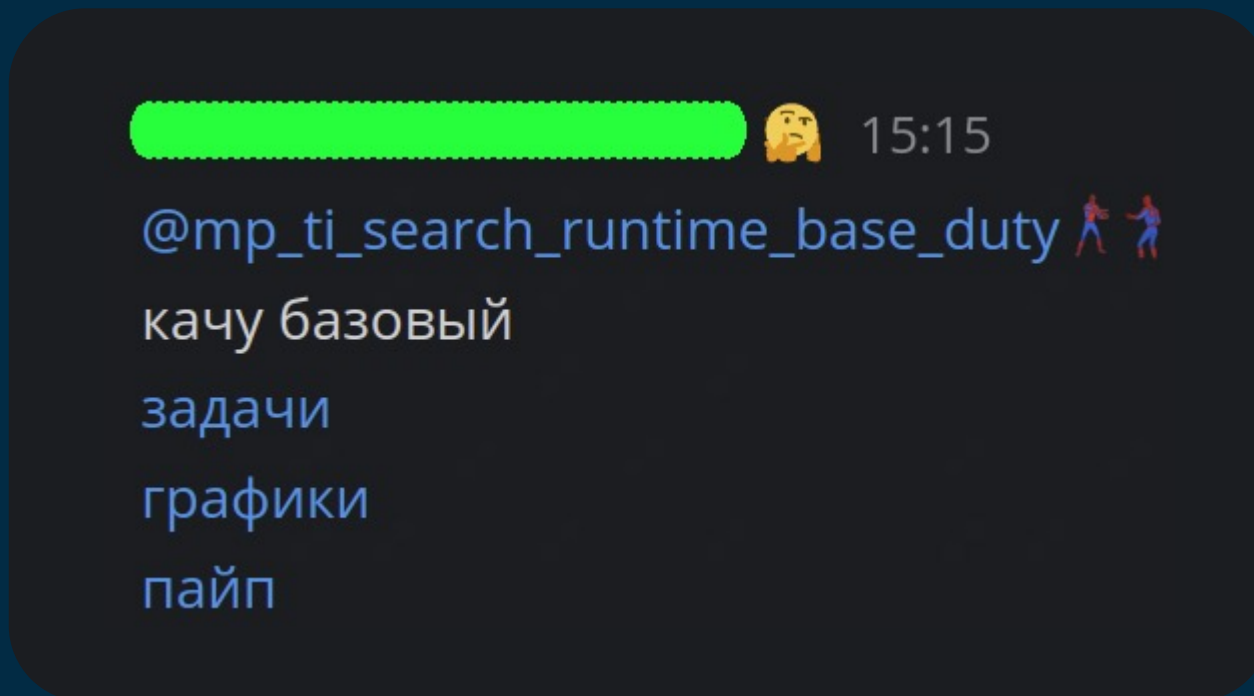


История одного  
рефакторинга

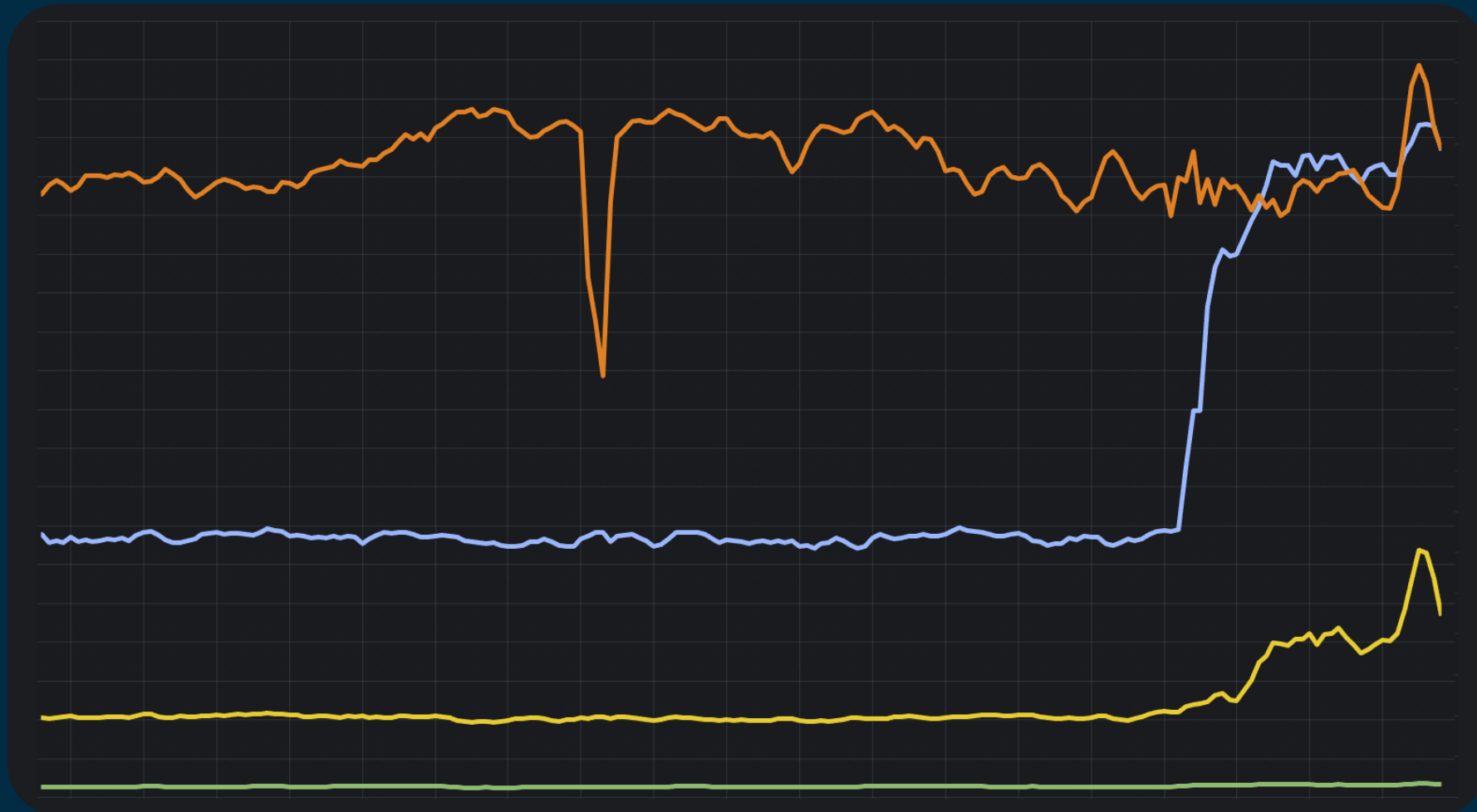


# Как (не) надо катить релизы

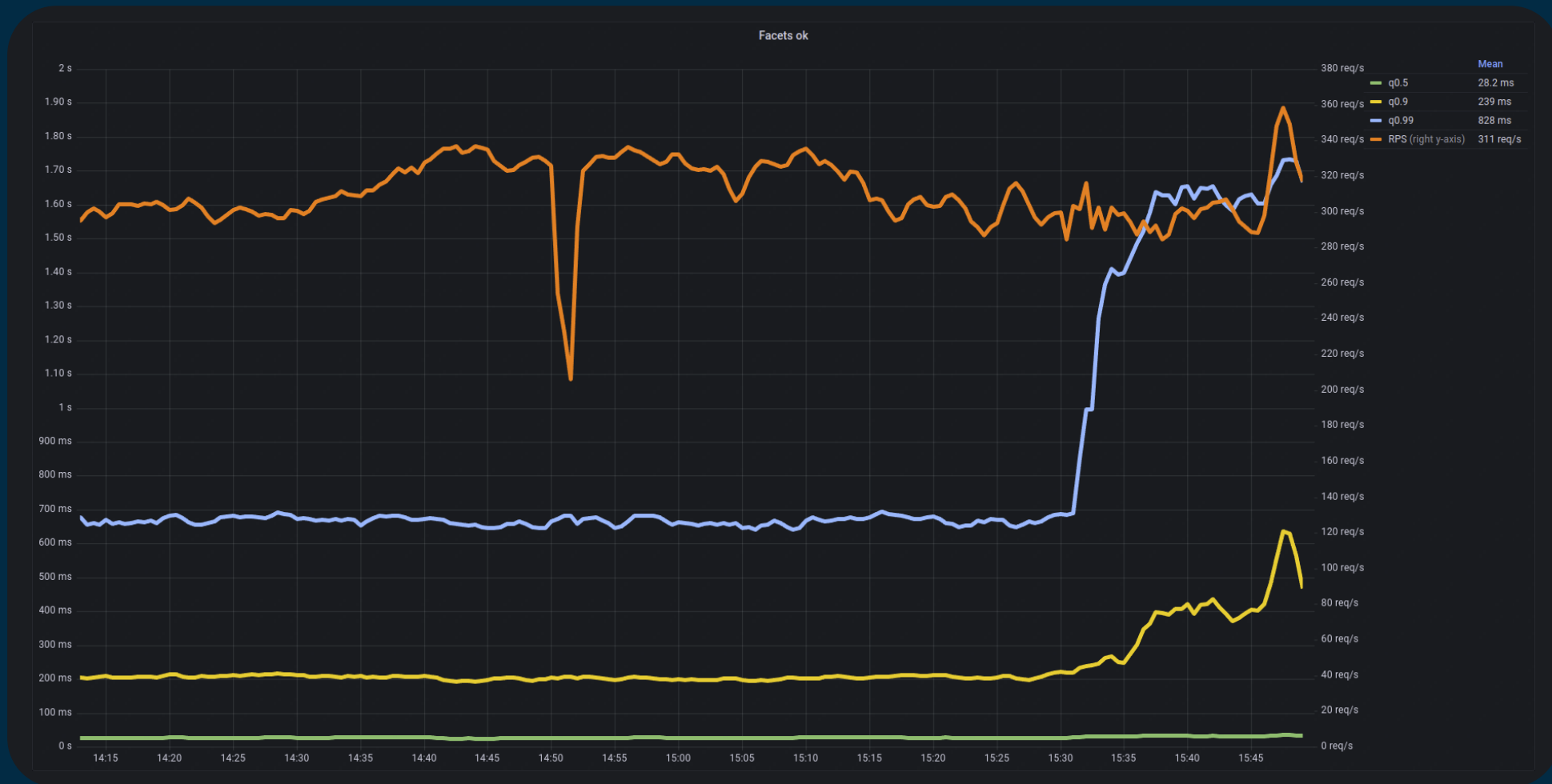
# Шаг 1. Катим релиз



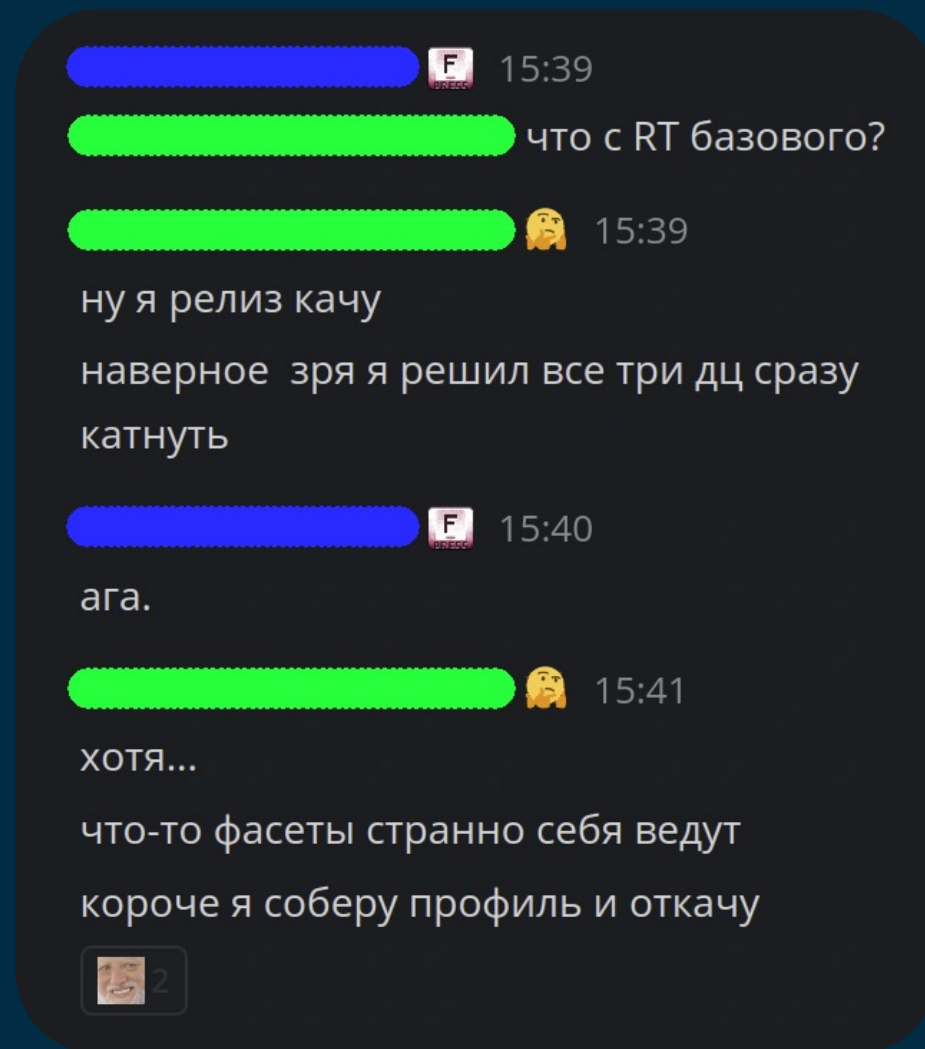
## Шаг 2. Наблюдаем рост!



## Шаг 2. Наблюдаем рост таймингов...



## Шаг 3. Смотрим в чат



## Шаг 4. Зовём виноватого

 🤔 16:25

релиз откатан и в подвешенном состоянии.  
Петя ищет почему он взорвал  
`TerminateAfterCollector.collect` в  
самостоятельном режиме 😊





## Шаг 4. Осуждаем себя

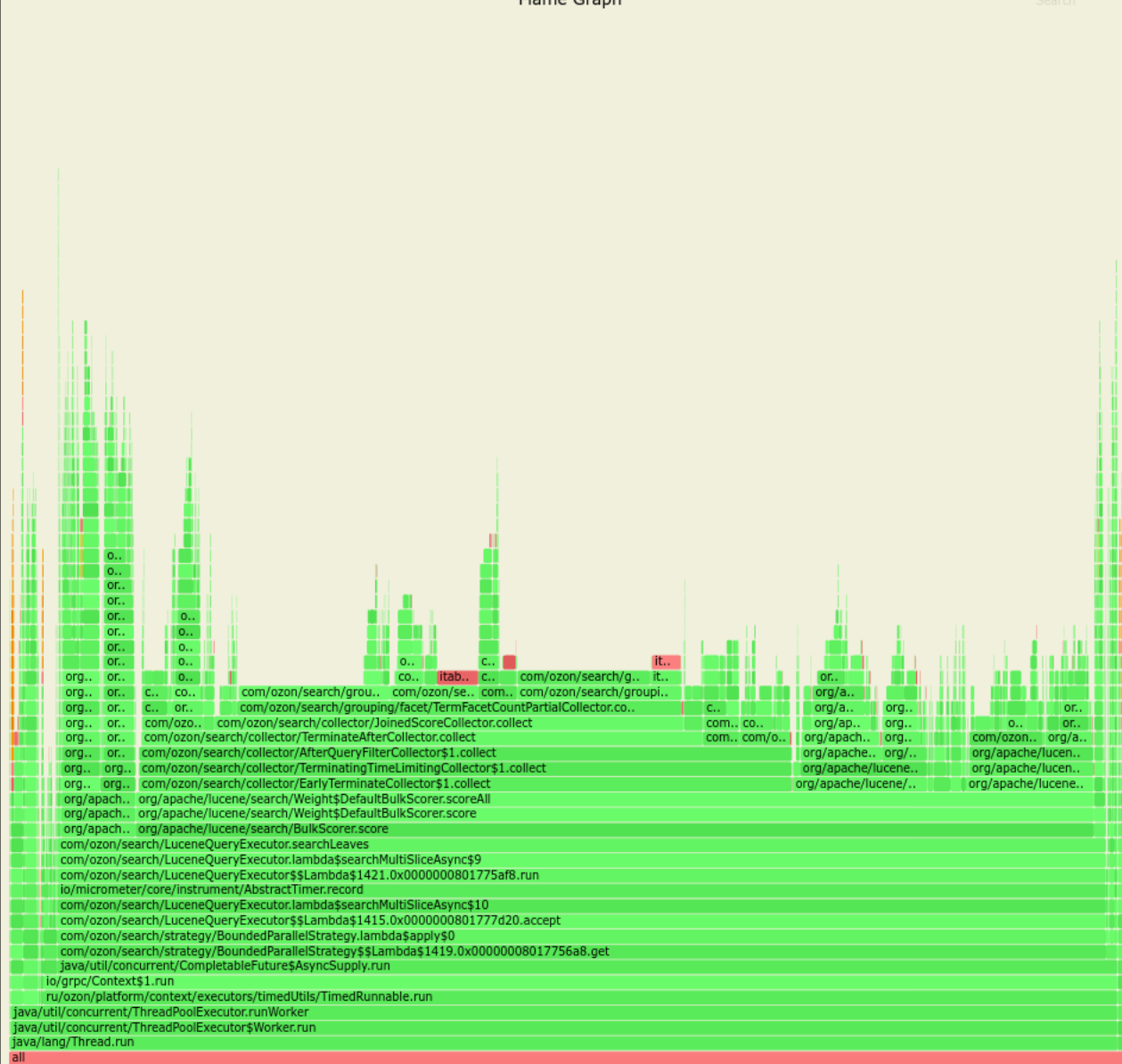
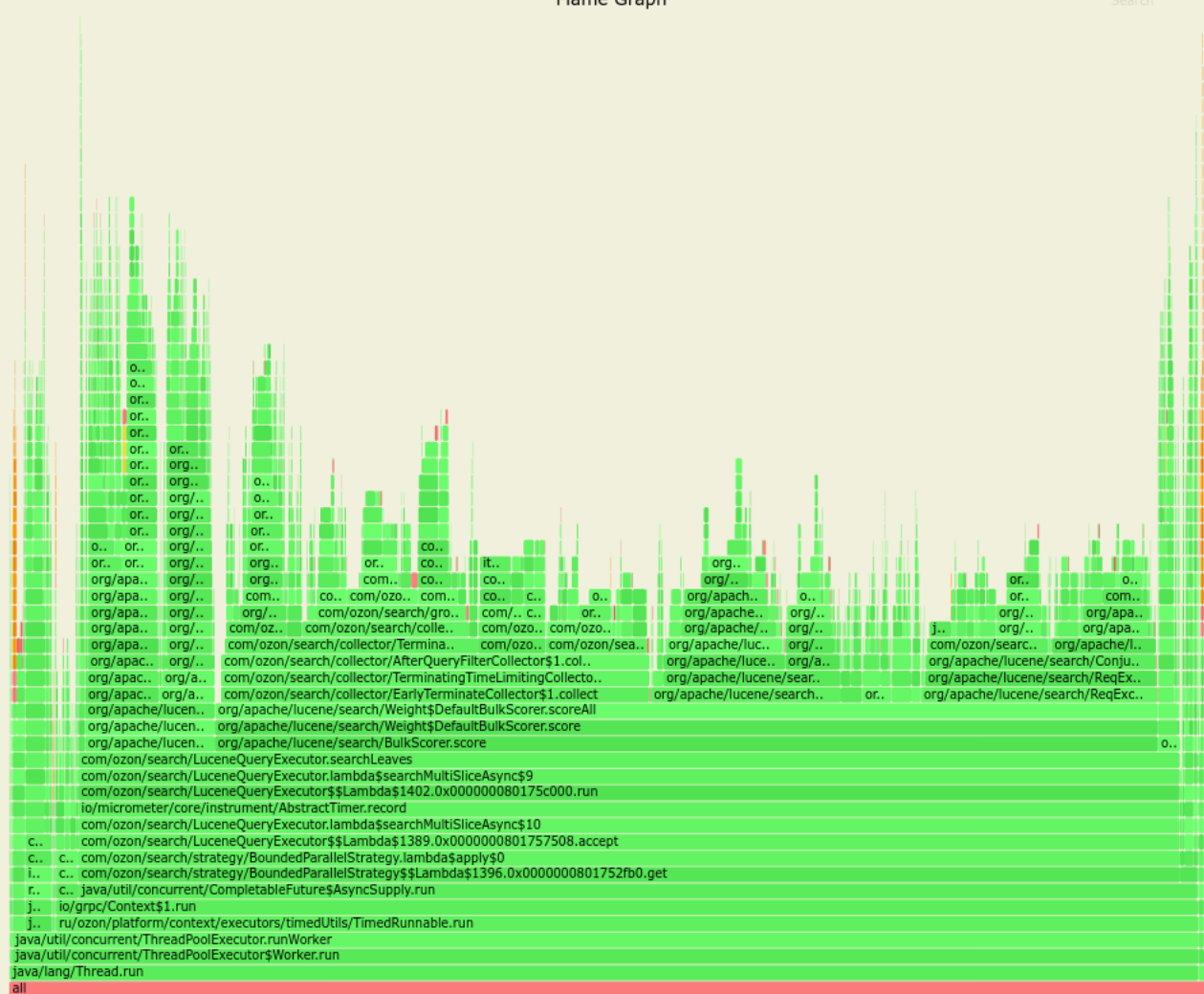
 112 files +1923 -1356





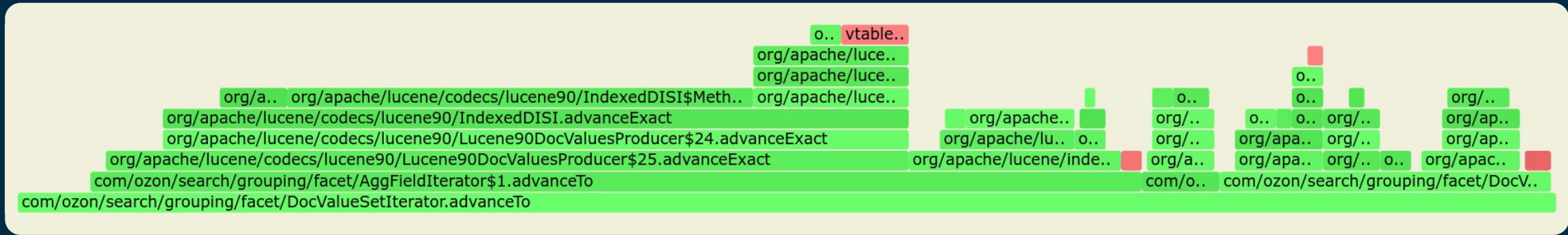


# Давайте искать причину



# Было

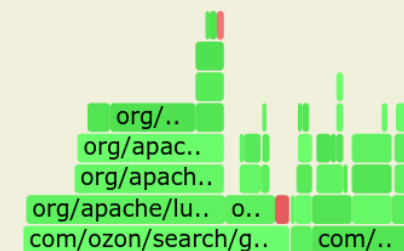
2,52%



# Стало

13,43%

`com/ozon/search/grouping/facet/DocValueSetIterator.advanceTo`

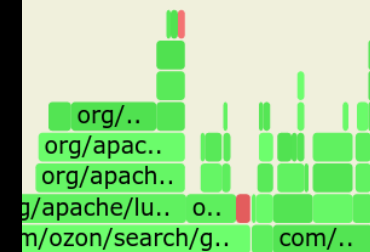


# Стало

13,43%

com/ozon/search/grouping/fa

SetIterator.advanceTo



# Где?





Все мы любим  
рефакторинги!

# Игра

Найди до чего докопаться

○ ○ ○

```
1 @SuppressWarnings("NotNullFieldNotInitialized")
2 protected TReader reader;
3
4 ...
5
6 @Override
7 public final boolean advanceTo(int docid) throws IOException {
8     Objects.requireNonNull(reader);
9
10    var result = reader.advanceTo(docId);
11    ord = result ? 0 : FacetUtils.NO_MORE_ORDS;
12    return result;
13 }
```

# Игра

Найди до чего докопаться

○ ○ ○

```
1 @SuppressWarnings("NotNullFieldNotInitialized")
2 protected TReader reader;
3
4 ...
5
6 @Override
7 public final boolean advanceTo(int docid) throws IOException {
8     Objects.requireNonNull(reader);
9
10    var result = reader.advanceTo(docId);
11    ord = result ? 0 : FacetUtils.NO_MORE_ORDS;
12    return result;
13 }
```

# Игра

Прочитать доку

## requireNonNull

```
public static <T> T requireNonNull(T obj)
```

Checks that the specified object reference is not null. This method is designed primarily for doing parameter validation in methods and constructors, as demonstrated below:

```
public Foo(Bar bar) {  
    this.bar = Objects.requireNonNull(bar);  
}
```

### Type Parameters:

T - the type of the reference

### Parameters:

obj - the object reference to check for nullity

### Returns:

obj if not null

### Throws:

NullPointerException - if obj is null

# Игра

Отрефачить



```
1 @SuppressWarnings("NotNullFieldNotInitialized")
2 protected TReader reader;
3
4 ...
5
6 @Override
7 public final boolean advanceTo(int docId) throws IOException {
8     var result = Objects.requireNonNull(reader).advanceTo(docId);
9     ord = result ? 0 : FacetUtils.NO_MORE_ORDS;
10    return result;
11 }
```



А дальше  
вы знаете

# Сравним байткод

Было

○○○

```
1 public final advanceTo(I)Z throws java/io/IOException
2     ALOAD 0
3     GETFIELD com/ozon/search/grouping/facet/DocValueSetIterator.reader : Lcom/ozon/search/grouping/facet/DocValueSetReader;
4     INVOKESTATIC java/util/Objects.requireNonNull (Ljava/lang/Object;)Ljava/lang/Object;
5     POP
6
7     ALOAD 0
8     GETFIELD com/ozon/search/grouping/facet/DocValueSetIterator.reader : Lcom/ozon/search/grouping/facet/DocValueSetReader;
9     ILOAD 1
10    INVOKEINTERFACE com/ozon/search/grouping/facet/DocValueSetReader.advanceTo (I)Z (itf)
11    ISTORE 2
```

# Сравним байткод

Стало

○○○

```
1 public final advanceTo(I)Z throws java/io/IOException
2     ALOAD 0
3     GETFIELD com/ozon/search/grouping/facet/DocValueSetIterator.reader : Lcom/ozon/search/grouping/facet/DocValueSetReader;
4     INVOKESTATIC java/util/Objects.requireNonNull (Ljava/lang/Object;)Ljava/lang/Object;
5     CHECKCAST com/ozon/search/grouping/facet/DocValueSetReader
6
7
8
9     ILOAD 1
10    INVOKEINTERFACE com/ozon/search/grouping/facet/DocValueSetReader.advanceTo (I)Z (itf)
11    ISTORE 2
```



# Сравним байткод

Было

○○○

```
1 public final advanceTo(I)Z throws java/io/IOException
2     ALOAD 0
3     GETFIELD com/ozon/search/grouping/facet/DocValueSetIterator.reader : Lcom/ozon/search/grouping/facet/DocValueSetReader;
4     INVOKESTATIC java/util/Objects.requireNonNull (Ljava/lang/Object;)Ljava/lang/Object;
5     POP
6
7     ALOAD 0
8     GETFIELD com/ozon/search/grouping/facet/DocValueSetIterator.reader : Lcom/ozon/search/grouping/facet/DocValueSetReader;
9     ILOAD 1
10    INVOKEINTERFACE com/ozon/search/grouping/facet/DocValueSetReader.advanceTo (I)Z (itf)
11    ISTORE 2
```

# Сравним байткод

Стало

○○○

```
1 public final advanceTo(I)Z throws java/io/IOException
2     ALOAD 0
3     GETFIELD com/ozon/search/grouping/facet/DocValueSetIterator.reader : Lcom/ozon/search/grouping/facet/DocValueSetReader;
4     INVOKESTATIC java/util/Objects.requireNonNull (Ljava/lang/Object;)Ljava/lang/Object;
5     CHECKCAST com/ozon/search/grouping/facet/DocValueSetReader
6
7
8
9     ILOAD 1
10    INVOKEINTERFACE com/ozon/search/grouping/facet/DocValueSetReader.advanceTo (I)Z (itf)
11    ISTORE 2
```

# И обнаруживаем этого товарища

checkcast выполняет  
проверку типа  
объекта на стеке



## checkcast

### Operation

Check whether object is of given type

### Format

```
checkcast  
indexbyte1  
indexbyte2
```

### Forms

checkcast = 192 (0xc0)

### Operand Stack

..., *objectref* →  
..., *objectref*

### Description

The *objectref* must be of type `REFERENCE`. The unsigned *indexbyte1* and *indexbyte2* are used to construct an index into the run-time constant pool of the current class ([§2.6](#)), where the value of the index is  $(\text{indexbyte1} \ll 8) \mid \text{indexbyte2}$ . The run-time constant pool entry at the index must be a symbolic reference to a class, array, or interface type. If *objectref* is `NULL`, then the operand stack is unchanged.

Otherwise, the named class, array, or interface type is resolved ([§5.4.3.1](#)). If *objectref* can be cast to the resolved class, array, or interface type, the operand stack is unchanged; otherwise, the *checkcast* instruction throws a `ClassCastException`.

The following rules are used to determine whether an *objectref* that is not `NULL` can be cast to the resolved type. If *S* is the type of the object referred to by *objectref*, and *T* is the resolved class, array, or interface type, then *checkcast* determines whether *objectref* can be cast to type *T* as follows:

- If *S* is a class type, then:
  - If *T* is a class type, then *S* must be the same class as *T*, or *S* must be a subclass of *T*.
  - If *T* is an interface type, then *S* must implement interface *T*.
- If *S* is an array type `SC[]`, that is, an array of components of type `SC`, then:
  - If *T* is a class type, then *T* must be `Object`.
  - If *T* is an interface type, then *T* must be one of the interfaces implemented by arrays (JLS §4.10.3).
  - If *T* is an array type `TC[]`, that is, an array of components of type `TC`, then one of the following must be true:
    - *TC* and *SC* are the same primitive type.
    - *TC* and *SC* are reference types, and type `SC` can be cast to `TC` by recursive application of these rules.

### Linking Exceptions

During resolution of the symbolic reference to the class, array, or interface type, any of the exceptions documented in [§5.4.3.1](#) can be thrown.

### Run-time Exception

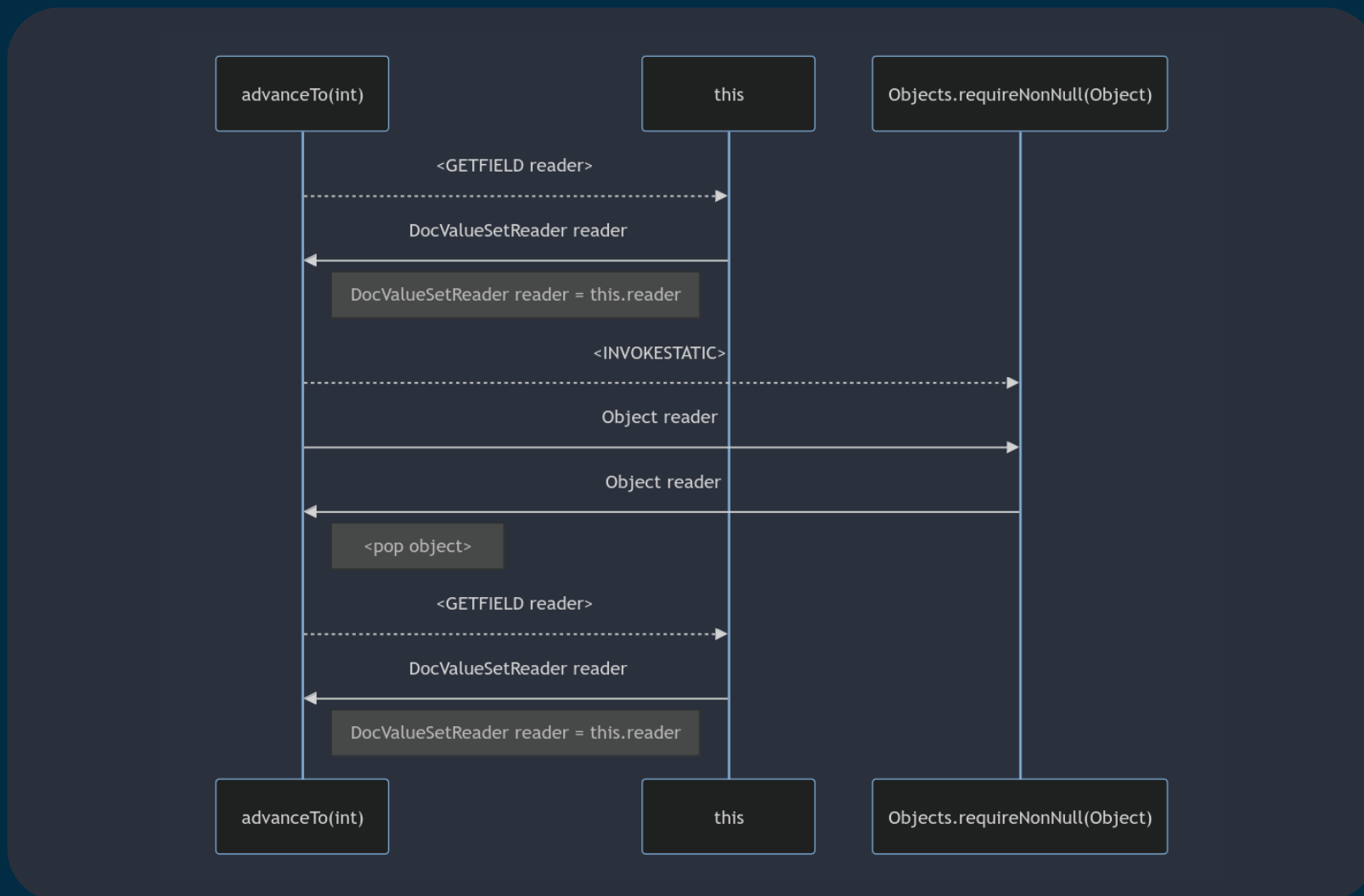
Otherwise, if *objectref* cannot be cast to the resolved class, array, or interface type, the *checkcast* instruction throws a `ClassCastException`.

### Notes

The *checkcast* instruction is very similar to the *instanceof* instruction ([§instanceof](#)). It differs in its treatment of `NULL`, its behavior when its test fails (*checkcast* throws an exception, *instanceof* pushes a result code), and its effect on the operand stack.

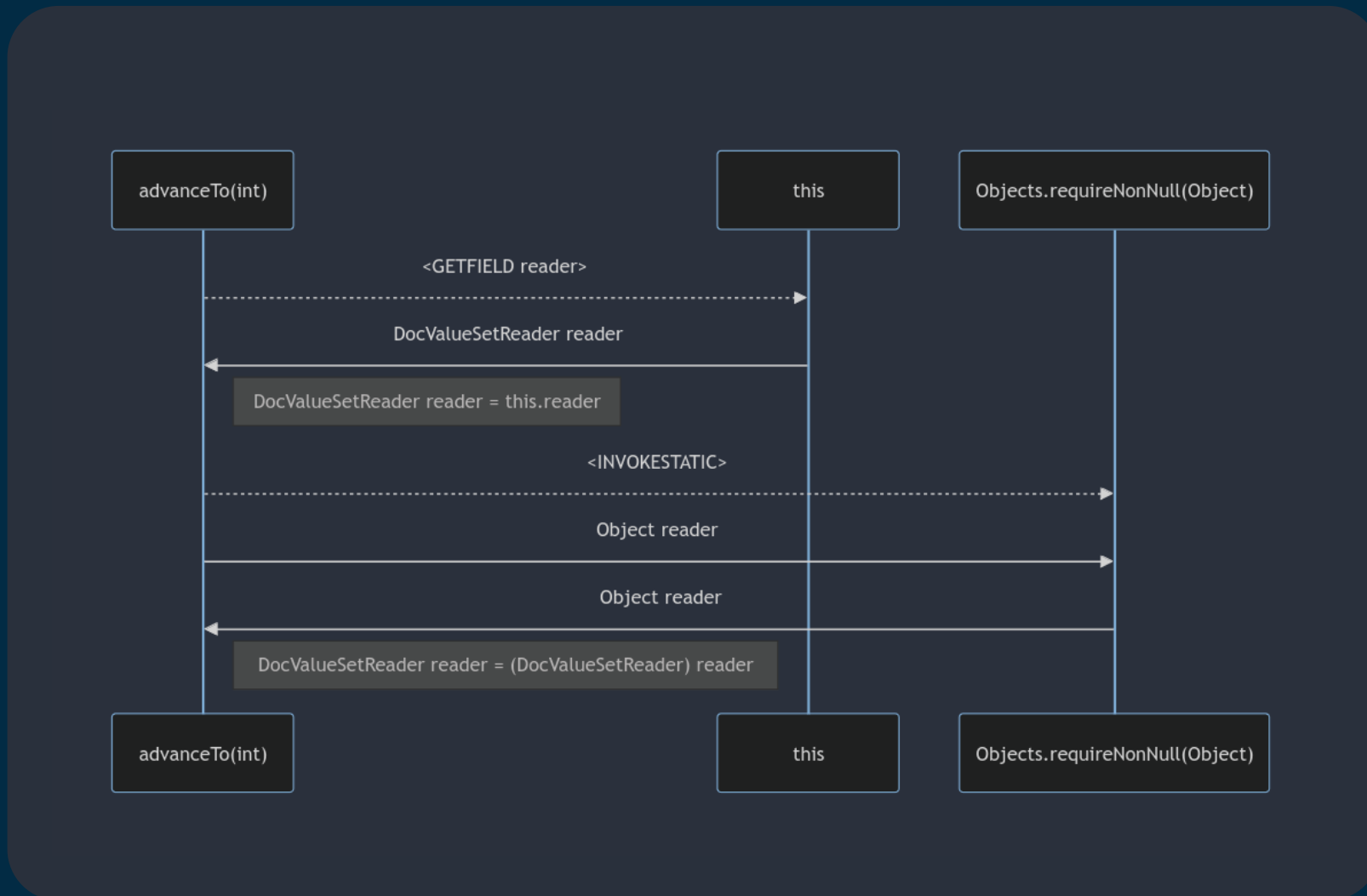
# Было

Два чтения поля конкретного типа



# Стало

Одно чтение поля конкретного типа и потеря этого типа



# Генерики типобезопасны

А рантайм не очень

○○○



```
1 @SuppressWarnings("NotNullFieldNotInitialized")
2 protected TReader reader;
3
4 ...
5
6 @Override
7 public final boolean advanceTo(int docId) throws IOException {
8     var result = ((TReader) Objects.requireNonNull(reader))
9         .advanceTo(docId);
10    ord = result ? 0 : FacetUtils.NO_MORE_ORDS;
11    return result;
12 }
```



Что можно  
улучшить?

# 8282143: Objects.requireNonNull should be ForceInline #7543

<> Code ▾

 Closed merykitty wants to merge 2 commits into `openjdk:master` from `merykitty:requireNonNullsForceInline` 

 Conversation 15  Commits 2  Checks 0  Files changed 1

+2 -0 



Changes from all commits ▾ File filter ▾ Conversations ▾ Jump to ▾ 

0 / 1 files viewed

Review in codespace

Review changes ▾

▾  2  src/java.base/share/classes/java/util/Objects.java 

 Viewed  

⋮ @@ -227,6 +227,7 @@ public static <T> int compare(T a, T b, Comparator<? super T> c) {

```
227     * @return {@code obj} if not {@code null}
228     * @throws NullPointerException if {@code obj} is {@code null}
229     */
```

```
227     * @return {@code obj} if not {@code null}
228     * @throws NullPointerException if {@code obj} is {@code null}
229     */
```

230 + @ForceInline

```
230     public static <T> T requireNonNull(T obj) {
231         if (obj == null)
232             throw new NullPointerException();
```

```
231     public static <T> T requireNonNull(T obj) {
232         if (obj == null)
233             throw new NullPointerException();
```

⋮ @@ -252,6 +253,7 @@ public static <T> T requireNonNull(T obj) {

```
252     * @return {@code obj} if not {@code null}
253     * @throws NullPointerException if {@code obj} is {@code null}
254     */
```

```
253     * @return {@code obj} if not {@code null}
254     * @throws NullPointerException if {@code obj} is {@code null}
255     */
```

256 + @ForceInline

```
255     public static <T> T requireNonNull(T obj, String message) {
256         if (obj == null)
257             throw new NullPointerException(message);
```

```
257     public static <T> T requireNonNull(T obj, String message) {
258         if (obj == null)
259             throw new NullPointerException(message);
```



# Но только попробовать...



JDK / JDK-8180450

## secondary\_super\_cache does not scale well

Log In

Open ▾

### Details

Type: ● Bug

Resolution: Unresolved

Priority: 3 P3

Fix Version/s: tbd

Affects Version/s: 9, 10, 11, 12, 13, 17, 20, 21, 22

Component/s: hotspot

Labels: hs-comp-triaged jdk21-defer-request jdk21-defer-yes oracle-triage-14 performance

Subcomponent: compiler

# Но только попробовать...



JDK / JDK-8180450

## secondary\_super\_cache does not scale well

Resolved ▾

### Details

Type:	Bug	Resolution:	Fixed
Priority:	P3	Fix Version/s:	23
Affects Version/s:	9, 10, 11, 12, 13, 17, 20, 21, 22		
Component/s:	hotspot		
Labels:	<a href="#">amazon-interest</a> <a href="#">hs-comp-triaged</a> <a href="#">jdk21-defer-request</a> <a href="#">jdk21-defer-yes</a> <a href="#">oracle-triage-14</a> <a href="#">performance</a>		
Subcomponent:	compiler		
Resolved In Build:	b19		

# Имя проблеме — type pollution

DEVOXX  
United Kingdom

## Pandora's box

- Netty
- Smallrye Mutiny
- Smallrye Common
- Hibernate ORM
- Hibernate Reactive
- Vert.x
- Vert.x Web
- Quarkus
- Infinispan
- Camel
- Java Class Library
- ...



# RedHat Type Pollution Agent



 [https://github.com/RedHatPerf/  
type-pollution-agent](https://github.com/RedHatPerf/type-pollution-agent)





# Как это исправили мы?

С помощью старого советского...



# [OSRCH-27888] Resolve `DocValueSetIterator.advanceTo` miscompilation

Edit

Code



Merged Portnov Petr Vladimirovich requested to merge OSRCH-27888 into release/2023w3-1 1 year ago

Overview 0 Commits 1 Pipelines 1 Changes 1

Compare release/2023w3-1 and latest version

1 file +5 -4



src/main/java/com/ozon/search/grouping/facet/DocValueSetIterator.java

+5 -4 Viewed

<pre> 1 package com.ozon.search.grouping.facet; 2 3 import java.io.IOException; 4 - import java.util.Objects; 5 import javax.annotation.Nullable; 6 import lombok.NonNull; 7 import org.apache.lucene.index.DocValues; 8 9 @@ -16,10 +15,10 @@ public abstract class DocValueSetIterator&lt;TValue, TReader extends 10 DocValueSetRea 11 12 protected final MappedField mappedField; 13 protected int ord; 14 15 - @SuppressWarnings("NotNullFieldNotInitialized") 16 protected TReader reader; 17 18 - DocValueSetIterator(MappedField mappedField) { 19     this.mappedField = mappedField; 20 } 21 22 @@ -33,7 +32,9 @@ public abstract class DocValueSetIterator&lt;TValue, TReader extends DocValueSetRea 23 } 24 25 public final boolean advanceTo(int docId) throws IOException { 26 -     var result = Objects.requireNonNull(reader).advanceTo(docId); 27 28     ord = result ? 0 : (int) SortedSetDocValues.NO_MORE_ORDS; 29     return result; 30 } </pre>	<pre> 1 package com.ozon.search.grouping.facet; 2 3 import java.io.IOException; 4 import javax.annotation.Nullable; 5 import lombok.NonNull; 6 import org.apache.lucene.index.DocValues; 7 8 @@ -16,10 +15,10 @@ public abstract class DocValueSetIterator&lt;TValue, TReader extends 9 DocValueSetRea 10 11 protected final MappedField mappedField; 12 protected int ord; 13 14 + @Nullable 15 protected TReader reader; 16 17 + protected DocValueSetIterator(MappedField mappedField) { 18     this.mappedField = mappedField; 19 } 20 21 @@ -33,7 +32,9 @@ public abstract class DocValueSetIterator&lt;TValue, TReader extends DocValueSetRea 22 } 23 24 public final boolean advanceTo(int docId) throws IOException { 25 +     assert reader != null : "caller should ensure that `reader` has been set via call to 26 `setNextReader`"; 27 + 28     var result = reader.advanceTo(docId); 29 30     ord = result ? 0 : (int) SortedSetDocValues.NO_MORE_ORDS; 31     return result; 32 } </pre>
---	--



# Правильный рефакторинг

assert'ы — наше всё

○○○

```
1 @Nullable
2 protected TReader reader;
3
4 @Override
5 public final boolean advanceTo(int docId) throws IOException {
6     var reader = this.reader;
7     assert reader != null : "caller should ensure that `reader` has been set via call to `setNextReader`";
8     var result = reader.advanceTo(docId);
9     ...
10 }
11
12 public final void setNextReader(LeafReaderContext readerContext) throws IOException {
13     reader = getReader(readerContext);
14 }
```

# Выводы

01 Чем сложнее рантайм, тем больше в нём «особенностей»



02 Телеметрия спасает время и деньги

03 assert'ы хороши для описания контрактов

04 Для разных проблем разный тулинг

05 UPD: чудеса случаются







# Выводы и советы





Перформанс —  
это не просто,  
но **результаты**  
**того стоят**

01

Важно вдумчиво  
и системно относиться  
к производительности  
продукта



02

Важно отталкиваться  
от своих требований,  
сценариев, окружения  
и целей



03

Чем более сложная  
система, тем больше  
в ней может быть  
неожиданностей



# Выводы

Заключительная  
грустная история /s



# Выводы



**Aleksey Shipilëv**

@shipilev



Я потыкал, вслепую у меня это воспроизвести не удалось. Но я думаю, что бэкпортировать этот патчик не помешает:

[bugs.openjdk.org/browse/JDK-828...](https://bugs.openjdk.org/browse/JDK-828...)

Translate post

7:20 PM · Jan 17, 2023 · **145** Views



# Выводы

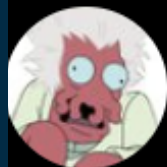


**Jarvis Pogrammer** @PROgrm\_JARvis · Jan 17, 2023

Спасибо за линк, завёл PR с бэкпортом



69



**Aleksey Shipilëv** @shipilev · Jan 17, 2023

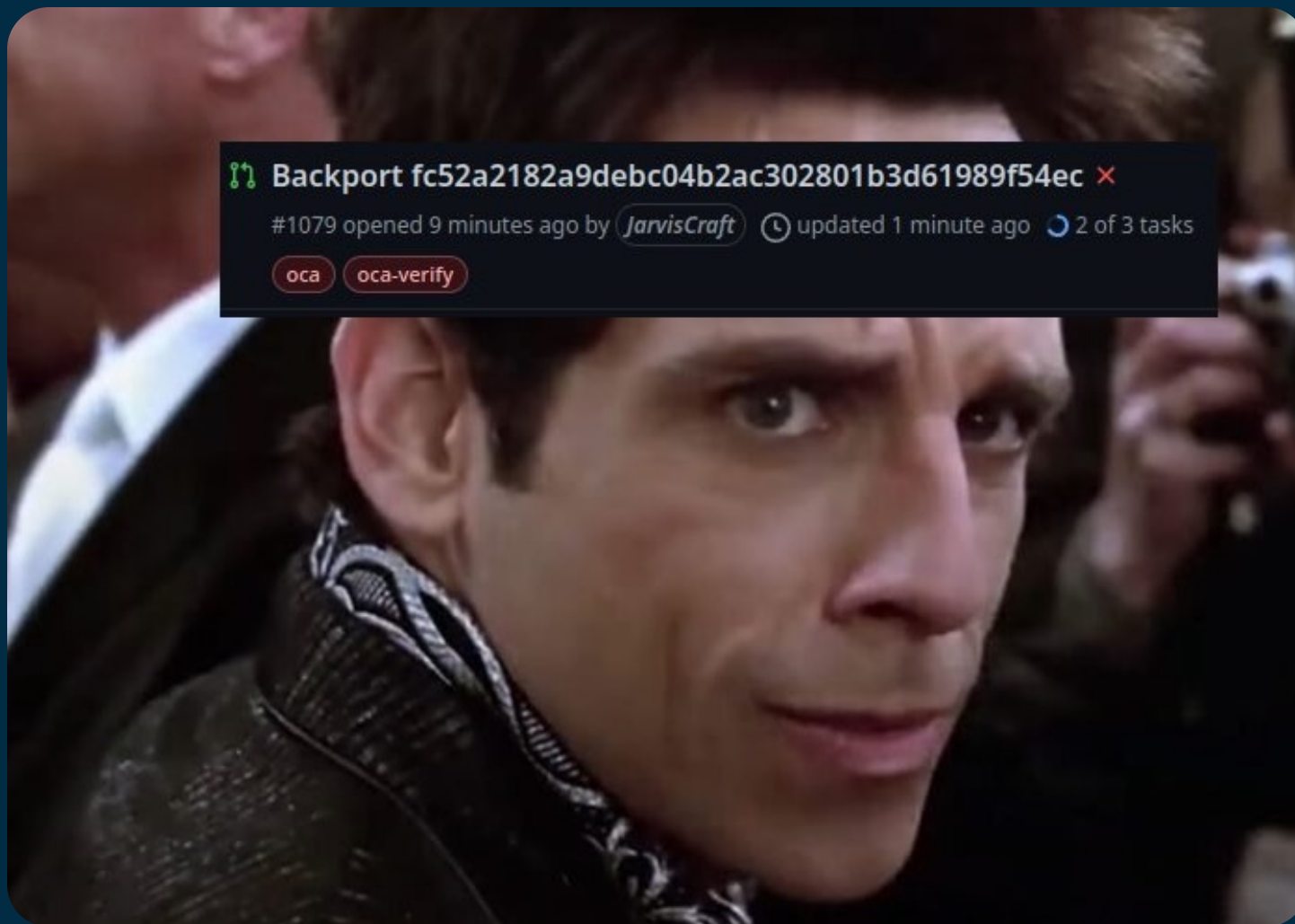
Так вот он! [github.com/openjdk/jdk17u...](https://github.com/openjdk/jdk17u...)



54



# Выводы



# Выводы





Спасибо  
за внимание!

Пётр Портнов  
pportnov@ozon.ru

