Thread Dump - Intelligence Report

File: *inbound-thread dumps.txt*

Timestamp: 2024-10-13 07:08:22

Thread Count Summary

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

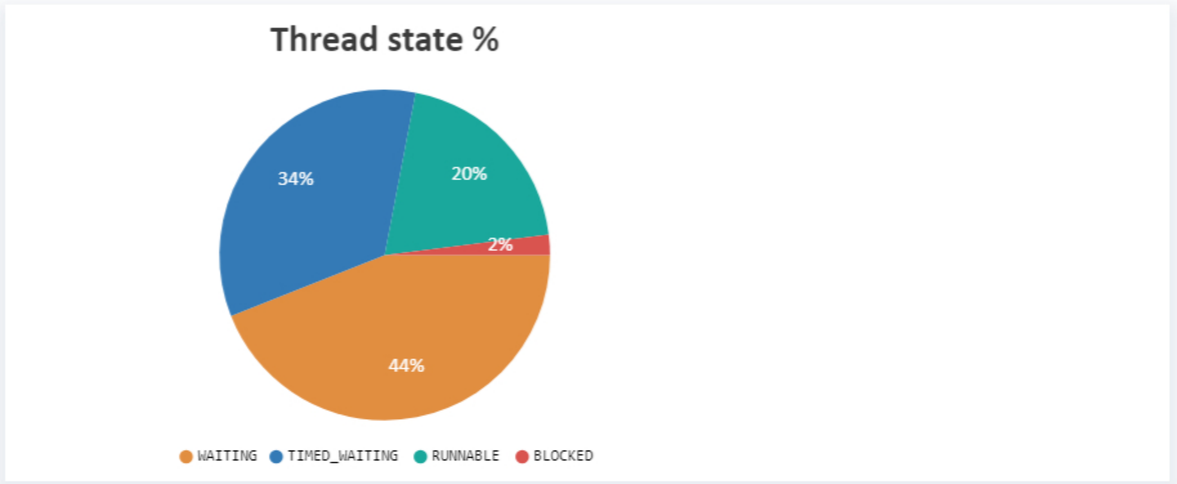
Total Threads count: 104

46
WAITING
View Details

35
TIMED_WAITING
View Details

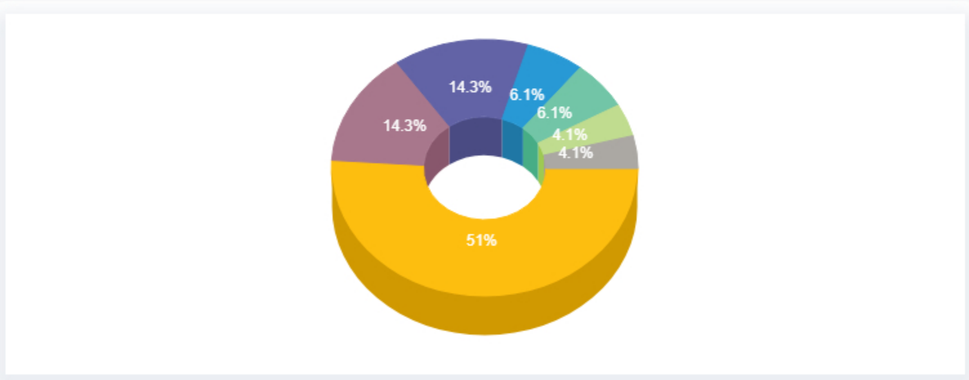
21
RUNNABLE
View Details

2
BLOCKED
View Details



Thread Pools

Threads with similar names are grouped in this section



	Thread Pool	Count	States
●	[prod-b2b-inbound-as2-qeew].ObjectStoreManager-M..	**25** threads	WAITING:25
●	qtp1898563665	**7** threads	TIMED_WAITING:5
●	[prod-b2b-inbound-as2	**7** threads	RUNNABLE:6 WAITING:1
●	connection-pool-delays-thread	**3** threads	TIMED_WAITING:3
●	Grizzly-IdleTimeoutFilter	**3** threads	TIMED_WAITING:3
●	servo-gauge	**2** threads	TIMED_WAITING:1 WAITING:1

Show all thread groups >>

Daemon vs non-Daemon

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

70

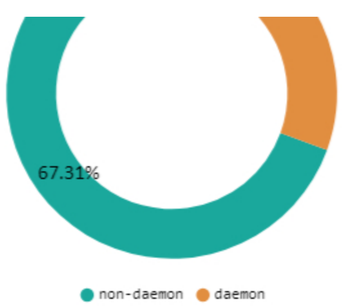
NON-DAEMON

View Details

34

DAEMON

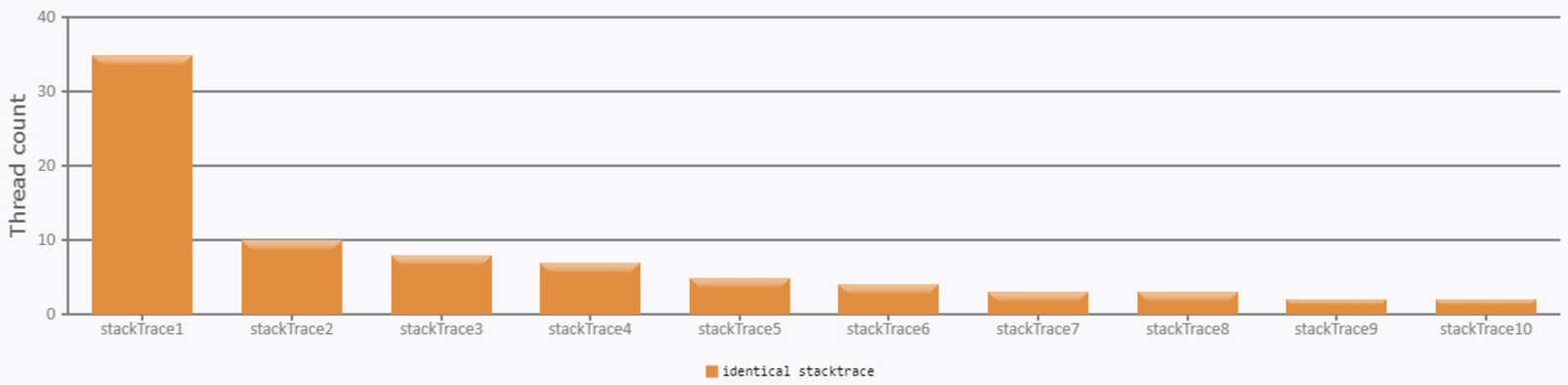
View Details



Threads with identical stack trace

Become Performance Expert! Training from FastThread Architect!

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
35 WAITING threads	java.lang.Thread.State: WAITING (parking) at sun.misc.Unsafe.park(Native Method) - parking to wait for <0x00000000e5532110> (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.
10 TIMED_WAITING threads	java.lang.Thread.State: TIMED_WAITING (parking) at sun.misc.Unsafe.park(Native Method) - parking to wait for <0x00000000e5530df8> (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.
8 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.
7 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 org.apache.logging.log4j.core.async.TimeoutBlockingWaitStrategy.awaitNanos(TimeoutBlockingWaitStrategy.java:130) at org.apache.logging.log4j.core.async.TimeoutBlockingWaitStrategy.waitFor(TimeoutBlockingWaitStrategy.java:86) ... See complete stacktrace.
5 RUNNABLE threads	java.lang.Thread.State: RUNNABLE See complete stacktrace.
4 TIMED_WAITING threads	java.lang.Thread.State: TIMED_WAITING (parking) at sun.misc.Unsafe.park(Native Method) - parking to wait for <0x00000000e0c86c20> (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.
3 TIMED_WAITING threads	java.lang.Thread.State: TIMED_WAITING (on object monitor) at java.lang.Object.wait(Native Method) at org.glassfish.grizzly.utils.DelayedExecutor$DelayedRunnable.run(DelayedExecutor.java:187) - locked <0x00000000e489cc08> (a org.glassfish.grizzly.utils.DelayedExecutor) at org.glassfish.grizzly.threadpool.AbstractThreadPool$Worker.doWork(AbstractThreadPool.java:593) ... See complete stacktrace.
3 TIMED_WAITING threads	java.lang.Thread.State: TIMED_WAITING (on object monitor) at java.lang.Object.wait(Native Method) at org.glassfish.grizzly.utils.DelayedExecutor$DelayedRunnable.run(DelayedExecutor.java:187) - locked <0x00000000e489cdd8> (a org.glassfish.grizzly.utils.DelayedExecutor) at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)

	... See complete stacktrace.
2 RUNNABLE threads	stacktrace See complete stacktrace.
2 TIMED_WAITING threads	java.lang.Thread.State: TIMED_WAITING (on object monitor) at java.lang.Object.wait(Native Method) at org.glassfish.grizzly.utils.DelayedExecutor$DelayedRunnable.run(DelayedExecutor.java:187) - locked <0x00000000e41047b8> (a org.glassfish.grizzly.utils.DelayedExecutor) at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) ... See complete stacktrace.

Last executed methods

Methods that threads were executing when thread dump was captured is reported. Learn All roads lead to Rome pattern

Thread Count	Method	Percentage
56 threads	sun.misc.Unsafe.park(Native Method) To see stack trace click here.	54%
22 threads	java.lang.Object.wait(Native Method) To see stack trace click here.	21%
10 threads	sun.nio.ch.EPollArrayWrapper.epollWait(Native Method) To see stack trace click here.	10%
2 threads	java.lang.Thread.sleep(Native Method) To see stack trace click here.	2%
2 threads	java.net.SocketInputStream.socketRead0(Native Method) To see stack trace click here.	2%

Show all methods >>

CPU consuming threads

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

Thread	CPU consuming thread's stacktrace
qtp1898563665-25 nativeId: 188	java.lang.Thread.State: RUNNABLE at sun.management.DiagnosticCommandImpl.executeDiagnosticCommand(Native Method) at sun.management.DiagnosticCommandImpl.access$000(DiagnosticCommandImpl.java:40) at sun.management.DiagnosticCommandImpl$Wrapper.execute(DiagnosticCommandImpl.java:139) at sun.management.DiagnosticCommandImpl.invoke(DiagnosticCommandImpl.java:230) ... See complete stacktrace.

Need help diagnosing high CPU consumption? Learn our Effective Tips

line #576 of *org.apache.activemq.transport.failover.FailoverTransport* file in *oneway()* method is blocking 2 threads.

Blocking threads makes application unresponsive, learn Traffic Jam pattern

How to understand this graph? Watch this video

GC Threads

Garbage collection threads count reported. Learn Scavengers pattern

Not reported in the thread dump

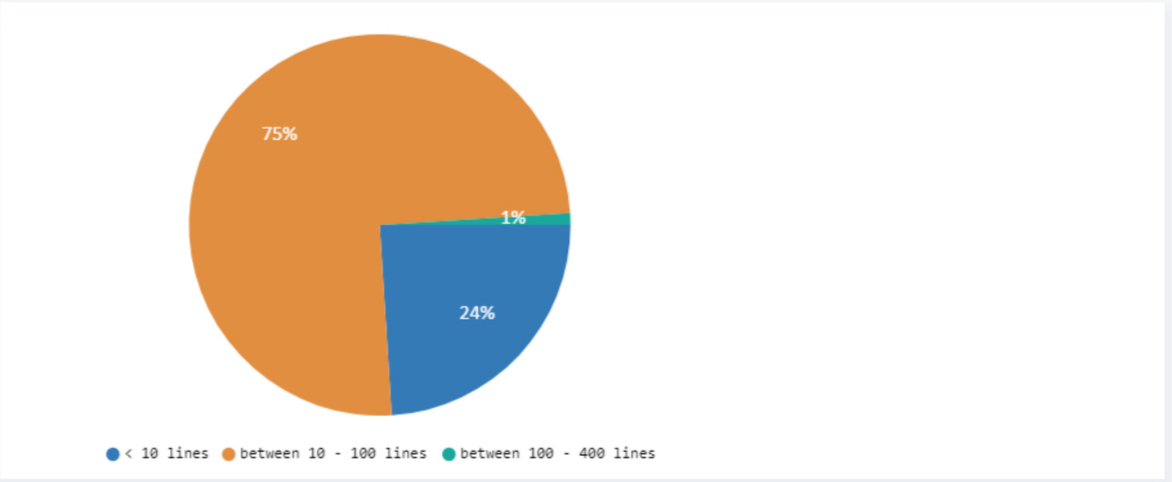
Threads Stack Length

Threads Stack Length

Lengthy stacks can cause StackOverflowError. Learn more

No Problem in Stack trace length.

Stack Length	Thread count
< 10 lines	25
between 10 - 100 lines	78
between 100 - 400 lines	1



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

Reverse Call stack

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

- (97) root
  - (91) java.lang.Thread.run(Thread.java:750)
    - (68) java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    - (7) com.lmax.disruptor.BatchEventProcessor.run(BatchEventProcessor.java:125)
    - (4) org.eclipse.jetty.util.thread.QueuedThreadPool$Runner.run(QueuedThreadPool.java:1034)

- (4) org.eclipse.jetty.util.thread.QueuedThreadPool$Runner.run(QueuedThreadPool.java:1018)
- (3) org.glassfish.grizzly.threadpool.AbstractThreadPool$Worker.run(AbstractThreadPool.java:573)
- (2) io.netty.util.concurrent.FastThreadLocalRunnable.run(FastThreadLocalRunnable.java:30)
- (1) org.apache.activemq.transport.tcp.TcpTransport.run(TcpTransport.java:215)
- (1) sun.nio.fs.LinuxWatchService$Poller.run(LinuxWatchService.java:314)
- (1) org.tanukisoftware.wrapper.WrapperManager.run(WrapperManager.java:6332)

(1) java.util.TimerThread.run(Timer.java:505)

- (1) java.util.TimerThread.mainLoop(Timer.java:526)

(1) org.quartz.core.QuartzSchedulerThread.run(QuartzSchedulerThread.java:427)

- (1) java.lang.Object.wait(Native Method)

(1) org.quartz.simpl.SimpleThreadPool$WorkerThread.run(SimpleThreadPool.java:568)

- (1) java.lang.Object.wait(Native Method)

(1) org.tanukisoftware.wrapper.WrapperManager$3.run(WrapperManager.java:1074)

- (1) java.lang.Thread.sleep(Native Method)

(1) java.lang.ref.Finalizer$FinalizerThread.run(Finalizer.java:188)

- (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)

My Patterns

Manage my Patterns

Create, apply, and visualize custom domain patterns for precise analysis. Learn More

No Pattern Found

Divide & Conquer Panel

View More

Search Panel

View More