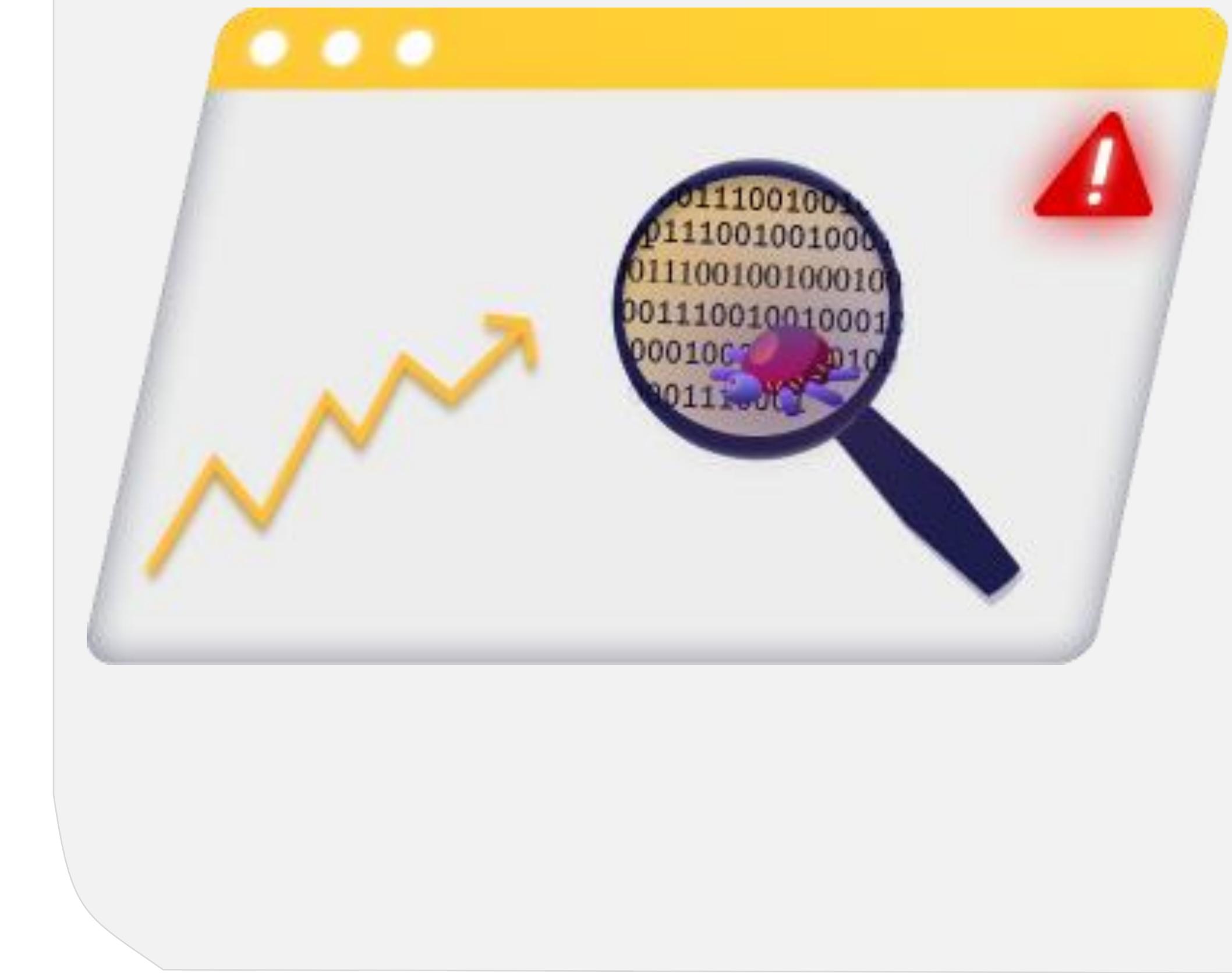


# Анализ дампа: с чего начать?

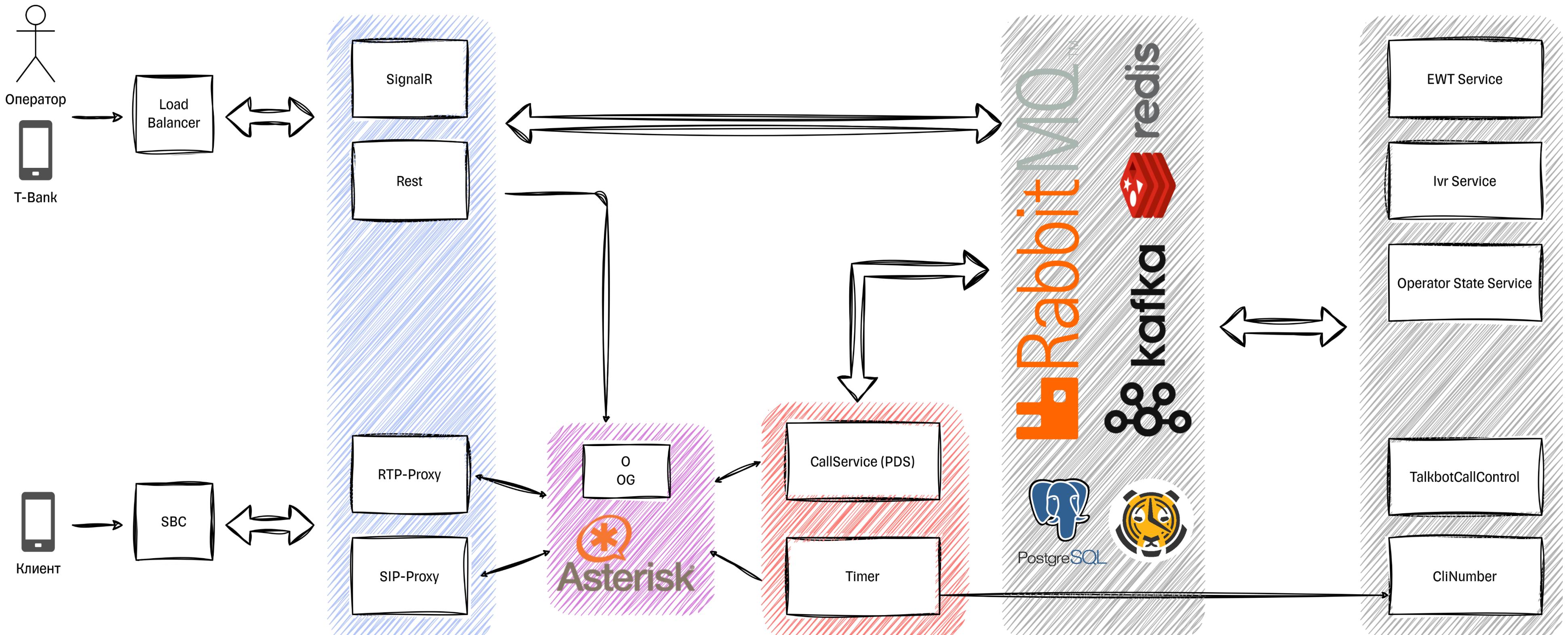
Виктор Греков @vait11



# О проекте

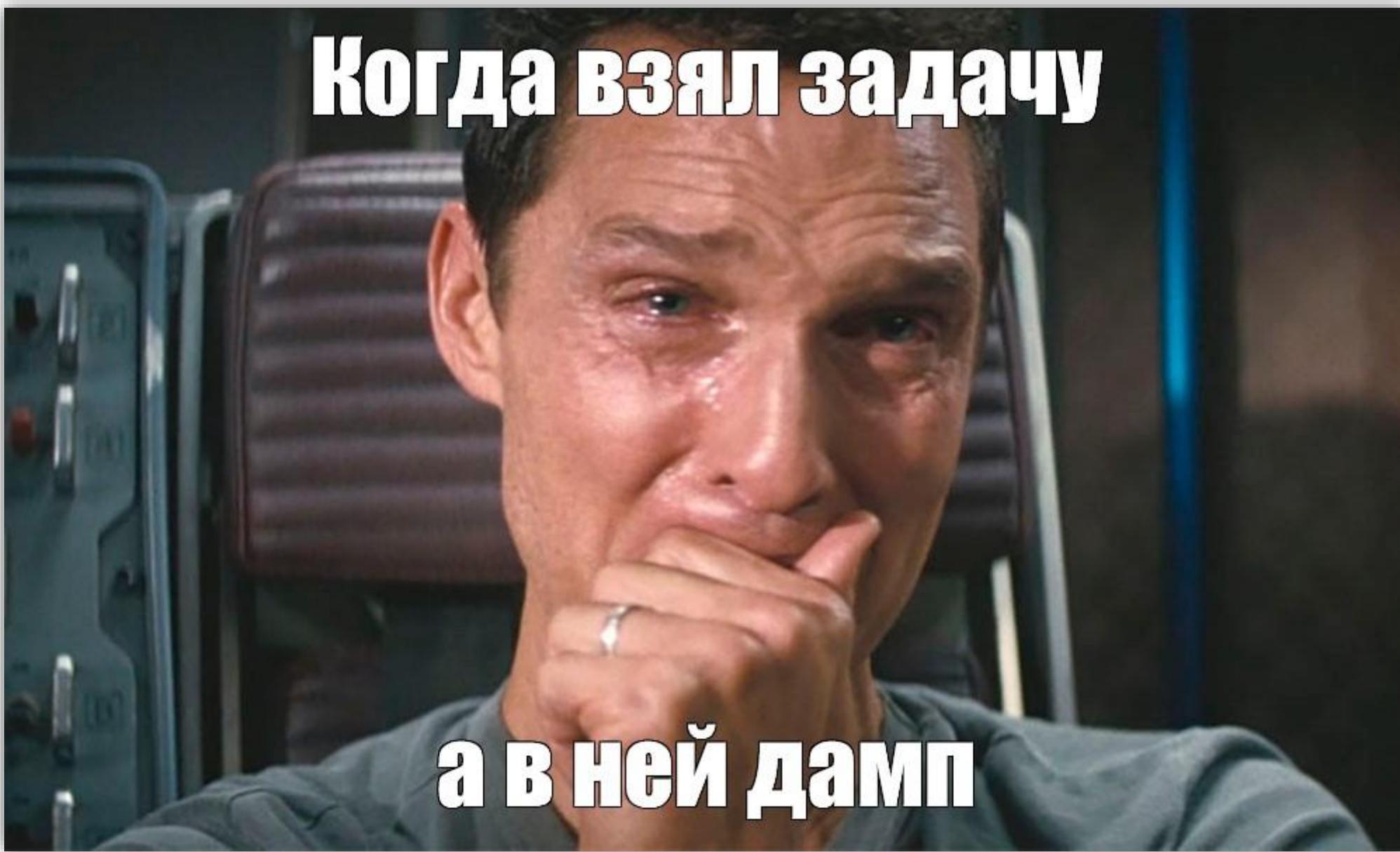
- Обработка входящих и исходящих звонков
- Обрабатываем **>2 млн.** звонков в день
- Критичная для бизнеса система
- SLA – 99,9%
- Нагрузка при любом сбое (UX)

# О проекте



# **Мой путь**

# Мой путь



# Мой путь



Можем в дампы

# **Матчасть**

# **Матчасть**

Программа это?

# Матчасть

Программа это?

Набор инструкций для обработки  
данных.

# Матчасть

Программа это?

Набор инструкций для обработки данных.

Данные хранятся где?

# Матчасть

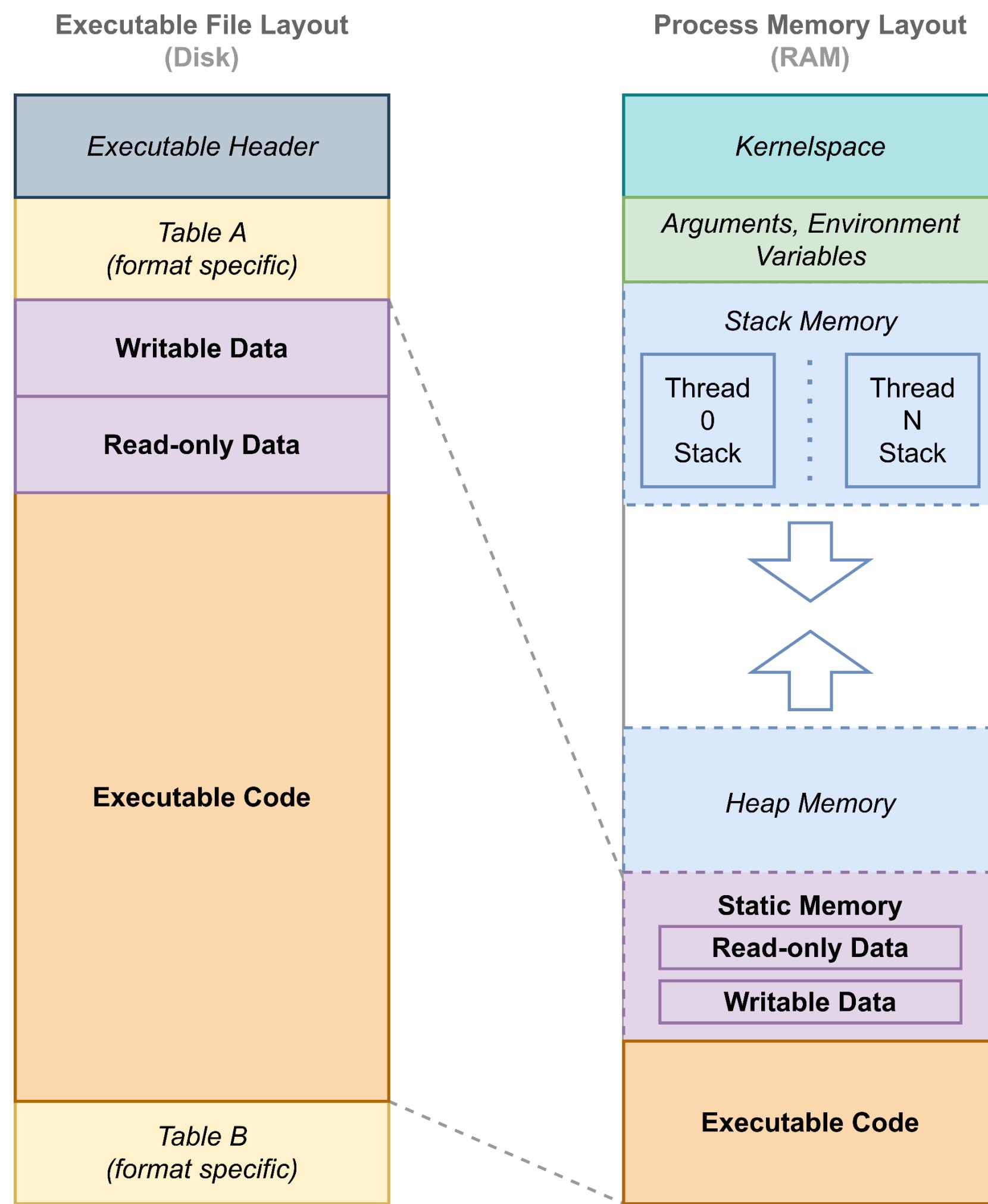
Программа это?

Набор инструкций для обработки данных.

Данные хранятся где?

В памяти (процесса).

# Матчность



# **Матчасть**

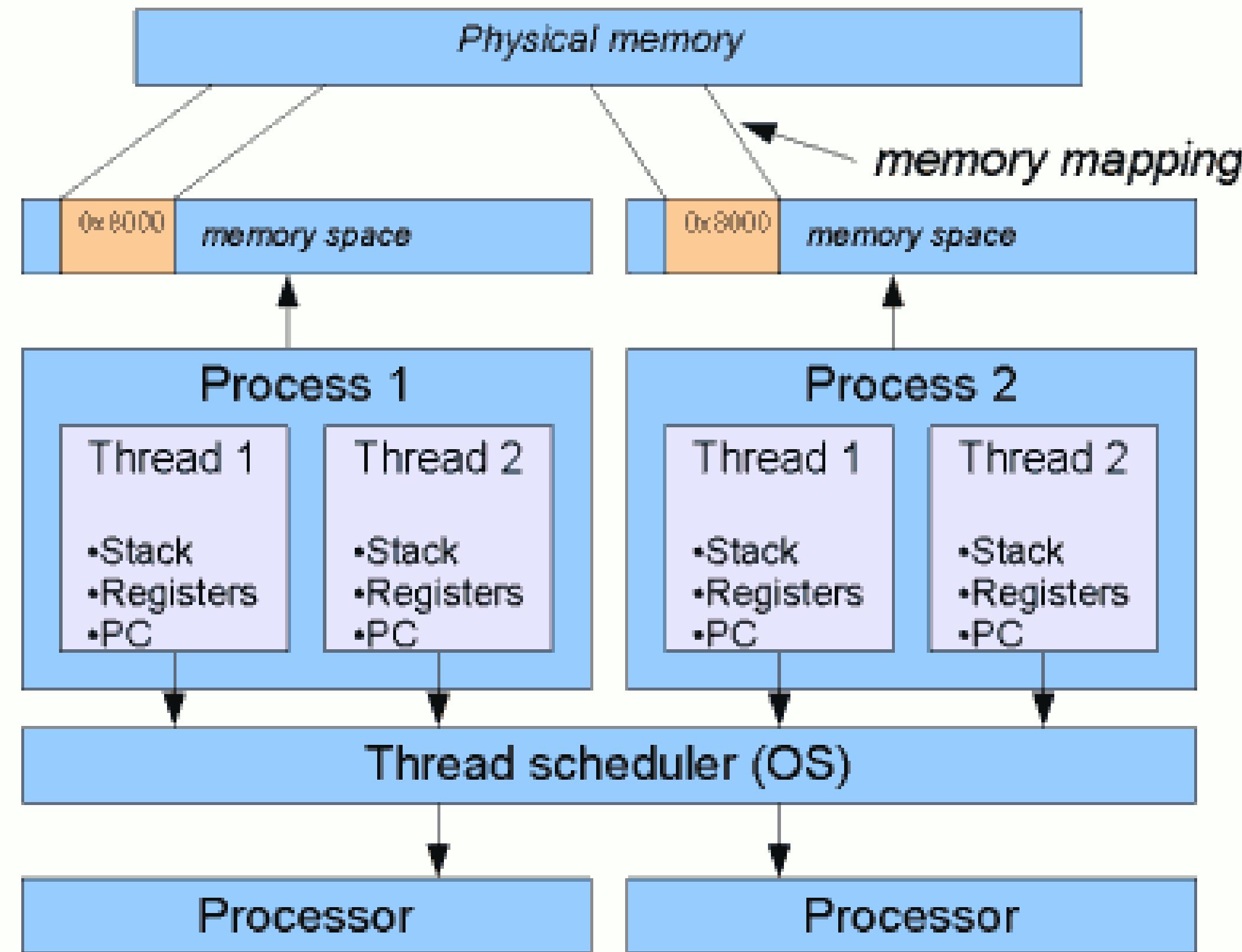
**Поток это?**

# Матчасть

Поток это?

Единица, которой операционная система выделяет процессорное время.

# Матчасть



# **Матчасть**

**Дамп это?**

# **Матчасть**

**Дамп это?**

**Содержимое рабочей памяти процесса в определённый момент времени.**

# Память .NET

# Память .NET

- Управляемая
- Не управляемая

# **Проблемы**

# Возникающие проблемы

- Утекающая память

# Возникающие проблемы

- Утекающая память
- Блокировка потоков

# Возникающие проблемы

- Утекающая память
- Блокировка потоков
- Истощение пула

# Возникающие проблемы

- Утекающая память
- Блокировка потоков
- Истощение пула
- Множественные подключения

# Возникающие проблемы

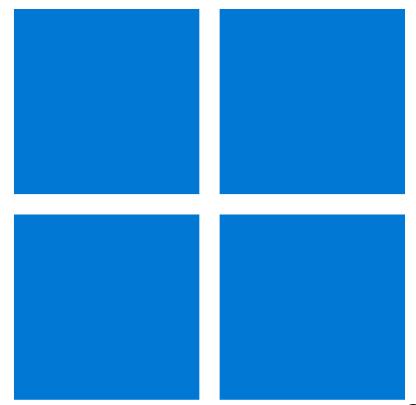
- Память

- Потоки

# **Инструменты**

# Как снимать дамп? Windows

- Диспетчер Задач
- Process Explorer (Sysinternals)
- ProcDump (Sysinternals)
- dotnet-dump



27

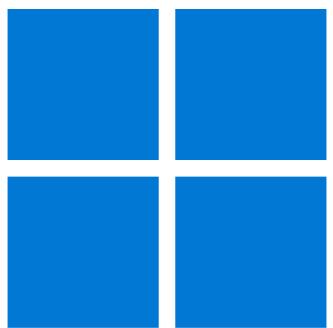
# Как снимать дамп? Unix/Mac версия

- dotnet-dump
- dotnet-trace
- dotnet-stack
- dotnet-gcdump



# Как снимать дамп? Unix/Mac версия

- dotnet-dump
- dotnet-trace
- dotnet-stack
- dotnet-gcdump



Mac OS

# Как снимать дамп? Unix версия

- ProcDump



# Как снимать дамп? Docker версия

- dotnet-monitor



# Как снимать дамп? dotnet-monitor

<https://domain/diag/dotnet-monitor/<docker-host>/dump?type=Full&uid={UID}>

Основное API:

**..*/swagger/index.html*** - swagger - тут можно найти все доступные entrypoint

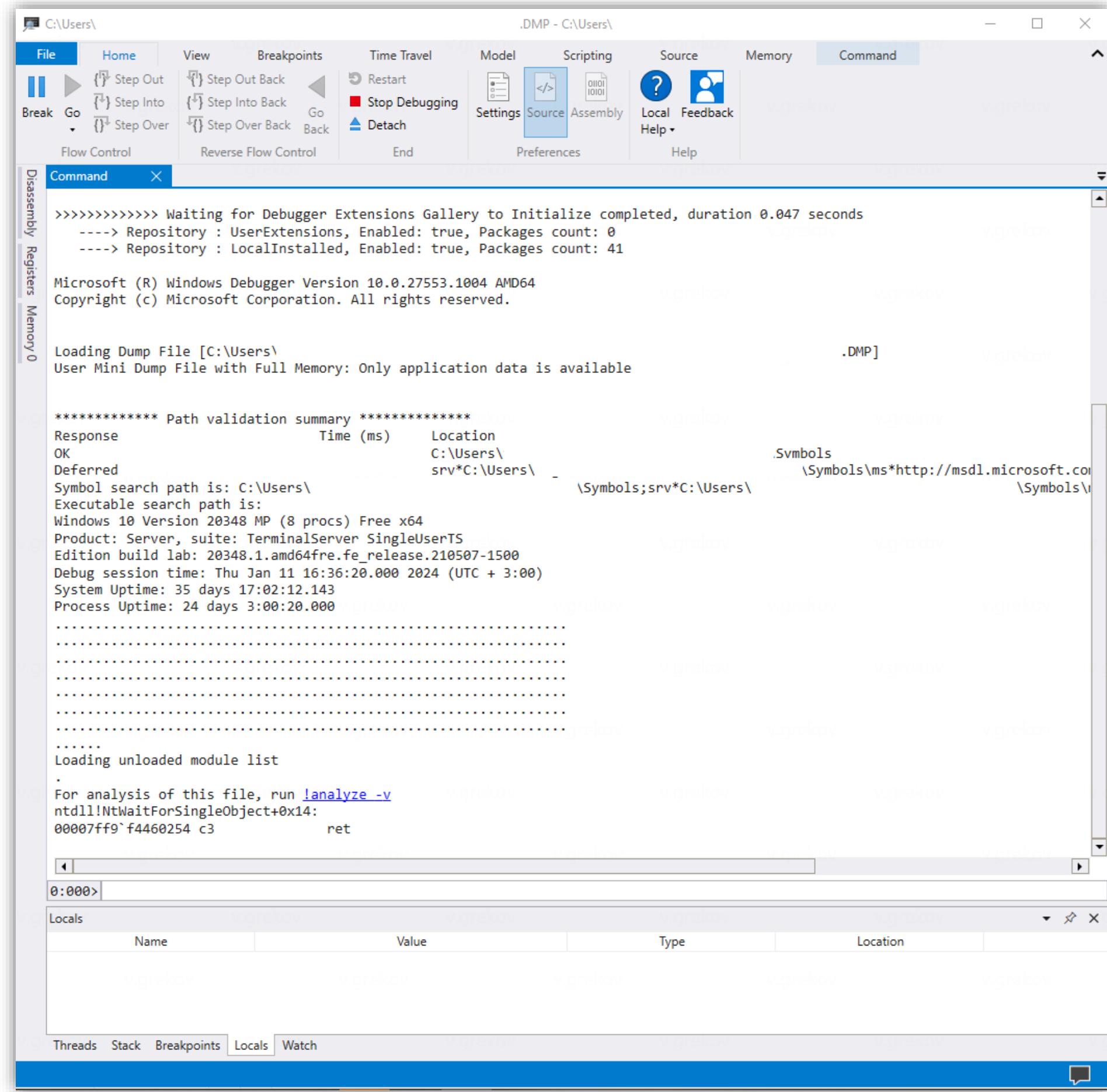
**..*/processes*** - отобразить процессы доступные для дампа

**..*/process?uid={UID}*** - отобразить информацию о процессе

**..*/dump?type=Full&uid={UID}*** - снятие полного дампа процесса по его UID

# Инструменты.

# WinDbg (Microsoft)



# Инструменты.

# dotMemory (JetBrains)

This view shows an overview of snapshot data and results of automatic memory inspections

**Largest Size**

Type	Size
All objects	139,76 MB
Byte[]	52,72 MB
Char[]	16,25 MB
SByte[]	10,34 MB
RawResult[]	8,09 MB
String	4,50 MB

**Largest Retained Size**

Type	Size
All objects	139,76 MB
FileTarget	57,02 MB
Object[]	21,14 MB
TaskExceptionHolder	8,67 MB
PhysicalConnection	7,90 MB
AsyncTargetWrapper	4,43 MB

**String duplicates** 1,73 MB wasted by 36 328 objects

A large number of strings with the same value is inefficient from the point of memory usage

Value	Wasted	Count	Length
"data source=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCI	161,8 KB	331	240
"t	57,5 KB	796	26
"Нс	42,9 KB	298	63
"До	39,9 KB	293	59
"An existing connection was forcibly closed by the remote host."	31,4 KB	221	62

**Sparse arrays** 54,36 MB wasted by 3 608 objects

Partially filled arrays are inefficient from the point of memory usage

Type	Wasted	Fill rate	Length
Byte[]	26,60 MB	17,4%	33 782 190
Char[]	3,36 MB	70,7%	12 029 278
SByte[]	905,9 KB	55,8%	2 097 152
RawResult[]	556,8 KB	0,6%	71 677
Total	224,0 KB	0,0%	4 096

**Finalizable objects** 7 109 objects of 31 type(s) found

Objects that were queued for finalization or already finalized since the previous snapshot. It might be more efficient if these objects were disposed via the IDisposable interface.

Type	Count
TaskExceptionHolder	1
SafeWaitHandle	1
OracleLogicalTransaction	1
WeakReference<KestrelConnection>	1
ThreadLocal<ConcurrentBag<WorkStealingQueue<Package>>	1

**Event handlers leak** 496 B retained by 6 objects

Objects subscribed to an event of another object and never unsubscribed from this event

Type	Count
InterceptorExtensions+InterceptionHelper	1
JsonSerializeExtensions+<>c_DisplayClass4_0	1
SimpleInjectorServiceCollectionExtensions+<>c_DisplayClass18_1	1

**WPF binding leak** No issues found

**Dependency property leak** No issues found

**WPF collection binding leak** No issues found

**x:Name WPF leak** No issues found

**Types**

Type	Address	References count	Bytes	Retained bytes
TelephonyServerService	00000237868297c8	11	104	27 424
_defaultPredicate Expression1<Func<TelephonyServiceDao, Boolean>>	00000237868298c8	2	32	184
_logger TcsLogger	00000237868299e8	1	24	648
_cacheClient MemoryStorage	000002380685f948	4	56	796 040
_telephonyServerDb TelephonyServerDb	0000023786817388	3	40	112
_asteriskTelephonyServiceConfigDb AsteriskTelephonyServiceConfigDb	0000023786822e48	3	40	72
_telephonyServiceRestartDb TelephonyServiceRestartDb	0000023786823680	3	40	40
_contextFactory TelephoneContextFactory	000002370634f940	4	48	48
_logger TcsLogger	000002370634ffcc8	1	24	648
_properties PropertyInfo[0]	0000023506371050	24	24	24
_validator TelephoneServiceValidator	0000023786829c18	7	72	26 488
<Rules>k_BackingField TrackingCollection<ValidationRuleInternal<TelephoneServiceDao>>	0000023786829c90	1	40	6 458
_innerCollection List<ValidationRuleInternal<TelephoneServiceDao>>	0000023786829cb8	1	32	6 418
Fields				
_items IValidationRuleInternal<TelephoneServiceDao>[8]	0000023786838368	6	88	6 386
[0] PropertyRule<TelephoneServiceDao, String>	0000023786835b00	9	136	720
[1] PropertyRule<TelephoneServiceDao, String>	0000023786836300	9	136	2 056
[2] PropertyRule<TelephoneServiceDao, AsteriskTelephonyServiceConfigDao>	0000023786837020	9	136	940
[3] PropertyRule<TelephoneServiceDao, TelephoneTypeEnum>	0000023786837940	9	136	770
[4] PropertyRule<TelephoneServiceDao, String>	0000023786838228	9	136	906
[5] PropertyRule<TelephoneServiceDao, String>	0000023786838650	9	136	906
_classLevelCascadeMode Func<CascadeMode>	0000023786829d08	1	64	64
_ruleLevelCascadeMode Func<CascadeMode>	0000023786829d48	1	64	64
_telephonyServerDb TelephonyServerDb	0000023786817388	3	40	112
_fileStorageService FileStorageService	0000023786824290	4	48	712
_serviceGroupLinkDb TelephonyServiceGroupLinkDb	0000023786824dd0	3	40	88
_configValidators Dictionary<TelephoneTypeEnum, AsteriskTelServiceConfigValidator>	000002378682a1a0	2	80	19 958
_splunkAdapter BusinessLogAdapterEnrichDataAdapter	0000023806875db8	2	32	592
_atsQueueFallbackService AtsQueueFallbackService	000002378681e018	7	72	720
_wssUriDb TeleCrudRepository<WssUriDao>	0000023786825bd0	3	40	72
? _callTransferConfig <Unresolved>@00003f05	00000237868290a0	2	48	564

# Инструменты.

# Microsoft.Diagnostics.Runtime

```
//Для поиска родителя
session.CreateReferenceMapping();
Console.WriteLine($"End {nameof(session.CreateReferenceMapping)} {DateTime.Now}");
var atsActivityEventProcessorType = session.Heap.GetTypeByName("EventProcessor");

foreach (var entry in session.Heap.EnumerateObjects().Where(x => x.Type != null && x.Type == atsActivityEventProcessorType))
{
    counter++;
    var _taskQueue = entry.GetFieldValue("_taskQueue").obj;
    var _tasks = _taskQueue?.GetFieldValue("_tasks").obj;
    var _defaultTarget = _tasks?.GetFieldValue("_defaultTarget").obj;
    var _spscTarget = _tasks?.GetFieldValue("_spscTarget").obj;

    var target = _defaultTarget ?? _spscTarget;
    var _completionSource = target?.GetFieldValue("_completionSource").obj;
    var m_task = _completionSource?.GetFieldValue("m_task").obj;
    var m_stateFlags = m_task?.GetField<Int32>("m_stateFlags");
    var status = ((m_stateFlags & 2097152) == 0 ? ((m_stateFlags & 4194304) == 0 ? ((m_stateFlags & 16777216) == 0 ? ((m_stateFlags & 838
        var cacheEntryDyn = entry.GetDynamic();
        var ownersDyn = session.GetReferenceBy(cacheEntryDyn).ToArray();

        if (ownersDyn.Length != 1)
        {
            notOneParent++;

            if (ownersDyn[0].TypeName == ownersDyn[1].TypeName)
            {
                ++sameTypeOfParents;
            }
        }

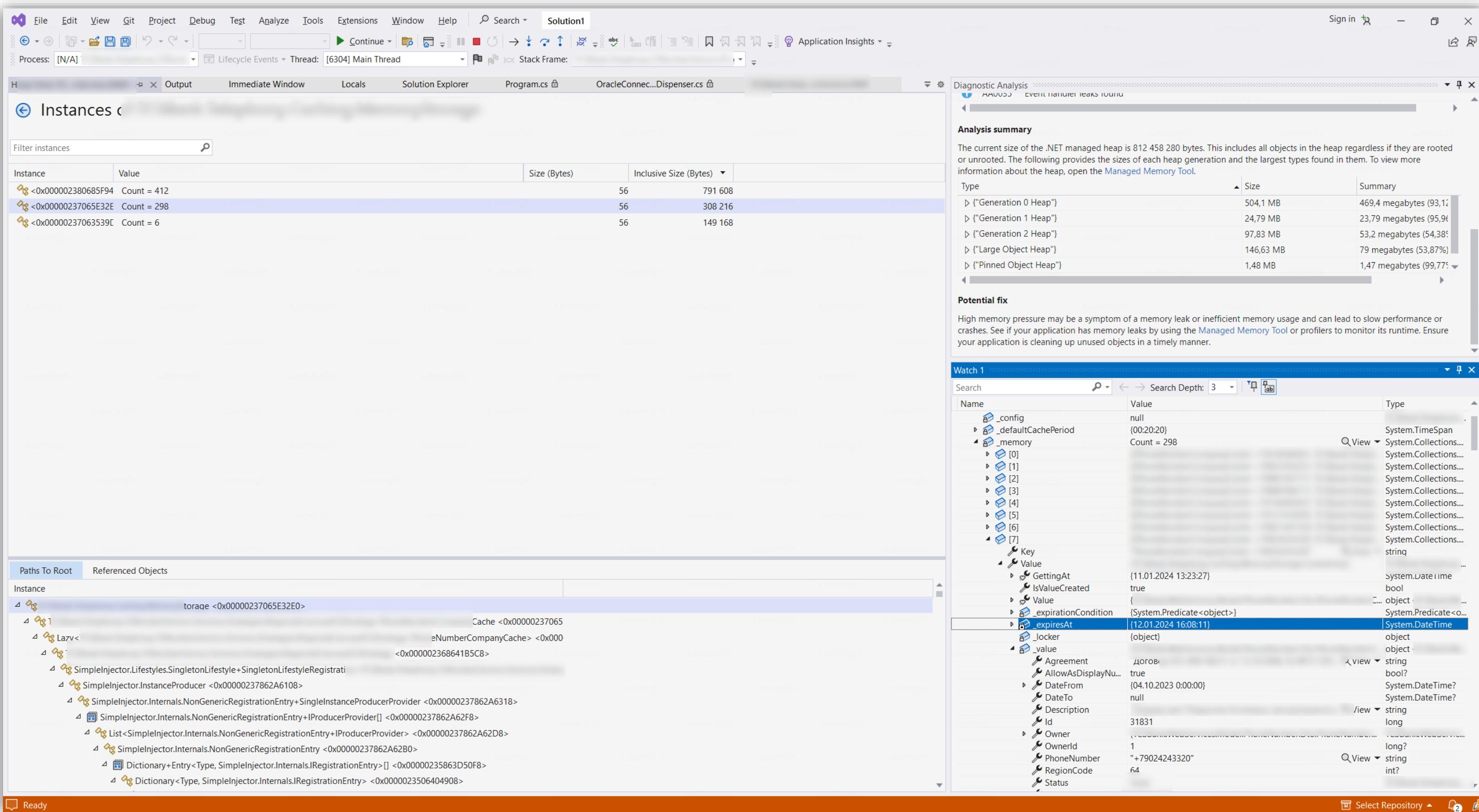
        if (status != TaskStatus.WaitingForActivation)
        {
            ++otherState;
        }

        if (status == TaskStatus.Faulted)
        {
            ++failed;
        }

        if (status != TaskStatus.WaitingForActivation || ownersDyn.Length != 1)
    }

    Console.ForegroundColor = ConsoleColor.Red;
    Console.WriteLine($"Total objects: {counter}");
    Console.WriteLine($"Total otherState: {otherState}");
    Console.WriteLine($"Total notOneParent: {notOneParent}");
    Console.WriteLine($"Total sameTypeOfParents: {sameTypeOfParents}");
    Console.WriteLine($"Total failed: {failed}");
}
```

# Другие инструменты



# Другие инструменты

## Еще раз от Microsoft

<https://learn.microsoft.com/en-us/dotnet/core/diagnostics/tools-overview>

\*dotnet-counters    \*dotnet-dump    \*dotnet-gcdump    \*dotnet-monitor  
\*dotnet-trace        \*dotnet-stack    \*dotnet-symbol

## Linux

<https://learn.microsoft.com/ru-ru/dotnet/core/diagnostics/debug-linux-dumps>

снятие и анализ дампов в Linux

## LLDB

<https://lldb.llvm.org/index.html>

<https://learn.microsoft.com/en-us/troubleshoot/developer/webapps/aspnetcore/practice-troubleshoot-linux/lab-1-2-analyze-core-dumps-lldb-debugger>

## PerfView

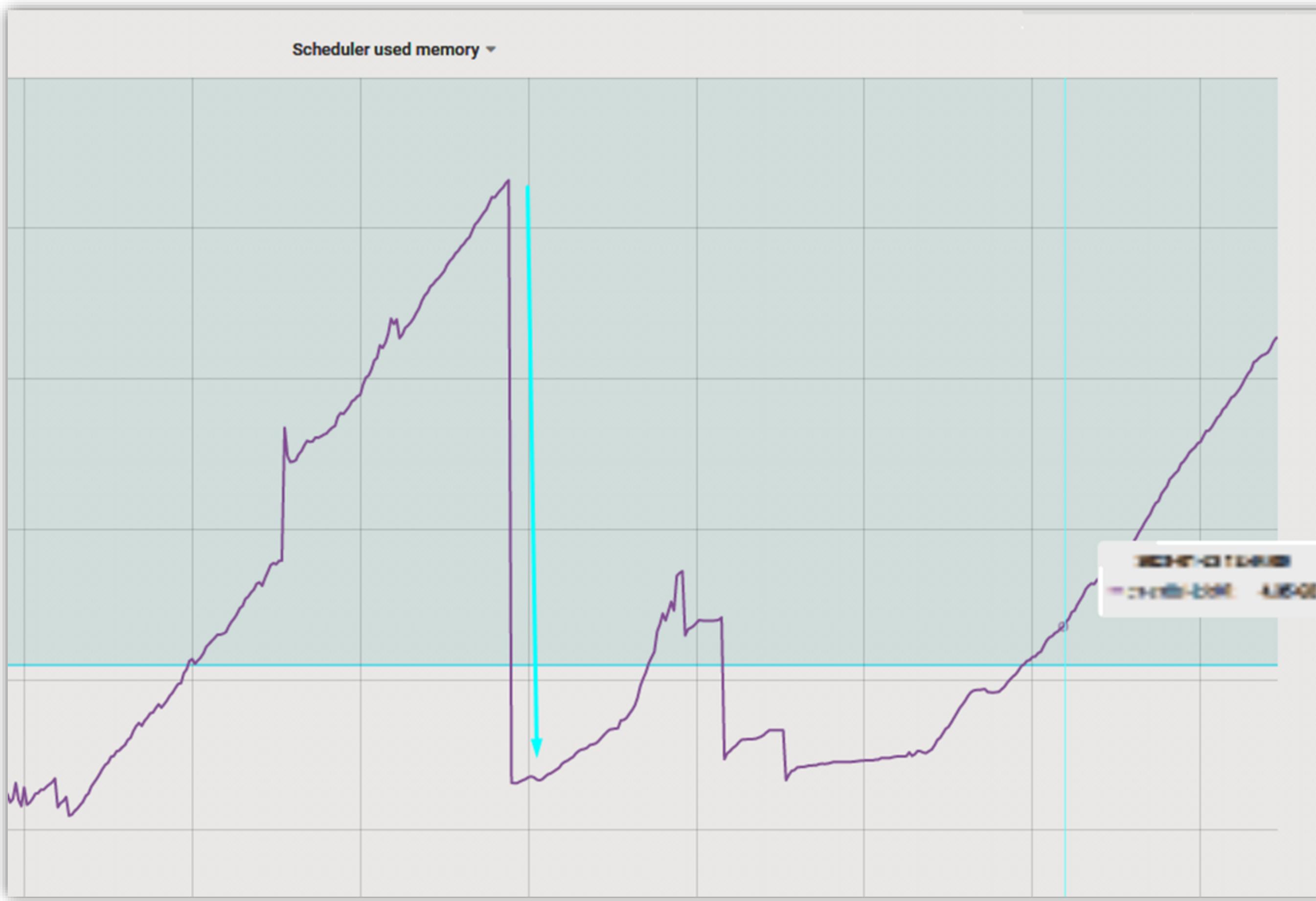
<https://learn.microsoft.com/ru-ru/shows/perfview-tutorial/>

<https://github.com/microsoft/perfview>

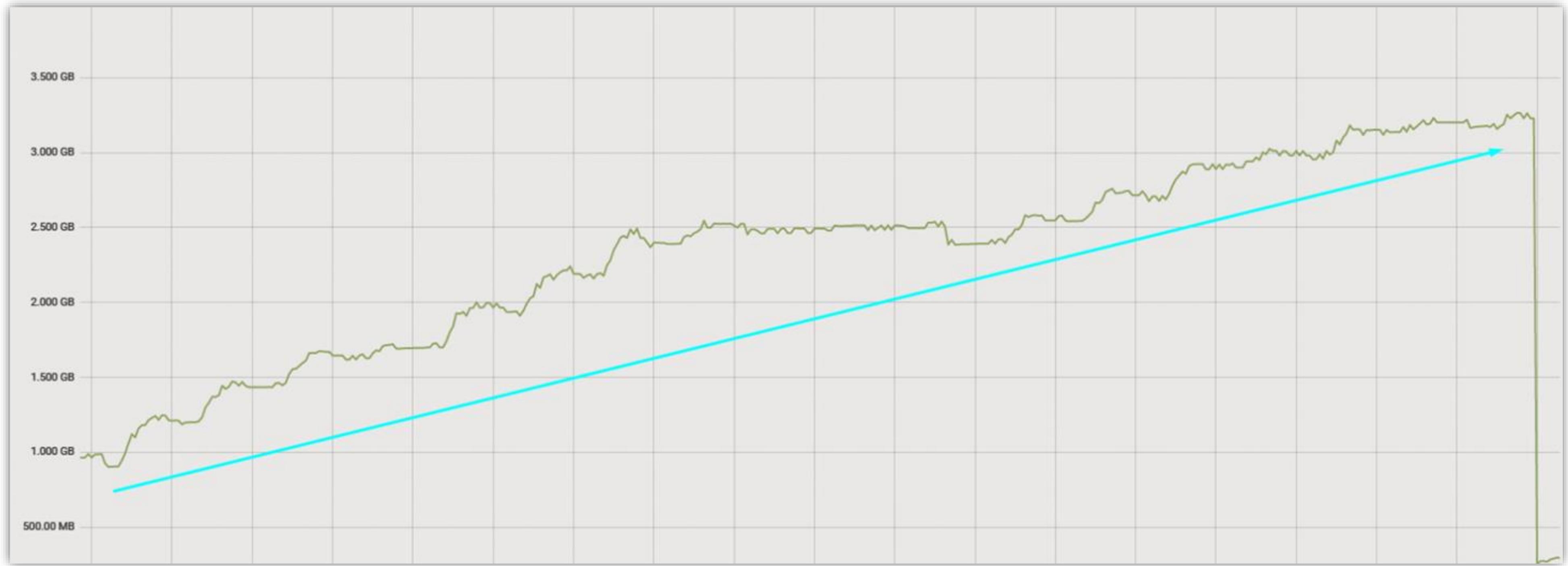


# **Примеры**

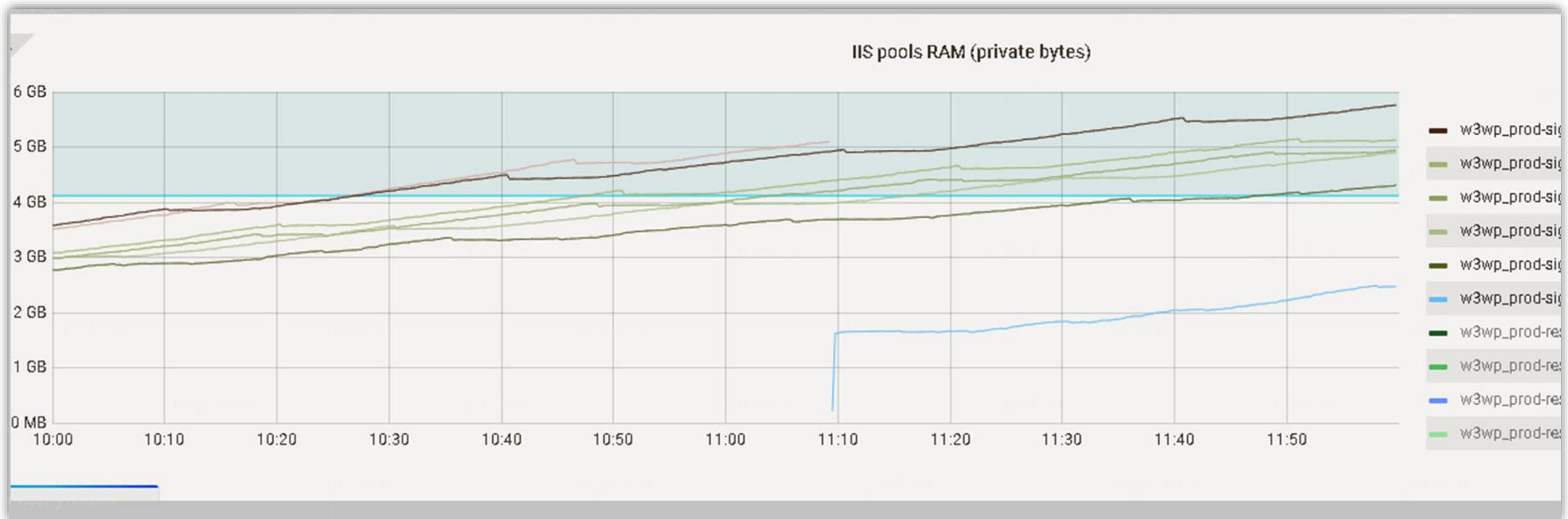
# Память



# Память



# Пример 1



# Пример 1

This view shows an overview of snapshot data and results of automatic memory inspections.

### Largest Size

All objects **3,68 GB**

1,49 GB	<a href="#">String</a>
198,17 MB	<a href="#">Hashtable+bucket[]</a>
154,50 MB	<a href="#">Object[]</a>
143,41 MB	<a href="#">ArrayList</a>
129,21 MB	<a href="#">String[]</a>

### Largest Retained Size

All objects **3,68 GB**

3,53 GB	1× <a href="#">MqListenerService</a>
22,78 MB	2× <a href="#">MemoryStorage</a>
20,51 MB	8 437× <a href="#">Topic</a>
18,06 MB	96× <a href="#">PhysicalConnection</a>
6,61 MB	952× <a href="#">HttpRequest</a>

### String duplicates [?](#)

A large number of strings with the same value is inefficient from the point of memory usage.

	Wasted	Count
"h-UserSingleConnectionHub"	42,33 MB	173370
"pc-Microsoft.AspNet.SignalR.Hubs.HubDispatcher"	29,16 MB	402274
"hg-UserSingleConnectionHub"	16,67 MB	213146
"pc-Microsoft.AspNet.SignalR.Hubs.HubDispatcher"	13,16 MB	116966
"hg-UserSingleConnectionHub"	12,72 MB	171025

### Sparse arrays [?](#)

Partially filled arrays are inefficient from the point of memory usage.

	Wasted	Filling
↳ Object[]	712,3 KB	65,2%
↳ Object[]	712,0 KB	65,2%
↳ Object[]	711,9 KB	65,2%
↳ Dictionary+Entry<Type, MruCache+MruCacheItem<Type, ObjectReflec	473,6 KB	0,0%

# Пример 1

Here you can view what objects have references to the instance. 

Page 1 of 343

Type	Address	References count	Bytes	Retained bytes
▲  MqListenerService ( Services)	00000030ebcd9d20	171 057	112	3 793 544 157
▷  _mqListenerService UserSingleConnectionHub	0000003005a7f1f8	3	136	1 348
▷  _mqListenerService UserSingleConnectionHub	0000002df8ded188	4	136	31 821
▷  _mqListenerService UserSingleConnectionHub	0000002ff336d500	4	136	31 067
▷  _mqListenerService UserSingleConnectionHub	0000002f7c6f5798	3	136	42 086
▷  _mqListenerService UserSingleConnectionHub	0000002e711f6318	4	136	30 165
▷  _mqListenerService UserSingleConnectionHub	0000002df336ca28	3	136	1 372
▷  _mqListenerService UserSingleConnectionHub	0000002f76c75538	3	136	1 372
▷  _mqListenerService UserSingleConnectionHub	0000002eed8ebec0	3	136	1 372
▷  _mqListenerService UserSingleConnectionHub	0000002dfe86a738	3	136	1 372
▷  _mqListenerService UserSingleConnectionHub	0000002eed8eba50	3	136	1 372
▷  _mqListenerService UserSingleConnectionHub	0000003101e2b8e0	3	136	1 348
▷  _mqListenerService UserSingleConnectionHub	00000030f8dea760	3	136	1

dotMemory

# Пример 1

Type	Address	References count	Bytes	Retained bytes
MqListenerService (alR.Services)	00000030ebcd9d20	171 057	112	3 793 544 157
_mqListenerService UserSingleConnectionHub (alR.Hubs)	0000003005a7f1f8	3	136	1 348
_target Action<UserDisconnectedRemoteWidgetMessage> (System)	0000003005a7f300	1	64	64
_target Action<ShutdownRemoteWidgetMessage> (System)	0000003005a7f2c0	1	64	64
_target Action<RemoteWidgetAliveMessage> (System)	0000003005a7f340	1	64	64
_mqListenerService UserSingleConnectionHub (alR.Hubs)	0000002df8ded188	4	136	31 821
_mqListenerService UserSingleConnectionHub (alR.Hubs)	0000002ff336d500	4	136	31 067
_mqListenerService UserSingleConnectionHub (alR.Hubs)	0000002f7c6f5798	3	136	42 086
_mqListenerService UserSingleConnectionHub (alR.Hubs)	0000002e711f6318	4	136	30 165
_mqListenerService UserSingleConnectionHub (alR.Hubs)	0000002df336ca28	3	136	1 372
_mqListenerService UserSingleConnectionHub (alR.Hubs)	0000002f76c75538	3	136	

dotMemory

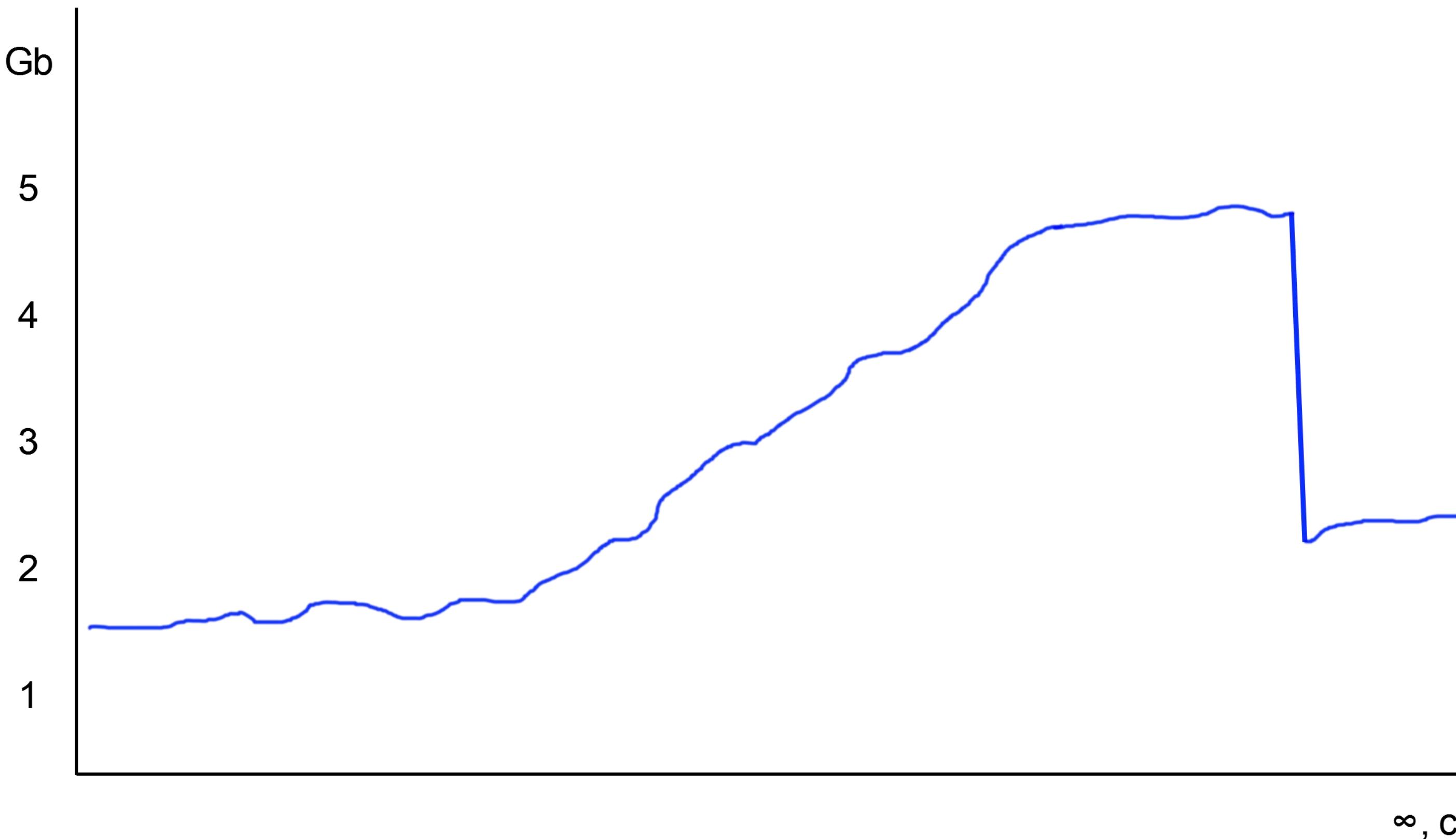
# Пример 1

```
/// <summary>
/// Constructor
/// </summary>
public UserSingleConnectionHub(IMessageQueuePublisher<IAstRabbitMqConnectionConfig> messageQueuePublisher,
                                IMqListenerService mqListenerService,
                                IUserRegistry<UserStateHub, IUserStateClient> userRegistry)
{
    _userIdLazy = new Lazy<UserId>(GetUserId);
    _messageQueuePublisher = messageQueuePublisher;
    _mqListenerService = mqListenerService;
    _mqListenerService.OnShutdown += OnShutdownRequest;
    _mqListenerService.OnUserDisconnected += async (e) => await OnUserDisconnected(e);
    _mqListenerService.OnWidgetAlive += OnWidgetAlive;
    _userRegistry = userRegistry;
}
```

# Пример 1. Выводы

`IDisposable.Dispose()`

## Пример 2



# Пример 2

```
t Ip
t Method
t MethodDirection In
t Parameters {
    "rolesIds": null,
    "userId": null,
    "sipNumber": null,
    "loginPart": null,
    "isActive": true,
    "dateOfSearch": null,
    "queueId": null,
    "appCode": null,
    "position": null,
    "count": null,
    "appUserId": null
}
t RequestMethod "GET"
t RequestUri "
```

## Пример 2

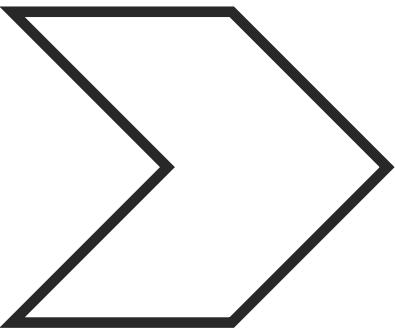
```
{  
    "rolesIds": null,  
    "userId": null,  
    "sipNumber": null,  
    "loginPart": null,  
    "isActive": true,  
    "dateOfSearch": null,  
    "queueId": null,  
    "appCode": null,  
    "position": null,  
    "count": null,  
    "appUserId": null,  
}
```

## Пример 2

```
{  
    "rolesIds": null,  
    "userId": null,  
    "sipNumber": null,  
    "loginPart": null,  
    "isActive": true,  
    "dateOfSearch": null,  
    "queueId": null,  
    "appCode": null,  
    "position": null,  
    "count": null,  
    "appUserId": null,  
}
```

## Пример 2

```
{  
    "rolesIds": null,  
    "userId": null,  
    "sipNumber": null,  
    "loginPart": null,  
    "isActive": true,  
    "dateOfSearch": null,  
    "queueId": null,  
    "appCode": null,  
    "position": null,  
    "count": null,  
    "appUserId": null,  
}
```



```
SELECT * FROM USER  
WHERE IS_ACTIVE = TRUE;
```

# **Пример 2. Выводы**

## Пример 2. Выводы

```
{  
    "rolesIds": null,  
    "userId": null,  
    "sipNumber": null,  
    "loginPart": null,  
    "isActive": true,  
    "dateOfSearch": null,  
    "queueId": null,  
    "appCode": null,  
    "position": null,  
    "count": null,  
    "appUserId": null,  
}
```

## Пример 2. Выводы

```
{  
    "rolesIds": null,  
    "userId": null,  
    "sipNumber": null,  
    "loginPart": null,  
    "isActive": true,  
    "dateOfSearch": null,  
    "queueId": null,  
    "appCode": null,  
    "position": null,  
    "count": null,  
    "appUserId": null,  
}
```

## Пример 2. Выводы

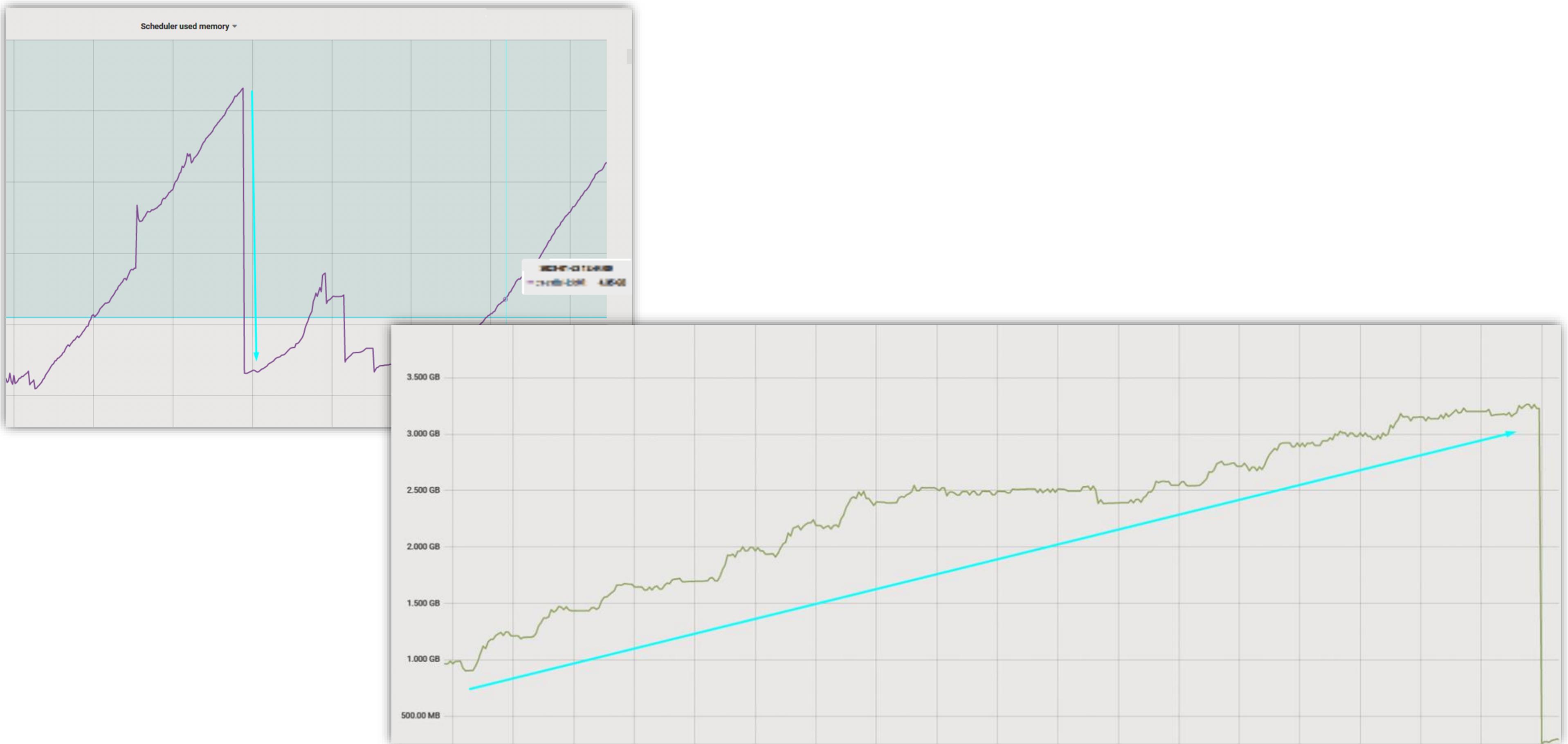
```
{  
    "rolesIds": null,  
    "userId": null,  
    "sipNumber": null,  
    "loginPart": null,  
    "isActive": true,  
    "dateOfSearch": null,  
    "queueId": null,  
    "appCode": null,  
    "position": null,  
    "count": null,  
    "appUserId": null,  
}
```

## Пример 2. Выводы

```
{  
    "rolesIds": null,  
    "userId": null,  
    "sipNumber": null,  
    "loginPart": null,  
    "isActive": true,  
    "dateOfSearch": null,  
    "queueId": null,  
    "appCode": null,  
    "position": null,  
    "count": null,  
    "appUserId": null,  
}
```

Bad Request  
400

# Пример 3



# Пример 3



Type	Size	Committed	Private	Total WS	Private WS	Shareable WS	Shared WS	Locked WS	Blocks	Largest
Total	16 569 244 K	12 491 040 K	12 121 296 K	10 181 124 K	10 101 168 K	79 956 K	20 620 K		24394	
Free	137 422 384 224 K								506	136 848 817 968 K
Heap	13 055 132 K	10 611 604 K	10 611 540 K	8 661 720 K	8 661 716 K	4 K	4 K		17740	16 192 K
Image	400 608 K	398 308 K	38 052 K	86 304 K	7456 K	78 848 K	20 024 K		3089	24 980 K
Managed Heap	1 693 736 K	1 366 620 K	1 366 620 K	1 361 628 K	1 361 628 K				2009	393 216 K
Mapped File	6 772 K	6 772 K		896 K		896 K	396 K		16	2 900 K
Page Table	60 K	60 K	60 K	60 K	60 K					
Private Data	273 664 K	74 520 K	74 392 K	46 176 K	46 140 K	36 K	36 K		699	226 112 K
Shareable	5 816 K	2 524 K		172 K		172 K	160 K		30	1 568 K
Stack	1 094 656 K	30 632 K	30 632 K	24 168 K	24 168 K				811	4 096 K
Unusable	38 800 K									60 K

VMMMap

# Пример 3

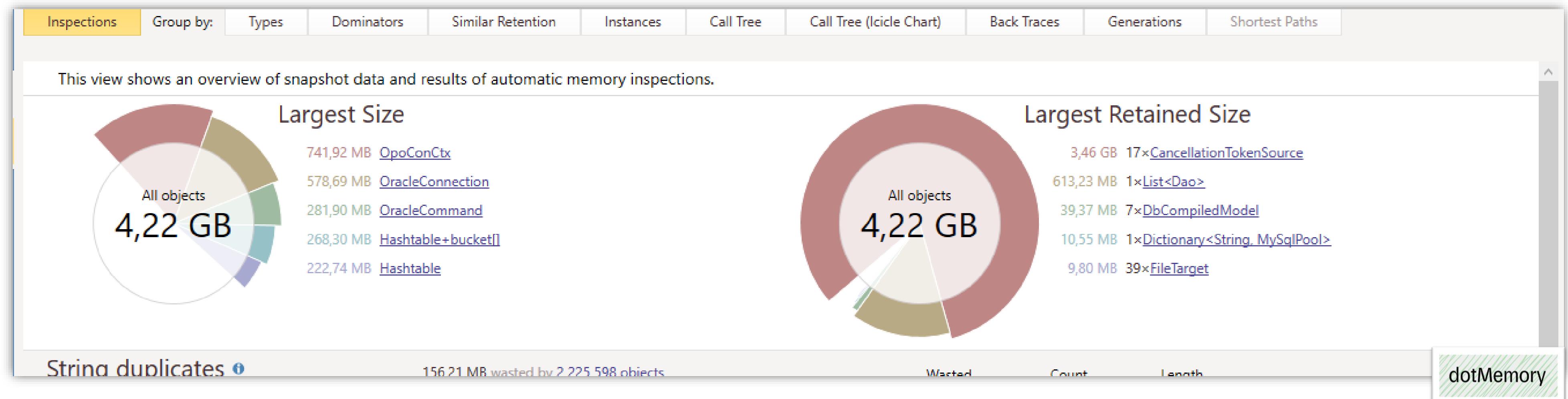
```
0000002b`f6492b90 00000000 00000000 00000000 00000000 .....  
0000002b`f6492ba0 00000000 00000000 00000000 00000000 .....  
0000002b`f6492bb0 00000000 00000000 00000000 00000000 .....  
0000002b`f6492bc0 00000000 00000000 00000000 00000000 .....  
0000002b`f6492bd0 00000000 00000000 00000000 00000000 .....  
0000002b`f6492be0 944523a0 00007ffe 00000000 00000000 .#E.....  
0000002b`f6492bf0 00000000 00000000 00000000 00000000 .....  
0000002b`f6492c00 00000000 00000000 00000b39 30b38f00 .....9....0  
0000002b`f6492c10 f6492990 0000002b 00000000 00000000 .)I.+.....  
0000002b`f6492c20 00000000 00000000 b43f2158 00007ffe .....X!?.  
0000002b`f6492c30 454c4553 0d205443 7247220a 4270756f SELECT .."GroupB  
0000002b`f6492c40 2e223179 22314122 20534120 22314322 y1"."A1" AS "C1"  
0000002b`f6492c50 52460a0d 28204d4f 4c455320 20544345 ..FROM ( SELECT  
0000002b`f6492c60 41090a0d 22284756 746c6946 22327265 ...AVG("Filter2"  
0000002b`f6492c70 3141222e 41202922 41222053 0a0d2231 ."A1") AS "A1"..  
0000002b`f6492c80 4f524609 2028204d 454c4553 0d205443 .FROM ( SELECT ..  
0000002b`f6492c90 2009090a 54534143 45202028 41525458 ... CAST( EXTRA  
0000002b`f6492ca0 20285443 20594144 4d4f5246 43202820 CT( DAY FROM ( C  
0000002b`f6492cb0 28545341 6c694622 31726574 4e222e22 AST("Filter1"."N  
0000002b`f6492cc0 5f545845 52455355 4154535f 445f4554 EXT_USER_STATE_D  
0000002b`f6492cd0 22455441 20534120 454d4954 4d415453 ATE" AS TIMESTAM  
0000002b`f6492ce0 29392850 202d2029 53414320 46222854 P(9)) - CAST("F  
0000002b`f6492cf0 65746c69 2e223172 45525022 53555f56 ilter1"."PREV_US  
0000002b`f6492d00 535f5245 45544154 5441445f 41202245 ER_STATE_DATE" ^  
0000002b`f6492d10 49542053 5453454d 28504d41 29292939 S TTM
```

WinDbg

```
0000002b`f6490880 00000000 00000000 0000002b 00000000 .....+....  
0000002b`f6490890 00000000 00000000 00000015 00000000 .....  
0000002b`f64908a0 f6491560 0000002b 00000001 00000000 `..I.+.....  
0000002b`f64908b0 00000000 0000057b f64908c0 0000002b ....{....I.+...  
0000002b`f64908c0 2d41524f 30343130 6e203a33 6164206f ORA-01403: no da  
0000002b`f64908d0 66206174 646e756f 0000000a 00000000 ta found.....  
0000002b`f64908e0 00000000 00000000 00000000 00000000 .....  
0000002b`f64908f0 00000000 00000000 00000000 00000000 .....  
0000002b`f6490900 00000000 00000000 00000000 00000000 .....
```

WinDbg

# Пример 3



# Пример 3

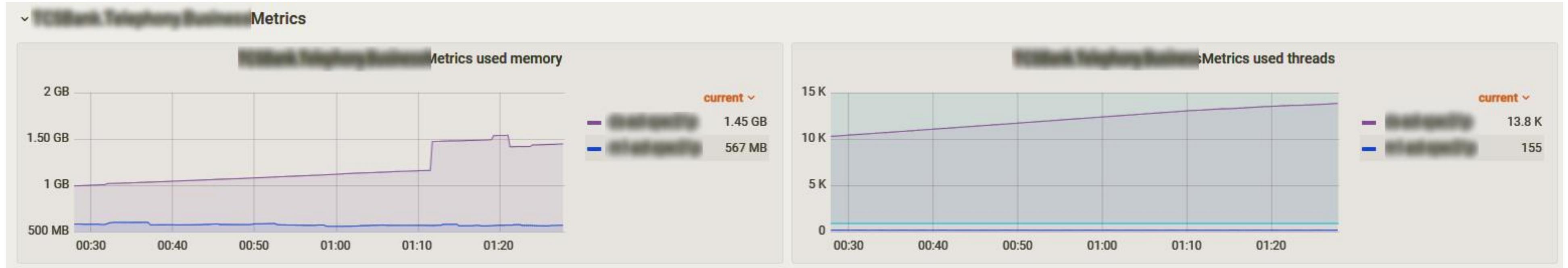
Type	Address	References count	Bytes	Retained bytes
CancellationTokenSource (System.Threading)	000000e98641b8f8	1	72	3 714 873 378
Fields				
m_registeredCallbacksLists SparselyPopulatedArray<CancellationCallbackInfo>[8] (System.Threading)	000000e9864308c0	1	88	3 714 873 306
[2] SparselyPopulatedArray<CancellationCallbackInfo> (System.Threading)	000000e986430918	2	32	3 714 873 218
m_head SparselyPopulatedArrayFragment<CancellationCallbackInfo> (System.Threading)	000000e986430938	2	48	16 814
m_freeCount Int32 -246 = 0xFFFFF0A				
m_elements CancellationCallbackInfo[4] (System.Threading)	000000e986430968	4	56	16 766
[0] CancellationCallbackInfo (System.Threading)	000000e986430888	4	56	4 964
Callback Action<Object> (System)	000000e9b0b4a908	1	64	64
StateForCallback Action (System)	000000e9864306f8	1	64	4 572
_target OracleCommand (Oracle.DataAccess.Client)	000000e9864302d8	7	304	4 508
Fields				
m_MetaData MetaData (Oracle.DataAccess.Client)	000000e986430a30	7	128	704
m_Connection OracleConnection (Oracle.DataAccess.Client)	000000e98642ed08	22	624	2 582
m_Parameters OracleParameterCollection (Oracle.DataAccess.Client)	000000e986430408	4	64	776
m_CommandText String (System) Length: 1058 = Chars: 1018 @"SELECT \r\n\"Extent1\".\"ID\" AS \"REQU..."	000000e9b1fa5540		2 142	2 142
m_PooledCmdText String (System) Length: 1058 = Chars: 1018 @"SELECT \r\n\"Extent1\".\"ID\" AS \"REQU..."	000000e9b1fa5540		2 142	2 142
m_placeHolderColl ArrayList (System.Collections)	000000e9864309a0	1	40	142
m_expectedColumnTypes Type[18] (System)	000000e9b1db2168	18	168	

dotMemory

## **Пример 3. Выводы**

`CancellationTokenSource.CreateLinkedTokenSource()`

# Пример 4



# Пример 4

```
Thread 28
Current frame:
Child-SP      RetAddr      Caller, Callee
00000741c3fd4d0 00007ffff8754629e (MethodDesc 00007ffff8749df00 +0x15e StackExchange.Redis.ConnectionMultiplexer.GetServer(System.NetEndPoint, System.Object))
00000741c3fd4f0 00007ffffccda47ad (MethodDesc 00007ffffcc909d30 +0x1dd System.Linq.Parallel.PartitionedDataSource`1+ContiguousChunkLazyEnumerator[[System._Canon, mscorelib]].MoveNext())
00000741c3fd520 00007ffffe1e48261 (MethodDesc 00007ffffe11fa510 +0x11 System.Threading.CountdownEvent.Signal())
00000741c3fd570 00007ffff87545b5d (MethodDesc 00007ffff87787f68 +0x2d System.Linq.Parallel.SelectQueryOperator`2+SelectQueryOperatorEnumerator`1[[System._Canon, mscorelib],[System._Canon, mscorelib]].MoveNext())
00000741c3fd5b0 00007ffff87546117 (MethodDesc 00007ffff87787f70 +0x17 System.Linq.Parallel.SelectQueryOperator`2+SelectQueryOperatorEnumerator`1[[System._Canon, mscorelib],[System._Canon, mscorelib]].MoveNext())
00000741c3fd5e0 00007ffffccead2b8 (MethodDesc 00007ffffcc92ea40 +0x18 System.Linq.Parallel.FirstQueryOperator`1+FirstQueryOperatorEnumerator`1[[System._Canon, mscorelib],[System.Int32, mscorelib]].MoveNext())
00000741c3fd5f0 00007ffffccd3d53c (MethodDesc 00007ffffcc906880 +0x4c System.Linq.Parallel.SynchronousChannel`1[[System._Canon, mscorelib]].Init())
00000741c3fd610 00007ffff875460e4 (MethodDesc 00007ffff8778b630 +0x24 System.Linq.Parallel.StopAndGoSpoolingTask`2[[System._Canon, mscorelib],[System.Int32, mscorelib]].SpoolingFinally())
00000741c3fd6a0 00007ffffe1495ed2 (MethodDesc 00007ffffe120d218 +0x72 System.Threading.ExecutionContextSwitcher.Undo())
00000741c3fd6f0 00007ffffe1495af2 (MethodDesc 00007ffffe1088a00 +0x1a2 System.Threading.ExecutionContext.RunInternal(System.Threading.ExecutionContext, System.Threading.ContextCallback))
00000741c3fd710 00007ffffe14df2c6 (MethodDesc 00007ffffe108ff58 +0x46 System.Threading.Tasks.Task.Execute())
00000741c3fd730 00007ffffe1463fed (MethodDesc 00007ffffe1208bc0 +0x3d System.Threading.Tasks.Task.NewId())
00000741c3fd750 00007ffffe1495abc (MethodDesc 00007ffffe1088a00 +0x16c System.Threading.ExecutionContext.RunInternal(System.Threading.ExecutionContext, System.Threading.ContextCallback))
00000741c3fd760 00007ffffe1463a71 (MethodDesc 00007ffffe1090150 +0x81 System.Threading.Tasks.Task.FinishContinuations())
00000741c3fd850 00007ffffe14df591 (MethodDesc 00007ffffe108ffd8 +0x241 System.Threading.Tasks.Task.ExecuteWithThreadLocal(System.Threading.Tasks.Task ByRef))
00000741c3fd990 00007ffffe14e2efc (MethodDesc 00007ffffe11fa890 +0x1ac System.Threading.Tasks.TaskScheduler.TryRunInline(System.Threading.Tasks.Task, Boolean))
00000741c3fdd20 00007ffffe14e6a0f (MethodDesc 00007ffffe1208f90 +0x2cf System.Threading.ManualResetEventSlim.Wait(Int32, System.Threading.CancellationToken))
00000741c3fdbb0 00007ffffe14df815 (MethodDesc 00007ffffe10900d0 +0xb5 System.Threading.Tasks.Task.SpinThenBlockingWait(Int32, System.Threading.CancellationToken))
00000741c3fde20 00007ffffe1e4faa0 (MethodDesc 00007ffffe108fc28 +0x1d0 System.Threading.Tasks.Task.InternalRunSynchronously(System.Threading.Tasks.TaskScheduler, Boolean))
```

WinDbg

# Пример 4

```
Unit Test Explorer Unit Test Sessions AsyncHelperTests.cs NuGet - Solution MultiplexerWrapper.cs* MemoryStorage.cs RedisConnectionFactoryCommonMiscellaneous Files StackExchange.Redis.ConnectionMultiplexer OnManagedConnectionFailed<object> sender ConnectionFailedEventArgs e
internal void OnManagedConnectionFailed(object sender, ConnectionFailedEventArgs e)
{
    ConnectionMultiplexer connection = (ConnectionMultiplexer)sender;
    // Periodically check to see if we can reconnect to the proper master.
    // This is here in case we lost our subscription to a good sentinel instance
    // or if we miss the published master change
    if (connection.sentinelMasterReconnectTimer == null)
    {
        connection.sentinelMasterReconnectTimer = new Timer(_ =>
        {
            try
            {
                // Attempt, but do not fail here
                SwitchMaster(e.EndPoint, connection);
            }
            catch (Exception)
            {
            }
        }, null, TimeSpan.FromSeconds(0), TimeSpan.FromSeconds(1));
    }
}
```

# Пример 4

## 2.2.62

- Sentinel potential memory leak fix in OnManagedConnectionFailed handler (#1710 via alexSatov)
- fix issue where `GetOutstandingCount` could obscure underlying faults by faulting itself (#1792 via mgravell)
- fix issue #1719 with backlog messages becoming reordered (#1779 via TimLovellSmith)

## **Пример 4. Выводы**

**Не забывайте обновлять библиотеки**

# Пример 5



# Пример 5

This view shows an overview of snapshot data and results of automatic memory inspections

**Largest Size**

Object Type	Size
RawResult[]	28,88 MB
String	9,52 MB
Byte[]	9,17 MB
Object[]	2,87 MB
Char[]	2,53 MB

**Largest Retained Size**

Object Type	Size
PhysicalConnection	28,95 MB
Object[]	20,37 MB
AsyncConsumerWorkService+WorkPool	5,55 MB
SentryClient	3,21 MB
GCHeapHash	2,60 MB

**String duplicates** i

5,86 MB wasted by [96 159 objects](#)

A large number of strings with the same value is inefficient from the point of memory usage

String Value	Wasted	Count	Length
"	667,2 KB	3417	89
"amq.ctag-JYRTL_uD90QPEDcd2BaA3w"	279,8 KB	3412	31
"6.0.0.0"	258,5 KB	7353	7
"headerTraceld"	187,7 KB	4006	13
'	173,5 KB	3417	15

**Sparse arrays** i

38,05 MB wasted by [1 866 objects](#)

Partially filled arrays are inefficient from the point of memory usage

Type	Wasted	Filling	Length
ConcurrentDictionary+Node<String, LockEntry<FifoSemaphoreLock>>[]	1,27 MB	0,0%	166 717
ConcurrentDictionary+Node<String, MemoryStorage+CacheEntry>[]	1,09 MB	0,0%	143 357
RawResult[]	224,0 KB	0,0%	4 096
RawResult[]	224,0 KB	0,0%	4 096
RawResult[]	224,0 KB	0,0%	4 096

**Finalizable objects** i

571 objects of [30 type\(s\)](#) found

Objects that were queued for finalization or already finalized since the previous snapshot. It might be more efficient if these objects were disposed via the IDisposable interface.

Type	Queued	Finalized	Total
OracleLogicalTransaction	168	0	168
DynamicResolver+DestroyScout	70	0	70
ConditionalWeakTable+Container<ContextKey, ContextData>	49	0	49
Thread	41	0	41
SafeWaitHandle	36	0	36

**Event handlers leak** i

3,9 KB retained by [11 objects](#)

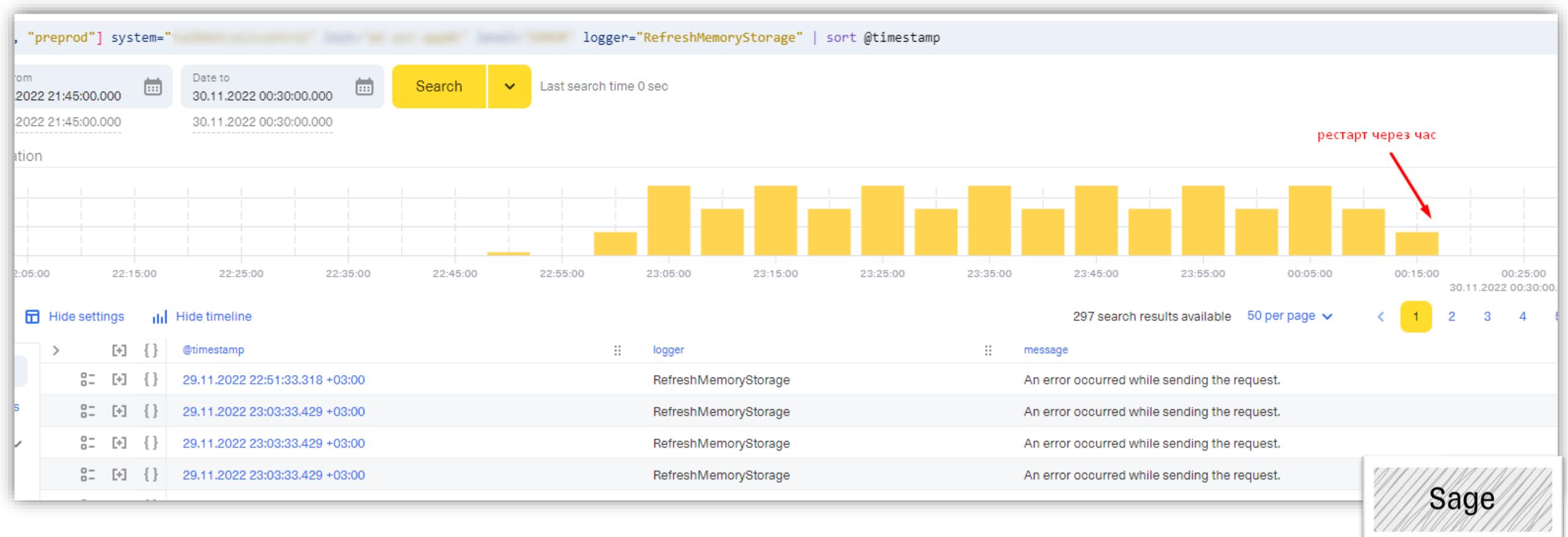
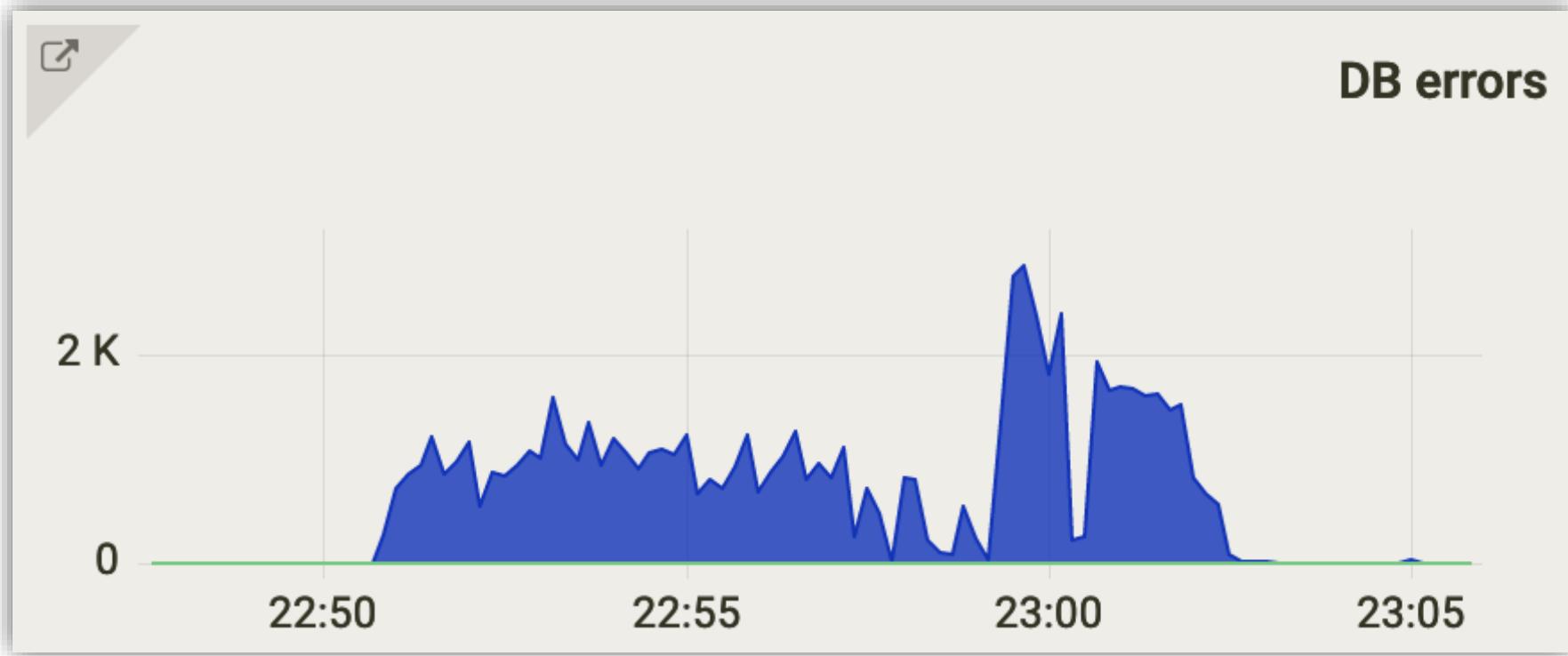
Objects subscribed to an event of another object and never unsubscribed from this event

Type	Retained Size	Count
Session	2,8 KB	2
trigger+<>c_DisplayClass5_0	648 B	1
InterceptorExtensions+InterceptionHelper	304 B	2
JsonSerializeExtensions+<>c_DisplayClass4_0	120 B	5
SimpleInjectorServiceCollectionExtensions+<>c_DisplayClass18_1	56 B	1

dotMemory

69

# Пример 5



# Пример 5

```
0:000> !Threads
ThreadCount: 2489
UnstartedThread: 0
BackgroundThread: 2464
PendingThread: 0
DeadThread: 24
Hosted Runtime: no



| DBG | ID  | OSID                 | ThreadOBJ        | State                   | GC Mode    | GC Alloc Context                  | Domain                           | Lock                           |
|-----|-----|----------------------|------------------|-------------------------|------------|-----------------------------------|----------------------------------|--------------------------------|
|     |     |                      |                  |                         |            |                                   |                                  | Count Apt Exception            |
| 0   | 1   | <a href="#">4b78</a> | 000001D81ED2F220 | <a href="#">2a020</a>   | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 MTA                     |
| 11  | 2   | <a href="#">1004</a> | 000001D81EE0A8E0 | <a href="#">2b220</a>   | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 MTA (Finalizer)         |
| 12  | 5   | <a href="#">486c</a> | 000001DD2C2E2A00 | <a href="#">302b220</a> | Preemptive | 000001DB208F7638:000001DB208F8EB0 | <a href="#">000001d81ed45f40</a> | -00001 MTA (Threadpool Worker) |
| 13  | 9   | <a href="#">29ec</a> | 000001DD2C2DB680 | <a href="#">102a220</a> | Preemptive | 000001D9A07A7DE0:000001D9A07A8A48 | <a href="#">000001d81ed45f40</a> | -00001 MTA (Threadpool Worker) |
| 14  | 15  | <a href="#">34ec</a> | 000001DD2C32E510 | <a href="#">302b220</a> | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 MTA (Threadpool Worker) |
| 17  | 23  | <a href="#">32d4</a> | 000001DD2BB78610 | <a href="#">202b220</a> | Preemptive | 000001D9A07B04B8:000001D9A07B1690 | <a href="#">000001d81ed45f40</a> | -00001 MTA                     |
| 16  | 28  | <a href="#">362c</a> | 000001DD2BB799C0 | <a href="#">202b220</a> | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 MTA                     |
| 18  | 32  | <a href="#">26cc</a> | 000001DD2BB7BA90 | <a href="#">302b220</a> | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 MTA (Threadpool Worker) |
| 19  | 46  | <a href="#">19c0</a> | 000001DD2DE812C0 | <a href="#">202b220</a> | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 MTA                     |
| 20  | 47  | <a href="#">361c</a> | 000001DD2DE81950 | <a href="#">202b220</a> | Preemptive | 000001DA20A7D100:000001DA20A7E500 | <a href="#">000001d81ed45f40</a> | -00001 MTA                     |
| 21  | 48  | <a href="#">2078</a> | 000001DD2DE81FE0 | <a href="#">202b220</a> | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 MTA                     |
| 22  | 49  | <a href="#">3e8c</a> | 000001DD2DE7D7B0 | <a href="#">202b220</a> | Preemptive | 000001DBA0915770:000001DBA0916B50 | <a href="#">000001d81ed45f40</a> | -00001 MTA                     |
| 23  | 50  | <a href="#">24fc</a> | 000001DD2DE83390 | <a href="#">202b220</a> | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 MTA                     |
| 24  | 51  | <a href="#">467c</a> | 000001DD2DE7E4D0 | <a href="#">202b220</a> | Preemptive | 000001DA20A7F120:000001DA20A80500 | <a href="#">000001d81ed45f40</a> | -00001 MTA                     |
| 25  | 52  | <a href="#">1e64</a> | 000001DD2DE7EB60 | <a href="#">202b220</a> | Preemptive | 000001D9A07DAD48:000001D9A07DC348 | <a href="#">000001d81ed45f40</a> | -00001 MTA                     |
| 26  | 53  | <a href="#">3858</a> | 000001DD2DE82670 | <a href="#">202b220</a> | Preemptive | 000001D8209536C8:000001D820954AC8 | <a href="#">000001d81ed45f40</a> | -00001 MTA                     |
| 27  | 54  | <a href="#">3964</a> | 000001DD2DE83A20 | <a href="#">202b220</a> | Preemptive | 000001D8A03FC438:000001D8A03FDA38 | <a href="#">000001d81ed45f40</a> | -00001 MTA                     |
| 28  | 55  | <a href="#">cc</a>   | 000001DD2DE7CA90 | <a href="#">202b220</a> | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 MTA                     |
| 29  | 154 | <a href="#">2714</a> | 000001DD2E63ADB0 | <a href="#">21220</a>   | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 Ukn                     |
| 30  | 155 | <a href="#">38a0</a> | 000001DD2E63B440 | <a href="#">21220</a>   | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 Ukn                     |
| 31  | 156 | <a href="#">3ebc</a> | 000001DD2E635860 | <a href="#">21220</a>   | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 Ukn                     |
| 32  | 157 | <a href="#">2cb0</a> | 000001DD2E63BAD0 | <a href="#">21220</a>   | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 Ukn                     |
| 33  | 158 | <a href="#">2764</a> | 000001DD2E635EF0 | <a href="#">21220</a>   | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 Ukn                     |
| 34  | 159 | <a href="#">1b4c</a> | 000001DD2E636580 | <a href="#">21220</a>   | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 Ukn                     |
| 35  | 160 | <a href="#">27b4</a> | 000001DD2E3CF840 | <a href="#">21220</a>   | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 Ukn                     |
| 36  | 161 | <a href="#">4af4</a> | 000001DD2E3D1280 | <a href="#">21220</a>   | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 Ukn                     |
| 37  | 228 | <a href="#">2960</a> | 000001DD2C8074C0 | <a href="#">302b220</a> | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 MTA (Threadpool Worker) |
| 38  | 104 | <a href="#">100c</a> | 000001DD2E2868E0 | <a href="#">302b220</a> | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 MTA (Threadpool Worker) |
| 39  | 170 | <a href="#">3104</a> | 000001DD2D82C180 | <a href="#">302b220</a> | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 MTA (Threadpool Worker) |
| 40  | 21  | <a href="#">29c8</a> | 000001DD2C4A67A0 | <a href="#">302b220</a> | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 MTA (Threadpool Worker) |
| 41  | 95  | <a href="#">3df8</a> | 000001DD2C4A46D0 | <a href="#">302b220</a> | Preemptive | 00000000000000:00000000000000     | <a href="#">000001d81ed45f40</a> | -00001 MTA (Threadpool Worker) |



WinDbg


```

# Пример 5

```
00000055247F7C0 00007ffe7b4f94a6 System.Threading.Tasks.Task.ExecuteWithThreadLocal(System.Threading.Tasks.Task ByRef, System.Threading.Thread) [/src/libraries/System.Private.CoreLib/src/System.Threading/PortableThreadPool.WorkerThread.cs @ 63]
00000055247F870 00007ffe78a5d03a System.Threading.ThreadPoolWorkQueue.Dispatch()
00000055247F900 00007ffe7cddc691 System.Threading.PortableThreadPool+WorkerThread.WorkerThreadStart() [/src/libraries/System.Private.CoreLib/src/System.Threading/PortableThreadPool.WorkerThread.cs @ 63]
00000055247FC50 00007ffed780aac3 [DebuggerU2MCatchHandlerFrame: 00000055247fc50]

S Thread Id: 0x3df8 (41)
    Child SP          IP Call Site
0000005572BE7E8 00007fff172d0544 [HelperMethodFrame 10BJ: 0000005572be7e8] System.Threading.Monitor.ReliableEnter(System.Object, Boolean ByRef)
0000005572BE940 00007ffe7b4edcfb MemoryStorage+CacheEntry.get_Value() [/builds/tinkoff-telephony/Caching/MemoryStorage.cs @ 534]
0000005572BE990 00007ffe7b4fdd61 MemoryStorage.GetOrAdd[[System._Canon, System.Private.CoreLib]](System.String, System.TimeSpan, System.Func`1<System._Canon>) [/builds/tinkoff-telephony/Caching/MemoryStorage.cs @ 534]
0000005572BE9F0 00007ffe7bba6fba .ConfigStorage.TryGetValue[[System._Canon, System.Private.CoreLib]](System.String, Boolean, System._Canon ByRef)
0000005572BEB70 00007ffe7ce2b7da .ConfigStorage+c_DisplayClass13_0`1[[System._Canon, System.Private.CoreLib]].b__1()
0000005572BEBE0 00007ffe7bb8d503 ieExtensions.GetOrAdd[[System._Canon, System.Private.CoreLib]](System.String, Caching.ICacheClient, System.String, System.Func`2<System._Canon, System.TimeSpan>, System.Core.Configs.DependenceType)
0000005572BEC50 00007ffe7bab750 .ConfigStorage.FindObject[[System._Canon, System.Private.CoreLib]](System.String, System.Core.Configs.DependenceType)
0000005572BECF0 00007ffe7bab3fb .ConfigStorage.GetObject[[System._Canon, System.Private.CoreLib]](System.String, System.Core.Configs.DependenceType)
0000005572BED40 00007ffe7c16196f .ConfigFactoryBase`2+c_DisplayClass5_0[[System._Canon, System.Private.CoreLib],[System._Canon, System.Private.CoreLib]].b__0()
0000005572BED70 00007ffe7c1618f9 .ReloadPropInterceptor`1[[System._Canon, System.Private.CoreLib]].Intercept(TCSBank.Telephony.Ioc.Interceptors.Builder.IInvocation)
0000005572BEDD0 00007ffe7b4eab88 Interceptors.Builder.Interceptor+InterceptorProxy.Invoke(System.Reflection.MethodInfo, System.Object[]) [/builds/tinkoff-telephony/Interceptors/Builder/Interceptor+InterceptorProxy.cs @ 401]
0000005572BEE20 00007ffe7c88a7fa SOS Warning: Loading symbols for dynamic assemblies is not yet supported

unknown method>
0000005572BEE50 00007ffe7c88a574
0000005572BEEC0 00007ffe7c8798a0
0000005572BEFE0 00007ffe78b57e77
0000005572BF050 00007ffe7c8db3ad
0000005572BF0D0 00007ffe7accaa54
0000005572BF130 00007ffe7acca51c
0000005572BF1D0 00007ffe7acf5041
0000005572BF220 00007ffe7c8ceda4
0000005572BF300 00007ffe78b57e77
0000005572BF370 00007ffe7c926b29
0000005572BF3F0 00007ffe7accaa54
0000005572BF450 00007ffe7acca51c
0000005572BF4F0 00007ffe7bb87059
0000005572BF540 00007ffe7c92692a
0000005572BF590 00007ffe78b5140b
0000005572BF5F0 00007ffe7b4f94a6 System.Threading.ExecutionContext.RunFromThreadPoolDispatchLoop(System.Threading.Thread, System.Threading.ExecutionContext, System.Threading.ContextCallback, System.Object) [/src/libraries/System.Private.CoreLib/src/System.Threading/ExecutionContext.cs @ 2337]
0000005572BF6A0 00007ffe78a5d03a System.Threading.ThreadPoolWorkQueue.Dispatch()
0000005572BF730 00007ffe7cddc691 System.Threading.PortableThreadPool+WorkerThread.WorkerThreadStart() [/src/libraries/System.Private.CoreLib/src/System.Threading/PortableThreadPool.WorkerThread.cs @ 63]
0000005572BFA80 00007ffed780aac3 [DebuggerU2MCatchHandlerFrame: 0000005572bfa80]

S Thread Id: 0x2478 (42)
    Child SP          IP Call Site

```

WinDbg

# **Пример 5. Выводы**

## Пример 5. Выводы

KeepAlive = 10

# Пример 6

[+]	{}	@timestamp	logger	message	system	inst	assemblyVersion
0-	[+]	{}	ZooKeeperClient	Клиент Подключен. Ноды и наблюдатели восстановлены.			56.8.0.52752
0-	[+]	{}	ZooKeeperClient	Клиент отключен. Ноды и наблюдатели сохранены.			56.8.0.52752
0-	[+]	{}	ZooKeeperClient	The connection was closed.			56.8.0.52752
0-	[+]	{}	ZooKeeperClient	Поток завершен.			56.8.0.52752

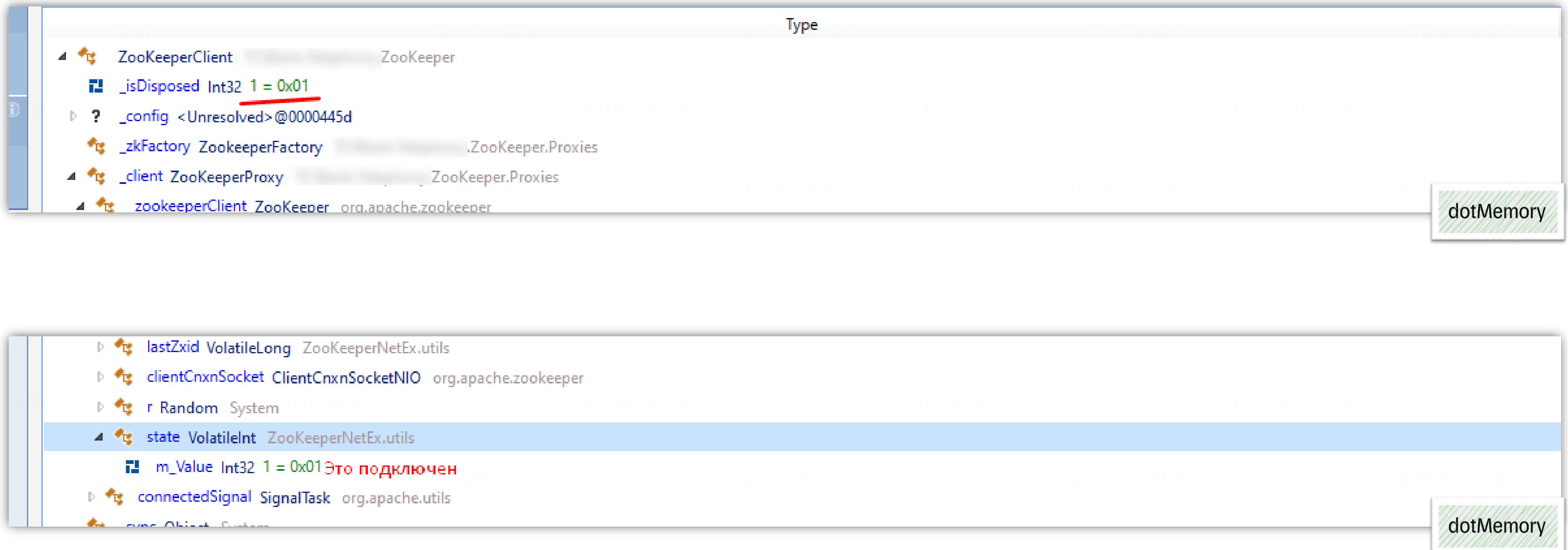
# Пример 6

Here you can view what objects are referenced from the instance as well as values of instance's fields [?](#)

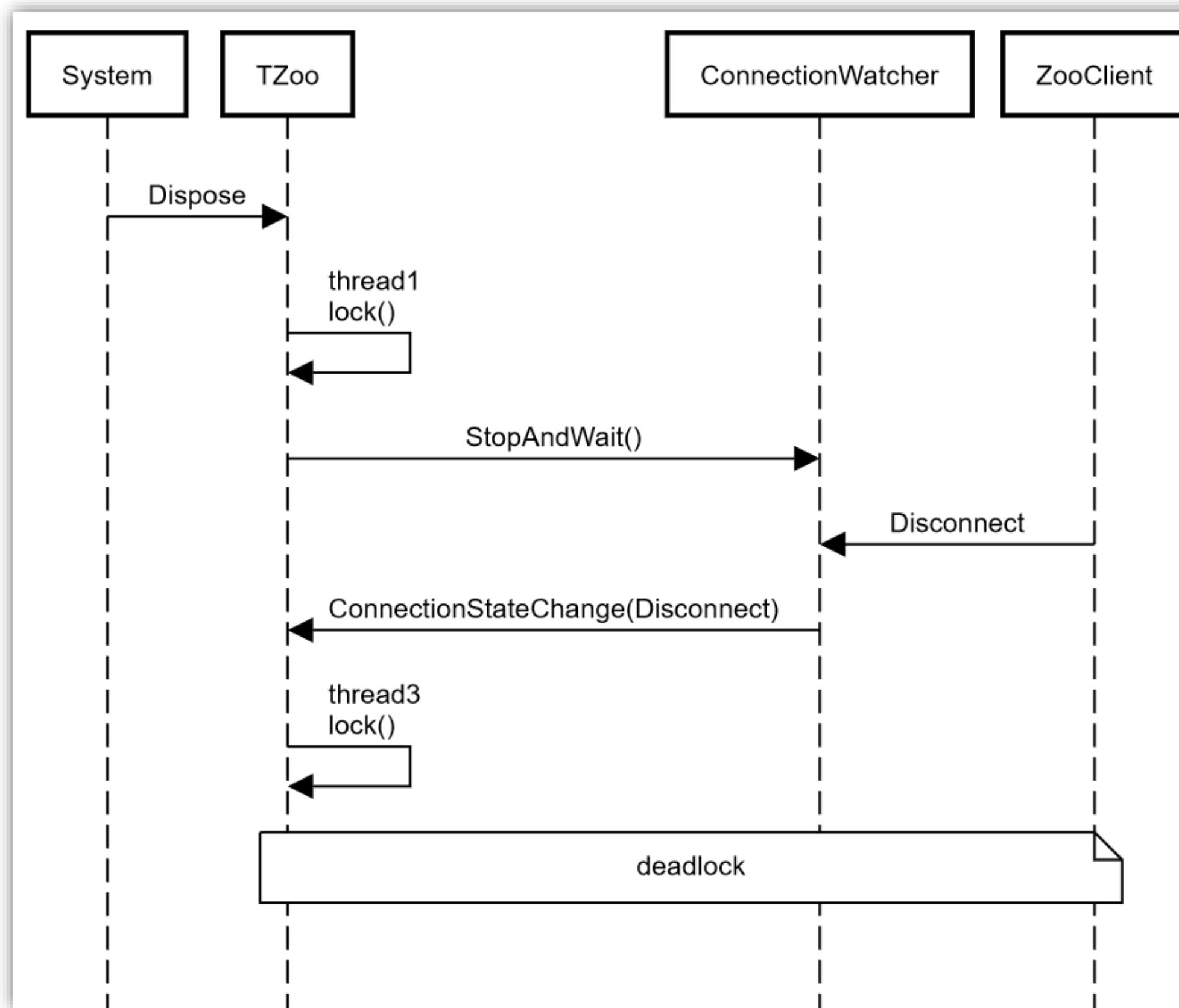
Type
ZooKeeperClient ZooKeeper
↳ _disposed Int32 1 = 0x01
↳ ? _config <Unresolved>@0000445d
↳ _zkFactory ZookeeperFactory ZooKeeper.Proxies
↳ _client ZooKeeperProxy ZooKeeper.Proxies
↳ _zookeeperClient ZooKeeper org.apache.zookeeper
↳ userDefinedSessionTimeout Int32 60 000 = 0xEA60
↳ watchManager ZooKeeper+ZKWatchManager org.apache.zookeeper
↳ cnxn ClientCnxn org.apache.zookeeper
↳ Fields
↳ authInfo List<ClientCnxn+AuthData> System.Collections.Generic
↳ pendingQueue LinkedList<ClientCnxn+Packet> System.Collections.Generic
↳ outgoingQueue LinkedList<ClientCnxn+Packet> System.Collections.Generic
↳ negotiatedSessionTimeout VolatileInt ZooKeeperNetEx.utils
↳ zooKeeper ZooKeeper org.apache.zookeeper
↳ watcher ZooKeeper+ZKWatchManager org.apache.zookeeper
↳ sessionPasswd Byte[16] System
↳ sendTask AsyncTaskMethodBuilder+AsyncStateMachineBox<VoidTaskResult, ClientCnxn+<startSendTask>d_61> System.Runtime.CompilerServices
↳ eventTask AsyncTaskMethodBuilder+AsyncStateMachineBox<VoidTaskResult, ClientCnxn+<startEventTask>d_37> System.Runtime.CompilerServices
↳ timer Timer System.Threading
↳ closing VolatileBool ZooKeeperNetEx.utils
↳ hostProvider DynamicHostProvider org.apache.zookeeper.client
↳ seenRwServerBefore VolatileBool ZooKeeperNetEx.utils
↳ waitingEventsSignal AwaitableSignal ZooKeeperNetEx.utils
↳ waitingEvents ConcurrentQueue<ClientCnxn+WatcherSetEventPair> System.Collections.Concurrent
↳ sessionState VolatileInt ZooKeeperNetEx.utils
↳ lastZxid VolatileLong ZooKeeperNetEx.utils
↳ clientCnxnSocket ClientCnxnSocketNIO org.apache.zookeeper
↳ r Random System
↳ state VolatileInt ZooKeeperNetEx.utils
↳ m_Value Int32 1 = 0x01 Это подключен
↳ connectedSignal SignalTask org.apache.utils
↳ sync Object

dotMemory

# Пример 6



# Пример 6



# Пример 6. Выводы



# Пример 6. Выводы



# Пример 6. Выводы

KISS  
keep it simple and straightforward

# Snippets

<https://github.com/vait/meetups/blob/main/spbdotnet-2024/Snippet.txt>

# Snippets

- !DumpHeap -stat

# Snippets

- !DumpHeap -stat
- !Threads

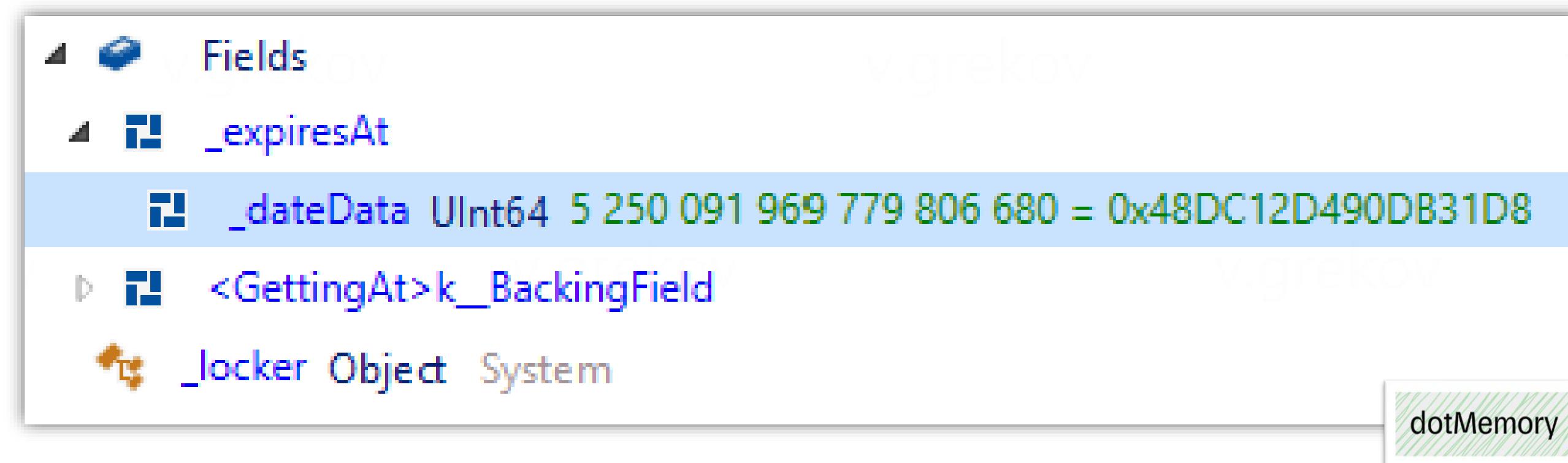
# Snippets

- !DumpHeap -stat
- !Threads
- !CLRStack -p

# Snippets

- !DumpHeap -stat
- !Threads
- !CLRStack -p
- ~\*e!clrstack

# Snippets



# Snippets

Fields:							
MT	Field	Offset	Type	VT	Attr	Value	Name
<a href="#">00007ff97bbdc9d8</a>	4000394	0	System.UInt64	1	instance	5250091969779806680	_dateData
<a href="#">00007ff97c0b5220</a>	400038f	1f8	System.UInt32[]	0	static	<a href="#">000002350635d368</a>	s_daysToMonth365
<a href="#">00007ff97c0b5220</a>	4000390	200	System.UInt32[]	0	static	<a href="#">000002350635d3b8</a>	s_daysToMonth366
<a href="#">00007ff97bc80cc8</a>	4000391	1e0	System.DateTime	1	static	<a href="#">00000239862a1208</a>	MinValue
<a href="#">00007ff97bc80cc8</a>	4000392	1e8	System.DateTime	1	static	<a href="#">00000239862a1210</a>	MaxValue
<a href="#">00007ff97bc80cc8</a>	4000393	1f0	System.DateTime	1	static	<a href="#">00000239862a1218</a>	UnixEpoch
<a href="#">00007ff97bb3bf28</a>	4000395	970	System.Boolean	1	static	1	s_systemSupportsLeapSeconds
<a href="#">0000000000000000</a>	4000396	968	FNPTR	0	static	<a href="#">00007ff9f2b744a0</a>	s_pfnGetSystemTimeAsFileTime
<a href="#">00007ff97c1dcc50</a>	4000397	208	...e+LeapSecondCache	0	static	<a href="#">00000238070cd7c8</a>	s_leapSecondCache

WinDbg

# Snippets

The screenshot shows the LINQPad interface. The top bar has tabs for 'Query 1\*' and '+'. Below the tabs are buttons for execution (play, stop, refresh), language selection ('Language' dropdown set to 'C# Statement(s)'), .NET framework selection ('.NET' dropdown set to 'Auto'), and connection settings ('Connection' dropdown set to '<None>'). The main query window contains the following C# code:

```
Console.WriteLine(DateTime.FromBinary((long)( 5250091969779806680 & 0xFFFFFFFFFFFFFF)));
```

The results pane at the bottom shows the output of the query: **11.01.2024 18:38:55**. The results tab is selected. The bottom right corner of the window has a 'LINQPad' logo.

# **Советы**

# Советы

- Мониторинг

# Советы

- Мониторинг
- Trunk Based Development

# Советы

- Мониторинг
- Trunk Based Development
- Нагрузочное тестирование

# Советы

- Мониторинг
- Trunk Based Development
- Нагрузочное тестирование
- BenchmarkDotNet

# Советы

- Мониторинг
- Trunk Based Development
- Нагрузочное тестирование
- BenchmarkDotNet
- Автоматизация

# **Итоги**

- Нет одного крутого инструмента для анализа

- Нет одного крутого инструмента для анализа
- WinDbg – сложный, но «низкоуровневый»

- Нет одного крутого инструмента для анализа
- WinDbg – сложный, но «низкоуровневый»
- dotMemory – удобный, но с ограничениями

- Нет одного крутого инструмента для анализа
- WinDbg – сложный, но «низкоуровневый»
- dotMemory – удобный, но с ограничениями
- MS Visual Studio тоже хорошо

- Знайте свой код

- Знайте свой код
- Большая часть проблем создана Вашими руками

- Знайте свой код
- Большая часть проблем создана Вашими руками
- Ставьте `keepalive` явно

- Знайте свой код
- Большая часть проблем создана Вашими руками
- Ставьте `keepalive` явно
- Работайте на опережение

# Что еще?

<https://github.com/microsoft/clrmd> - репозиторий для самых лучших программистов

<https://youtu.be/O8c5WwfbGFU> - Building your own debugging toolbox with ClrMD (Christophe Nasarre-Soulier)

<https://youtu.be/tSIFd0Clo0g> - Extend the new WinDbg to build your own dream debugging tool (Kevin Gosse)

<https://clck.ru/3DGQ7p> - Управление памятью в .NET для профессионалов (Конрад Кокоса, пер. DotNet.ru)

Воркшопа от Михаила Ярийчука с хардкором и кишочками

<https://youtu.be/IWRAK4zo9Vs> | <https://youtu.be/p8xCDDY6zx0>

<https://youtu.be/RogZNK3jAfQ> | <https://youtu.be/XGLIOje1Jzk>



# Спасибо!

Виктор Греков @vait11

