Thread Dump - Intelligence Report

File: *dump.out*

Timestamp: 2022-03-02 18:48:04

JVM Version: 64-Bit Server VM (25.191-b12 mixed mode)

Problems Detected

Our **machine learning (ML)** algorithms have detected problems in your application which may cause **application unresponsiveness**. To see the identified problems and recommended solutions, please subscribe to our plan.

Select Plan

Thread Count Summary

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

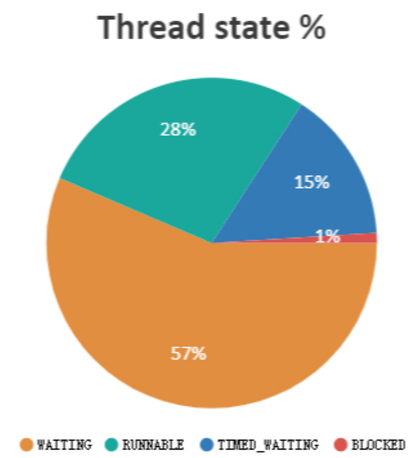
Total Threads count: 175

99
WAITING
View Details

49
RUNNABLE
View Details

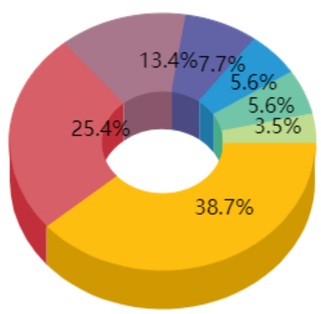
26
TIMED_WAITING
View Details

1
BLOCKED
View Details



Thread Group

Threads with similar names are grouped in this section



	Thread Group	Count	States
	pulsar	**55** threads	WAITING:21 TIMED_WAITING:11 RUNNABLE:
	amqp	**36** threads	WAITING:31 TIMED_WAITING:3
	bookkeeper	**19** threads	WAITING:14 TIMED_WAITING:3
	zk	**11** threads	WAITING:11
	Gang worker	**8** threads	RUNNABLE: 8
	broker	**8** threads	WAITING:8

Show all thread groups >>

Daemon vs non-Daemon

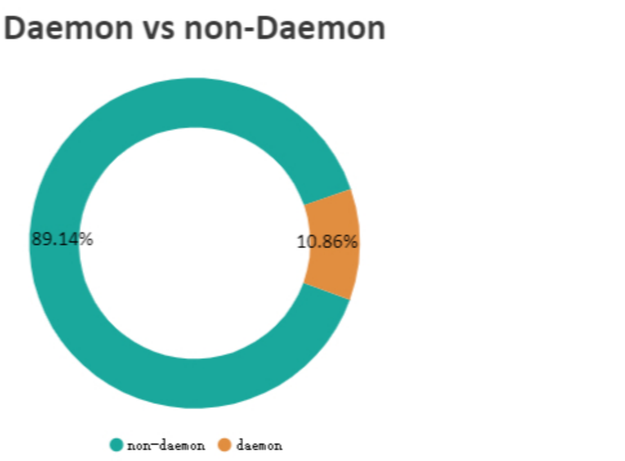
Learn more about daemon and non-daemon (i.e. user threads)

156
NON-DAEMON

View Details

19
DAEMON

View Details



org.apache.pulsar.client.impl.ProducerImpl is blocking 1 thread.

Blocking threads makes application unresponsive, learn Traffic Jam pattern

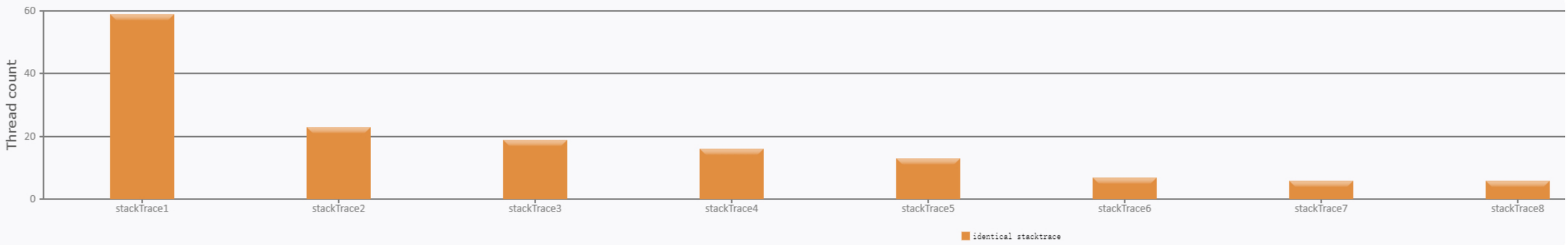
How to understand this graph? Watch this video

pulsar-client-io-57-14

pulsar-timer-60-1

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
59 WAITING threads	java.lang.Thread.State: WAITING (parking) at sun.misc.Unsafe.park(Native Method) - parking to wait for <0x00000000c14b0870> (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.
23 WAITING threads	java.lang.Thread.State: WAITING (parking) at sun.misc.Unsafe.park(Native Method) - parking to wait for <0x00000000c311c6b8> (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.
19 TIMED_WAITING threads	java.lang.Thread.State: TIMED_WAITING (parking) at sun.misc.Unsafe.park(Native Method) - parking to wait for <0x00000000c2ff8fc0> (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.
16 RUNNABLE threads	stacktrace See complete stacktrace.

13 RUNNABLE threads	java.lang.Thread.State: RUNNABLE at io.netty.channel.epoll.Native.epollWait0(Native Method) at io.netty.channel.epoll.Native.epollWait(Native.java:132) at io.netty.channel.epoll.EpollEventLoop.epollWait(EpollEventLoop.java:286) at io.netty.channel.epoll.EpollEventLoop.run(EpollEventLoop.java:351) ... See complete stacktrace.
7 RUNNABLE threads	java.lang.Thread.State: RUNNABLE at io.netty.channel.epoll.Native.epollWait(Native Method) at io.netty.channel.epoll.Native.epollWait(Native.java:148) at io.netty.channel.epoll.Native.epollWait(Native.java:141) at io.netty.channel.epoll.EpollEventLoop.epollWaitNoTimerChange(EpollEventLoop.java:290) ... See complete stacktrace.
6 RUNNABLE threads	java.lang.Thread.State: RUNNABLE See complete stacktrace.
6 WAITING threads	java.lang.Thread.State: WAITING (parking) at sun.misc.Unsafe.park(Native Method) - parking to wait for <0x00000000c2fef2d8> (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.
4 WAITING threads	java.lang.Thread.State: WAITING (parking) at sun.misc.Unsafe.park(Native Method) - parking to wait for <0x00000000c14dda28> (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.
4 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
121 threads	sun.misc.Unsafe.park(Native Method). To see stack trace click here.	69%
13 threads	io.netty.channel.epoll.Native.epollWait0(Native Method). To see stack trace click here.	7%
7 threads	io.netty.channel.epoll.Native.epollWait(Native Method). To see stack trace click here.	4%
6 threads	sun.nio.ch.EPollArrayWrapper.epollWait(Native Method). To see stack trace click here.	3%
2 threads	java.lang.Object.wait(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
21 JVM threads	See complete stacktrace.

GC Threads

Garbage collection threads count reported. Learn Scavengers pattern

15

GC Thread type	Count
G1 Concurrent Refinement Thread	5
Gang worker	8

GC threads

View Details

GC thread count is normal

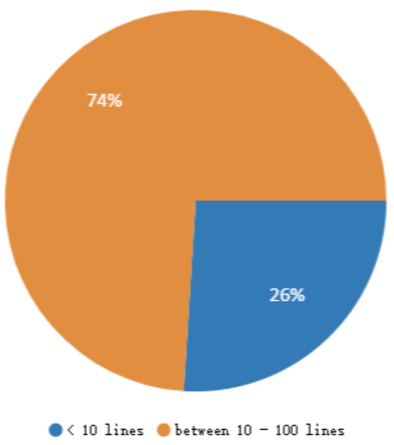
Concurrent Mark GC Thread	1
Concurrent GC	1
Total	15

Threads Stack Length

Lengthy stacks can cause StackOverflowError. Learn more

No Problem in Stack trace length.

Stack Length	Thread count
< 10 lines	45
between 10 - 100 lines	130



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.

(153) root

- (141) java.lang.Thread.run(Thread.java:748)
  - (136) io.netty.util.concurrent.FastThreadLocalRunnable.run(FastThreadLocalRunnable.java:30)
  - (5) java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
- (4) org.apache.zookeeper.ClientCnxn$EventThread.run(ClientCnxn.java:506)
  - (4) java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:442)
- (4) org.apache.zookeeper.ClientCnxn$SendThread.run(ClientCnxn.java:1223)
  - (4) org.apache.zookeeper.ClientCnxnSocketNIO.doTransport(ClientCnxnSocketNIO.java:345)
- (1) java.util.concurrent.ForkJoinWorkerThread.run(ForkJoinWorkerThread.java:157)
  - (1) java.util.concurrent.ForkJoinPool.runWorker(ForkJoinPool.java:1693)
- (1) java.lang.ref.Finalizer$FinalizerThread.run(Finalizer.java:216)
  - (1) java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:165)
- (1) java.lang.ref.Reference$ReferenceHandler.run(Reference.java:153)
  - (1) java.lang.ref.Reference.tryHandlePending(Reference.java:191)
- (1) org.apache.pulsar.PulsarBrokerStarter.main(PulsarBrokerStarter.java:356)
  - (1) org.apache.pulsar.PulsarBrokerStarter$BrokerStarter.join(PulsarBrokerStarter.java:287)