

Thread Dump - Intelligence Report


📁 File: *threaddump-1650435643192.tdump*

🕒 Timestamp: 2022-04-20 11:50:43

Thread Count Summary

📖 To learn about different thread states through real-life example, check out this [video tutorial](#)


Total Threads count: 37



29

RUNNABLE


[View Details](#)



6

TIMED_WAITING

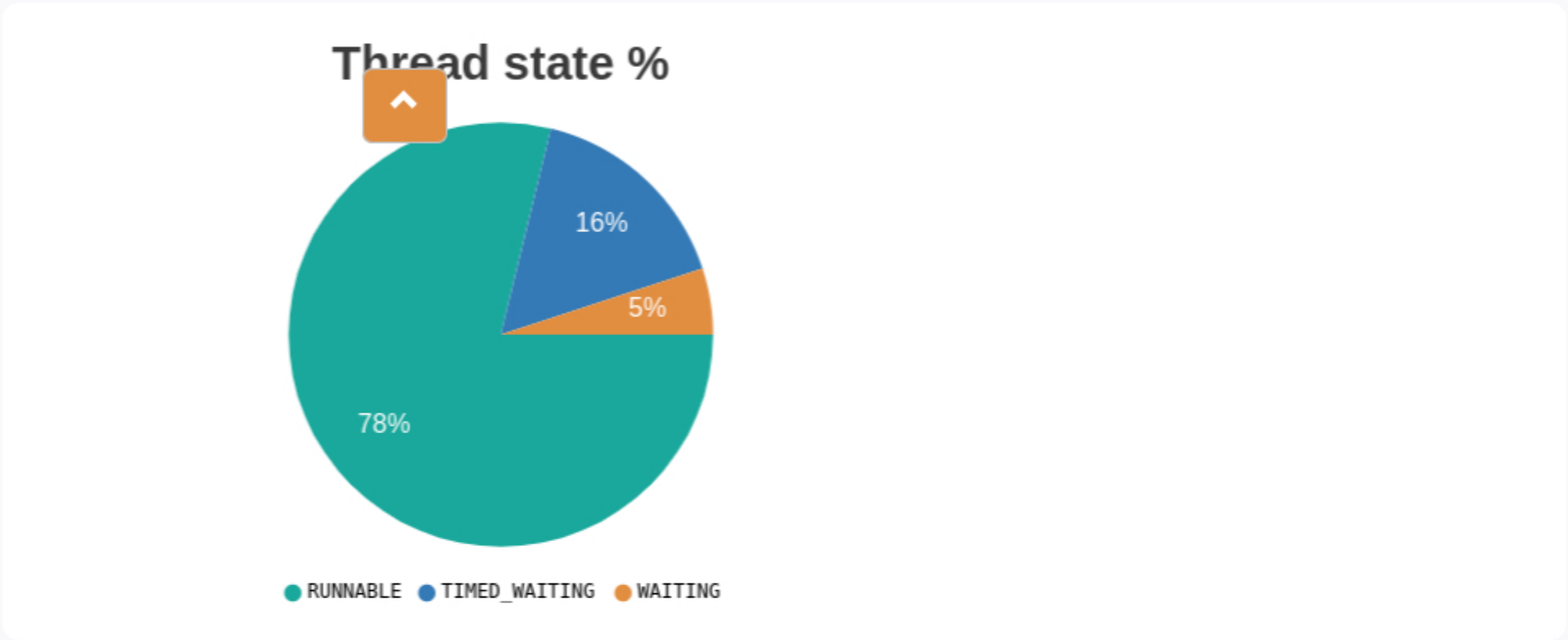
[View Details](#)



2

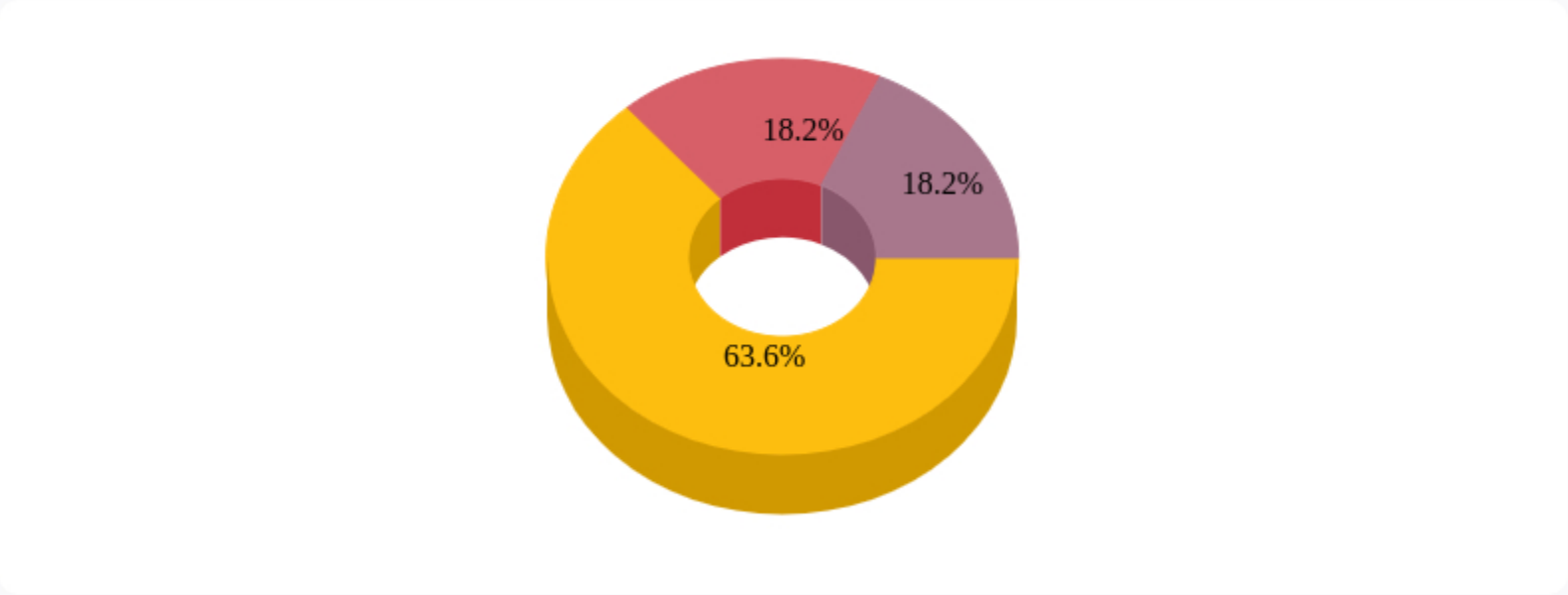
WAITING

[View Details](#)



Thread Group


Threads with similar names are grouped in this section



	Thread Group	Count	States
	GC Thread	7 threads	RUNNABLE:7
	G1 Conc	2 threads	RUNNABLE:2
	JMX server connection timeout	2 threads	TIMED_WAITING:2


Daemon vs non-Daemon

Learn more about [daemon and non-daemon \(i.e. user threads\)](#)



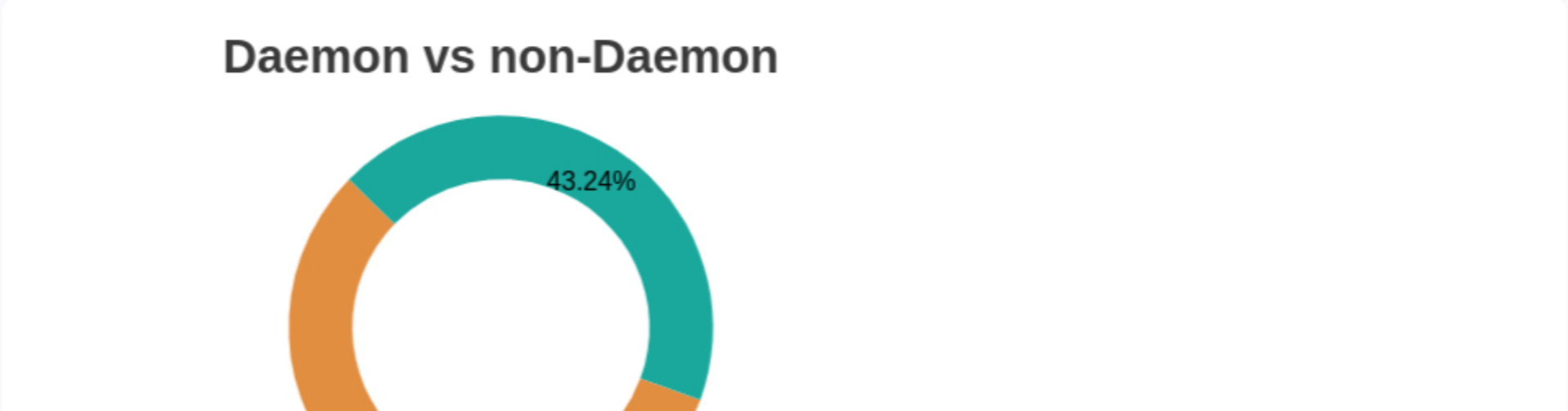
21

DAEMON



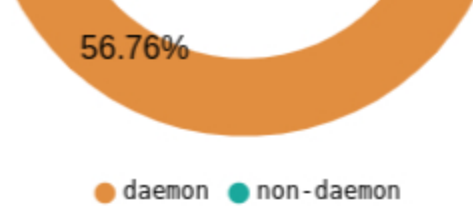
16

NON-DAEMON



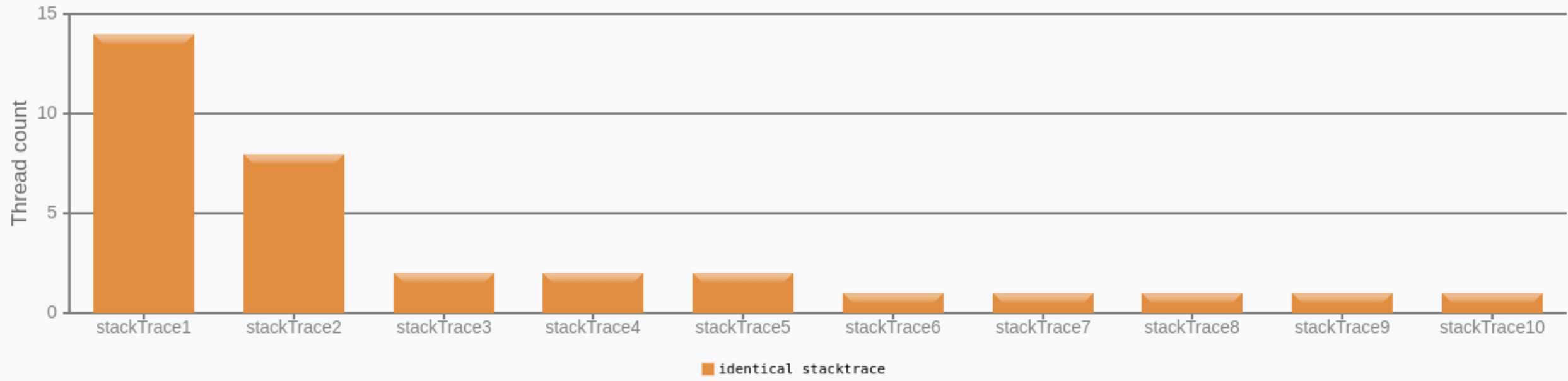
View Details

View Details



Threads with identical stack trace

Threads with identical stack traces are grouped here. If lot of threads start to exhibit identical stack trace it might be a concern, learn [RSI Pattern](#)



Thread Count	Identical Stack trace
14 RUNNABLE threads	stacktrace See complete stacktrace .
8 RUNNABLE threads	java.lang.Thread.State: RUNNABLE Locked ownable synchronizers: - None See complete stacktrace .
2 RUNNABLE threads	java.lang.Thread.State: RUNNABLE No compile task Locked ownable synchronizers: - None See complete stacktrace .
2 TIMED_WAITING threads	java.lang.Thread.State: TIMED_WAITING (on object monitor) at java.lang.Object.wait(java.base@17.0.2/Native Method) - waiting on at com.sun.jmx.remote.internal.ServerCommunicatorAdmin\$Timeout.run(java.management@17.0.2/ServerCommunicatorAdmin.java:171) - locked <0x000000070a4ff680> (a [I]) ... See complete stacktrace .
2 RUNNABLE threads	java.lang.Thread.State: RUNNABLE at sun.nio.ch.Net.poll(java.base@17.0.2/Native Method) at sun.nio.ch.NioSocketImpl.park(java.base@17.0.2/NioSocketImpl.java:181) at sun.nio.ch.NioSocketImpl.timedRead(java.base@17.0.2/NioSocketImpl.java:285) at sun.nio.ch.NioSocketImpl.implRead(java.base@17.0.2/NioSocketImpl.java:309) ... See complete stacktrace .
1 RUNNABLE threads	java.lang.Thread.State: RUNNABLE at java.io.FileOutputStream.writeBytes(java.base@17.0.2/Native Method) at java.io.FileOutputStream.write(java.base@17.0.2/FileOutputStream.java:349) at java.io.BufferedOutputStream.flushBuffer(java.base@17.0.2/BufferedOutputStream.java:81) at java.io.BufferedOutputStream.flush(java.base@17.0.2/BufferedOutputStream.java:142) ... See complete stacktrace .
1 TIMED_WAITING threads	java.lang.Thread.State: TIMED_WAITING (on object monitor) at java.lang.Object.wait(java.base@17.0.2/Native Method) - waiting on at com.sun.jmx.remote.internal.ArrayNotificationBuffer.fetchNotifications(java.management@17.0.2/ArrayNotificationBuffer.java:449) - locked <0x0000000709ccab78> (a com.sun.jmx.remote.internal.ArrayNotificationBuffer) ... See complete stacktrace .
1 TIMED_WAITING threads	java.lang.Thread.State: TIMED_WAITING (on object monitor) at java.lang.Object.wait(java.base@17.0.2/Native Method) - waiting on at java.lang.ref.ReferenceQueue.remove(java.base@17.0.2/ReferenceQueue.java:155) - locked <0x0000000709c02108> (a java.lang.ref.ReferenceQueue\$Lock) ... See complete stacktrace .
1 RUNNABLE threads	java.lang.Thread.State: RUNNABLE at sun.nio.ch.Net.accept(java.base@17.0.2/Native Method) at sun.nio.ch.NioSocketImpl.accept(java.base@17.0.2/NioSocketImpl.java:755) at java.net.ServerSocket.implAccept(java.base@17.0.2/ServerSocket.java:675) at java.net.ServerSocket.platformImplAccept(java.base@17.0.2/ServerSocket.java:641) ... See complete stacktrace .

1 RUNNABLE threads	java.lang.Thread.State: RUNNABLE at java.lang.ref.Reference.waitForReferencePendingList(java.base@17.0.2/Native Method) at java.lang.ref.Reference.processPendingReferences(java.base@17.0.2/Reference.java:253) at java.lang.ref.Reference\$ReferenceHandler.run(java.base@17.0.2/Reference.java:215) Locked ownable synchronizers: ... See complete stacktrace .
--------------------	--

Most used methods

Frequently executed methods are reported. If lot of threads executes same method, it may be a concern. Learn [All roads lead to Rome pattern](#)

Thread Count	Method	Percentage
6 threads	java.lang.Object.wait(java.base@17.0.2/Native Method). To see stack trace click here .	16% <div><div></div></div>
2 threads	sun.nio.ch.Net.poll(java.base@17.0.2/Native Method). To see stack trace click here .	5% <div><div></div></div>
1 threads	java.io.FileOutputStream.writeBytes(java.base@17.0.2/Native Method). To see stack trace click here .	3% <div><div></div></div>
1 threads	java.lang.ref.Reference.waitForReferencePendingList(java.base@17.0.2/Native Method). To see stack trace click here .	3% <div><div></div></div>
1 threads	java.util.TreeMap.put(java.base@17.0.2/TreeMap.java:826). To see stack trace click here .	3% <div><div></div></div>

[Show all methods >>](#)

CPU consuming threads

If application is consuming high CPU, investigate below threads. Learn [Athlete pattern](#)

Thread	CPU consuming thread's stacktrace
C2 CompilerThread0 nativeld: 393423	java.lang.Thread.State: RUNNABLE No compile task Locked ownable synchronizers: - None See complete stacktrace .
C1 CompilerThread0 nativeld: 393424	java.lang.Thread.State: RUNNABLE No compile task Locked ownable synchronizers: - None See complete stacktrace .
RMI TCP Accept-0 nativeld: 393730	java.lang.Thread.State: RUNNABLE at sun.nio.ch.Net.accept(java.base@17.0.2/Native Method) at sun.nio.ch.NioSocketImpl.accept(java.base@17.0.2/NioSocketImpl.java:755) at java.net.ServerSocket.implAccept(java.base@17.0.2/ServerSocket.java:675) at java.net.ServerSocket.platformImplAccept(java.base@17.0.2/ServerSocket.java:641) ... See complete stacktrace .
RMI TCP Connection(10)-10.20.40.178 nativeld: 466447	java.lang.Thread.State: RUNNABLE at java.util.TreeMap.put(java.base@17.0.2/TreeMap.java:826) at java.util.TreeMap.put(java.base@17.0.2/TreeMap.java:534) at javax.management.openmbean.CompositeDataSupport.makeMap(javax.management@17.0.2/CompositeDataSupport.java:143) at javax.management.openmbean.CompositeDataSupport.(javax.management@17.0.2/CompositeDataSupport.java:119) ... See complete stacktrace .
21 JVM threads	See complete stacktrace .

Blocking Threads - Transitive Graph

Threads that block other threads are displayed here. Blocking threads makes application unresponsive, learn [Traffic Jam pattern](#)

☒ No transitive blocks found

GC Threads

Garbage collection threads count reported. Learn [Scavengers pattern](#)

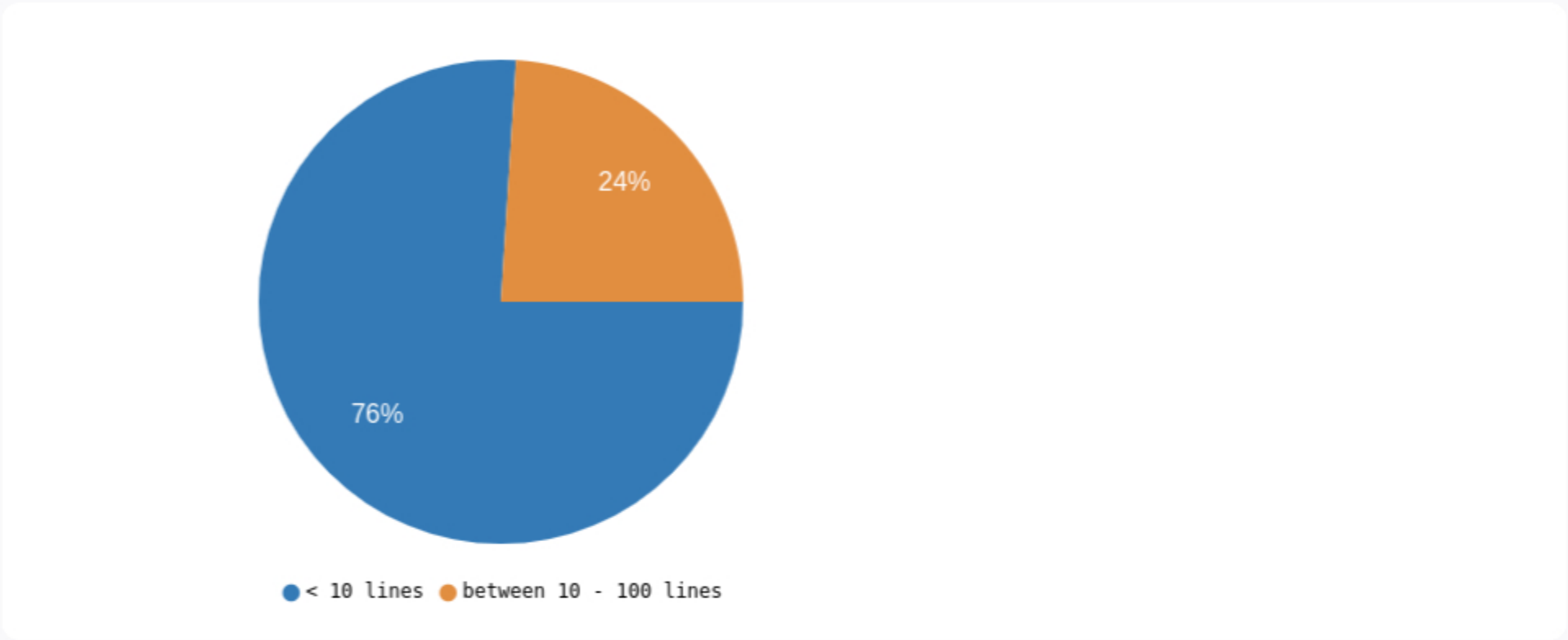
Not reported in the thread dump

Threads Stack Length

Lengthy stacks can cause StackOverflowError. [Learn more](#)

☒ No Problem in Stack trace length.

Stack Length	Thread count
< 10 lines	28
between 10 - 100 lines	9



Complex DeadLocks

Learn more about [Complex Deadlock](#)

☒ No Complex Deadlocks found

Dead Lock

Learn more about [Deadlock](#)

☒ No Deadlock found

Finalizer Thread

If finalizer thread is BLOCKED or WAITING for a prolonged period, it can result in OutOfMemoryError, to learn more visit [Leprechaun Trap pattern](#)

☒ No problem with Finalizer Thread.

Exception

Threads throwing commonly known Exceptions/Errors are reported here. [Learn more](#)

☒ No known exceptions are reported.

Bottom up Call Stack Tree

↓ Show Top Down call stack

All threads stacktrace are combined in to one single tree. Learn [it's benefits](#).



