



Oracle Weblogic Server

Oracle Weblogic Server

Thread Dump Analysis

“Threads stuck in java.net.SocketInputStream.socketRead0”

Copyright Notice: Protection of Intellectual Property

This document, and its contents, is the intellectual property of DigiTalk. It is protected under copyright law and international treaties. Unauthorized use, reproduction, distribution, or resale of this document or any of its content, in whole or in part, is strictly prohibited.

Any infringement of our copyright will result in legal action and may subject the violator to both civil and criminal penalties.

For permissions and inquiries, please contact digitalk.fmw@gmail.com

By accessing or using this document, you agree to abide by these terms and conditions.

Thank you for respecting our intellectual property rights.

DigiTalk

Reach us at digitalk.fmw@gmail.com

DigiTalk Channel: https://www.youtube.com/channel/UCCGTnI9vvF_ETMhGUXGdFWw

Playlists: <https://www.youtube.com/@digitalk.middleware/playlists>

Weblogic Server Architecture: <https://youtu.be/gNqeIfLjUqw>

Oracle Weblogic Server

DigiTalk Udemy Courses and Coupon Code

SOA Suite Administration

<https://www.udemy.com/course/mastering-oracle-soa-suite-12c-administration/?couponCode=739A60915F86847014EB>

Coupon Code: 739A60915F86847014EB

JBoss 8 Administration

<https://www.udemy.com/course/mastering-jboss-eap-8-administration-from-intro-to-advanced/?couponCode=BF65EB008CFE16686BD2>

Coupon Code: BF65EB008CFE16686BD2

OHS Administration

<https://www.udemy.com/course/mastering-oracle-ohs-http-12c-web-server-administration/?couponCode=8E990556B21AF3E1A316>

Coupon Code: 8E990556B21AF3E1A316

Weblogic Server Administration

<https://www.udemy.com/course/oracle-weblogic-server-12c-and-14c-administration/?couponCode=87BC1314AC7690FD5294>

Coupon Code: 87BC1314AC7690FD5294

You can write us on digitalk.fmw@gmail.com if coupon code expired.

Threads stuck in `java.net.SocketInputStream.socketRead0`

What does the `java.net.SocketInputStream.socketRead0()` API do?

Why does it frequently appear in various thread dumps?

Why is it highlighted in thread dump analysis tools?

Should you be concerned about it? And what solutions can address this issue?

Let's delve into these questions.

What does `SocketInputStream.socketRead0()` API do?

Imagine calling your spouse or partner on the phone:

When the call connects, their immediate cheerful response—like "Hello Honey (or darling or sweetie), How are you?"—illustrates an ideal scenario. However, if they're preoccupied with work, picking up kids, or at the gym, their response might be delayed—perhaps just a brief acknowledgment. In more extreme cases, if they're in a bad mood, their reaction could be unpredictable or the call might even hang up. The time you spend waiting from when the call connects until it concludes mirrors the essence of the `socketRead0()` API.

In your application's ecosystem, which interacts with multiple remote applications using protocols like SOAP, REST, HTTP, HTTPS, JDBC, and RMI, all connections pass through the JDK's `java.net` layer for essential TCP-IP and socket operations. Within this layer, the `SocketInputStream.socketRead0()` API is employed to read and receive data from these remote applications. Depending on various factors, such as network conditions and the responsiveness of remote applications, some responses may arrive promptly, while others may exhibit delays or even fail to respond altogether. During the period your application waits to complete reading the response data, its threads can become stuck in the `java.net.SocketInputStream.socketRead0()` API.

Oracle Weblogic Server

Sample Thread dump stacktrace

Here are some example thread dump stack traces that illustrate threads stuck in the `SocketInputStream.socketRead0()` API. Regardless of the protocol used, these threads demonstrate being stuck in `SocketInputStream.socketRead0()`.

RMI thread stuck in `SocketInputStream.socketRead0()` API

```
"RMI TCP Connection(3)-168.xxx.xx.xx" daemon prio=5 tid=0x0000000000008e4560 nid=0x498e20 runnable
[0x0000000000dedec000]
java.lang.Thread.State: RUNNABLE
at java.net.SocketInputStream.socketRead0(Native Method)
at java.net.SocketInputStream.read(Unknown Source)
at java.net.SocketInputStream.read(Unknown Source)
at java.io.BufferedInputStream.fill(Unknown Source)
at java.io.BufferedInputStream.read(Unknown Source)
- locked (0x000000006ad666789) (a java.io.BufferedInputStream)
at java.io.FilterInputStream.read(Unknown Source)
at sun.rmi.transport.tcp.TCPTransport.handleMessages(Unknown Source)
at sun.rmi.transport.tcp.TCPTransport$ConnectionHandler.run0(Unknown Source)
at sun.rmi.transport.tcp.TCPTransport$ConnectionHandler.run(Unknown Source)
at java.util.concurrent.ThreadPoolExecutor.runWorker(Unknown Source)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source)
at java.lang.Thread.run(Unknown Source)
```

Oracle Database connection stuck in `SocketInputStream.socketRead0()` API

```
"Thread-19" id=28 idx=0x6c tid=11456 prio=6 alive, in native, daemon
at jrokit/net/SocketNativeIO.readBytesPinned(Ljava/io/FileDescriptor;[BIII)I(Native Method)
at jrokit/net/SocketNativeIO.socketRead(SocketNativeIO.java:233)
at java/net/SocketInputStream.socketRead0(Ljava/io/FileDescriptor;[BIII)I(SocketInputStream.java)
at java/net/SocketInputStream.read(SocketInputStream.java:433)
```

Oracle Weblogic Server

```
at
java/net/ManagedSocketInputStreamHighPerformanceNew.read(ManagedSocketInputStreamHighPerformanceNew.java:434)
at java/net/SocketInputStream.read(SocketInputStream.java:232)
at
java/net/ManagedSocketInputStreamHighPerformanceNew.read(ManagedSocketInputStreamHighPerformanceNew.java:444)
at oracle/ons/InputBuffer.getNextString(InputBuffer.java:444)
at oracle/ons/ReceiverThread.run(ReceiverThread.java:433)
at jrockit/vm/RNLC2java(JJJJJ)V(Native Method)
```

RabbitMQ stuck in SocketInputStream.socketRead0() API

```
"AMQP Connection 168.xx.xxx.xxx:5554" prio=4 RUNNABLE
java.net.SocketInputStream.socketRead0(Native Method)
java.net.SocketInputStream.socketRead(SocketInputStream.java:333)
java.net.SocketInputStream.read(SocketInputStream.java:234)
java.net.SocketInputStream.read(SocketInputStream.java:232)
java.io.BufferedInputStream.fill(BufferedInputStream.java:234)
java.io.BufferedInputStream.read(BufferedInputStream.java:123)
java.io.DataInputStream.readUnsignedByte(DataInputStream.java:444)
com.rabbitmq.client.impl.Frame.readFrom(Frame.java:543)
com.rabbitmq.client.impl.SocketFrameHandler.readFrame(SocketFrameHandler.java:333)
com.rabbitmq.client.impl.AMQConnection$MainLoop.run(AMQConnection.java:544)
java.lang.Thread.run(Thread.java:555)
```

IBM DB2 statement execution stuck in SocketInputStream.socketRead0() API

```
"Thread-2020 id=234 idx=0x03c tid=234 prio=9 alive, in native, daemon
java.lang.Thread.State: RUNNABLE
at java.net.SocketInputStream.socketRead0(Native Method)
at java.net.SocketInputStream.read(SocketInputStream.java:140)
at com.ibm.db2.jcc.t4.z.b(z.java:1499)
at com.ibm.db2.jcc.t4.z.c(z.java:2189)
at com.ibm.db2.jcc.t4.sb.i(sb.java:1335)
at com.ibm.db2.jcc.am.yn.gb(yn.java:2446)
```

Oracle Weblogic Server

```
at com.ibm.db2.jcc.am.zn.pc(zn.java:3477)
at com.ibm.db2.jcc.am.zn.b(zn.java:5444)
at com.ibm.db2.jcc.am.zn.fc(zn.java:3334)
at com.ibm.db2.jcc.am.zn.execute(zn.java:2453)
at com.ibm.ws.rsadapter.jdbc.WSJdbcPreparedStatement.execute(WSJdbcPreparedStatement.java:618)
```

Solutions

Here are potential solutions to address the issue when a thread gets stuck in the `SocketInputStream.socketRead0` API and fails to recover:

1) Instrument timeout settings

Most applications neglect setting appropriate timeout settings for `SocketInputStream.socketRead0`, leading to prolonged API calls. Implementing timeouts is crucial to prevent prolonged stalls. Here are some timeout settings you can configure:

1.1. JVM Network settings

Configure these timeout properties to manage connections across various protocols using `java.net.URLConnection`:

- `sun.net.client.defaultConnectTimeout`: Timeout for establishing a connection.
- `sun.net.client.defaultReadTimeout`: Timeout for reading data from the input stream.

1.2. `setSoTimeout`

If directly programming with Sockets, utilize `setSoTimeout()` to set a timeout in milliseconds on the socket. This throws `java.net.SocketTimeoutException` if the remote application doesn't respond within the specified time.

1.3. JDBC

For JDBC connections, use `setQueryTimeout()` to specify the number of seconds the JDBC driver should wait for a query to complete before throwing `SQLException`.

Oracle Weblogic Server

1.4. Oracle JDBC

Configure `-Doracle.jdbc.ReadTimeout` during application startup to set a timeout in milliseconds for Oracle database connections.

Validate Network connectivity

Threads failing to recover from `SocketInputStream.socketRead0` API can also stem from problems with network connectivity or issues with load balancers. In some cases, the remote application may not send the necessary ACK or FIN packets, causing delays or stalls. Engaging network engineers or cloud hosting support teams is advisable to diagnose and resolve these issues.

On your part, using TCP/IP tracing tools like Wireshark can be invaluable. These tools allow you to analyze network packets exchanged between your application and the remote server. This analysis helps pinpoint whether the network issue resides on your end or on the remote application's side, facilitating targeted troubleshooting and resolution.

Collaborate with the remote application

In some cases, transaction slowdowns may arise due to performance issues within the remote application. In such situations, it's essential to communicate these slowdowns to the remote application team, raising awareness of the issue and collaborating with them to implement necessary fixes.

Non-blocking HTTP client

Consider utilizing non-blocking HTTP client libraries such as Grizzly or Netty, which avoid blocking operations that can cause threads to hang. Implementing this solution requires strategic code changes and thorough testing.

Please note, while this list is comprehensive, it may not encompass all potential solutions.

If you have additional timeout settings or solutions to suggest, please provide your feedback below. We welcome your recommendations and will update this information accordingly.

digitalkfmw@gmail.com



Oracle Weblogic Server

DISCLAIMER AND CONSENT

This document is being provided by DigiTalk as part of its effort to assist users in understanding and working with Oracle WebLogic Server. The Company wishes to emphasize that this document is not affiliated with Oracle Corporation ("Oracle") in any way, and the content contained herein is based solely on publicly available product documentation provided by Oracle.

While every effort has been made to ensure the accuracy and reliability of the information presented in this document, there is a possibility of typographical errors or inaccuracies. DigiTalk does not guarantee the correctness or completeness of the content provided in this document.

Users of this document are encouraged to cross-reference the information presented here with Oracle's official documentation available on their website or other authoritative sources. Any discrepancies or inaccuracies found in this document should be reported to us at digitalk.fmw@gmail.com.

By using this document, you acknowledge and consent to the following:

This document is not officially endorsed or verified by Oracle.

The Company makes no claims or guarantees about the accuracy or suitability of the information contained in this document.

Users are responsible for verifying and validating any information presented here for their specific use case.

DigiTalk disclaims any liability for any errors, omissions, or damages that may result from the use of this document.

If you discover any inaccuracies or errors in this document, please report them to digitalk.fmw@gmail.com, and the Company will endeavor to correct them as necessary.

This consent statement is provided to ensure transparency and understanding of the limitations of the information contained in this document. By using this document, you agree to abide by the terms and conditions outlined herein.