
Яндекс

Обо мне

Никита, разработчик в Яндекс Директе

Производительность по запросам

Производительность по запросам

```
〉system.processes; SHOW PROCESSLIST
```

Производительность по запросам

```
〉system.processes; SHOW PROCESSLIST
〉system.query_log
```

Производительность по запросам

```
〉system.processes; SHOW PROCESSLIST
〉system.query_log
〉system.query_thread_log
```

Производительность по запросам

- `system.processes; SHOW PROCESSLIST`
- `system.query_log`
- `system.query_thread_log`
- Текстовый лог сервера

Производительность по запросам

- `system.processes; SHOW PROCESSLIST`
- `system.query_log`
- `system.query_thread_log`
- Текстовый лог сервера
- `perf record`

Сложности

Сложности

〉 Нечеткая разметка затруднений

Сложности

- Нечеткая разметка затруднений
- `perf record` требует воспроизводимого окружения

Что мы хотим?

Что мы хотим?

Дано:

〉 Боевое окружение

Что мы хотим?

Дано:

〉 Боевое окружение

〉 Сервер под "живой" нагрузкой

Что мы хотим?

Дано:

〉 Боевое окружение

〉 Сервер под "живой" нагрузкой

Найти:

〉 Самые "горячие" строчки кода при выполнении запроса X

Poor man's profiler

Poor man's profiler

1. Останавливем процесс

Poor man's profiler

1. Останавливем процесс
2. Получаем stack trace

Poor man's profiler

1. Останавливем процесс
2. Получаем stack trace
3. Запускаем процесс обратно

Poor man's profiler

1. Останавливем процесс
2. Получаем stack trace
3. Запускаем процесс обратно
4. `sleep` и GOTO 1

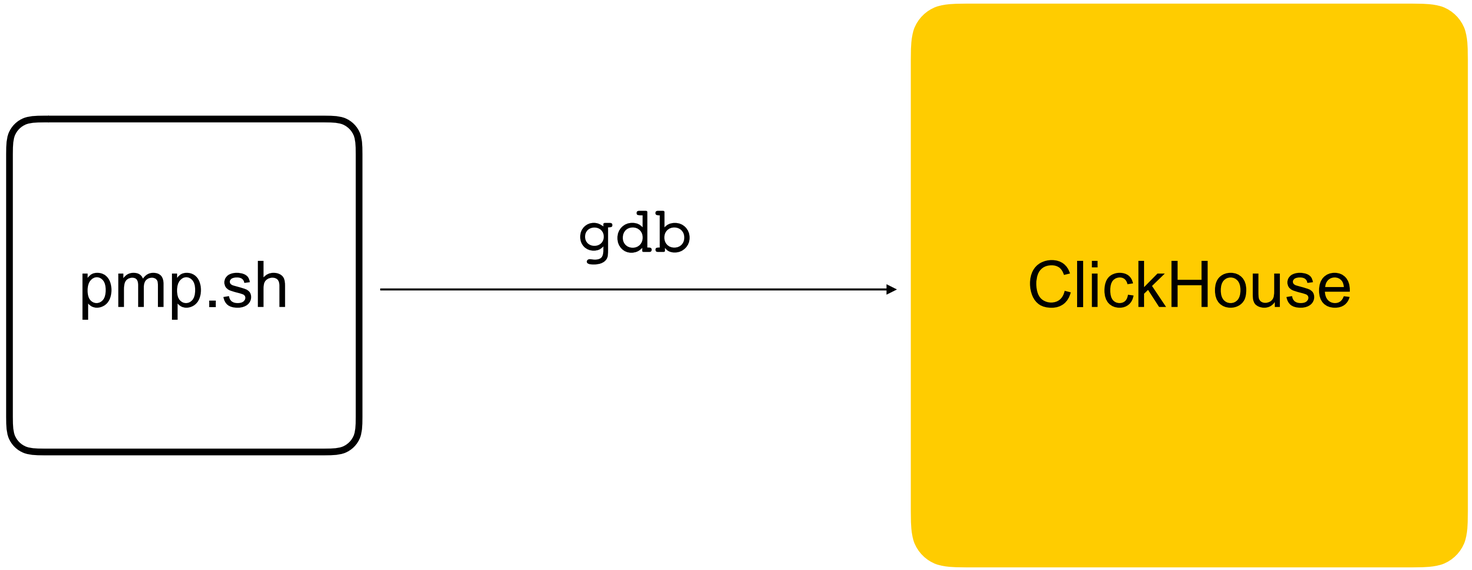
Poor man's profiler

1. Останавливем процесс
2. Получаем stack trace
3. Запускаем процесс обратно
4. `sleep` и GOTO 1
5. Агрегируем результаты

```
for x in $(seq 1 $nsamples)
  do
    gdb -ex "set pagination 0" -ex "thread apply all bt" -batch -p $pid
    sleep 1
  done
```

poormansprofiler.org

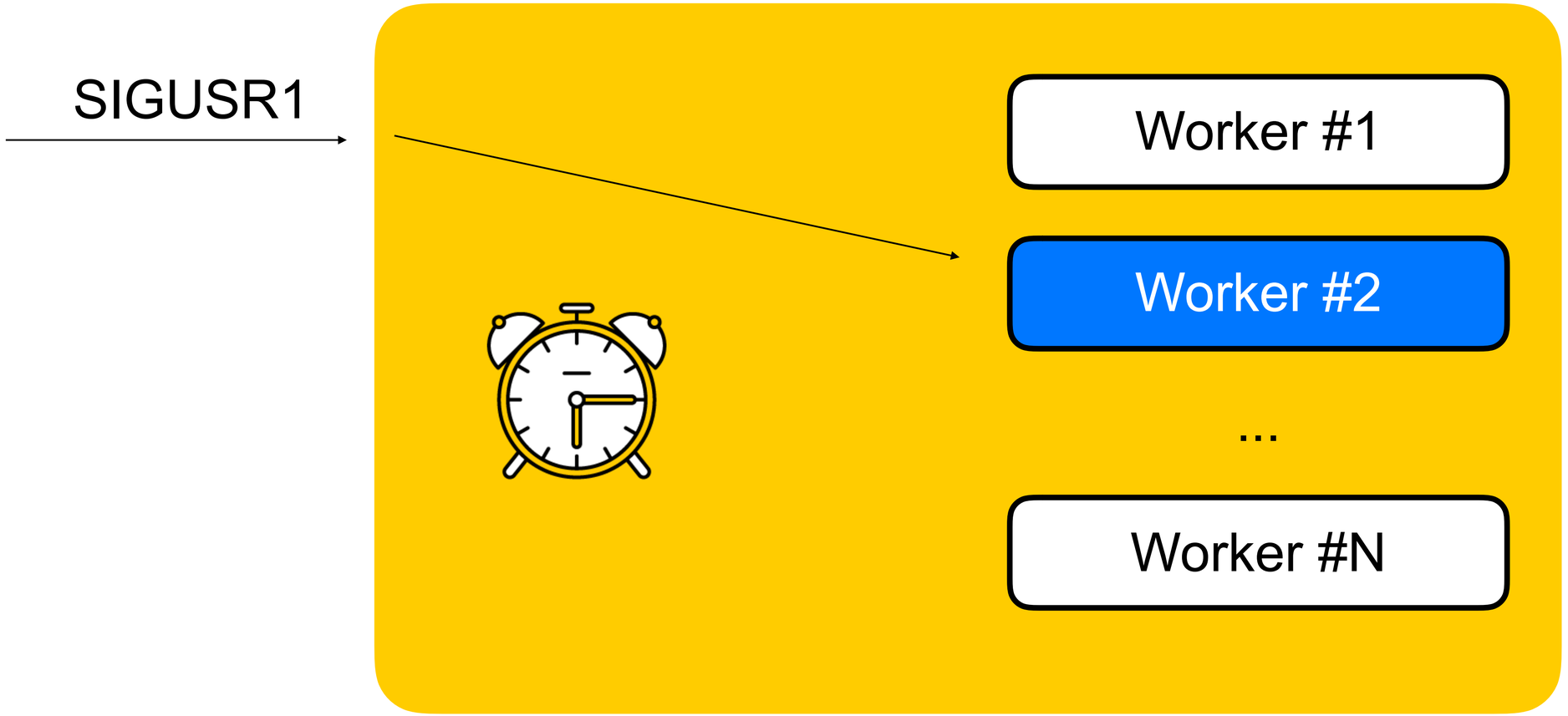
Применение "в лоб"



Применение "в лоб"

- Дорогие stack trace'ы
- Нет query_id

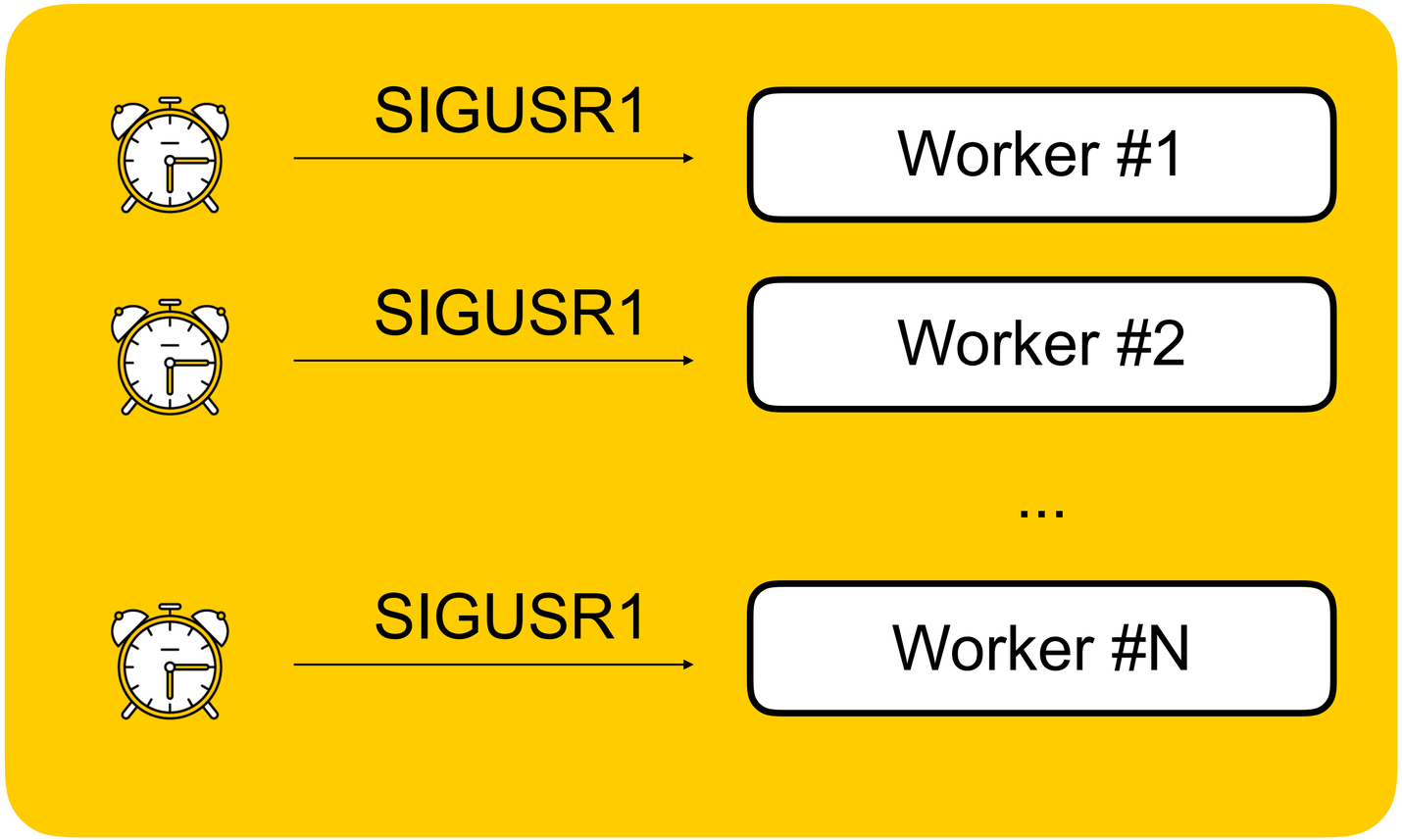
Таймер внутри + сигналы



Таймер внутри + сигналы

〉 Некоторые запросы не получат статистики

Таймер для каждого треда



Таймер для каждого треда

〉Куда писать stack trace'ы?

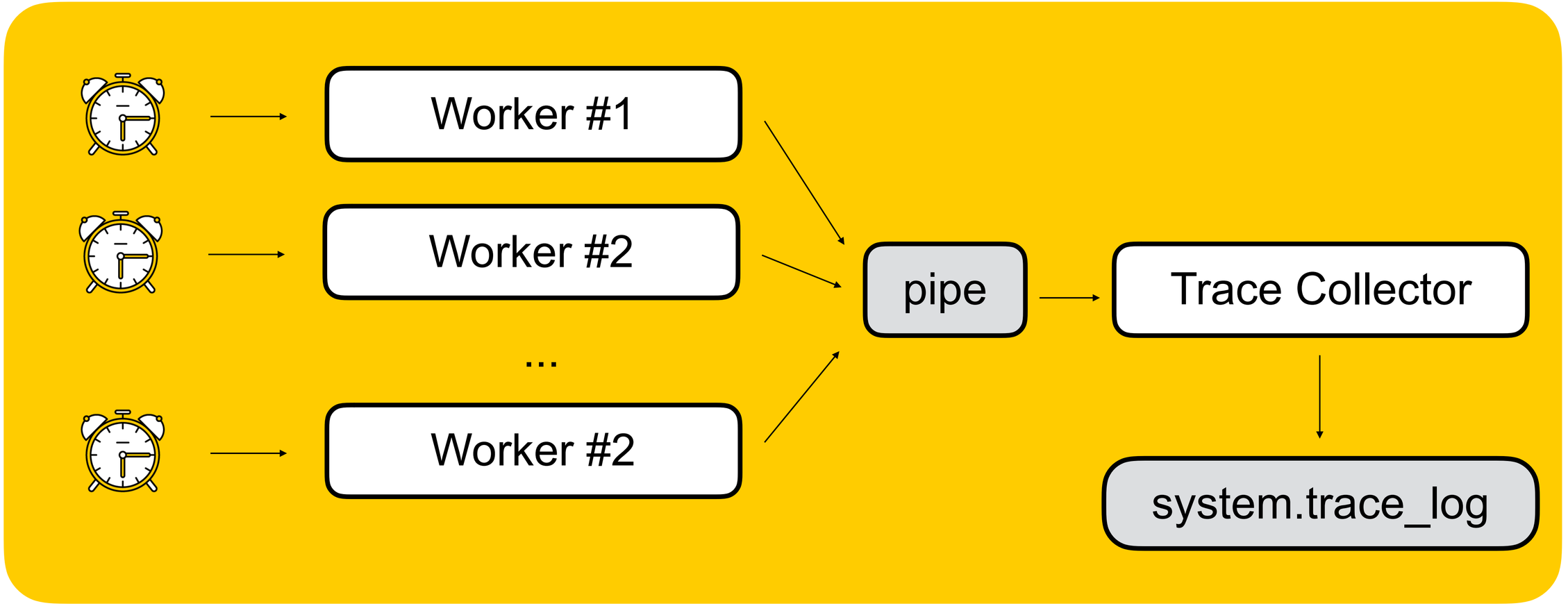
Таймер для каждого треда

〉Куда писать stack trace'ы? **В ClickHouse**

Таймер для каждого треда

〉Куда писать stack trace'ы? В ClickHouse
〉Вызов не signal safe и reentrant функций

Запись в pipe



Технические трудности

Технические трудности

〉 libunwind для раскрутки stack trace'ов

Технические трудности

〉libunwind для раскрутки stack trace'ов

1. Сборка под sanitizers с nongnu libunwind

Как проверить валидность указателя?

Как проверить валидность указателя?

```
bool check_read(void * addr) {
  do {
    ret = write (pipe_fd, addr, 1);
  } while ( errno == EINTR );

  return !ret;
}
```

Технические трудности

〉libunwind для раскрутки stack trace'ов

1. Сборка под sanitizers с nongnu libunwind
2. Внедрение LLVM libunwind

Технические трудности

〉libunwind для раскрутки stack trace'ов

1. Сборка под sanitizers с nongnu libunwind
2. Внедрение LLVM libunwind
3. 17 регистр в x86_64

Технические трудности

〉libunwind для раскрутки stack trace'ов
1. Сборка под sanitizers с nongnu libunwind
2. Внедрение LLVM libunwind
3. 17 регистр в x86_64

〉Перезапуск syscalls

```
void SleepForSeconds(int seconds) {
  ::sleep(seconds);
}

SleepForSeconds(1);
```

```
void SleepForSeconds(int seconds) {
  while (seconds = ::sleep(seconds));
}

SleepForSeconds(1);
```

```
void SleepForSeconds(int seconds) {
  struct timerspec delta = {.tv_sec = seconds, .tv_nsec = 0};
  while (::nanosleep(&delta, &delta));
}

SleepForSeconds(1);
```

```
void SleepForSeconds(int seconds) {
  struct timerspec current_time;
  clock_gettime(CLOCK_REALTIME, &current_time);

  struct timerspec finish_time = current_time;
  finish_time.tv_sec += seconds;

  while (::clock_nanosleep(CLOCK_REALTIME, TIMER_ABSTIME, &finish_time, nullptr));
}

SleepForSeconds(1);
```

Технические трудности

- libunwind для раскрутки stack trace'ов
  1. Сборка под sanitizers с nongnu libunwind
  2. Внедрение LLVM libunwind
  3. 17 регистр в x86_64
- Перезапуск syscalls
- Совместимость с Ubuntu 12.04

Технические трудности

- libunwind для раскрутки stack trace'ов
  1. Сборка под sanitizers с nongnu libunwind
  2. Внедрение LLVM libunwind
  3. 17 регистр в x86_64
- Перезапуск syscalls
- Совместимость с Ubuntu 12.04
- Read after close

```
constexpr int FD_READ = 0;
constexpr int FD_WRITE = 1;

int fd[2];
pipe(fd);

close(fd[FD_READ]);

int data;
read(fd[FD_READ], &data, sizeof(data)); // <-
```

Результат

```
SELECT
    timer_type,
    count(*)
FROM system.trace_log
GROUP BY timer_type
```

timer_type	count()
Real	191379
CPU	81532

Результат

```
SELECT
    length(trace),
    bar(count(*), 0, 100000, 20)
FROM system.trace_log
GROUP BY length(trace)
ORDER BY count(*) DESC
```

length(trace)	bar(count(), 0, 100000, 20)
22	
12	
18	
32	
29	
27	
25	
19	
3	

Результат

```
SELECT
    count,
    symbolizeTrace(trace)
FROM
(
    SELECT
        count(*) AS count,
        trace
    FROM system.trace_log
    GROUP BY trace
    ORDER BY count(*) DESC
    LIMIT 1
)
FORMAT Vertical

Row 1:
──────
count:                 97707
symbolizeTrace(trace): 0. ./clickhouse-server(typeinfo for DB::FunctionHierar
erarchy>+0x16) [0x78e3f8e]
1. ./clickhouse-server(xmlRemoveRef+0x16) [0x6fef8f6]
2. [0xf5a65390]
3. [0xf5a6135e]
4. ./clickhouse-server() [0x838d2ef]
5. ./clickhouse-server(typeinfo name for DB::VisitorImplHelper<DB::GatherUtil
Utils::NullableArraySink<DB::GatherUtils::NumericArraySink<short> >&, DB::ICo
Source<DB::GatherUtils::NumericArraySource<int> >, DB::GatherUtils::NullableA
erUtils::NumericArraySource<float> >, DB::GatherUtils::NullableArraySource<DB
nericArraySource>, DB::GatherUtils::ConstSource<DB::GatherUtils::NumericArray
ort> >, DB::GatherUtils::ConstSource<DB::GatherUtils::NumericArraySource<unsi
atherUtils::ConstSource<DB::GatherUtils::NumericArraySource<signed char> >, D
e<DB::GatherUtils::NumericArraySource<int> >, DB::GatherUtils::ConstSource<DB
```

Яндекс

Спасибо

Никита Лапков

C++ Разработчик

 laplab@yandex-team.ru

 @laplab