

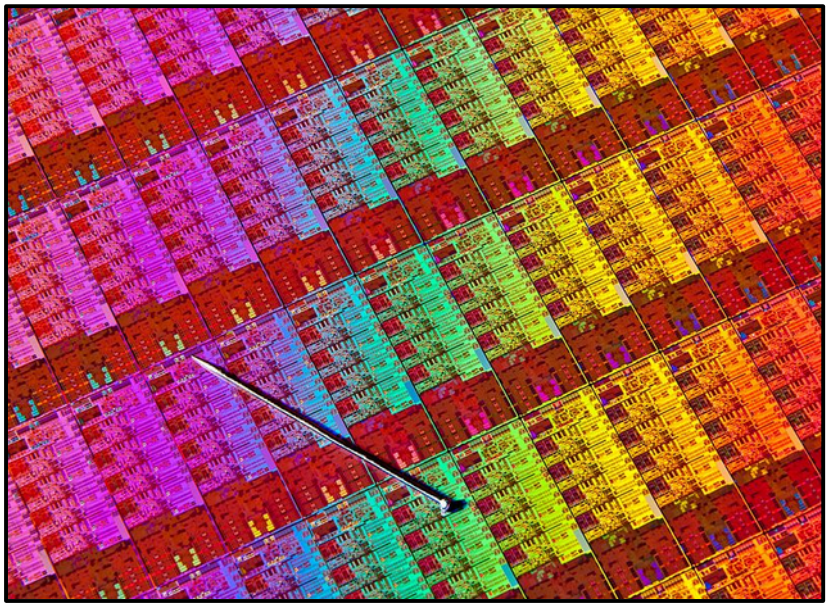


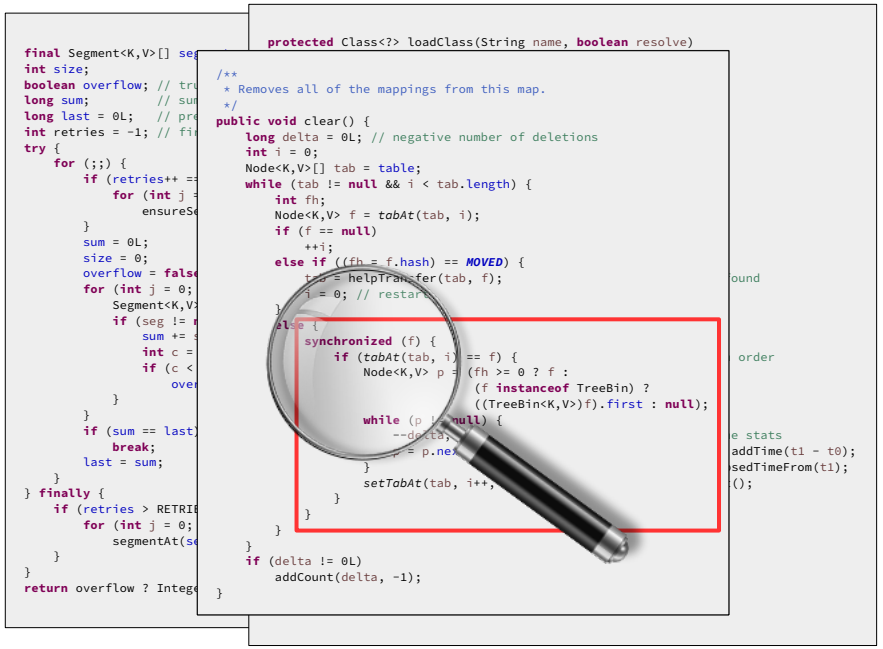
# Efficient Tracing and Versatile Analysis of Lock Contention in Java Applications on the Virtual Machine Level

**Peter Hofer**
David Gnedt
Andreas Schörgenhumer
Hanspeter Mössenböck

16 March 2016

# The Need for Analyzing Lock Contention




*Images: Intel Haswell Chip, Intel Free Press, licensed under Creative Commons BY-SA 2.0 / System Lock, Yuri Samoilov, CC BY 2.0*

# Java Intrinsic Object Locks

```
class SynchronizedArrayList {
  ArrayList list = new ArrayList();


  void get(int index) {
    synchronized(list) {
      return list.get(index);
    }
  }

  void add(int index, Object obj) {
    synchronized(list) {
      return list.add(index, obj);
    }
  }
}
```

# java.util.concurrent Locks

```
class SynchronizedArrayList {
  ArrayList list = new ArrayList();
  final ReentrantLock lock = new ReentrantLock();

  void get(int index) {
    lock.lock();
    try { return list.get(index);
    } finally { lock.unlock(); }
  }

  void add(int index, Object obj) {
    lock.lock();
    try { return list.add(index, obj);
    } finally { lock.unlock(); }
  }
}
```

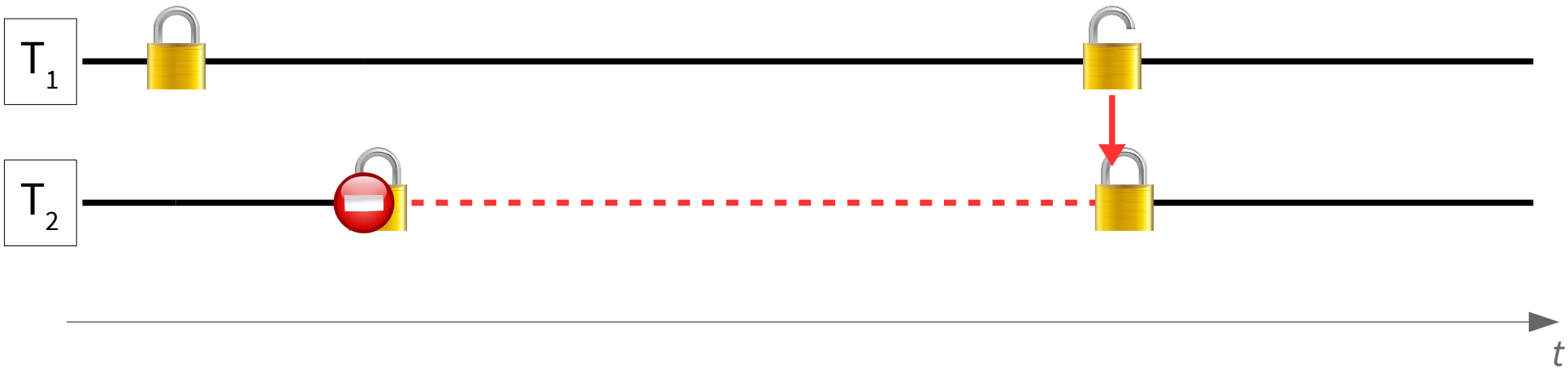
# Java Synchronization

**Locking is cheap:** fast compare-and-set operations

**Contention is expensive**

no progress in blocked threads

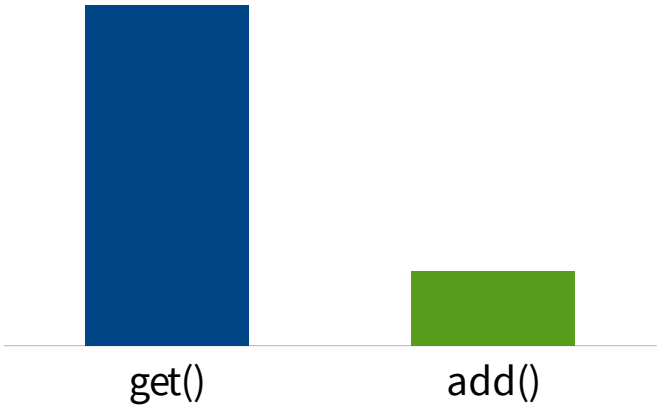
threads must coordinate



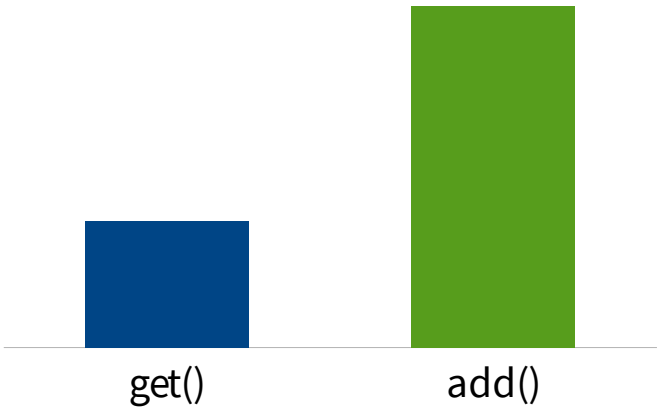

*Images: KDE Oxygen Icon Project, licensed under the LGPL*

# Blame

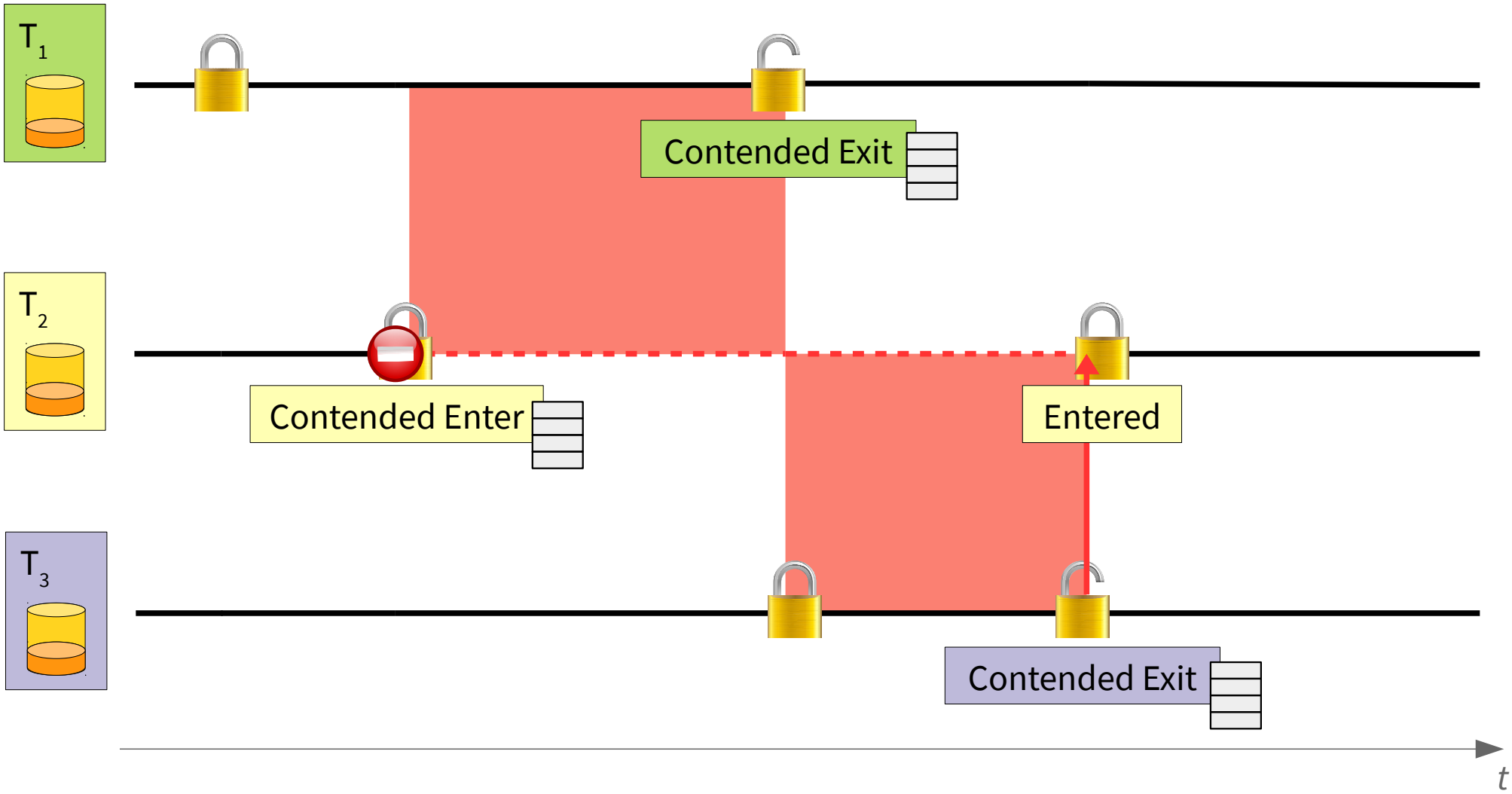
Time spent waiting **at**...




Time spent waiting **for**...



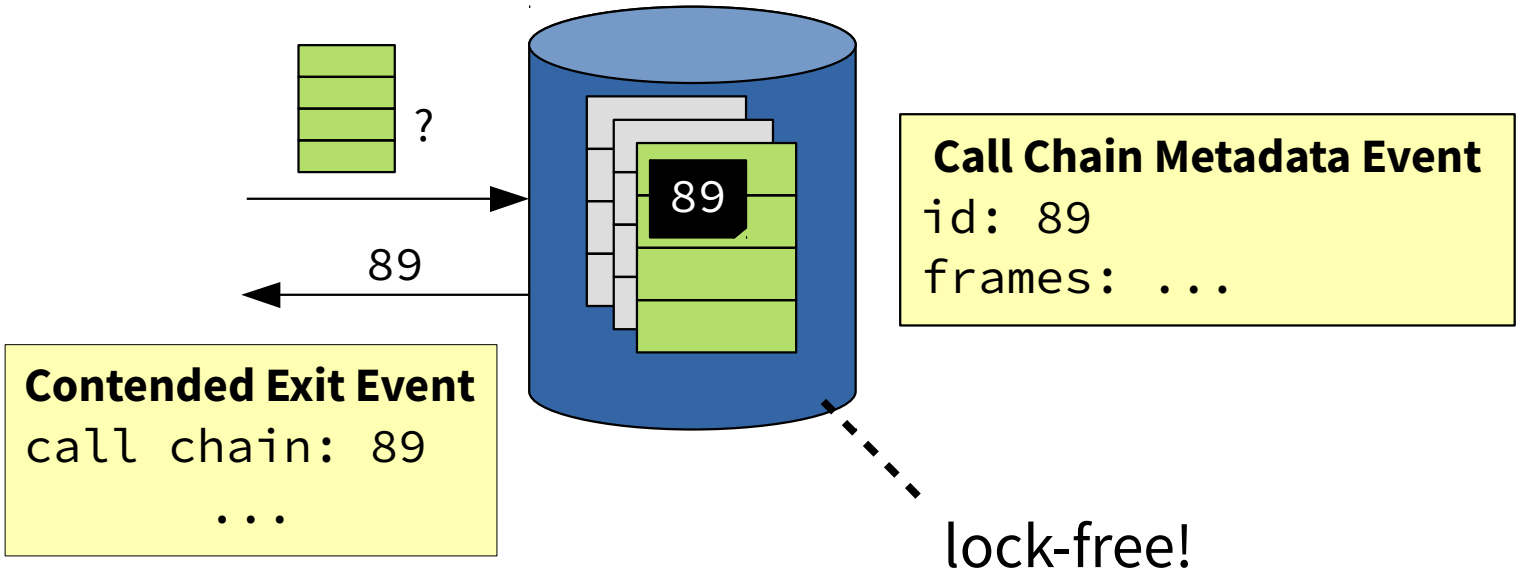
# Tracing Monitors in the HotSpot Java VM



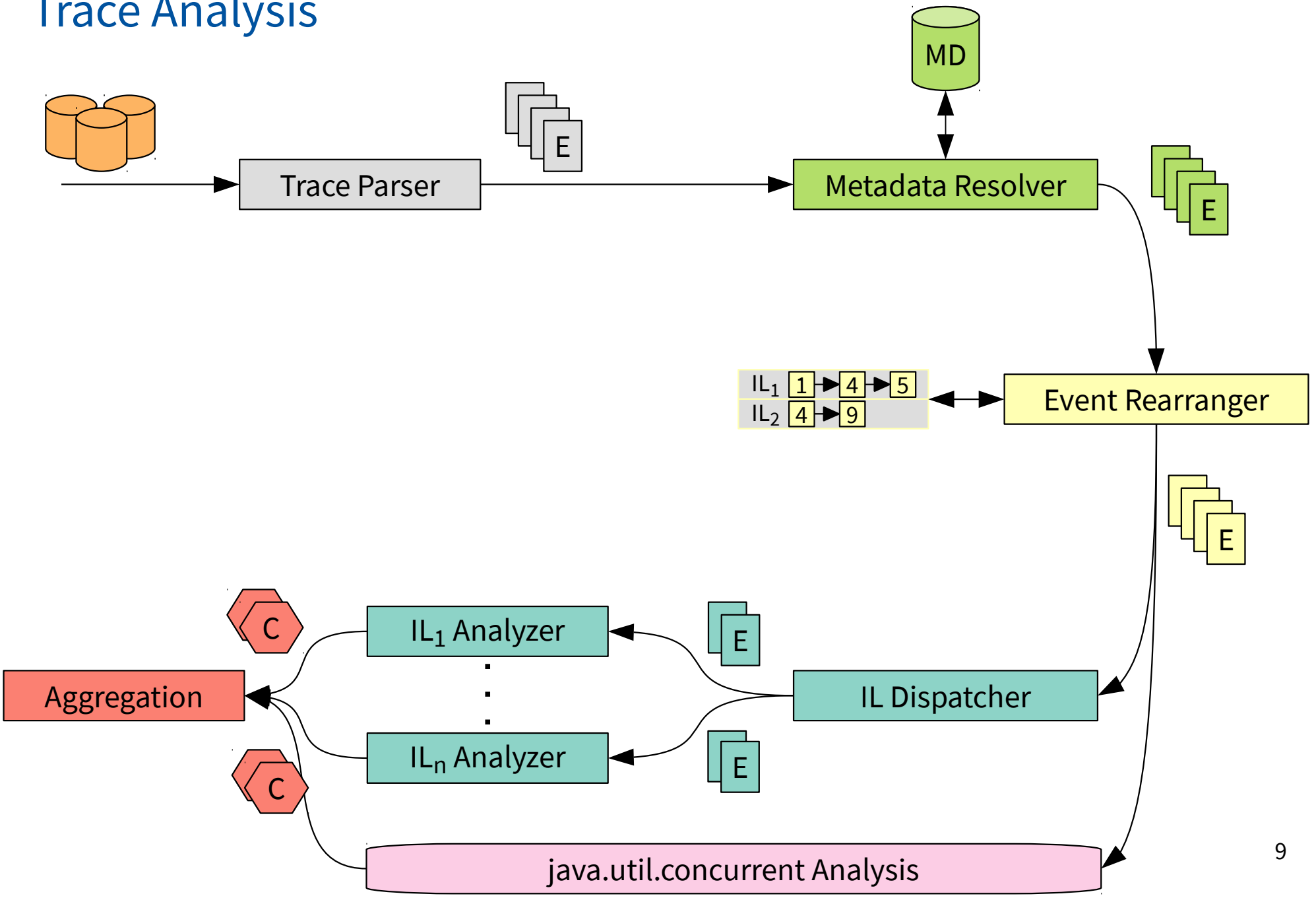
# Metadata

Metadata recorded in special events

- Call chains
  - execution state – no identity
  - many identical call chains

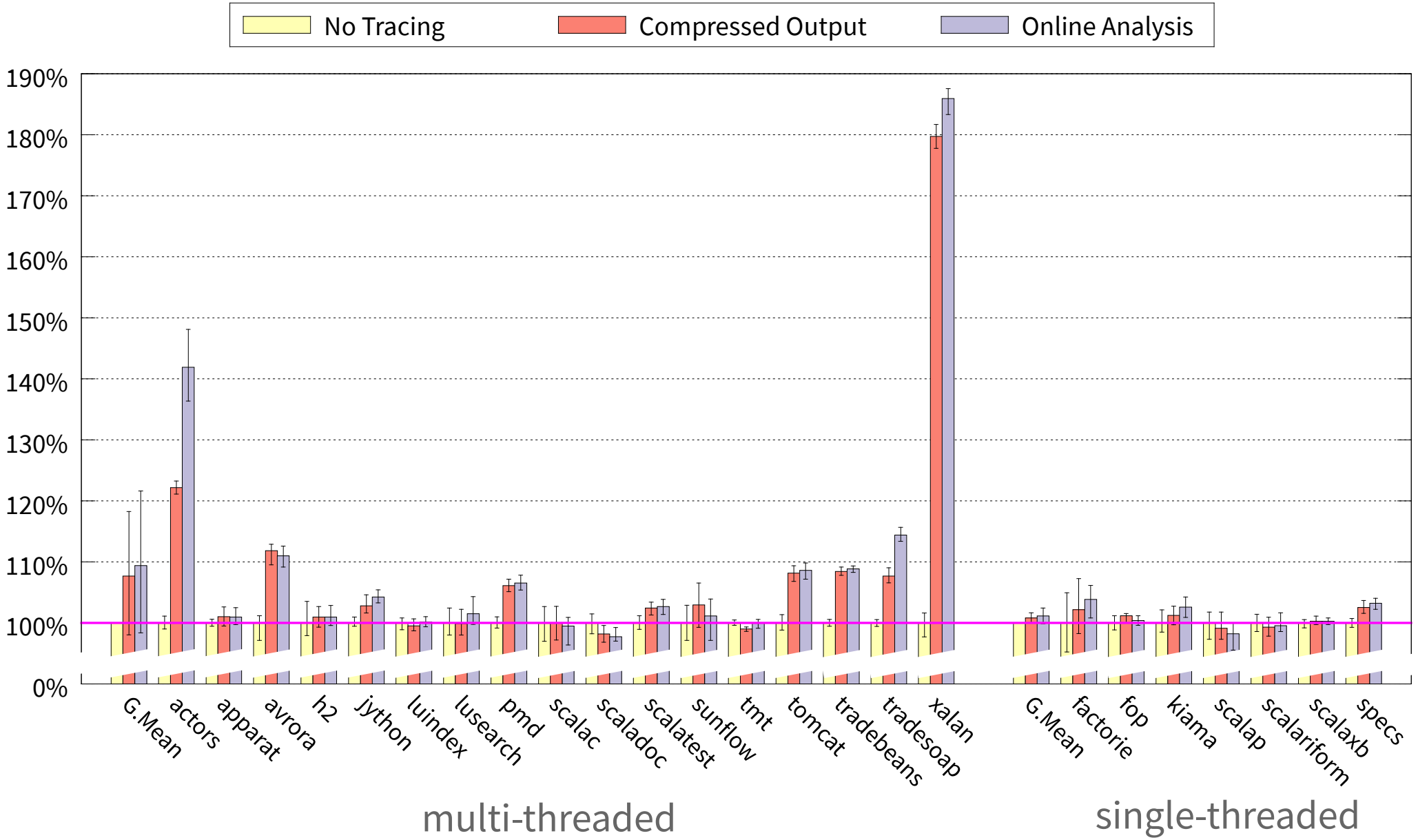


# Trace Analysis



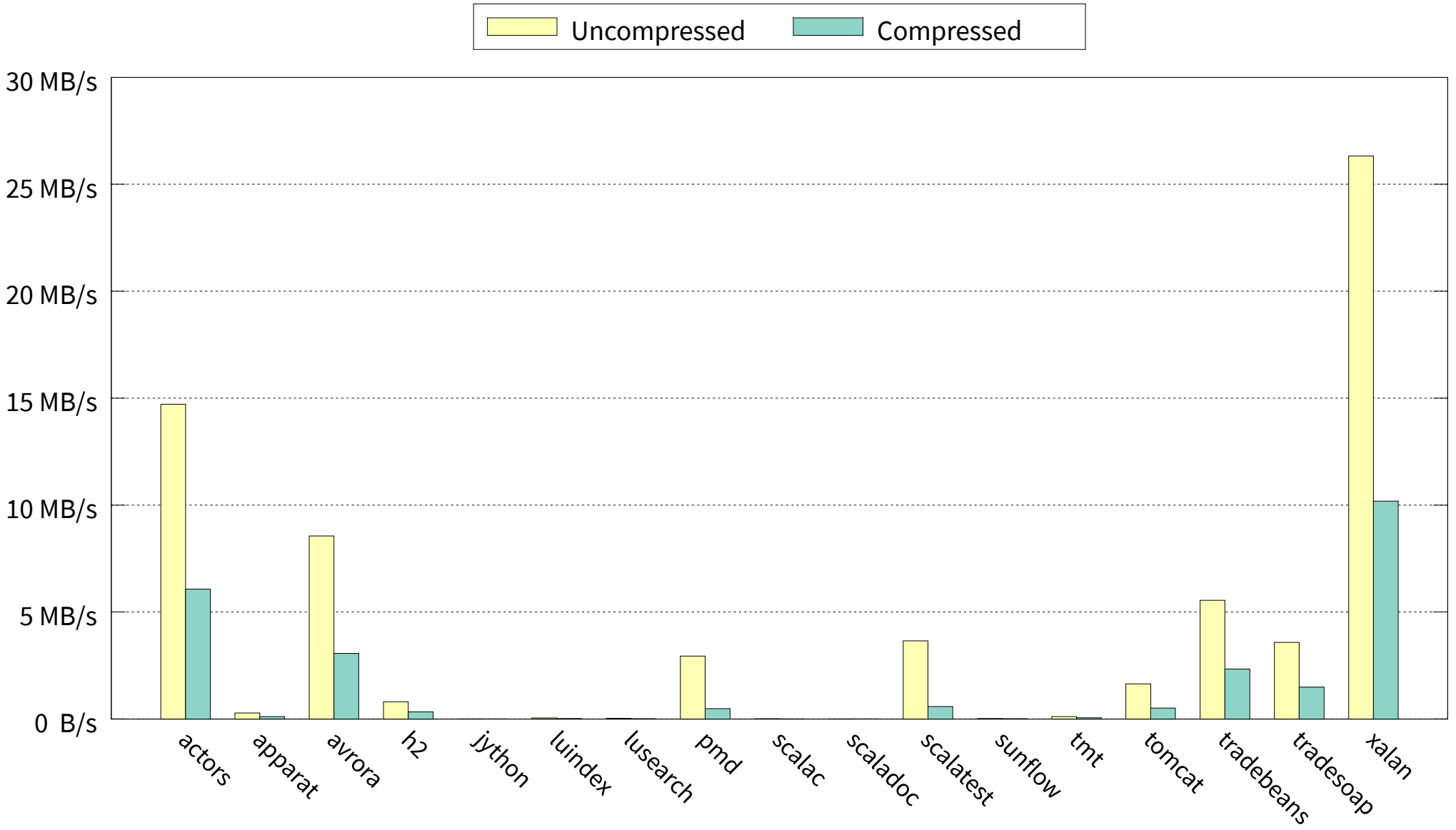
# Overhead

DaCapo and scalabench benchmarks (except batik and eclipse)
2x Xeon E5-2670v2 = 20 cores + HT, Oracle Linux 7, OpenJDK 8u45-b14



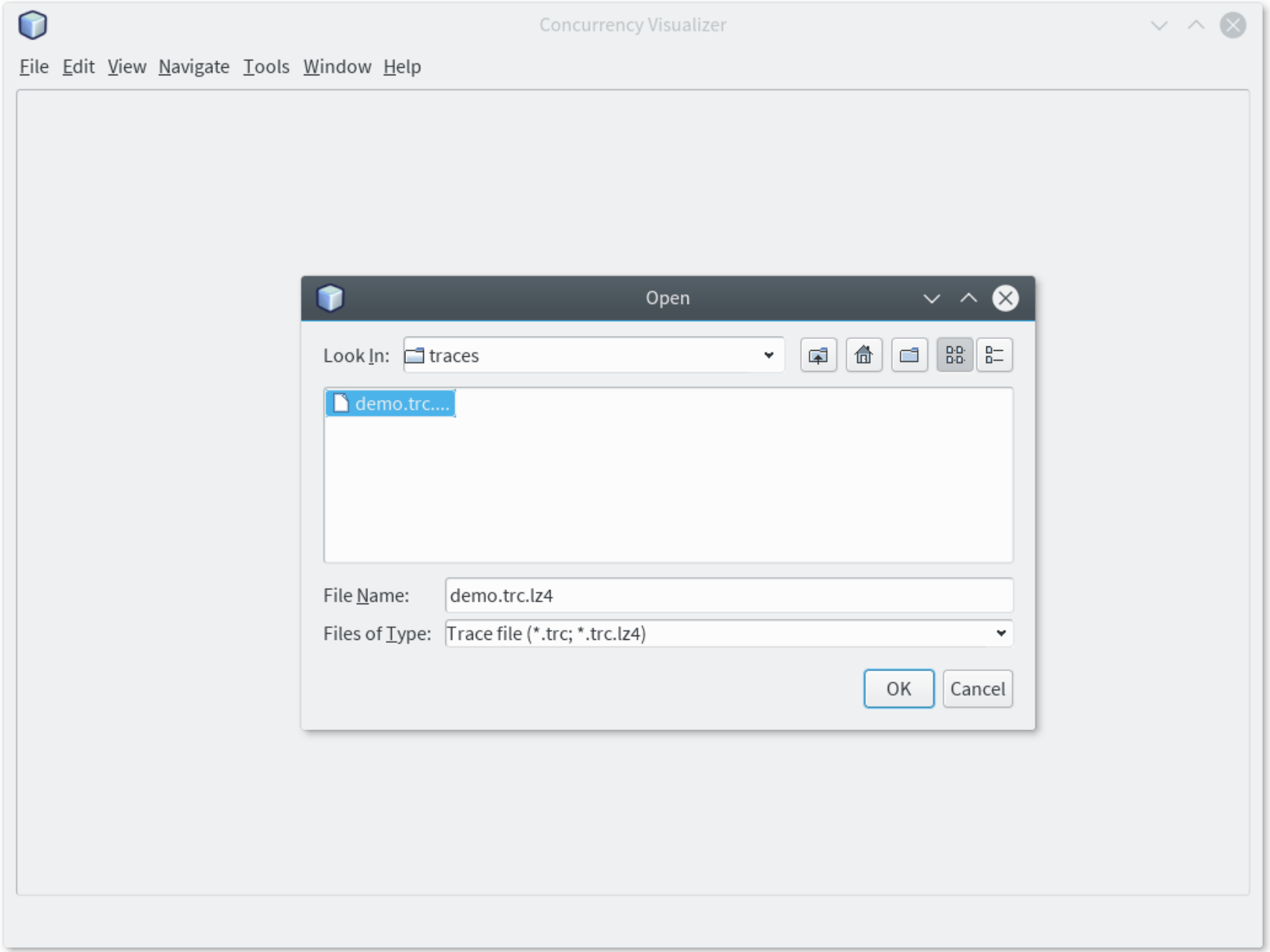
# Bytes per Second

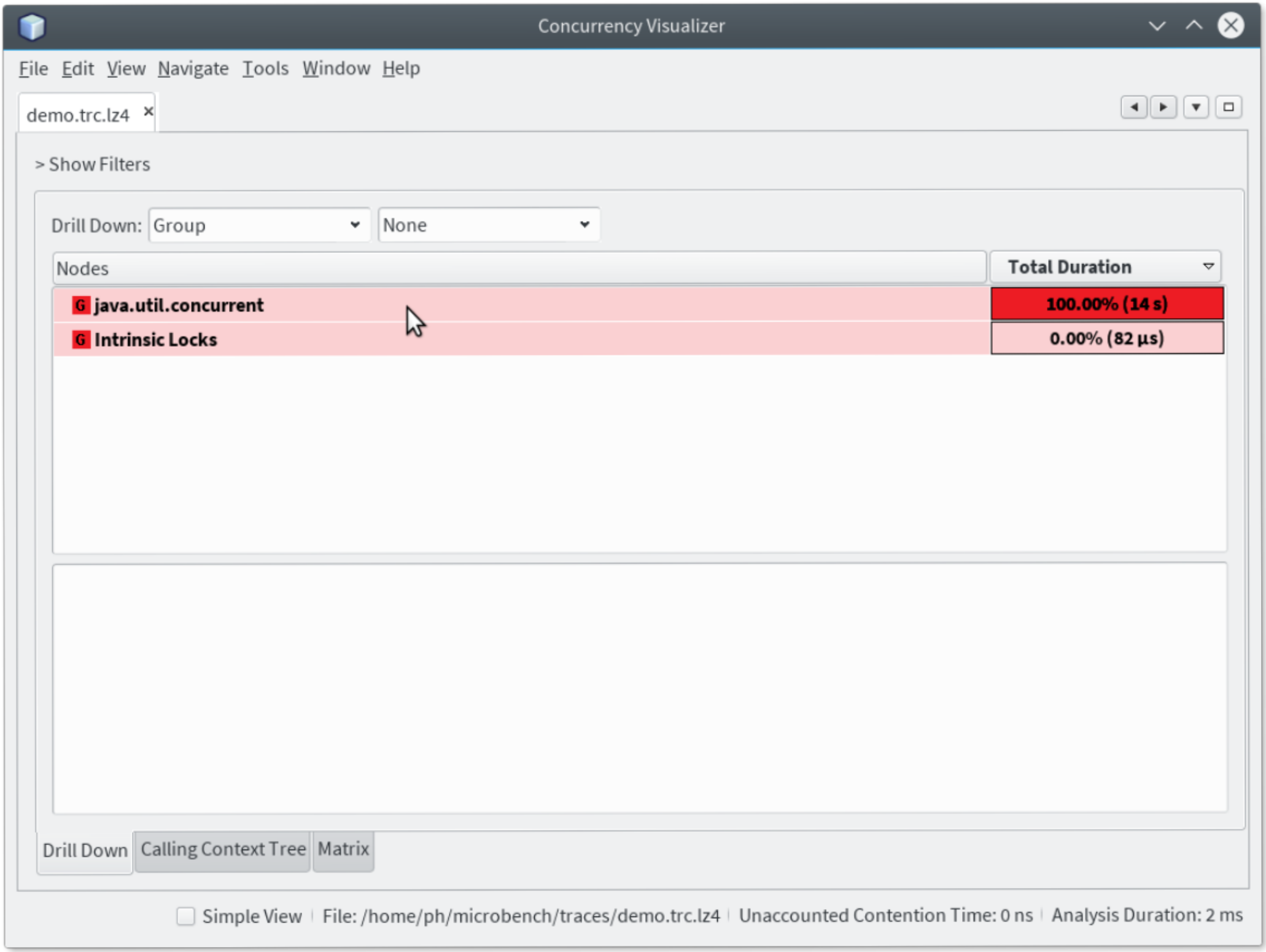
DaCapo and scalabench benchmarks (except batik and eclipse)
2x Xeon E5-2670v2 = 20 cores + HT, Oracle Linux 7, OpenJDK 8u45-b14

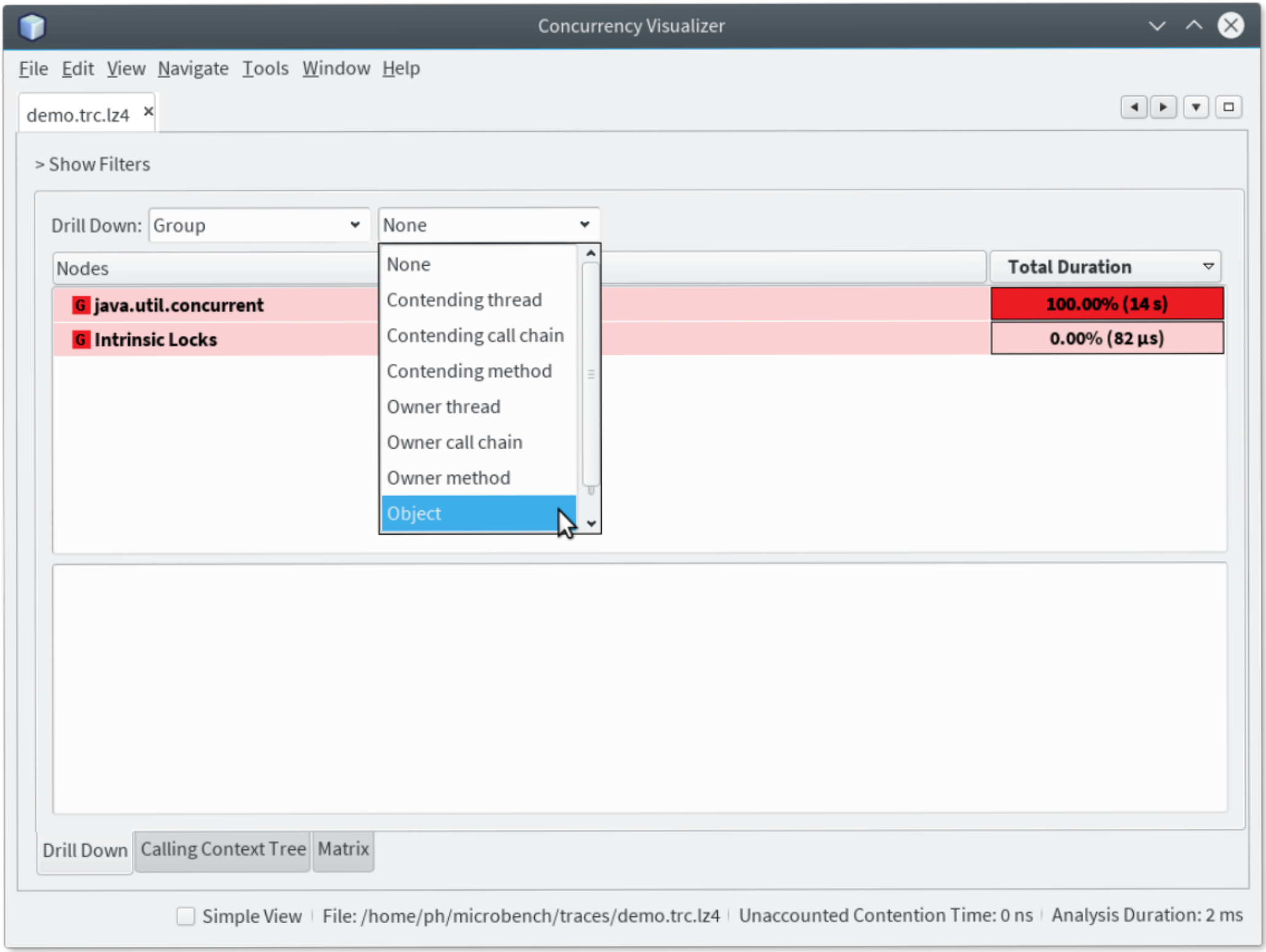


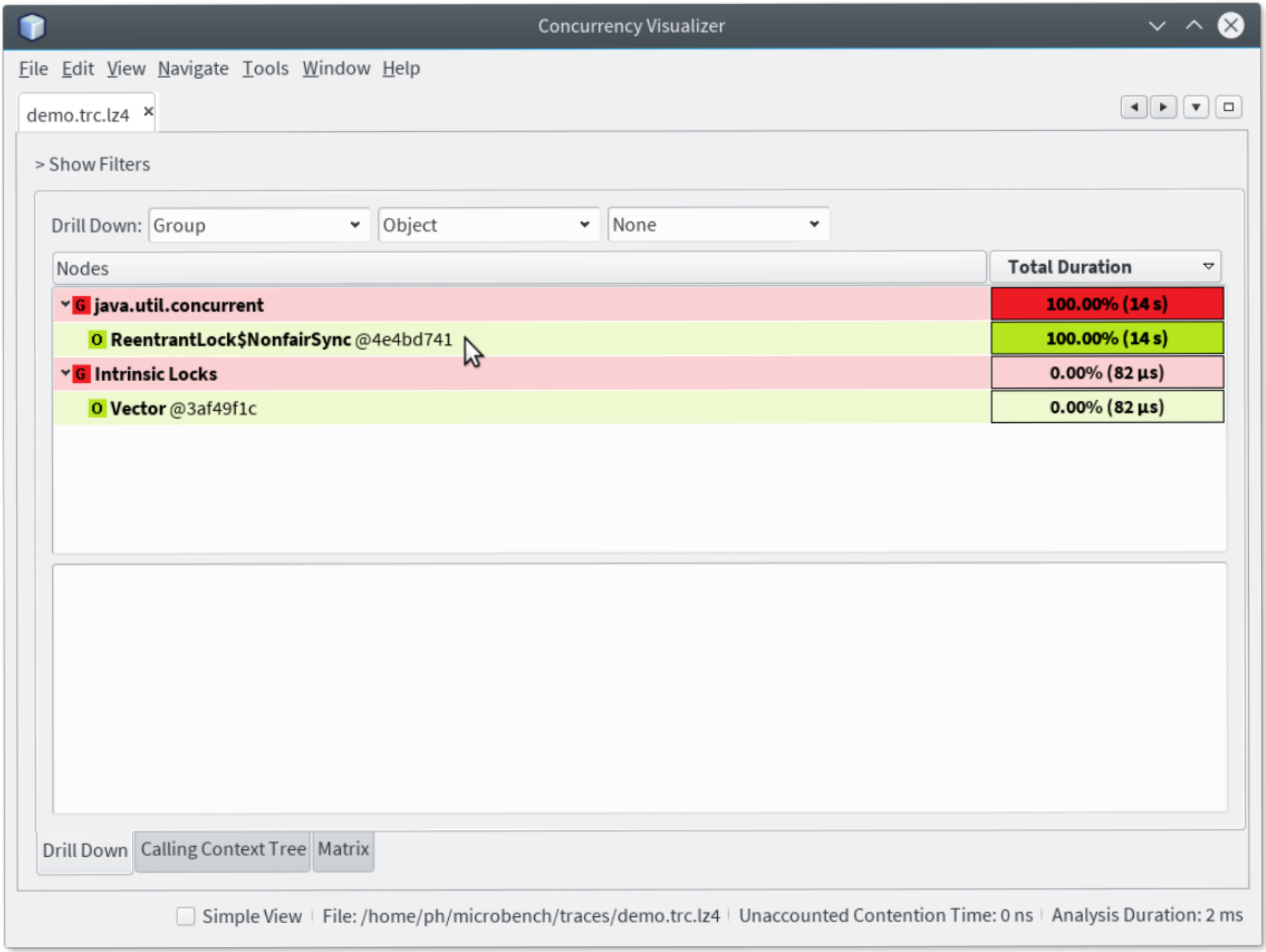

multi-threaded

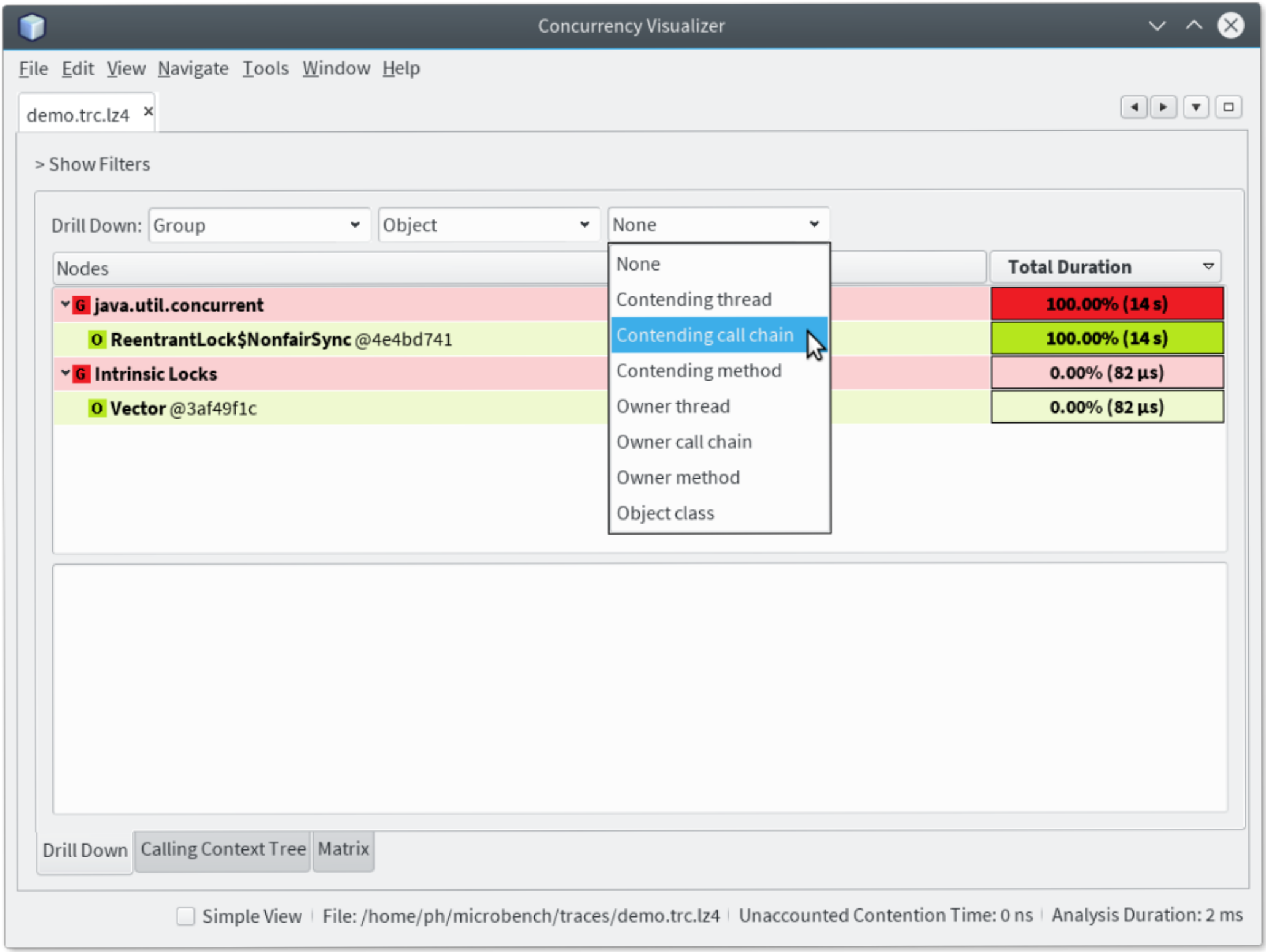
# Demonstration

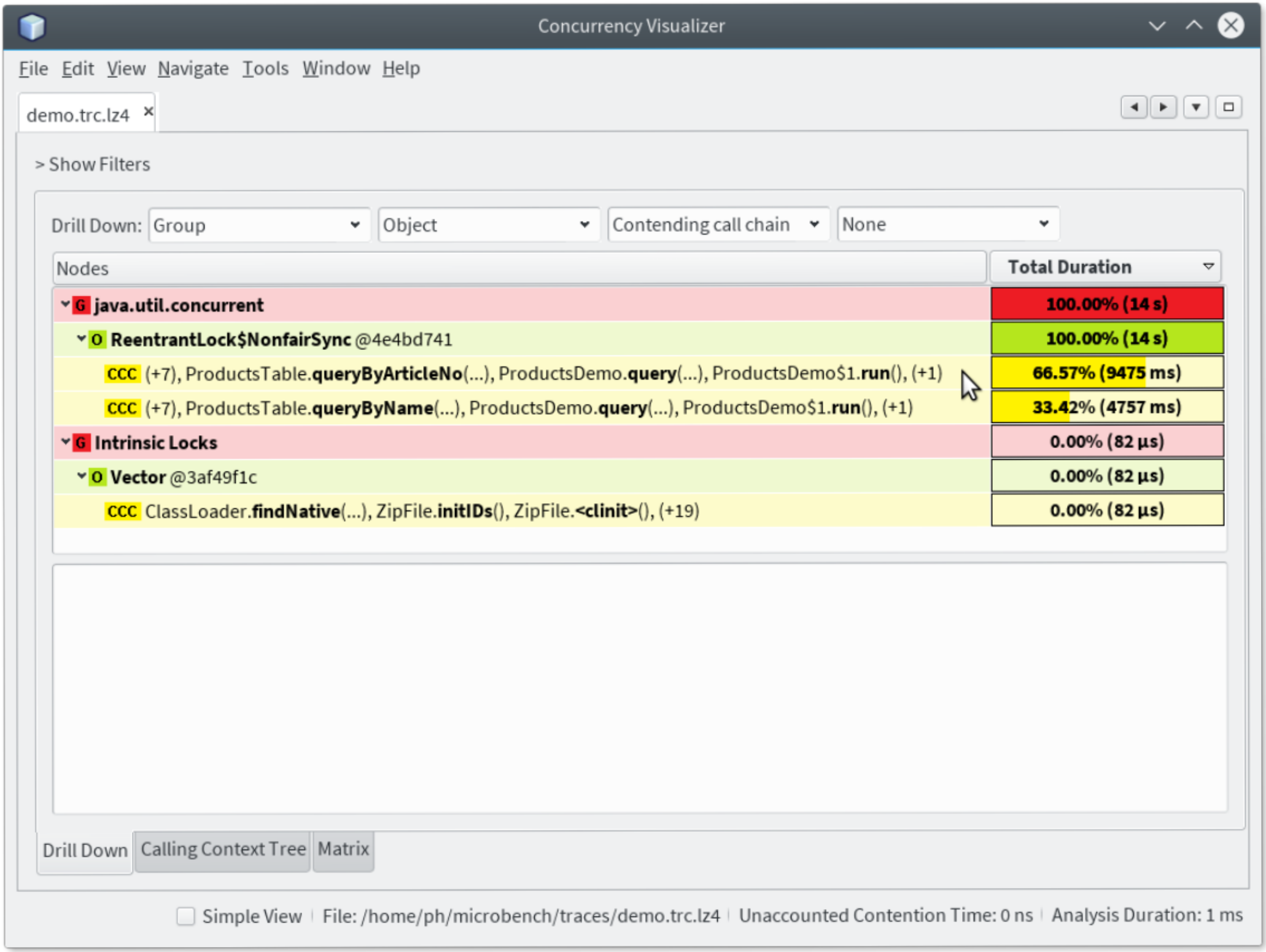


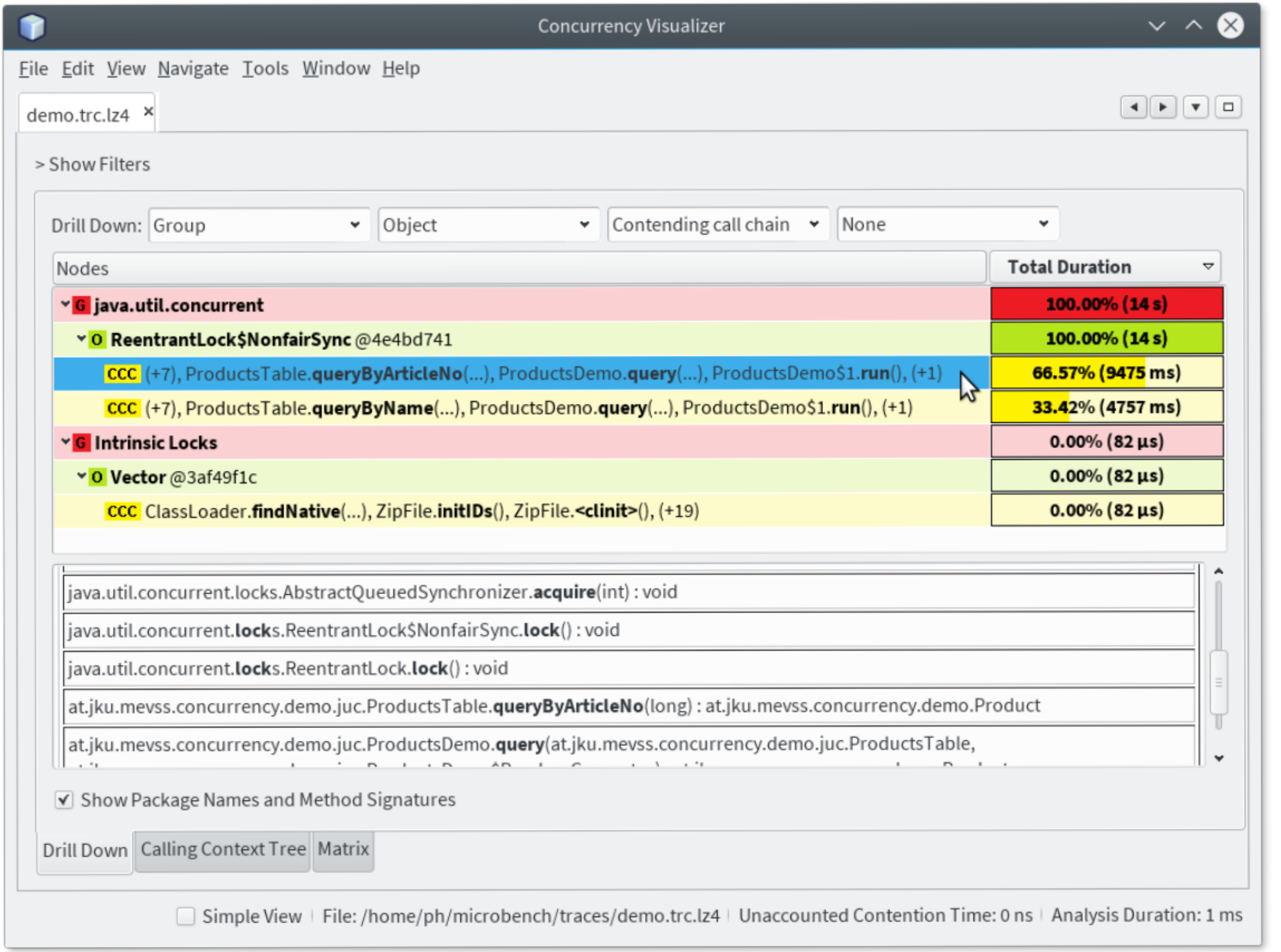

Concurrency Visualizer
File Edit View Navigate Tools Window Help
demo.trc.lz4
> Show Filters
Drill Down: Group None
Nodes
Total Duration
java.util.concurrent
100.00% (14 s)
Intrinsic Locks
0.00% (82 µs)
Drill Down
Calling Context Tree
Matrix
Simple View
File: /home/ph/microbench/traces/demo.trc.lz4
Unaccounted Contention Time: 0 ns
Analysis Duration: 2 ms

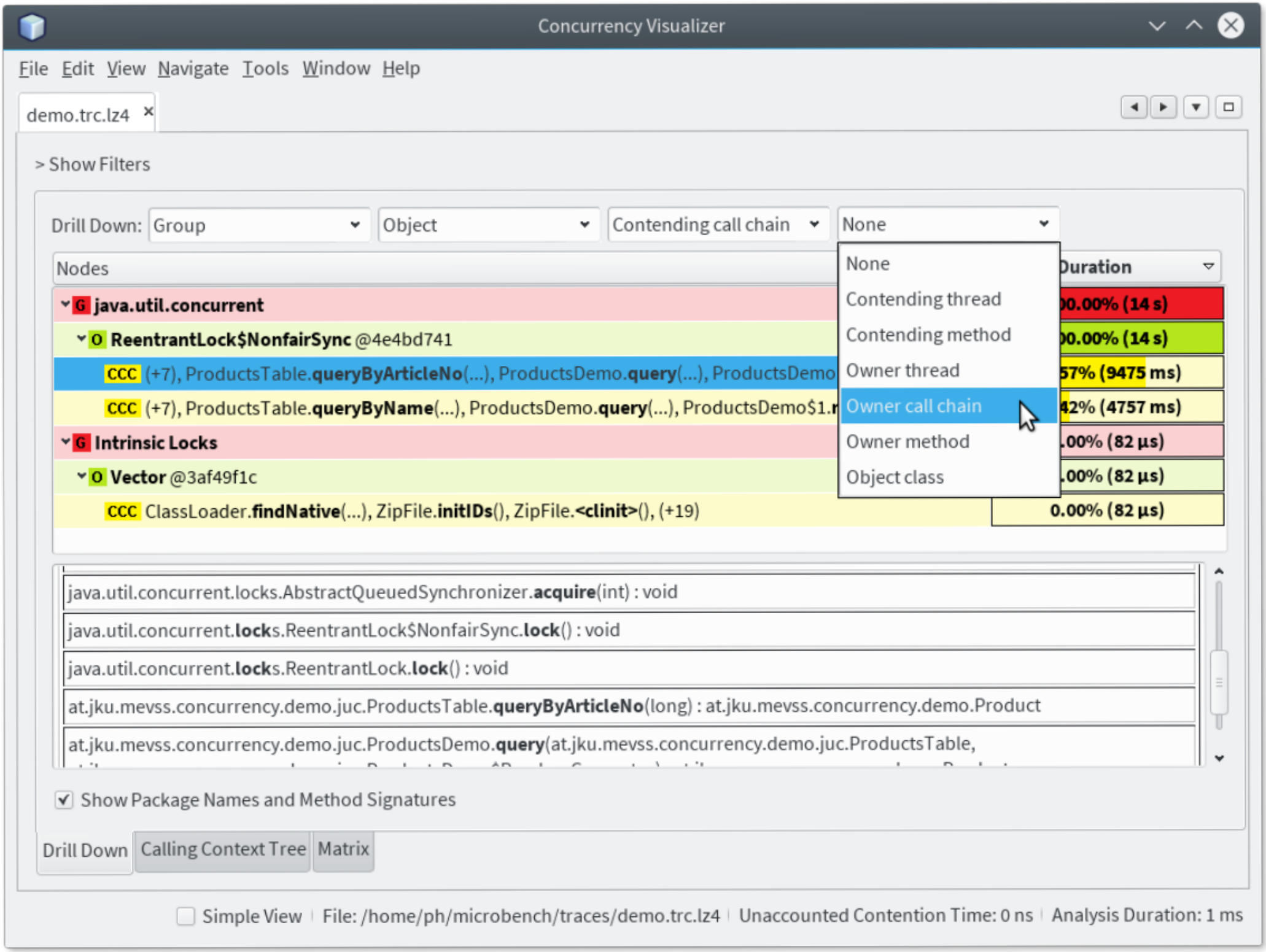

Concurrency Visualizer
File Edit View Navigate Tools Window Help
demo.trc.lz4
> Show Filters
Drill Down: Group
None
Nodes
Total Duration
java.util.concurrent
100.00% (14 s)
Intrinsic Locks
0.00% (82 µs)
None
Contending thread
Contending call chain
Contending method
Owner thread
Owner call chain
Owner method
Object
Drill Down
Calling Context Tree
Matrix
Simple View
File: /home/ph/microbench/traces/demo.trc.lz4
Unaccounted Contention Time: 0 ns
Analysis Duration: 2 ms

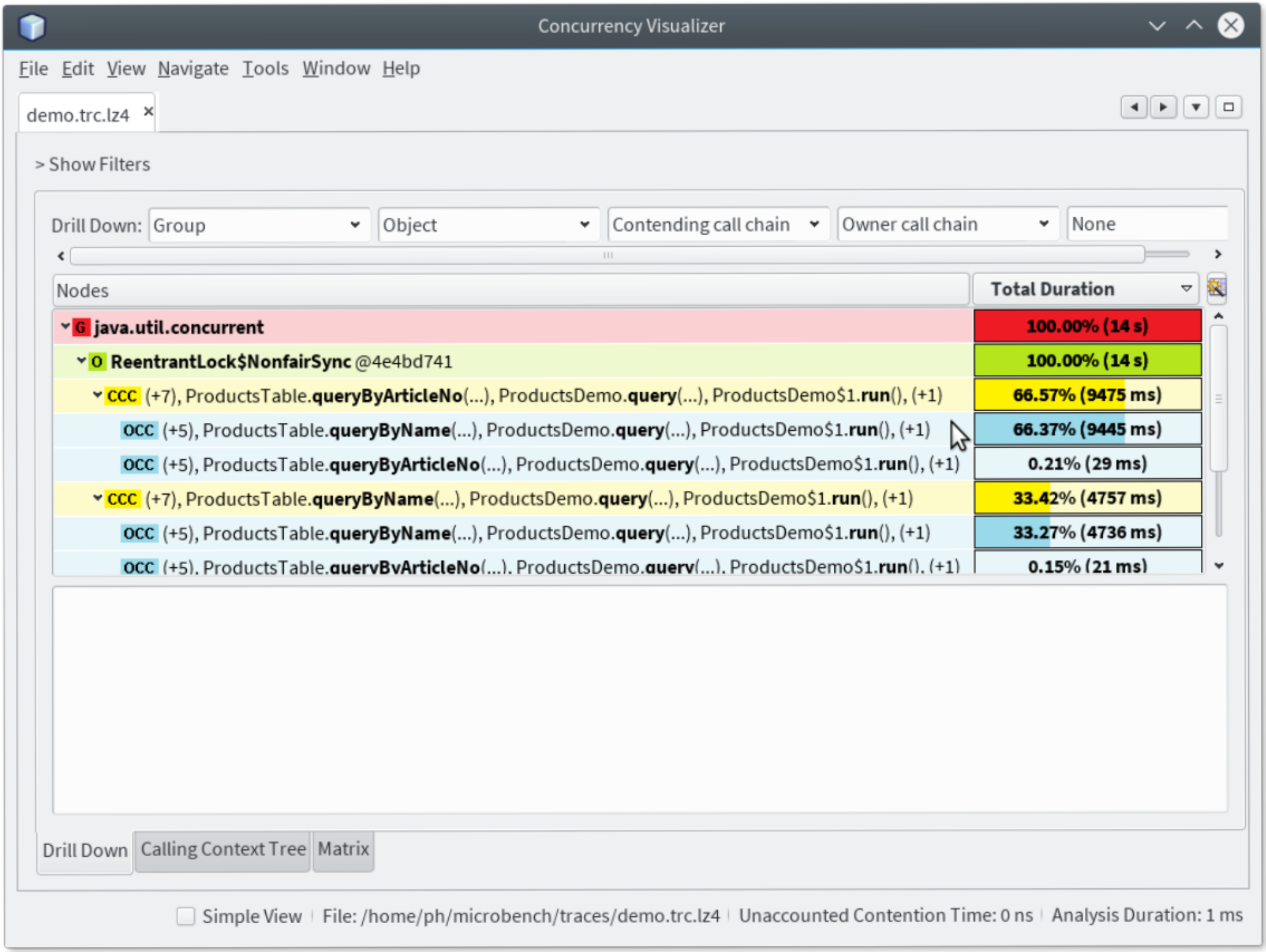

Concurrency Visualizer
File Edit View Navigate Tools Window Help
demo.trc.lz4
> Show Filters
Drill Down: Group Object None
Nodes
Total Duration
G java.util.concurrent
100.00% (14 s)
O ReentrantLock$NonfairSync @4e4bd741
100.00% (14 s)
G Intrinsic Locks
0.00% (82 µs)
O Vector @3af49f1c
0.00% (82 µs)
Drill Down
Calling Context Tree
Matrix
Simple View
File: /home/ph/microbench/traces/demo.trc.lz4
Unaccounted Contention Time: 0 ns
Analysis Duration: 2 ms

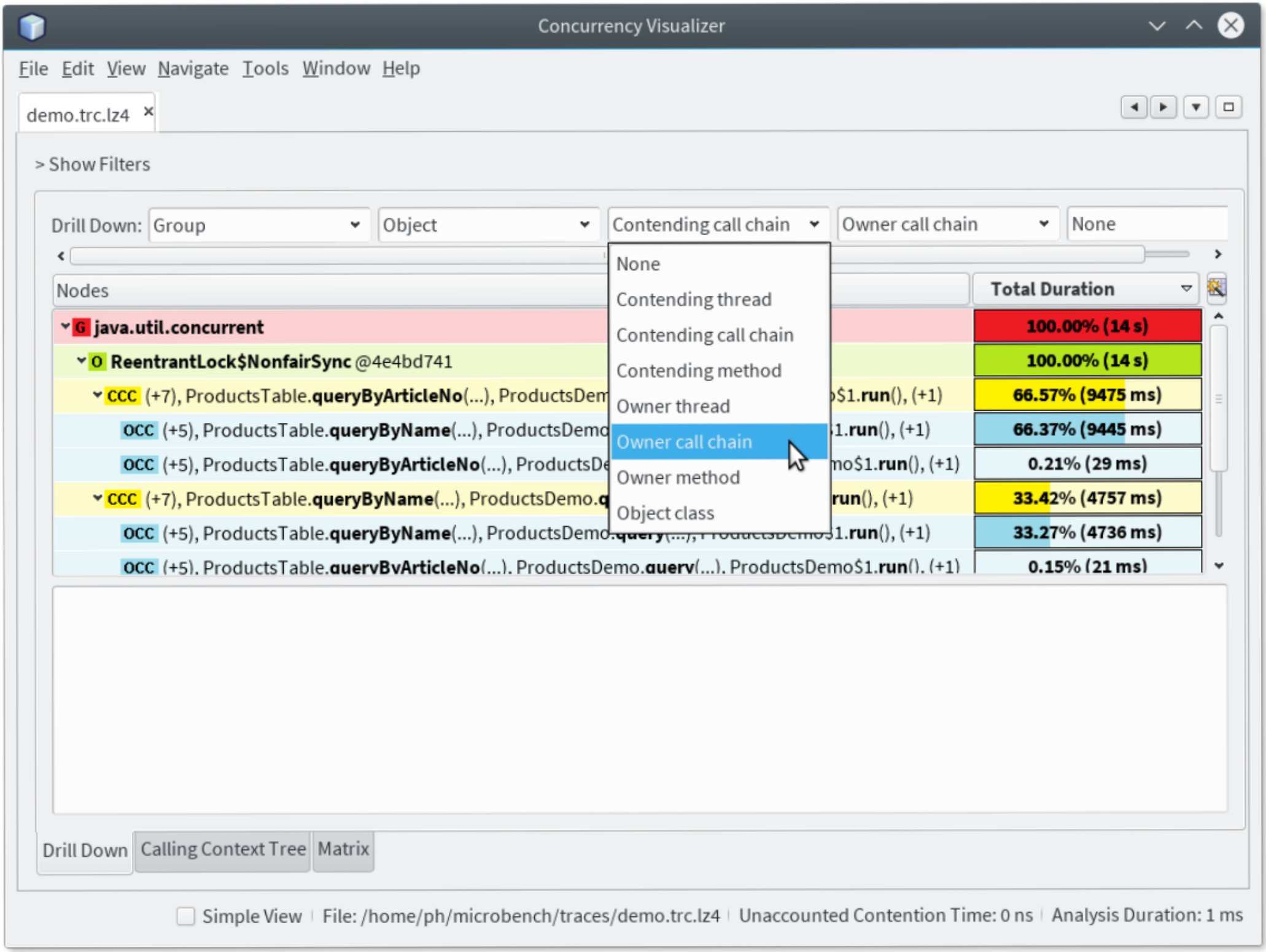

Concurrency Visualizer
File Edit View Navigate Tools Window Help
demo.trc.lz4
> Show Filters
Drill Down: Group Object None
None
Contending thread
Contending call chain
Contending method
Owner thread
Owner call chain
Owner method
Object class
Nodes
Total Duration
java.util.concurrent
100.00% (14 s)
ReentrantLock$NonfairSync @4e4bd741
100.00% (14 s)
Intrinsic Locks
0.00% (82 µs)
Vector @3af49f1c
0.00% (82 µs)
Drill Down
Calling Context Tree
Matrix
Simple View
File: /home/ph/microbench/traces/demo.trc.lz4
Unaccounted Contention Time: 0 ns
Analysis Duration: 2 ms

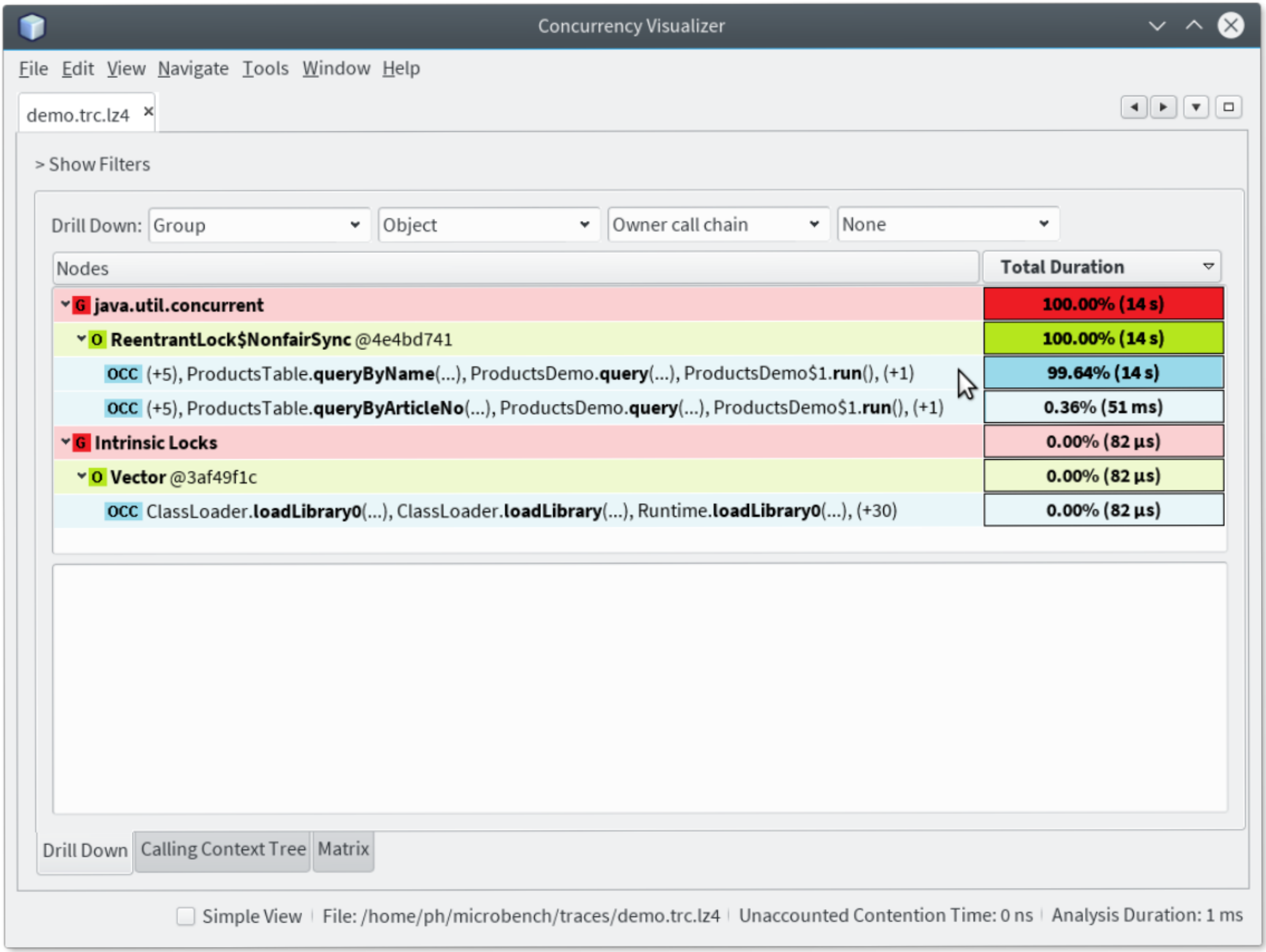

Concurrency Visualizer
File Edit View Navigate Tools Window Help
demo.trc.lz4
> Show Filters
Drill Down: Group Object Contending call chain None
Nodes
Total Duration
G java.util.concurrent
100.00% (14 s)
O ReentrantLock$NonfairSync @4e4bd741
100.00% (14 s)
CCC (+7), ProductsTable.queryByArticleNo(...), ProductsDemo.query(...), ProductsDemo$1.run(), (+1)
66.57% (9475 ms)
CCC (+7), ProductsTable.queryByName(...), ProductsDemo.query(...), ProductsDemo$1.run(), (+1)
33.42% (4757 ms)
G Intrinsic Locks
0.00% (82 µs)
O Vector @3af49f1c
0.00% (82 µs)
CCC ClassLoader.findNative(...), ZipFile.initIDs(), ZipFile.<clinit>(), (+19)
0.00% (82 µs)
Drill Down
Calling Context Tree
Matrix
Simple View
File: /home/ph/microbench/traces/demo.trc.lz4
Unaccounted Contention Time: 0 ns
Analysis Duration: 1 ms

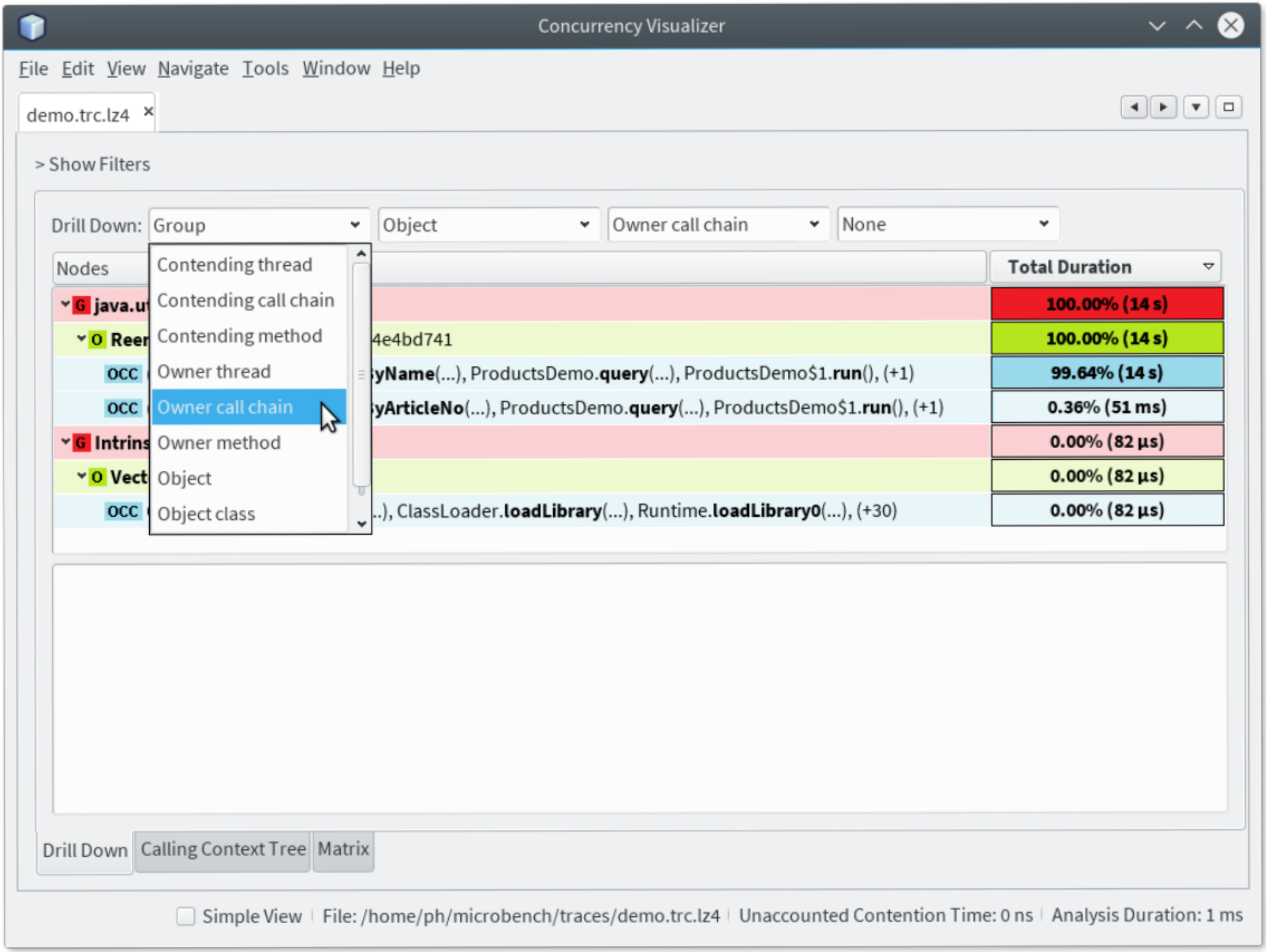

Concurrency Visualizer
File Edit View Navigate Tools Window Help
demo.trc.lz4
> Show Filters
Drill Down: Group Object Contending call chain None
Nodes
Total Duration
G java.util.concurrent
100.00% (14 s)
O ReentrantLock$NonfairSync @4e4bd741
100.00% (14 s)
CCC (+7), ProductsTable.queryByArticleNo(...), ProductsDemo.query(...), ProductsDemo$1.run(), (+1)
66.57% (9475 ms)
CCC (+7), ProductsTable.queryByName(...), ProductsDemo.query(...), ProductsDemo$1.run(), (+1)
33.42% (4757 ms)
G Intrinsic Locks
0.00% (82 µs)
O Vector @3af49f1c
0.00% (82 µs)
CCC ClassLoader.findNative(...), ZipFile.initIDs(), ZipFile.<clinit>(), (+19)
0.00% (82 µs)
java.util.concurrent.locks.AbstractQueuedSynchronizer.acquire(int) : void
java.util.concurrent.locks.ReentrantLock$NonfairSync.lock() : void
java.util.concurrent.locks.ReentrantLock.lock() : void
at.jku.mevss.concurrency.demo.juc.ProductsTable.queryByArticleNo(long) : at.jku.mevss.concurrency.demo.Product
at.jku.mevss.concurrency.demo.juc.ProductsDemo.query(at.jku.mevss.concurrency.demo.juc.ProductsTable,
Show Package Names and Method Signatures
Drill Down
Calling Context Tree
Matrix
Simple View
File: /home/ph/microbench/traces/demo.trc.lz4
Unaccounted Contention Time: 0 ns
Analysis Duration: 1 ms

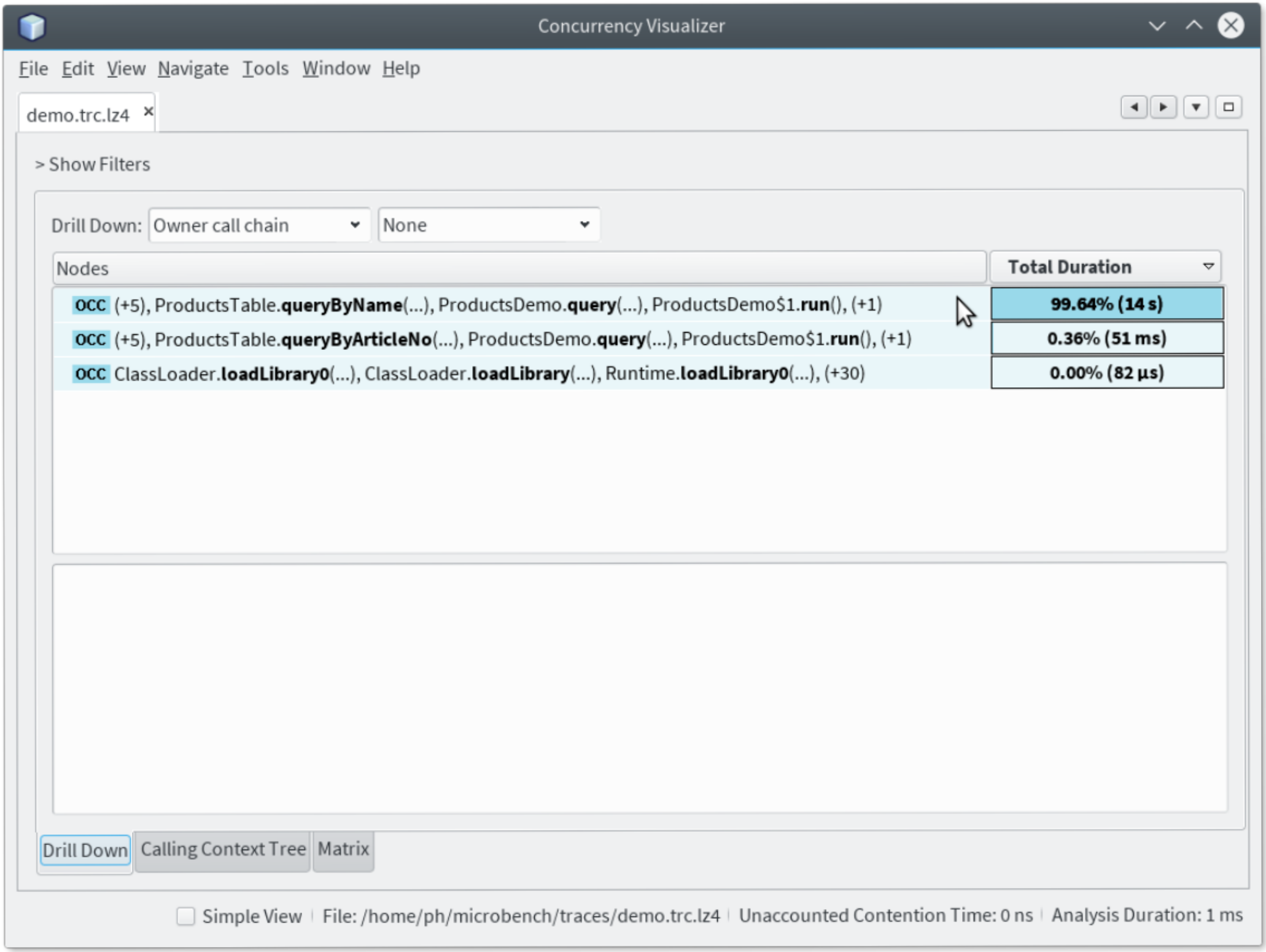

Concurrency Visualizer
File Edit View Navigate Tools Window Help
demo.trc.lz4
> Show Filters
Drill Down: Group Object Contending call chain None
None
Contending thread
Contending method
Owner thread
Owner call chain
Owner method
Object class
Nodes
Duration
G java.util.concurrent
O ReentrantLock$NonfairSync @4e4bd741
CCC (+7), ProductsTable.queryByArticleNo(...), ProductsDemo.query(...), ProductsDemo
CCC (+7), ProductsTable.queryByName(...), ProductsDemo.query(...), ProductsDemo$1.r
G Intrinsic Locks
O Vector @3af49f1c
CCC ClassLoader.findNative(...), ZipFile.initIDs(), ZipFile.<clinit>(), (+19)
0.00% (82 µs)
java.util.concurrent.locks.AbstractQueuedSynchronizer.acquire(int) : void
java.util.concurrent.locks.ReentrantLock$NonfairSync.lock() : void
java.util.concurrent.locks.ReentrantLock.lock() : void
at.jku.mevss.concurrency.demo.juc.ProductsTable.queryByArticleNo(long) : at.jku.mevss.concurrency.demo.Product
at.jku.mevss.concurrency.demo.juc.ProductsDemo.query(at.jku.mevss.concurrency.demo.juc.ProductsTable,
Show Package Names and Method Signatures
Drill Down
Calling Context Tree
Matrix
Simple View
File: /home/ph/microbench/traces/demo.trc.lz4
Unaccounted Contention Time: 0 ns
Analysis Duration: 1 ms

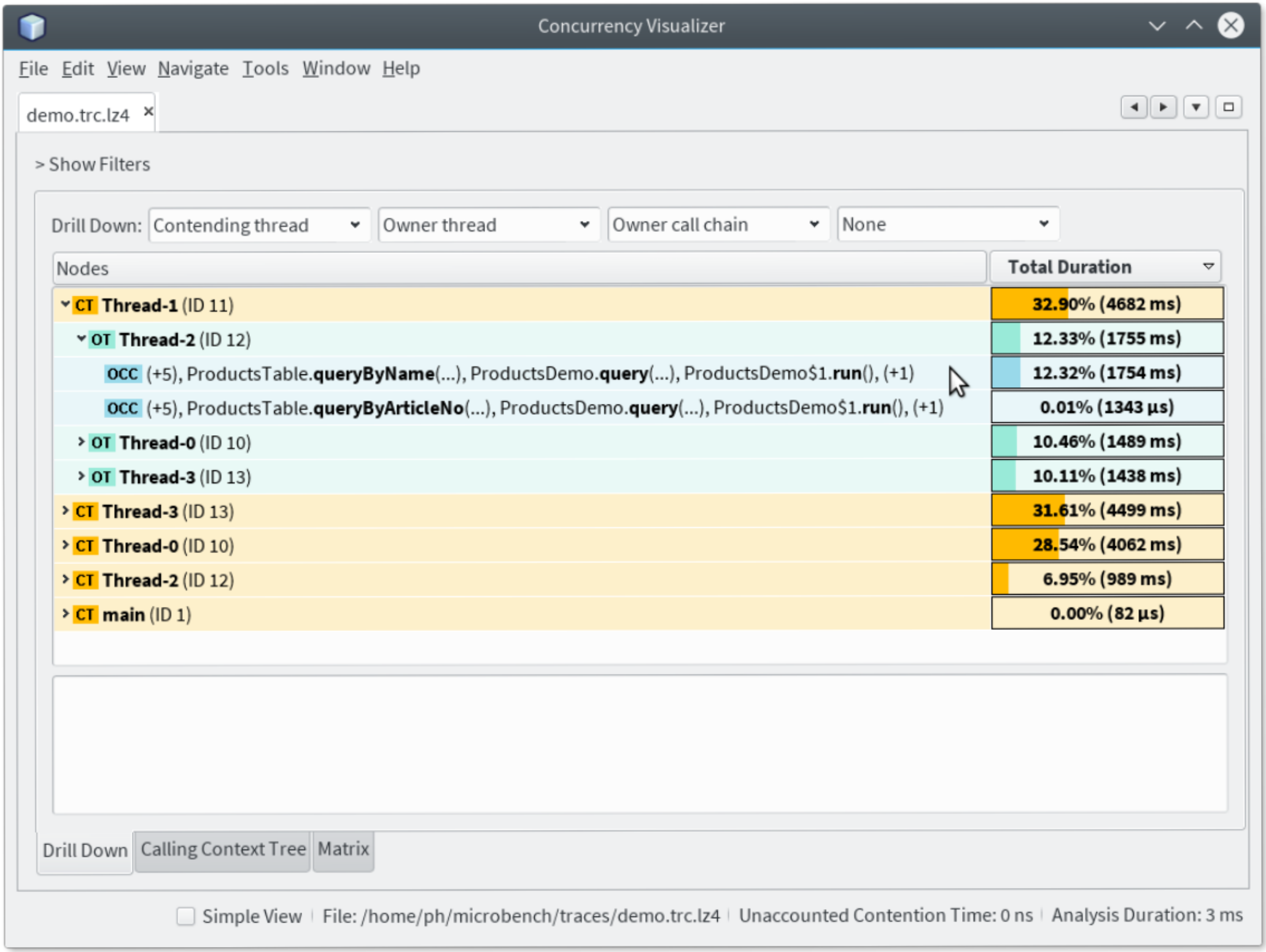

Concurrency Visualizer
File Edit View Navigate Tools Window Help
demo.trc.lz4
> Show Filters
Drill Down: Group Object Contending call chain Owner call chain None
Nodes
Total Duration
G java.util.concurrent
100.00% (14 s)
O ReentrantLock$NonfairSync @4e4bd741
100.00% (14 s)
CCC (+7), ProductsTable.queryByArticleNo(...), ProductsDemo.query(...), ProductsDemo$1.run(), (+1)
66.57% (9475 ms)
OCC (+5), ProductsTable.queryByName(...), ProductsDemo.query(...), ProductsDemo$1.run(), (+1)
66.37% (9445 ms)
OCC (+5), ProductsTable.queryByArticleNo(...), ProductsDemo.query(...), ProductsDemo$1.run(), (+1)
0.21% (29 ms)
CCC (+7), ProductsTable.queryByName(...), ProductsDemo.query(...), ProductsDemo$1.run(), (+1)
33.42% (4757 ms)
OCC (+5), ProductsTable.queryByName(...), ProductsDemo.query(...), ProductsDemo$1.run(), (+1)
33.27% (4736 ms)
OCC (+5), ProductsTable.queryByArticleNo(...), ProductsDemo.query(...), ProductsDemo$1.run(), (+1)
0.15% (21 ms)
Drill Down
Calling Context Tree
Matrix
Simple View
File: /home/ph/microbench/traces/demo.trc.lz4
Unaccounted Contention Time: 0 ns
Analysis Duration: 1 ms


Concurrency Visualizer
File Edit View Navigate Tools Window Help
demo.trc.lz4
> Show Filters
Drill Down: Group Object Contending call chain Owner call chain None
None
Contending thread
Contending call chain
Contending method
Owner thread
Owner call chain
Owner method
Object class
Nodes
Total Duration
G java.util.concurrent
100.00% (14 s)
O ReentrantLock$NonfairSync @4e4bd741
100.00% (14 s)
CCC (+7), ProductsTable.queryByArticleNo(...), ProductsDem
66.57% (9475 ms)
OCC (+5), ProductsTable.queryByName(...), ProductsDemo
66.37% (9445 ms)
OCC (+5), ProductsTable.queryByArticleNo(...), ProductsDe
0.21% (29 ms)
CCC (+7), ProductsTable.queryByName(...), ProductsDemo.q
33.42% (4757 ms)
OCC (+5), ProductsTable.queryByName(...), ProductsDemo
33.27% (4736 ms)
OCC (+5), ProductsTable.queryByArticleNo(...), ProductsDemo.query(...), ProductsDemo$1.run(), (+1)
0.15% (21 ms)
Drill Down
Calling Context Tree
Matrix
Simple View
File: /home/ph/microbench/traces/demo.trc.lz4
Unaccounted Contention Time: 0 ns
Analysis Duration: 1 ms


Concurrency Visualizer
File Edit View Navigate Tools Window Help
demo.trc.lz4
> Show Filters
Drill Down: Group Object Owner call chain None
Nodes
Total Duration
G java.util.concurrent
100.00% (14 s)
O ReentrantLock$NonfairSync @4e4bd741
100.00% (14 s)
OCC (+5), ProductsTable.queryByName(...), ProductsDemo.query(...), ProductsDemo$1.run(), (+1)
99.64% (14 s)
OCC (+5), ProductsTable.queryByArticleNo(...), ProductsDemo.query(...), ProductsDemo$1.run(), (+1)
0.36% (51 ms)
G Intrinsic Locks
0.00% (82 µs)
O Vector @3af49f1c
0.00% (82 µs)
OCC ClassLoader.loadLibrary0(...), ClassLoader.loadLibrary(...), Runtime.loadLibrary0(...), (+30)
0.00% (82 µs)
Drill Down
Calling Context Tree
Matrix
Simple View
File: /home/ph/microbench/traces/demo.trc.lz4
Unaccounted Contention Time: 0 ns
Analysis Duration: 1 ms


Concurrency Visualizer
File Edit View Navigate Tools Window Help
demo.trc.lz4
> Show Filters
Drill Down: Group
Object
Owner call chain
None
Contending thread
Contending call chain
Contending method
Owner thread
Owner call chain
Owner method
Object
Object class
Nodes
Total Duration
100.00% (14 s)
100.00% (14 s)
99.64% (14 s)
0.36% (51 ms)
0.00% (82 µs)
0.00% (82 µs)
0.00% (82 µs)
Drill Down
Calling Context Tree
Matrix
Simple View
File: /home/ph/microbench/traces/demo.trc.lz4
Unaccounted Contention Time: 0 ns
Analysis Duration: 1 ms


Concurrency Visualizer
File Edit View Navigate Tools Window Help
demo.trc.lz4
> Show Filters
Drill Down: Owner call chain None
Nodes
Total Duration
OCC (+5), ProductsTable.queryByName(...), ProductsDemo.query(...), ProductsDemo$1.run(), (+1)
99.64% (14 s)
OCC (+5), ProductsTable.queryByArticleNo(...), ProductsDemo.query(...), ProductsDemo$1.run(), (+1)
0.36% (51 ms)
OCC ClassLoader.loadLibrary0(...), ClassLoader.loadLibrary(...), Runtime.loadLibrary0(...), (+30)
0.00% (82 µs)
Drill Down
Calling Context Tree
Matrix
Simple View
File: /home/ph/microbench/traces/demo.trc.lz4
Unaccounted Contention Time: 0 ns
Analysis Duration: 1 ms


Concurrency Visualizer
File Edit View Navigate Tools Window Help
demo.trc.lz4
> Show Filters
Drill Down: Contending thread Owner thread Owner call chain None
Nodes
Total Duration
CT Thread-1 (ID 11) 32.90% (4682 ms)
OT Thread-2 (ID 12) 12.33% (1755 ms)
OCC (+5), ProductsTable.queryByName(...), ProductsDemo.query(...), ProductsDemo$1.run(), (+1) 12.32% (1754 ms)
OCC (+5), ProductsTable.queryByArticleNo(...), ProductsDemo.query(...), ProductsDemo$1.run(), (+1) 0.01% (1343 µs)
OT Thread-0 (ID 10) 10.46% (1489 ms)
OT Thread-3 (ID 13) 10.11% (1438 ms)
CT Thread-3 (ID 13) 31.61% (4499 ms)
CT Thread-0 (ID 10) 28.54% (4062 ms)
CT Thread-2 (ID 12) 6.95% (989 ms)
CT main (ID 1) 0.00% (82 µs)
Drill Down
Calling Context Tree
Matrix
Simple View
File: /home/ph/microbench/traces/demo.trc.lz4
Unaccounted Contention Time: 0 ns
Analysis Duration: 3 ms

**Read** our paper

for more details on tracing and analysis,
especially java.util.concurrent

**Download** our JDK and visualization tool:

**http://mevss.jku.at/** → Tools → Lock Contention Tracing

**Ask** questions!