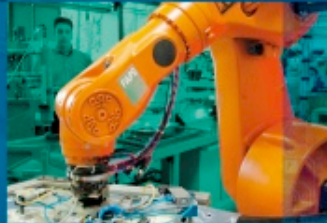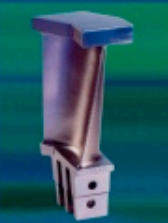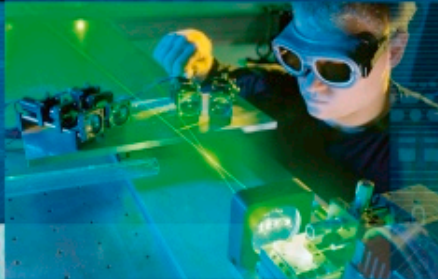# A microbenchmark-based analysis of Kieker's garbage collection activity

Andreas Kumlehn
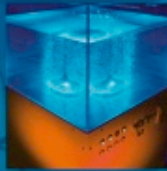
Agenda

1. Overhead benchmarking with MooBench
2. Analyzing the GCs
3. Profiling Kieker

MooBench setup

- **Parameters**
  - ☐ Two workloads: 0 mus and 500 mus
  - ☐ Single threaded driver – no ThreadMXBean
  - ☐ Logging to disk

- **Evaluated tools**
  - ☐ Kieker
  - ☐ inspectIT: raw timer to monitor each invocation
  - ☐ SWBB
  - ☐ noProbe

MooBench results – workload 500 mus
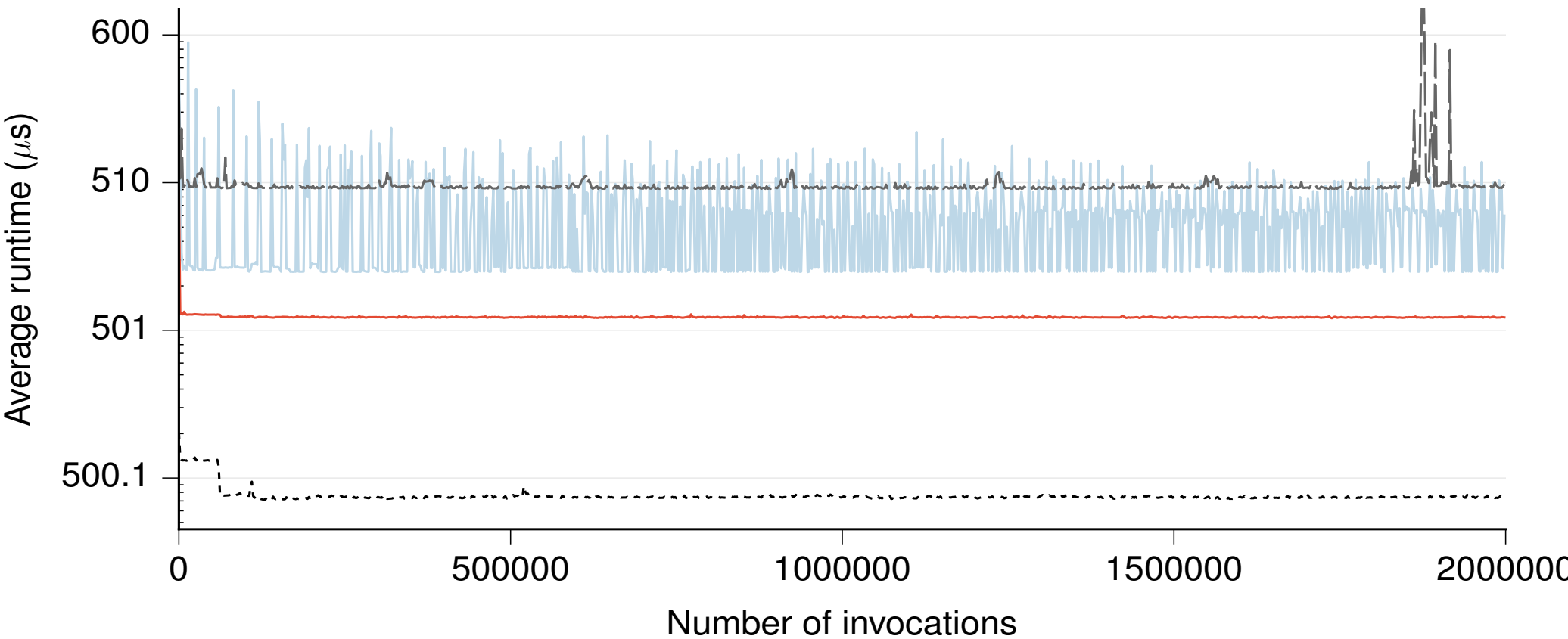
- Note: log-scale
- Kieker: Spikes due to GCs
- inspectIT: massive DB delay at the end

MooBench results – workload 0 mus
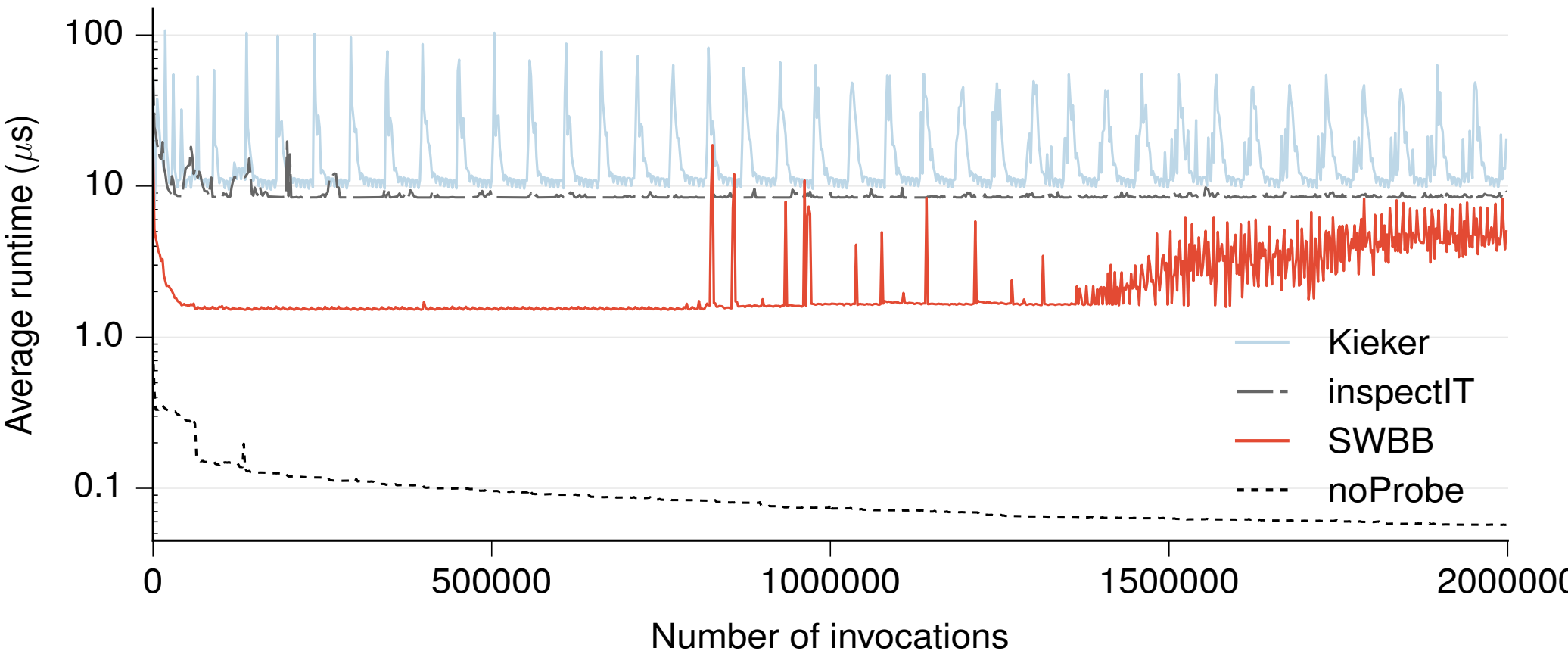
- Note: log-scale
- Similar overall patterns

GC protocols – the numbers of Kieker and inspectIT

Count and time	Kieker	inspectIT
number of GCs	43	66
full GCs	1	1
total time	7.35s	0.22s
... in full GCs	0.11s	0.04s
... in minor GCs	7.23	0.18

- Memory freeing behavior is similar in both tools

Visualizing GC protocols - inspectIT
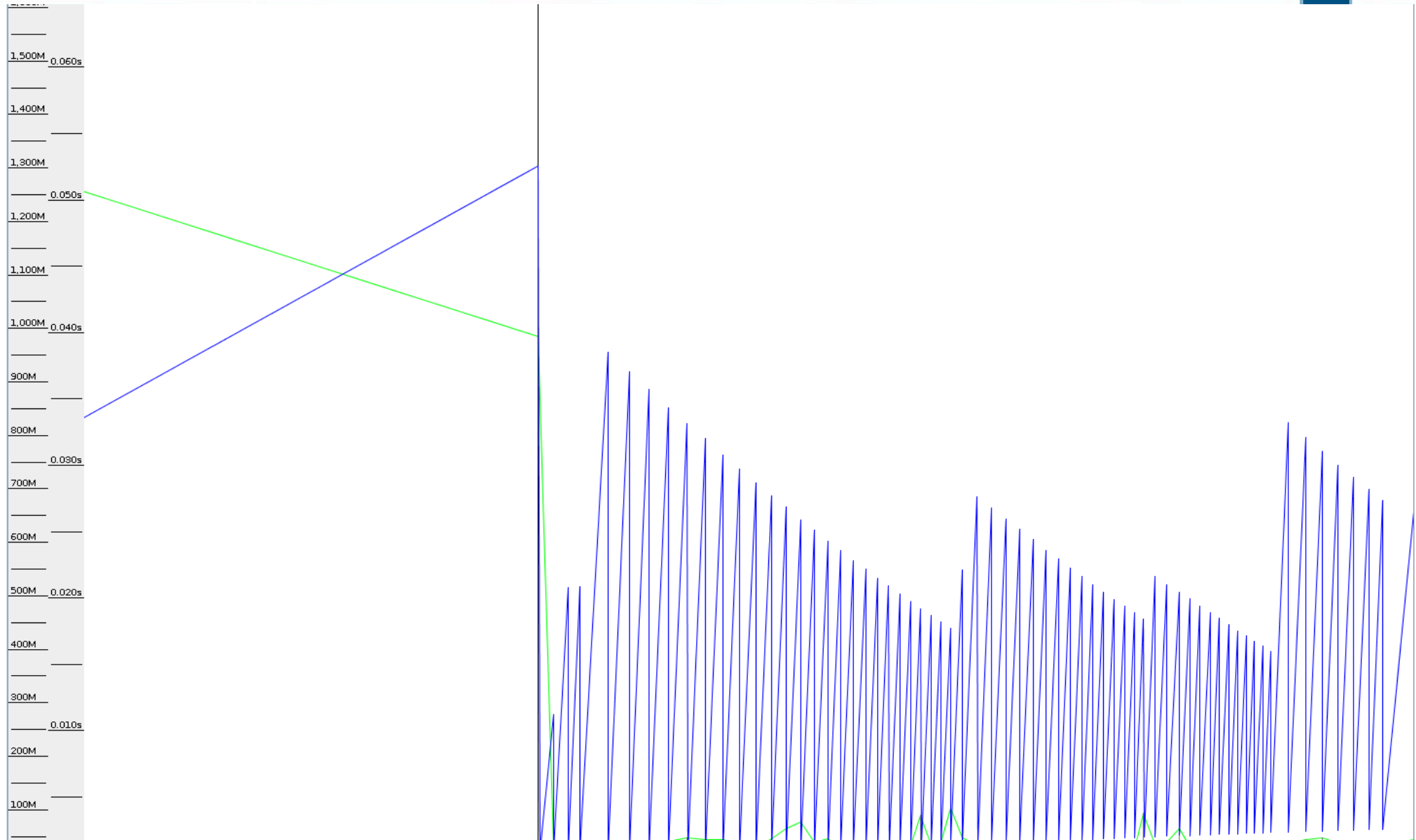
Visualizing GC protocols - Kieker
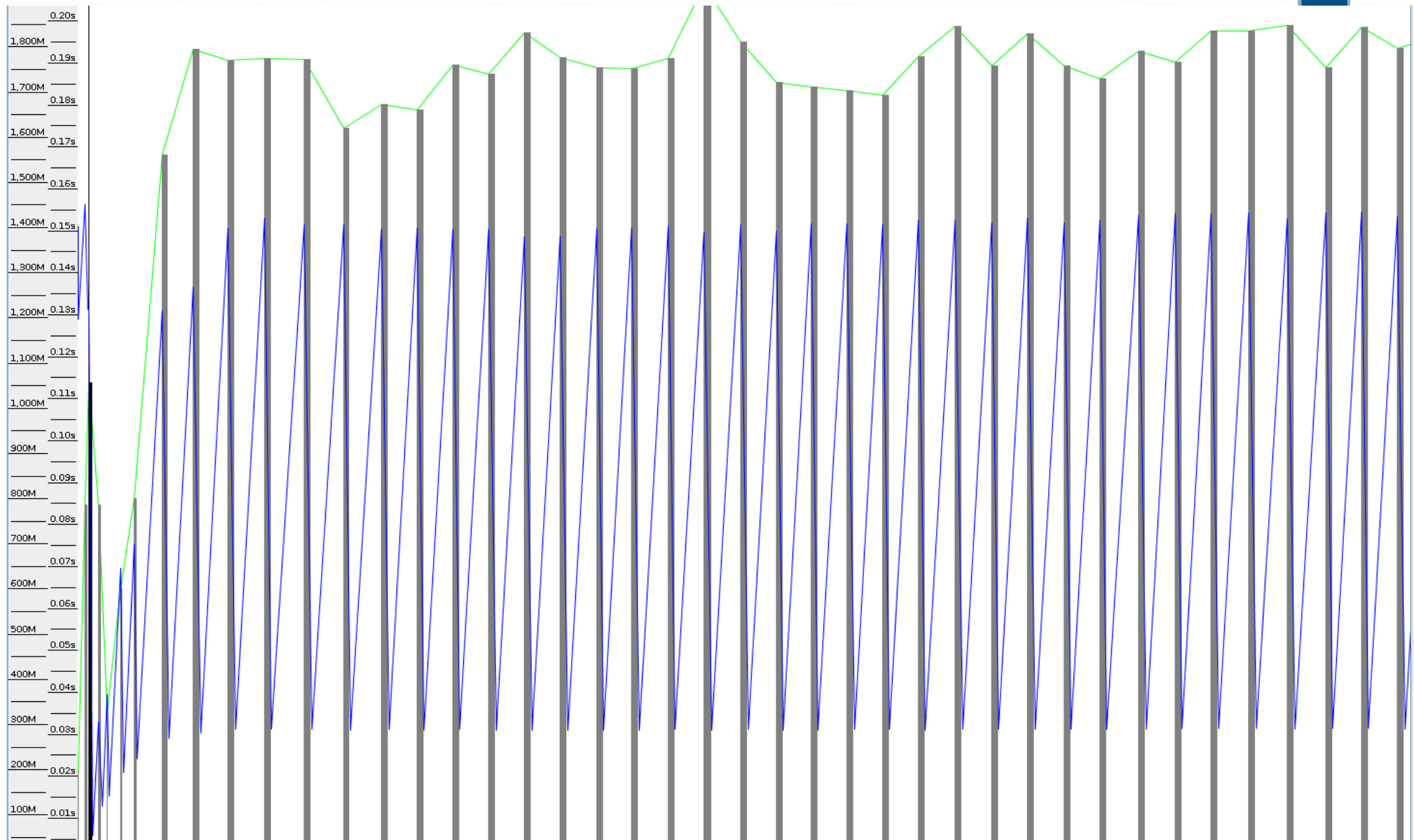
Profiling Kieker – Which objects are on the heap?

- Ca 1.450.000 invocations of the monitoredMethod
- Massive amount of nio-related objects
- Ratio 3:1

Name	Instance count ▼	Size
sun.misc.Cleaner	4,477,578	179 MB
ava.nio.DirectByteBuffer$Deallocator	4,477,578	143 MB
ava.lang.Long	4,357,586	104 MB
ava.nio.DirectByteBuffer	3,050,826	195 MB
ava.lang.Integer	2,905,058	46,480 kB
kieker.common.record.flow.trace.operation.AfterOperatio...	1,452,778	81,355 kB
kieker.common.record.flow.trace.operation.BeforeOperat...	1,452,777	81,355 kB
org.aspectj.runtime.reflect.JoinPointImpl	1,452,529	46,480 kB
nooBench.monitoredApplication.MonitoredClassSimple$...	1,452,529	34,860 kB
ava.util.concurrent.locks.AbstractQueuedSynchronizer$N...	370,167	11,845 kB
kieker.common.record.flow.trace.TraceMetadata	145,277	9,297 kB
ava.lang.ThreadLocal$ThreadLocalMap$Entry	145,255	4,648 kB
ava.lang.String	1,523	36,552 bytes
ava.lang.StringBuilder	488	11,712 bytes

Allocation call tree

- ## Entry point for further evaluation and optimization

Recorded allocations of: **All classes**

Liveness mode: **Live objects**

Aggregation level: m Methods   View mode: ≡ Tree

- m 64.2% - 42,753 kB - 1,076,398 alloc. kieker.monitoring.writer.AbstractAsyncThread.run
  - m 64.2% - 42,732 kB - 1,075,726 alloc. kieker.monitoring.writer.filesystem.async.AbstractFsWriterThread.consume
    - m 64.2% - 42,724 kB - 1,075,461 alloc. kieker.monitoring.writer.filesystem.async.BinaryFsWriterThread.write
      - m 64.2% - 42,724 kB - 1,075,461 alloc. java.nio.ByteBuffer.allocateDirect
    - m 0.0% - 3,744 bytes - 132 alloc. kieker.monitoring.writer.filesystem.async.AbstractFsWriterThread.getFilename
      - m 0.0% - 2,496 bytes - 90 alloc. java.lang.String.format
      - m 0.0% - 672 bytes - 18 alloc. java.text.DateFormat.format
      - m 0.0% - 144 bytes - 6 alloc. java.lang.StringBuilder.toString
      - m 0.0% - 144 bytes - 6 alloc. java.util.Date.<init>
      - m 0.0% - 144 bytes - 6 alloc. java.lang.StringBuilder.<init>
    - m 0.0% - 3,704 bytes - 133 alloc. kieker.monitoring.writer.filesystem.async.BinaryFsWriterThread.prepareFile
      - m 0.0% - 2,672 bytes - 90 alloc. kieker.common.util.filesystem.BinaryCompressionMethod$1.getDataOutputStream
      - m 0.0% - 936 bytes - 37 alloc. java.io.File.<init>
      - m 0.0% - 96 bytes - 6 alloc. java.io.DataOutputStream.close
- m 35.8% - 23,798 kB - 710,559 alloc. mooBench.benchmark.BenchmarkingThreadNanoWithTimestamps.run
  - m 35.8% - 23,798 kB - 710,559 alloc. mooBench.monitoredApplication.MonitoredClassSimple.monitoredMethod
    - m 34.4% - 22,866 kB - 669,441 alloc. kieker.monitoring.probe.aspectj.flow.operationExecution.AbstractAspect.operation
      - m 31.2% - 20,773 kB - 630,620 alloc. org.aspectj.lang.ProceedingJoinPoint.proceed
        - m 31.2% - 20,773 kB - 630,620 alloc. mooBench.monitoredApplication.MonitoredClassSimple$AjcClosure1.run
          - m 31.0% - 20,608 kB - 623,767 alloc. mooBench.monitoredApplication.MonitoredClassSimple.monitoredMethod_aroundBody0
            - Further allocation information is unavailable because the profiled JVM has terminated
      - m 0.2% - 164 kB - 6,853 alloc. org.aspectj.runtime.internal.Conversions.longObject