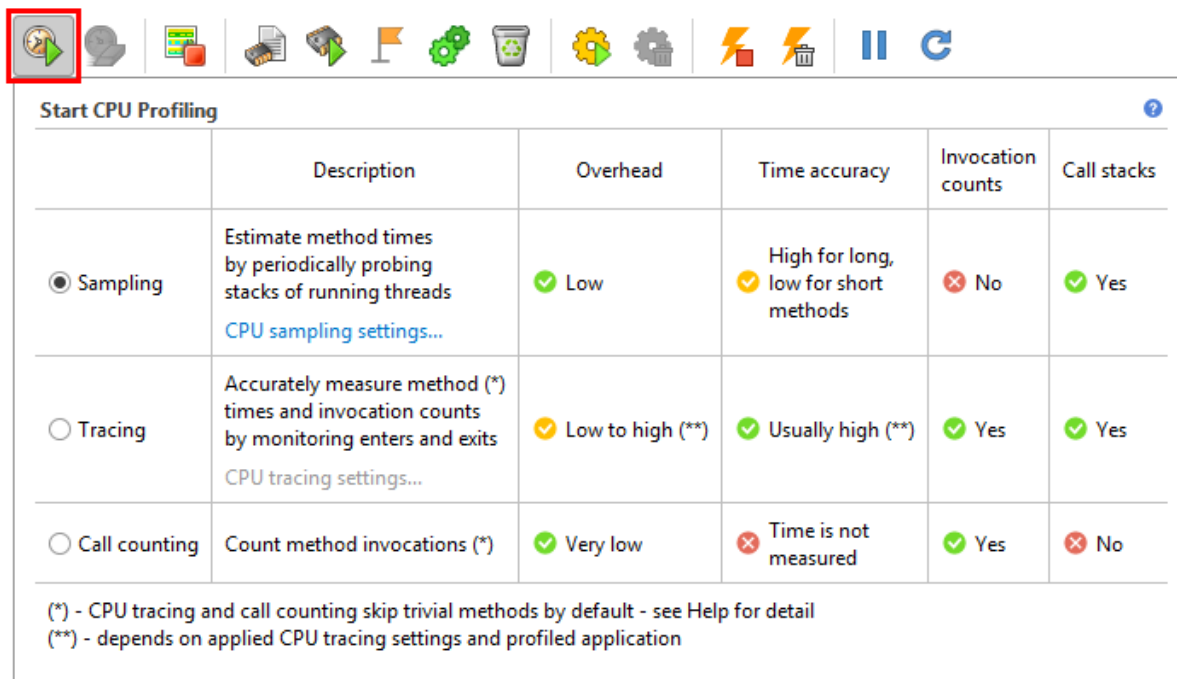


# Yourkit príručka

## Cpu Profiling:

Poznáme 3 typy CPU profilingu: Sampling, Tracing, Call Counting, ktoré si môžeme zvoliť pri spúšťaní profileru v eclipse.

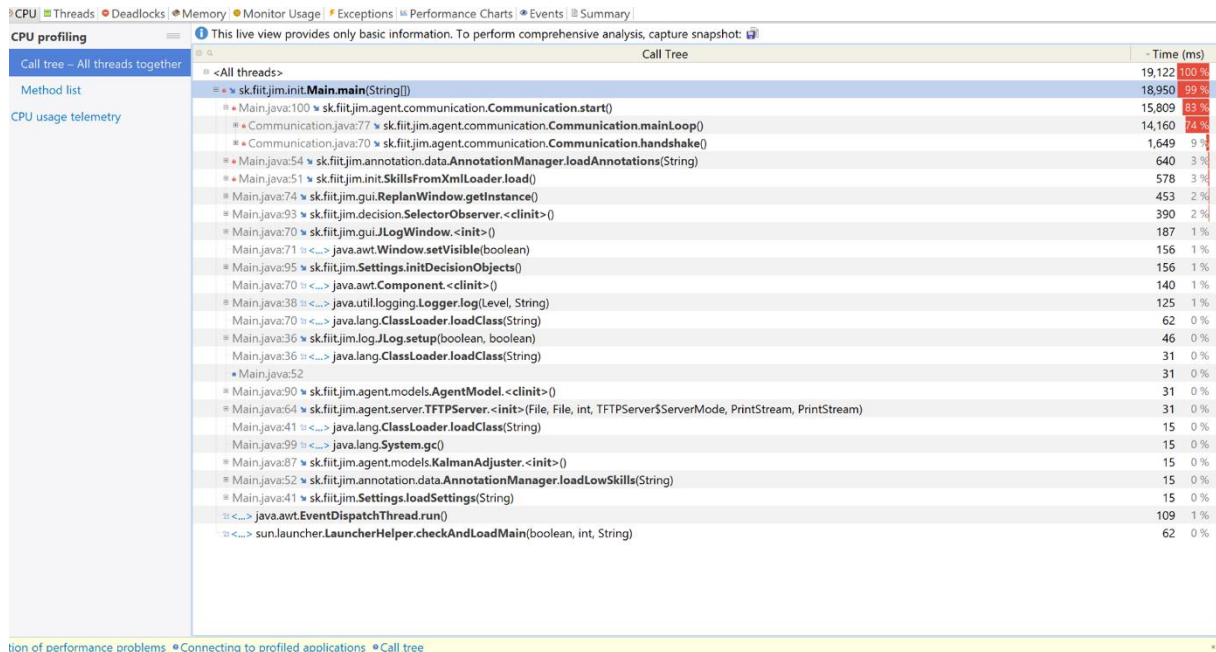


Start CPU Profiling <span style="float: right;">?</span>					
	Description	Overhead	Time accuracy	Invocation counts	Call stacks
<input checked="" type="radio"/> Sampling	Estimate method times by periodically probing stacks of running threads <a href="#">CPU sampling settings...</a>	✓ Low	✓ High for long, low for short methods	✗ No	✓ Yes
<input type="radio"/> Tracing	Accurately measure method (*) times and invocation counts by monitoring enters and exits <a href="#">CPU tracing settings...</a>	✓ Low to high (**)	✓ Usually high (**)	✓ Yes	✓ Yes
<input type="radio"/> Call counting	Count method invocations (*)	✓ Very low	✗ Time is not measured	✓ Yes	✗ No

(\*) - CPU tracing and call counting skip trivial methods by default - see Help for detail  
(\*\*) - depends on applied CPU tracing settings and profiled application

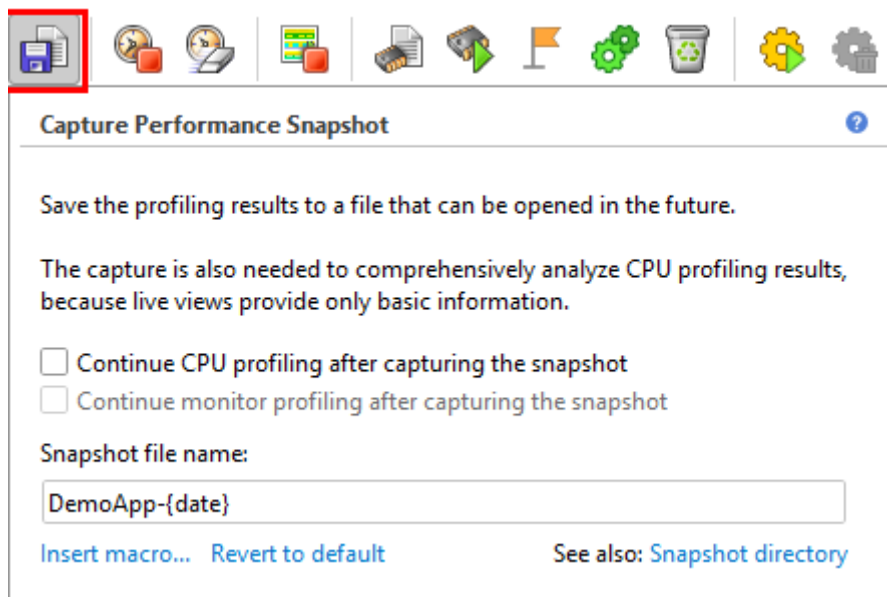
Obrázok 1 Sampling, Tracing, Call Counting

V momente ako je zavolaný CPU profiling môžeme výsledky vidieť v tzv. Call Tree, ktorý môžeme vidieť na nasledujúcom obrázku.



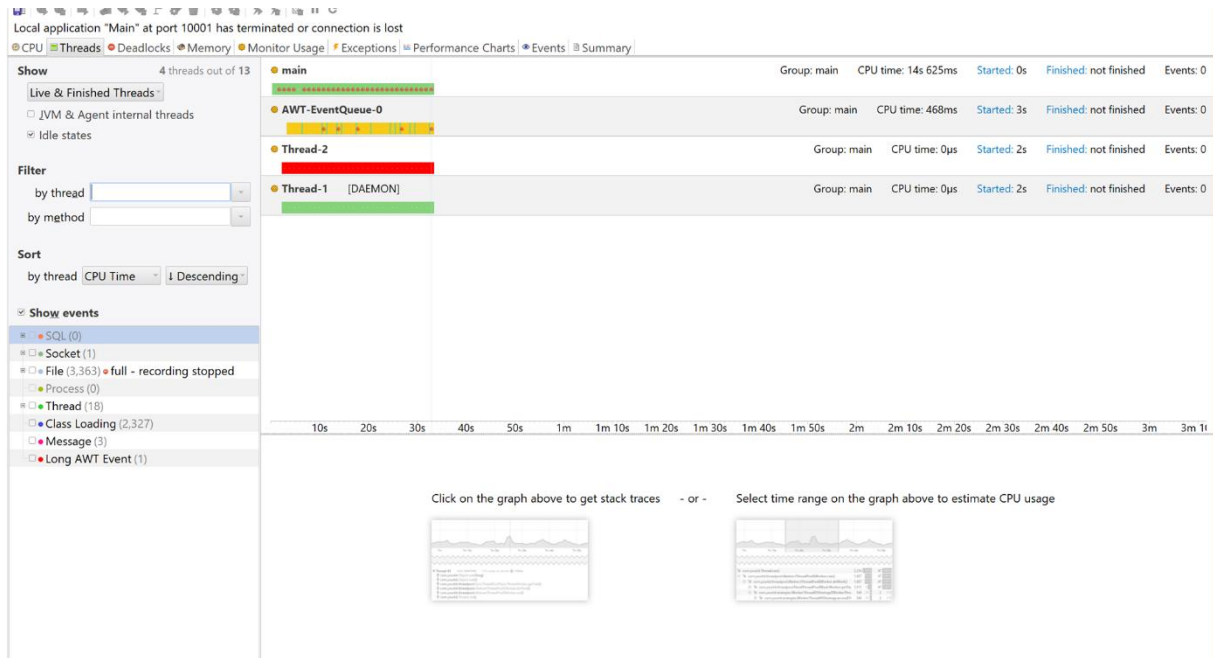
Obrázok 2 CPU profiling

Ak úloha, ktorú sme naplánovali, skončila (alebo bola vykonaná dostatočne dlho), môžeme vytvoriť snapshots CPU so všetkými zaznamenanými informáciami spôsobom, ktorý je zaznamenaný na nasledujúcom obrázku.



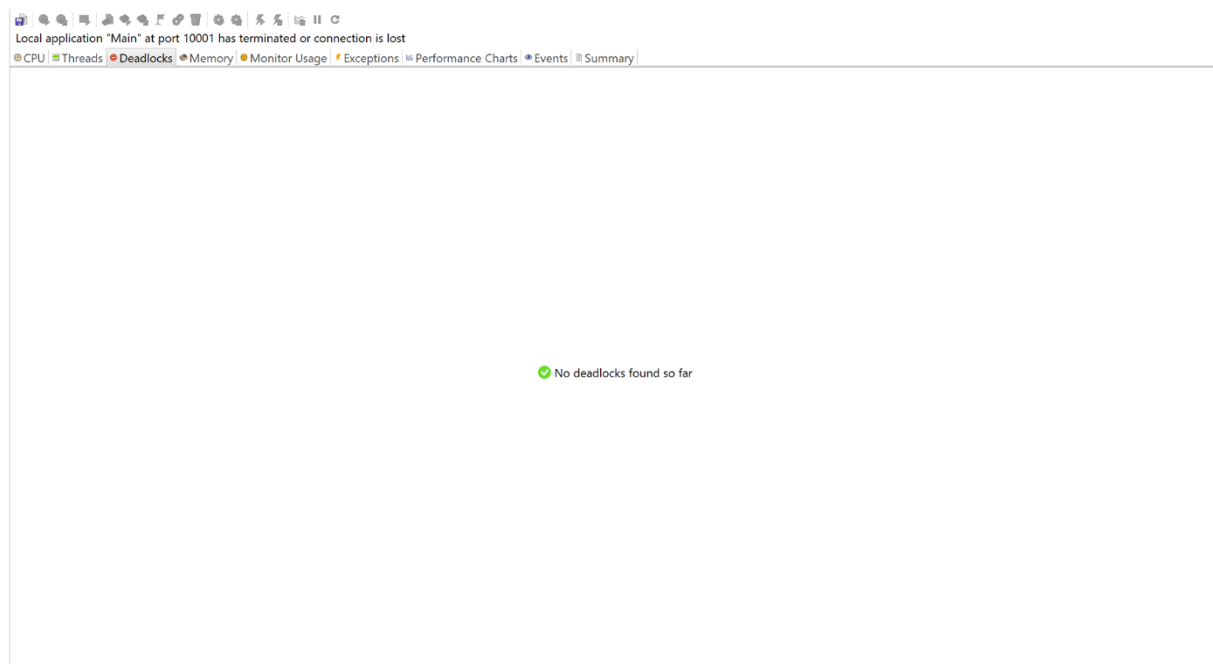
Obrázok 3 Snapshot

V aplikácii si taktiež môžeme pozrieť rozdelenie threadov.



Obrázok 4 Threads

Taktiež máme možnosť zistiť, či vrámci programu vzniká nejaký deadlock.



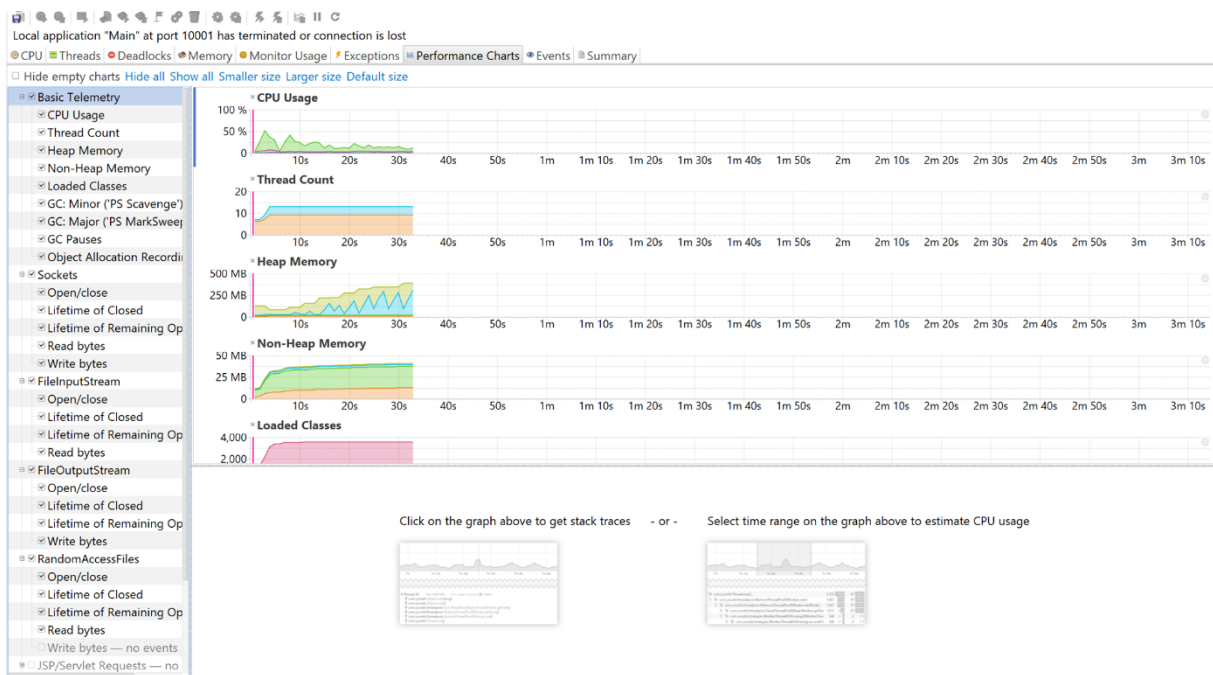
Obrázok 5 Deadlock

Jedno z ďalších vymožeností je taktiež, funkcionlita, kde si môžeme pozrieť ako prebieha alokácia pamäte, kedy a ako sa pamäť využíva.



Obrázok 6 Memory

Taktiež si vrámci aplikácie môžeme zobraziť tzv. Performance Charts na základe, ktorých máme celkový prehľad o využití CPU, Threadov, Heap memory, Non-heap memory a taktiež o počte lodanutých tried.



Obrázok 7 Performance Charts

Sekcia events nám zobrazuje, postup eventov v akom poradí sa vykonávajú na akom threade sa vykonávajú a rôzne iné detaily.

Local application "Main" at port 10001 has terminated or connection is lost

CPU Threads Deadlocks Memory Monitor Usage Exceptions Performance Charts Events Summary

Not all functionality is available in this live view. To perform comprehensive analysis, capture snapshot:

Configure included tables: Events by Table Event Timeline Event Call Tree CPU Usage Estimation

Event	Time Range	Time (ms)	Thread	Stack Trace	Detail
File.Open #1		0	<unknown>	PostVMInitHook.	Path="C:\ProgramData\Oracle\Java\oracle_jre_usage\17dfc292991c7c81.timestamp" Class="java.io.FileO
File.Write #1		0	<unknown>	PostVMInitHook.	Bytes=51 — Path="C:\ProgramData\Oracle\Java\oracle_jre_usage\17dfc292991c7c81.timestamp" Class=
File.Close #1		5	<unknown>	PostVMInitHook.	Path="C:\ProgramData\Oracle\Java\oracle_jre_usage\17dfc292991c7c81.timestamp" Class="java.io.FileO
File.Open #2		1	main	LauncherHelper.c	Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\init\Main.class" Class="java.io.FileInputStream"
File.Read #1		0	main	LauncherHelper.c	Bytes=1,024 — Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\init\Main.class" Class="java.io.FileR
File.Read #2		0	main	LauncherHelper.c	Bytes=2,048 — Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\init\Main.class" Class="java.io.FileR
File.Read #3		0	main	LauncherHelper.c	Bytes=981 — Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\init\Main.class" Class="java.io.FileR
File.Close #2		0	main	LauncherHelper.c	Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\init\Main.class" Class="java.io.FileInputStream"
File.Read #3		0	main	LauncherHelper.c	Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\log\LogType.class" Class="java.io.FileInputStream"
File.Read #4		0	main	LauncherHelper.c	Bytes=1,024 — Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\log\LogType.class" Class="java.io.F
File.Read #5		0	main	LauncherHelper.c	Bytes=145 — Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\log\LogType.class" Class="java.io.Fil
File.Close #3		0	main	LauncherHelper.c	Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\log\LogType.class" Class="java.io.FileInputStream"
File.Open #4		0	main	LauncherHelper.c	Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\agent\parsing\ParsedDataObserver.class" Class="je
File.Open #6		0	main	LauncherHelper.c	Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\agent\parsing\ParsedDataObserver.class" Class="je
File.Close #4		0	main	LauncherHelper.c	Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\agent\parsing\ParsedDataObserver.class" Class="je
File.Open #5		0	main	Main.main	Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\log\Log.class" Class="java.io.FileInputStream"
File.Read #7		0	main	Main.main	Bytes=1,024 — Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\log\Log.class" Class="java.io.FileIn
File.Read #8		0	main	Main.main	Bytes=1,820 — Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\log\Log.class" Class="java.io.FileIn
File.Close #5		0	main	Main.main	Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\log\Log.class" Class="java.io.FileInputStream"
File.Open #6		0	main	Main.main	Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\log\JmConsoleFormatter.class" Class="java.io.FileI
File.Read #9		0	main	Main.main	Bytes=1,024 — Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\log\JmConsoleFormatter.class" Cl
File.Read #10		0	main	Main.main	Bytes=1,536 — Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\log\JmConsoleFormatter.class" Cl
File.Close #6		0	main	Main.main	Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\log\JmConsoleFormatter.class" Class="java.io.Fil
File.Open #7		0	main	Main.main	Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\log\JmHtmlFormatter.class" Class="java.io.FileInp
File.Read #11		0	main	Main.main	Bytes=1,024 — Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\log\JmHtmlFormatter.class" Class=
File.Read #12		0	main	Main.main	Bytes=2,048 — Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\log\JmHtmlFormatter.class" Class=
File.Read #13		0	main	Main.main	Bytes=1,822 — Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\log\JmHtmlFormatter.class" Class=
File.Close #7		0	main	Main.main	Path="C:\workspace\Bareko\agent\jvm\bin\sk\fiitj\j\log\JmHtmlFormatter.class" Class="java.io.FileInp
File.Open #8		0	main	JLog.setup	Path="j\j\logs.html" Class="java.io.FileOutputStream"

Obrázok 8 Events

## Sekcia summary, nám udáva základný všeobecný prehľad.

Local application "Main" at port 10001 has terminated or connection is lost

CPU Threads Deadlocks Memory Monitor Usage Exceptions Performance Charts Events Summary

Runtime & Agent

Java Virtual Machine: Java HotSpot(TM) 64-Bit Server VM: 1.8.0\_161; 25.161-b12; mixed mode

Vendor: Oracle Corporation

Start time: April 22, 2018 03:53:59 PM

Uptime: 32s

CPU time: 25s

Command line: C:\Program Files\Java\jre1.8.0\_161\bin\javaw.exe -agentpath:C:\ProgramData\YourKit\2017.02.75.8EA557DF\64\ypagent.dll=sampling\_ide\_project\_name=Jim,sessionname=Main,profiler\_dir=YJPQUOTE...

VM arguments: -agentpath:C:\ProgramData\YourKit\2017.02.75.8EA557DF\64\ypagent.dll=sampling\_ide\_project\_name=Jim,sessionname=Main,profiler\_dir=YJPQUOTED433a5c50726f772616d2046696c65735c596f75...

Class path: C:\Program Files\Java\jre1.8.0\_161\lib\resources.jar;C:\Program Files\Java\jre1.8.0\_161\lib\rt.jar;C:\Program Files\Java\jre1.8.0\_161\lib\jse.jar;C:\Program Files\Java\jre1.8.0\_161\lib\jce.jar;C:\Program Files\...

Boot class path: C:\Program Files\Java\jre1.8.0\_161\lib\resources.jar;C:\Program Files\Java\jre1.8.0\_161\lib\rt.jar;C:\Program Files\Java\jre1.8.0\_161\lib\sunrsasign.jar;C:\Program Files\Java\jre1.8.0\_161\lib\jse.jar;C:\Progra...

Library path: C:\Program Files\Java\jre1.8.0\_161\bin;C:\Windows\Sun\Java\bin;C:\Windows\system32;C:\Windows;C:\Program Files\Java\jre1.8.0\_161\bin;server;C:\Program Files\Java\jre1.8.0\_161\bin;C:\Program Files\...

System properties: ...

Agent version: YourKit Java Profiler 2017.02-b75

Agent mode: Loaded on start

Heap Memory	Non-Heap Memory	Garbage Collector
Used: 305 MB	Used: 39 MB	Collections: 16
Allocated: 385 MB	Allocated: 40 MB	Time: 0s
Limit: 1.8 GB	Limit: unknown	

Classes	Threads	Operating System
Currently loaded: 3,527	Currently live: 13	Name: Windows 10
Total unloaded: 0	Currently live daemons: 9	Version: 10.0.16299
	Peak: 14	Architecture: amd64
	Total created: 18	Processors: 4

Automatic Deobfuscator

If profiled application was obfuscated, the profiler can automatically restore original names of classes, methods and fields.

[Configure deobfuscator...](#)

Obrázok 9 Summary