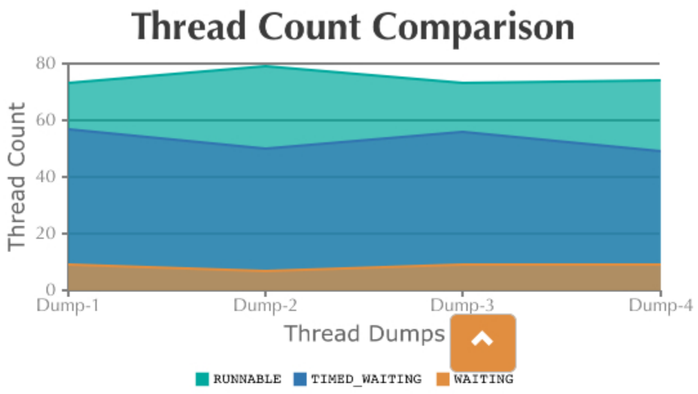
Comparative Summary

Thread count

To learn about different thread states through real-life example, check out this video tutorial

	Dump 1	Dump 2	Dump 3	Dump 4
RUNNABLE	73	79	73	74
TIMED_WAITING	57	50	56	49
WAITING	9	7	9	9
Total	**139**	**136**	**138**	**132**



Thread States

One view to see all threads states across all dumps. Learn more

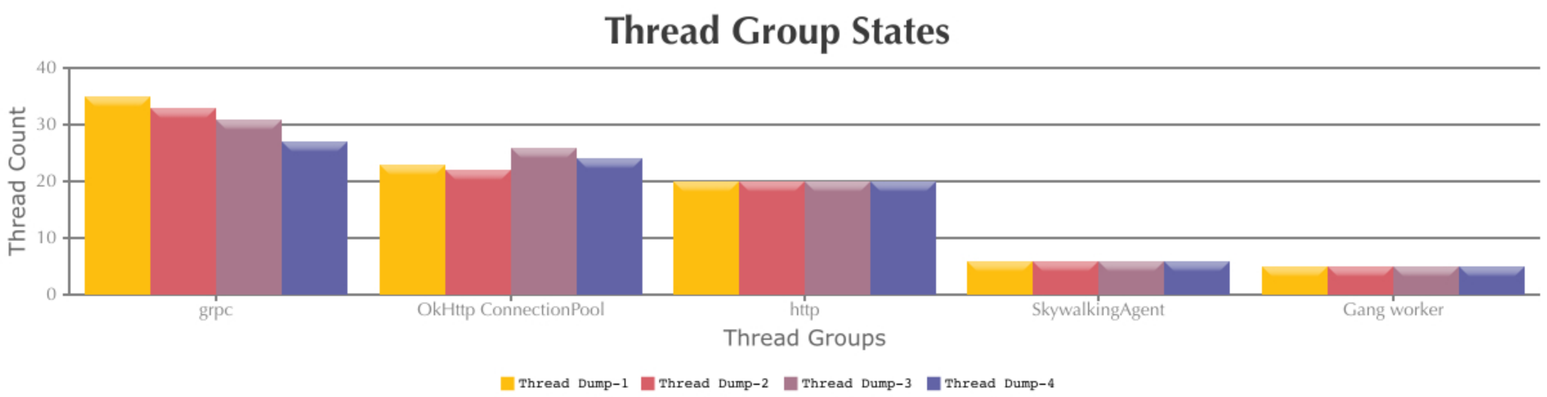
RUNNABLE WAITING TIMED_WAITING

Thread	Dump 1	Dump 2	Dump 3	Dump 4
jdbc:jtds:sqlserver://10....				
jTDS TimerThread				
Abandoned connection clea...				
grpc-nio-worker-ELG-1-31			-	-
OkHttp ConnectionPool				
OkHttp ConnectionPool				
grpc-nio-worker-ELG-1-21				
commons-pool-EvictionTime...				
DestroyJavaVM				
container-0				

See all threads >>

Thread Groups size

If there is lot of fluctuation in a particular thread group count, than thread group should be investigated further.



	Dump 1	Dump 2	Dump 3	Dump 4
grpc	35	33	31	27
OkHttp ConnectionPool	23	22	26	24
http	20	20	20	20
SkywalkingAgent	6	6	6	6
Gang worker	5	5	5	5

Waiting Threads

Waiting threads with same stacktrace between dumps are displayed here.

Code	WAITING Threads
wait() method in **java.lang.Object** file	2 threads are WAITING: Finalizer Reference Handler
park() method in **sun.misc.Unsafe** file	3 threads are WAITING: pool-1-thread-1 elk-thread-pool-1 spring.cloud.inetutils

Timed Waiting Threads

TIMED WAITING threads with same stacktrace between dumps are displayed here.

Code	TIMED_WAITING Threads
wait() method in **java.lang.Object** file	13 threads are TIMED_WAITING: Abandoned connection cleanup thread commons-pool-EvictionTimer OkHttp ConnectionPool OkHttp ConnectionPool OkHttp ConnectionPool OkHttp ConnectionPool OkHttp ConnectionPool Okio Watchdog OkHttp ConnectionPool OkHttp ConnectionPool OkHttp ConnectionPool OkHttp ConnectionPool OkHttp ConnectionPool

park() method in **sun.misc.Unsafe** file	12 threads are TIMED_WAITING: jdbc:jtds:sqlserver://10.1.249.109:databaseName=HHubGlobalDB_pools housekeeper SkywalkingAgent-5-JVMService-consume-0 SkywalkingAgent-4-JVMService-produce-0 SkywalkingAgent-3-GRPCChannelManager-0 SkywalkingAgent-6-ServiceAndEndpointRegisterClient-0 SkywalkingAgent-8-ProfileSendSnapshotService-0 SkywalkingAgent-7-ProfileGetTaskService-0 http-nio-80-exec-2 http-nio-80-exec-7 ForkJoinPool.commonPool-worker-57 ApplicationTokenRefreshService http-nio-80-exec-13
sleep() method in **java.lang.Thread** file	5 threads are TIMED_WAITING: DataCarrier.DEFAULT.Consumser.0.Thread container-0 ContainerBackgroundProcessor[StandardEngine[Tomcat]] kafka-agent http-nio-80-AsyncTimeout

Blocked Threads

BLOCKED threads with same stacktrace between dumps are displayed here. Blocked threads can make application unresponsive. Learn atherosclerosis pattern

No threads are BLOCKED on the same lock across all dumps.

Bottom up Call Stack Tree

Show Top Down call stack

All threads stacktrace are combined in to one single tree. Learn it's benefits.

- (417) root
  - (369) java.lang.Thread.run(Thread.java:748)
    - (155) java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    - (122) org.apache.skywalking.apm.dependencies.io.netty.util.concurrent.FastThreadLocalRunnable.run(FastThreadLocalRunnable.java:30)
    - (64) org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
    - (8) org.apache.kafka.clients.producer.internals.Sender.run(Sender.java:238)
    - (8) org.apache.tomcat.util.net.NioEndpoint$Poller.run(NioEndpoint.java:793)
    - (4) org.apache.coyote.AbstractProtocol$AsyncTimeout.run(AbstractProtocol.java:1133)
    - (4) org.apache.tomcat.util.net.NioEndpoint$Acceptor.run(NioEndpoint.java:455)
    - (4) org.apache.catalina.core.ContainerBase$ContainerBackgroundProcessor.run(ContainerBase.java:1357)
  - (6) ch.qos.logback.core.AsyncAppenderBase$Worker.run(AsyncAppenderBase.java:289)
    - (6) java.util.concurrent.ArrayBlockingQueue.take(ArrayBlockingQueue.java:403)
  - (4) java.util.concurrent.ForkJoinWorkerThread.run(ForkJoinWorkerThread.java:157)
    - (4) java.util.concurrent.ForkJoinPool.runWorker(ForkJoinPool.java:1693)
  - (4) org.apache.tomcat.util.net.NioBlockingSelector$BlockPoller.run(NioBlockingSelector.java:298)

- (4) sun.nio.ch.SelectorImpl.select(SelectorImpl.java:97)
- (4) org.springframework.boot.web.embedded.tomcat.TomcatWebServer$1.run(TomcatWebServer.java:182)
  - (4) org.apache.catalina.core.StandardServer.await(StandardServer.java:427)
- (4) java.util.TimerThread.run(Timer.java:505)
  - (4) java.util.TimerThread.mainLoop(Timer.java:552)
- (4) net.sourceforge.jtds.util.TimerThread.run(TimerThread.java:115)
  - (4) java.lang.Object.wait(Native Method)
- (4) com.mysql.cj.jdbc.AbandonedConnectionCleanupThread.run(AbandonedConnectionCleanupThread.java:43)
  - (4) java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:144)
- (4) okio.AsyncTimeout$Watchdog.run(AsyncTimeout.java:312)
  - (1) okio.AsyncTimeout.awaitTimeout(AsyncTimeout.java:347)
  - (3) okio.AsyncTimeout.awaitTimeout(AsyncTimeout.java:361)
- (4) org.apache.skywalking.apm.commons.datacarrier.consumer.ConsumerThread.run(ConsumerThread.java:55)
  - (4) java.lang.Thread.sleep(Native Method)
- (4) java.lang.ref.Finalizer$FinalizerThread.run(Finalizer.java:216)
  - (4) java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:165)
- (4) java.lang.ref.Reference$ReferenceHandler.run(Reference.java:153)
  - (4) java.lang.ref.Reference.tryHandlePending(Reference.java:191)
- (2) ch.qos.logback.core.AsyncAppenderBase$Worker.run(AsyncAppenderBase.java:290)
  - (2) ch.qos.logback.core.spi.AppenderAttachableImpl.appendLoopOnAppenders(AppenderAttachableImpl.java:51)

Thread Dump 1 - Intelligence Report

File: *jstack.tar (2).gz-4*

Thread Count Summary

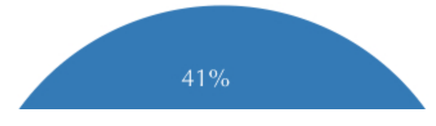
To learn about different thread states through real-life example, check out this video tutorial

Total Threads count: 139

73
RUNNABLE

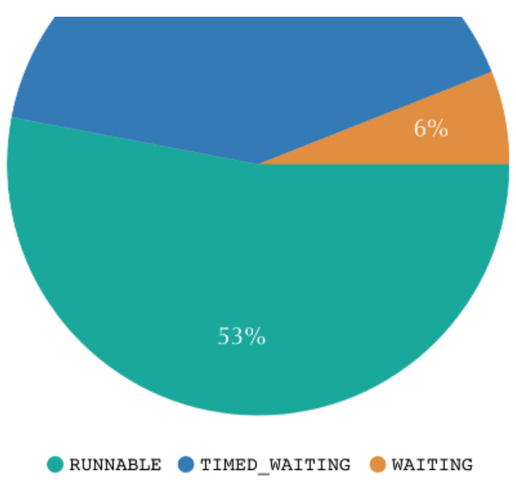
57
TIMED_WAITING

Thread state %



View Details

View Details



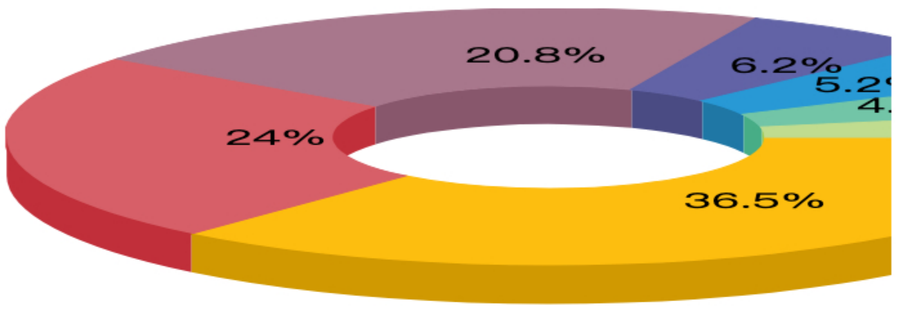
9

WAITING

View Details

Thread Group

Threads with similar names are grouped in this section



Legends	Thread Group	Count	States
●	grpc	**35** threads	RUNNABLE:34 — TIMED_WAITING:1
●	OkHttp ConnectionPool	**23** threads	TIMED_WAITING:23
●	http	**20** threads	TIMED_WAITING:17 — RUNNABLE:3
●	SkywalkingAgent	**6** threads	TIMED_WAITING:6
●	Gang worker	**5** threads	RUNNABLE:5
●	G1 Concurrent Refinement Thread	**4** threads	RUNNABLE:4

Show all thread groups >>

Daemon vs non-Daemon

Learn more about daemon and non-daemon (i.e. user threads)

119

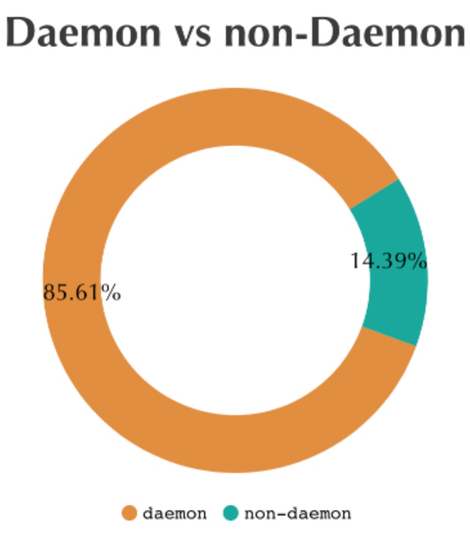
DAEMON

View Details

20

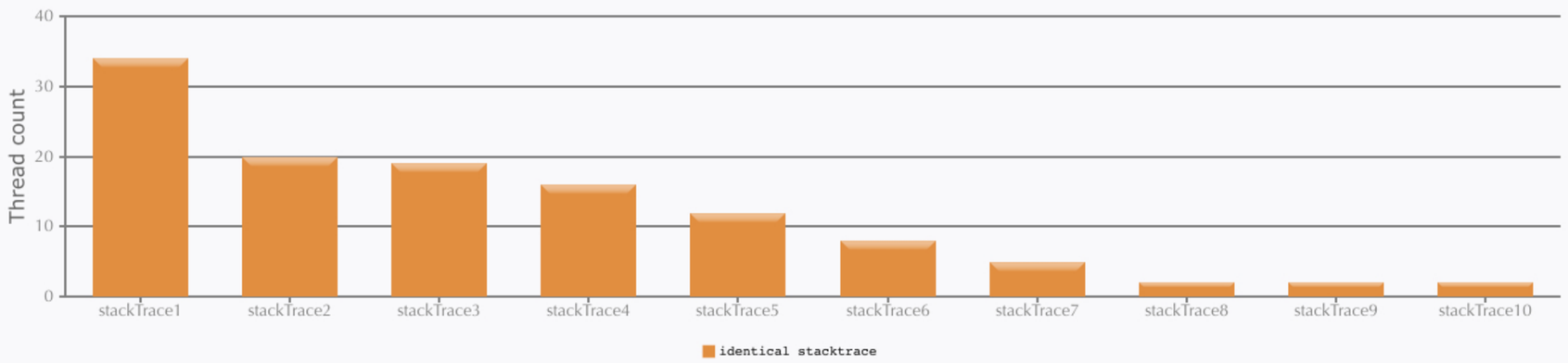
NON-DAEMON

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 HSI Pattern



Thread Count	Identical Stack trace
34 RUNNABLE threads	java.lang.Thread.State: RUNNABLE at sun.nio.ch.EPollArrayWrapper.epollWait(Native Method) at sun.nio.ch.EPollArrayWrapper.poll(EPollArrayWrapper.java:269) at sun.nio.ch.EPollSelectorImpl.doSelect(EPollSelectorImpl.java:93) at sun.nio.ch.SelectorImpl.lockAndDoSelect(SelectorImpl.java:86) ... See complete stacktrace.
20 RUNNABLE threads	java.lang.Thread.State: RUNNABLE Locked ownable synchronizers: -None See complete stacktrace.
19 TIMED_WAITING threads	java.lang.Thread.State: TIMED_WAITING (on object monitor) at java.lang.Object.wait(Native Method) at java.lang.Object.wait(Object.java:460) at okhttp3.ConnectionPool$1.run(ConnectionPool.java:67) - locked <0x00000000d2311dd8> (a okhttp3.ConnectionPool) ... See complete stacktrace.
16 TIMED_WAITING threads	java.lang.Thread.State: TIMED_WAITING (parking) at sun.misc.Unsafe.park(Native Method) - parking to wait for <0x00000000b96f4db0> (a java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject) at java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:215) at java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.awaitNanos(AbstractQueuedSynchronizer.java:2078) ... See complete stacktrace.
12 RUNNABLE threads	stacktrace See complete stacktrace.
8 TIMED_WAITING threads	java.lang.Thread.State: TIMED_WAITING (parking) at sun.misc.Unsafe.park(Native Method) - parking to wait for <0x00000000bb0e4848> (a java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject) at java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:215) at java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.awaitNanos(AbstractQueuedSynchronizer.java:2078) ... See complete stacktrace.
5 TIMED_WAITING threads	java.lang.Thread.State: TIMED_WAITING (parking) at sun.misc.Unsafe.park(Native Method) - parking to wait for <0x00000000b4ffb610> (a java.util.concurrent.SynchronousQueue$TransferStack) at java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:215) at java.util.concurrent.SynchronousQueue$TransferStack.awaitFulfill(SynchronousQueue.java:460) ... See complete stacktrace.
2 RUNNABLE threads	java.lang.Thread.State: RUNNABLE at sun.nio.ch.EPollArrayWrapper.epollWait(Native Method) at sun.nio.ch.EPollArrayWrapper.poll(EPollArrayWrapper.java:269) at sun.nio.ch.EPollSelectorImpl.doSelect(EPollSelectorImpl.java:93) at sun.nio.ch.SelectorImpl.lockAndDoSelect(SelectorImpl.java:86) ... See complete stacktrace.
2 WAITING threads	java.lang.Thread.State: WAITING (parking) at sun.misc.Unsafe.park(Native Method) - parking to wait for <0x00000000b9716f78> (a java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject) at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175) at java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(AbstractQueuedSynchronizer.java:2039) ... See complete stacktrace.
2 RUNNABLE threads	java.lang.Thread.State: RUNNABLE at sun.nio.ch.EPollArrayWrapper.epollWait(Native Method) at sun.nio.ch.EPollArrayWrapper.poll(EPollArrayWrapper.java:269) at sun.nio.ch.EPollSelectorImpl.doSelect(EPollSelectorImpl.java:93)

at sun.nio.ch.SelectorImpl.lockAndDoSelect(SelectorImpl.java:86)
...
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
39 threads	sun.nio.ch.EPollArrayWrapper.epollWait(Native Method). To see stack trace click here.	28%
35 threads	sun.misc.Unsafe.park(Native Method). To see stack trace click here.	25%
25 threads	java.lang.Object.wait(Native Method). To see stack trace click here.	18%
5 threads	java.lang.Thread.sleep(Native Method). To see stack trace click here.	4%
2 threads	java.net.PlainSocketImpl.socketConnect(Native Method). To see stack trace click here.	1%

Show all methods >>

CPU consuming threads

If application is consuming high CPU, investigate below threads. Learn Athlete pattern

Thread	CPU consuming thread's stacktrace
31 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

11
GC threads

View Details

GC Thread type	Count
Concurrent Mark GC Thread	1
Gang worker	5
Concurrent GC	1
G1 Concurrent Refinement Thread	4
Total	11

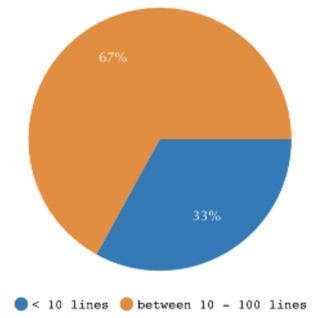
☑ GC thread count is normal

Threads Stack Length

Lengthy stacks can cause StackOverflowError. Learn more

☑ No Problem in Stack trace length.

Stack Length	Thread count
< 10 lines	46
between 10 - 100 lines	93



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.

- (107) root
  - (95) java.lang.Thread.run(Thread.java:748)
    - (38) java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    - (34) org.apache.skywalking.apm.dependencies.io.netty.util.concurrent.FastThreadLocalRunnable.run(FastThreadLocalRunnable.java:30)
    - (16) org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
    - (2) org.apache.kafka.clients.producer.internals.Sender.run(Sender.java:238)
    - (2) org.apache.tomcat.util.net.NioEndpoint$Poller.run(NioEndpoint.java:793)
    - (1) org.apache.coyote.AbstractProtocol$AsyncTimeout.run(AbstractProtocol.java:1133)
    - (1) org.apache.tomcat.util.net.NioEndpoint$Acceptor.run(NioEndpoint.java:455)
    - (1) org.apache.catalina.core.ContainerBase$ContainerBackgroundProcessor.run(ContainerBase.java:1357)
  - (2) ch.qos.logback.core.AsyncAppenderBase$Worker.run(AsyncAppenderBase.java:289)
    - (2) java.util.concurrent.ArrayBlockingQueue.take(ArrayBlockingQueue.java:403)
  - (1) java.util.concurrent.ForkJoinWorkerThread.run(ForkJoinWorkerThread.java:157)
    - (1) java.util.concurrent.ForkJoinPool.runWorker(ForkJoinPool.java:1693)
  - (1) org.apache.tomcat.util.net.NioBlockingSelector$BlockPoller.run(NioBlockingSelector.java:298)
    - (1) sun.nio.ch.SelectorImpl.select(SelectorImpl.java:97)
  - (1) org.springframework.boot.web.embedded.tomcat.TomcatWebServer$1.run(TomcatWebServer.java:182)
    - (1) org.apache.catalina.core.StandardServer.await(StandardServer.java:427)
  - (1) java.util.TimerThread.run(Timer.java:505)
    - (1) java.util.TimerThread.mainLoop(Timer.java:552)
  - (1) net.sourceforge.jtds.util.TimerThread.run(TimerThread.java:115)
    - (1) java.lang.Object.wait(Native Method)
  - (1) com.mysql.cj.jdbc.AbandonedConnectionCleanupThread.run(AbandonedConnectionCleanupThread.java:43)
    - (1) java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:144)
  - (1) okio.AsyncTimeout$Watchdog.run(AsyncTimeout.java:312)
    - (1) okio.AsyncTimeout.awaitTimeout(AsyncTimeout.java:347)
  - (1) org.apache.skywalking.apm.commons.datacarrier.consumer.ConsumerThread.run(ConsumerThread.java:55)
    - (1) java.lang.Thread.sleep(Native Method)
  - (1) java.lang.ref.Finalizer$FinalizerThread.run(Finalizer.java:216)
    - (1) java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:165)
  - (1) java.lang.ref.Reference$ReferenceHandler.run(Reference.java:153)
    - (1) java.lang.ref.Reference.tryHandlePending(Reference.java:191)

Thread Dump 2 - Intelligence Report

File: *jstack.tar (2).gz-3*

Thread Count Summary

To learn about different thread states through real-life example, check out this video tutorial

Total Threads count: 136

79
RUNNABLE
View Details

50
TIMED_WAITING
View Details

7
WAITING
View Details

Thread Group

Threads with similar names are grouped in this section

Show all thread groups >>

Daemon vs non-Daemon

Learn more about daemon and non-daemon (i.e. user threads)

116

DAEMON

View Details

20

NON-DAEMON

View Details

Threads Stack Length

Lengthy stacks can cause StackOverflowError. Learn more

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

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]

Show all methods >>

CPU consuming threads

If application is consuming high CPU, investigate below threads. Learn Athlete pattern

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

11

GC threads

View Details

GC thread count is normal

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

All threads stacktrace are combined in to one single tree. Learn it's benefits.

- (104) root
  - (92) java.lang.Thread.run(Thread.java:748)
    - (37) java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    - (32) org.apache.skywalking.apm.dependencies.io.netty.util.concurrent.FastThreadLocalRunnable.run(FastThreadLocalRunnable.java:30)
    - (16) org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
    - (2) org.apache.kafka.clients.producer.internals.Sender.run(Sender.java:238)
    - (2) org.apache.tomcat.util.net.NioEndpoint$Poller.run(NioEndpoint.java:793)
    - (1) org.apache.coyote.AbstractProtocol$AsyncTimeout.run(AbstractProtocol.java:1133)
    - (1) org.apache.tomcat.util.net.NioEndpoint$Acceptor.run(NioEndpoint.java:455)

- (1) org.apache.catalina.core.ContainerBase$ContainerBackgroundProcessor.run(ContainerBase.java:1357)
- (2) ch.qos.logback.core.AsyncAppenderBase$Worker.run(AsyncAppenderBase.java:290)
  - (2) ch.qos.logback.core.spi.AppenderAttachableImpl.appendLoopOnAppenders(AppenderAttachableImpl.java:51)
- (1) java.util.concurrent.ForkJoinWorkerThread.run(ForkJoinWorkerThread.java:157)
  - (1) java.util.concurrent.ForkJoinPool.runWorker(ForkJoinPool.java:1693)
- (1) org.apache.tomcat.util.net.NioBlockingSelector$BlockPoller.run(NioBlockingSelector.java:298)
  - (1) sun.nio.ch.SelectorImpl.select(SelectorImpl.java:97)
- (1) org.springframework.boot.web.embedded.tomcat.TomcatWebServer$1.run(TomcatWebServer.java:182)
  - (1) org.apache.catalina.core.StandardServer.await(StandardServer.java:427)
- (1) java.util.TimerThread.run(Timer.java:505)
  - (1) java.util.TimerThread.mainLoop(Timer.java:552)
- (1) net.sourceforge.jtds.util.TimerThread.run(TimerThread.java:115)
  - (1) java.lang.Object.wait(Native Method)
- (1) com.mysql.cj.jdbc.AbandonedConnectionCleanupThread.run(AbandonedConnectionCleanupThread.java:43)
  - (1) java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:144)
- (1) okio.AsyncTimeout$Watchdog.run(AsyncTimeout.java:312)
  - (3) okio.AsyncTimeout.awaitTimeout(AsyncTimeout.java:361)
- (1) org.apache.skywalking.apm.commons.datacarrier.consumer.ConsumerThread.run(ConsumerThread.java:55)
  - (1) java.lang.Thread.sleep(Native Method)
- (1) java.lang.ref.Finalizer$FinalizerThread.run(Finalizer.java:216)
  - (1) java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:165)
- (1) java.lang.ref.Reference$ReferenceHandler.run(Reference.java:153)
  - (1) java.lang.ref.Reference.tryHandlePending(Reference.java:191)

Thread Dump 3 - Intelligence Report

File: *jstack.tar (2).gz-2*

Thread Count Summary

To learn about different thread states through real-life example, check out this video tutorial

Total Threads count: 138

73

56

RUNNABLE

View Details

TIMED_WAITING

View Details

9

WAITING

View Details

Thread Group

Threads with similar names are grouped in this section

Show all thread groups >>

Daemon vs non-Daemon

Learn more about daemon and non-daemon (i.e. user threads)

118

DAEMON

View Details

20

NON-DAEMON

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

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

Show all methods >>

CPU consuming threads

If application is consuming high CPU, investigate below threads. Learn Athlete pattern

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

11

GC threads

View Details

GC thread count is normal

Threads Stack Length

Lengthy stacks can cause StackOverflowError. Learn more

No Problem in Stack trace length.

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

All threads stacktrace are combined in to one single tree. Learn it's benefits.

- (106) root
  - (94) java.lang.Thread.run(Thread.java:748)
    - (41) java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    - (30) org.apache.skywalking.apm.dependencies.io.netty.util.concurrent.FastThreadLocalRunnable.run(FastThreadLocalRunnable.java:30)
    - (16) org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
    - (2) org.apache.kafka.clients.producer.internals.Sender.run(Sender.java:238)
    - (2) org.apache.tomcat.util.net.NioEndpoint$Poller.run(NioEndpoint.java:793)
    - (1) org.apache.coyote.AbstractProtocol$AsyncTimeout.run(AbstractProtocol.java:1133)
    - (1) org.apache.tomcat.util.net.NioEndpoint$Acceptor.run(NioEndpoint.java:455)
    - (1) org.apache.catalina.core.ContainerBase$ContainerBackgroundProcessor.run(ContainerBase.java:1357)
  - (2) ch.qos.logback.core.AsyncAppenderBase$Worker.run(AsyncAppenderBase.java:289)
    - (2) java.util.concurrent.ArrayBlockingQueue.take(ArrayBlockingQueue.java:403)
  - (1) java.util.concurrent.ForkJoinWorkerThread.run(ForkJoinWorkerThread.java:157)
    - (1) java.util.concurrent.ForkJoinPool.runWorker(ForkJoinPool.java:1693)
  - (1) org.apache.tomcat.util.net.NioBlockingSelector$BlockPoller.run(NioBlockingSelector.java:298)
    - (1) sun.nio.ch.SelectorImpl.select(SelectorImpl.java:97)
  - (1) org.springframework.boot.web.embedded.tomcat.TomcatWebServer$1.run(TomcatWebServer.java:182)
    - (1) org.apache.catalina.core.StandardServer.await(StandardServer.java:427)
  - (1) java.util.TimerThread.run(Timer.java:505)
    - (1) java.util.TimerThread.mainLoop(Timer.java:552)
  - (1) net.sourceforge.jtds.util.TimerThread.run(TimerThread.java:115)
    - (1) java.lang.Object.wait(Native Method)
  - (1) com.mysql.cj.jdbc.AbandonedConnectionCleanupThread.run(AbandonedConnectionCleanupThread.java:43)
    - (1) java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:144)

- (1) okio.AsyncTimeout$Watchdog.run(AsyncTimeout.java:312)
  - (1) okio.AsyncTimeout.awaitTimeout(AsyncTimeout.java:361)
- (1) org.apache.skywalking.apm.commons.datacarrier.consumer.ConsumerThread.run(ConsumerThread.java:55)
  - (1) java.lang.Thread.sleep(Native Method)
- (1) java.lang.ref.Finalizer$FinalizerThread.run(Finalizer.java:216)
  - (1) java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:165)
- (1) java.lang.ref.Reference$ReferenceHandler.run(Reference.java:153)
  - (1) java.lang.ref.Reference.tryHandlePending(Reference.java:191)

Thread Dump 4 - Intelligence Report

File: ***jstack.tar (2).gz-1***

Thread Count Summary

To learn about different thread states through real-life example, check out this video tutorial

Total Threads count: 132

74
RUNNABLE
View Details

49
TIMED_WAITING
View Details

9
WAITING
View Details

Thread Group

Threads with similar names are grouped in this section

Show all thread groups >>

Daemon vs non-Daemon

Learn more about daemon and non-daemon (i.e. user threads)

112

DAEMON

View Details

20

NON-DAEMON

View Details

Threads Stack Length

Lengthy stacks can cause StackOverflowError. Learn more

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

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

Show all methods >>

CPU consuming threads

If application is consuming high CPU, investigate below threads. Learn Athlete pattern

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

11

GC threads

View Details

GC thread count is normal

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

All threads stacktrace are combined in to one single tree. Learn it's benefits.

- (100) root
  - (88) java.lang.Thread.run(Thread.java:748)
    - (39) java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    - (26) org.apache.skywalking.apm.dependencies.io.netty.util.concurrent.FastThreadLocalRunnable.run(FastThreadLocalRunnable.java:30)
    - (16) org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
    - (2) org.apache.kafka.clients.producer.internals.Sender.run(Sender.java:238)
    - (2) org.apache.tomcat.util.net.NioEndpoint$Poller.run(NioEndpoint.java:793)
    - (1) org.apache.coyote.AbstractProtocol$AsyncTimeout.run(AbstractProtocol.java:1133)
    - (1) org.apache.tomcat.util.net.NioEndpoint$Acceptor.run(NioEndpoint.java:455)
    - (1) org.apache.catalina.core.ContainerBase$ContainerBackgroundProcessor.run(ContainerBase.java:1357)
  - (2) ch.qos.logback.core.AsyncAppenderBase$Worker.run(AsyncAppenderBase.java:289)
    - (2) java.util.concurrent.ArrayBlockingQueue.take(ArrayBlockingQueue.java:403)
  - (1) java.util.concurrent.ForkJoinWorkerThread.run(ForkJoinWorkerThread.java:157)
    - (1) java.util.concurrent.ForkJoinPool.runWorker(ForkJoinPool.java:1693)
  - (1) org.apache.tomcat.util.net.NioBlockingSelector$BlockPoller.run(NioBlockingSelector.java:298)
    - (1) sun.nio.ch.SelectorImpl.select(SelectorImpl.java:97)
  - (1) org.springframework.boot.web.embedded.tomcat.TomcatWebServer$1.run(TomcatWebServer.java:182)
    - (1) org.apache.catalina.core.StandardServer.await(StandardServer.java:427)
  - (1) java.util.TimerThread.run(Timer.java:505)
    - (1) java.util.TimerThread.mainLoop(Timer.java:552)
  - (1) net.sourceforge.jtds.util.TimerThread.run(TimerThread.java:115)
    - (1) java.lang.Object.wait(Native Method)
  - (1) com.mysql.cj.jdbc.AbandonedConnectionCleanupThread.run(AbandonedConnectionCleanupThread.java:43)
    - (1) java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:144)
  - (1) okio.AsyncTimeout$Watchdog.run(AsyncTimeout.java:312)
    - (1) okio.AsyncTimeout.awaitTimeout(AsyncTimeout.java:361)
  - (1) org.apache.skywalking.apm.commons.datacarrier.consumer.ConsumerThread.run(ConsumerThread.java:55)
    - (1) java.lang.Thread.sleep(Native Method)
  - (1) java.lang.ref.Finalizer$FinalizerThread.run(Finalizer.java:216)
    - (1) java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:165)
  - (1) java.lang.ref.Reference$ReferenceHandler.run(Reference.java:153)
    - (1) java.lang.ref.Reference.tryHandlePending(Reference.java:191)