Google Cloud Sprint Bootcamp

Team Members

1. Alok Bhatt
2. Vibhuti Talekar
3. M M Kishore.
4. Pragya Kimothi
5. Sapna Bisht

Google Cloud Sprint Bootcamp: India 2022

Week 2 | Assignment 2

Under the Guidance

of

Jason JunChul

Ques1. Follow the below steps and answer the questions :

a. **Create a java file in Linux environment : vi SyncExec.java**
b. **Add code [1] to the above file.**
c. **Compile the above file : javac SyncExec.java**
d. **Execute the code : java SyncExec**
   - **How will you troubleshoot the issue this code runs into ? Please share an approach.**
   - **What different commands can you use to infer the behavior of the issue.**
   - **Please share the RCA of the issue**

```
public class SyncExec {
 public static Object Lck1 = new Object();
 public static Object Lck2 = new Object();

 public static void main(String args[]) {
   SyncThread1 T1 = new SyncThread1();
   SyncThread2 T2 = new SyncThread2();
   T1.start();
   T2.start();
 }

 private static class SyncThread1 extends Thread {
   public void run() {
     synchronized (Lck1) {
       System.out.println("Thread 1: Holding lock 1...");

       try { Thread.sleep(10); }
       catch (InterruptedException e) {}
       System.out.println("Thread 1: Waiting for lock 2...");

       synchronized (Lck2) {
         System.out.println("Thread 1: Holding lock 1 & 2...");
       }
     }
   }
 }
 private static class SyncThread2 extends Thread {
   public void run() {
     synchronized (Lck2) {
       System.out.println("Thread 2: Holding lock 2...");

      try { Thread.sleep(10); }
       catch (InterruptedException e) {}
       System.out.println("Thread 2: Waiting for lock 1...");

       synchronized (Lck1) {
         System.out.println("Thread 2: Holding lock 1 & 2...");
       }
     }
   }
 }
```

}

Observation

On running the program we can observe that Thread1 and Thread2 acquires lock1 and lock2 simultaneously and both are waiting for the second lock. The program is not getting completed and to stop the program “ctrl + d”(windows) or “ctrl + z”(linux) depending on the operating system.

Erroneous output

```
wolverine@wolverine-HP-Laptop-15-da0077tx:~/Documents/Google/cloud_sprint-22$ java SyncExec
Thread 1: Holding lock 1...
Thread 2: Holding lock 2...
Thread 1: Waiting for lock 2...
Thread 2: Waiting for lock 1...
```

Debugging

Find the PID(process ID) of the java program.

Make sure the program is running in the background

```
ps aux
```

The name of the java class “SyncExec” is visible in the output from the “ps aux” command
We can also use pipe ‘|’ operator to filter out unwanted processes. For example:

```
ps aux | grep "java"
```

Using the process name(last column) corresponding PID(second column) can be found.
Note the PID for further processes**.**

Output

```
wolverine@wolverine-HP-Laptop-15-da0077tx:~/Documents/Google/cloud_sprint-22$ ps aux | tail -15
root        8114  0.0  0.0      0     0 ?        I    08:01   0:00 [kworker/5:2]
wolveri+    8120  0.0  0.8 2423852 69240 ?       Sl   08:02   0:00 /usr/lib/firefox/firefox-bin -contentproc -childID 48 -isForBrowser
 4350 true tab
wolveri+    8168  0.0  0.7 2411988 58424 ?       Sl   08:02   0:00 /usr/lib/firefox/firefox-bin -contentproc -childID 49 -isForBrowser
 4350 true tab
wolveri+    8199  0.1  0.3 4852560 25896 pts/1   Sl+  08:02   0:00 java SyncExec
wolveri+    8224  0.0  0.0   9836  4032 pts/2    Ss   08:03   0:00 /bin/bash
root        8304  0.0  0.0      0     0 ?        I    08:04   0:00 [kworker/2:3-events]
root        8305  0.0  0.0      0     0 ?        I    08:04   0:00 [kworker/2:4]
wolveri+    8313  0.0  0.7 2411988 58520 ?       Sl   08:04   0:00 /usr/lib/firefox/firefox-bin -contentproc -childID 50 -isForBrowser
```

Using Jstack

Using Jstack we can analyze the stack trace to find out what could be the problem and why the program isn't getting terminated. This command is recommended for debugging java-based processes.

Replace <pid> with the process id provided retrieved form "Process Status" or "ps" command.

```
jstack <pid>
```

Example:

```
jstack 8199
```

Output

```
Found one Java-level deadlock:
=============================
"Thread-1":
  waiting to lock monitor 0x00007f86a0003ac8 (object 0x00000000d7072c58, a java.lang.Object),
  which is held by "Thread-0"
"Thread-0":
  waiting to lock monitor 0x00007f86a0006568 (object 0x00000000d7072c68, a java.lang.Object),
  which is held by "Thread-1"

Java stack information for the threads listed above:
===================================================
"Thread-1":
        at SyncExec$SyncThread2.run(SyncExec.java:37)
        - waiting to lock <0x00000000d7072c58> (a java.lang.Object)
        - locked <0x00000000d7072c68> (a java.lang.Object)
"Thread-0":
        at SyncExec$SyncThread1.run(SyncExec.java:22)
        - waiting to lock <0x00000000d7072c68> (a java.lang.Object)
        - locked <0x00000000d7072c58> (a java.lang.Object)

Found 1 deadlock.
```

Deadlock detected !

another way to detect:

Using strace

Replace <pid> with the process id provided retrieved form "Process Status" or "ps" command.

```
strace -p <pid>
```

Example:

```
strace -p 9067
```

Output

```
wolverine@wolverine-HP-Laptop-15-da0077tx:~/Documents/Google/cloud_sprint-22$ sudo strace -p 9067
strace: Process 9067 attached
futex(0x7f79a948a9d0, FUTEX_WAIT, 9068, NULL
```

The futex() system call provides a method for waiting until a certain condition becomes true.

The program is in wait state !

What is Happening?

- The code seems to be structured properly but there is an issue in acquiring the locks.
- Thread1 acquires lock 1 first and Thread2 acquires lock2 first..
- Thread1 waits for lock2 whereas Thread2 waits for lock1 to enter into their corresponding critical state.
- Since both threads are not ready to release their initial locks.

The deadlock may not occur in all situations.For example: If a computer is very slow to process Thread2, Thread1 will get executed first by acquiring and releasing both locks before Thread2 begins its execution, so that deadlock does not occour.

Rectifying deadlock

The lock needs to be acquired in a sequence. We are acquiring the locks in the following order:

1. Lck1
2. Lck2

Therefore when Thread 1/ Thread 2 acquires lock 1, the other Thread has to wait until the critical section accessed by the previous Thread is completed. This sequence of acquiring locks make sure deadlock doesn't happen during execution of the program.

New Code -

```
public class SyncExec {
  public static Object Lck1 = new Object();
  public static Object Lck2 = new Object();

  public static void main(String args[]) {
     SyncThread1 T1 = new SyncThread1();
     SyncThread2 T2 = new SyncThread2();
     T1.start();
     T2.start();
  }

  private static class SyncThread1 extends Thread {
     public void run() {
        synchronized (Lck1) {
           System.out.println("Thread 1: Holding lock 1...");

           try { Thread.sleep(10); }
```

```
            catch (InterruptedException e) {}
            System.out.println("Thread 1: Waiting for lock 2...");

            synchronized (Lck2) {
               System.out.println("Thread 1: Holding lock 1 & 2...");
            }
         }
      }
   }
   private static class SyncThread2 extends Thread {
      public void run() {
         synchronized (Lck1) {
            System.out.println("Thread 2: Holding lock 1...");

            try { Thread.sleep(10); }
            catch (InterruptedException e) {}
            System.out.println("Thread 2: Waiting for lock 1...");

            synchronized (Lck2) {
               System.out.println("Thread 2: Holding lock 1 & 2...");
            }
         }
      }
   }
}
```

Output

```
wolverine@wolverine-HP-Laptop-15-da0077tx:~/Documents/Google/cloud_sprint-22$ java SyncExec
Thread 1: Holding lock 1...
Thread 1: Waiting for lock 2...
Thread 1: Holding lock 1 & 2...
Thread 2: Holding lock 1...
Thread 2: Waiting for lock 1...
Thread 2: Holding lock 1 & 2...
```

No more Deadlocks :)

Ques 2.How will you troubleshoot a situation when a user is not able to SSH to a remote VM. What are the possible scenarios this issue can occur in ?

A user is unable to ssh into remote VM due to the below scenarios taking place:

1.SSH client is not installed

Before diving into troubleshooting,we must ensure that both our client as well as server must have OpenSSH client installed.If it is absent in either of the client or server,or in case absent in both there is no way any user will be able to ssh into the remote VM.

Possible Solution:
To install SSH client on the remote server as well as the client(depending on the situation).

2.Incorrect Credentials

SSH connections can be refused when the user mistypes the Username and Password.Incorrect IP addresses can also cause this error.SSH port not open or incorrect SSH port open can also cause error.

Possible Solution:
Ensure that the password ,username and ip address is correctly typed.

3.SSH service Down

When a user tries to ssh to a remote VM ,the SSH service of the VM could be down.It can checked using command

sudo service ssh status

Possible Solution:
Enable SSH Service using command like
systemctl start sshd

4.Firewall preventing SSH Connection

Firewall protects the host from hazardous connections.SSH can refuse connections due to firewall restrictions.

Possible solution:
Allow SSH connection through Firewall with the correct port number.

5.SSH port is Closed

If SSH port is closed or if it's state is not LISTEN then the server will refuse connection

Possible solution:
Open the ssh port

6.Passwordless Remote Server File Permission

When the user has passwordless login enabled a public key authentication is used to SSH into an instance.This requires the public key to be uploaded into the remote server.If the public key is uploaded incorrectly it can cause errors.

Possible solutions:
Ensure correct keys are uploaded to the correct server.
Edit the file permissions using chmod command
-Set permissions 700 to .ssh directory
chmod 700 .ssh
-Set permissions 640 to .ssh/authorised_keys directory
chmod 640 .ssh/authorised_keys

7.Remote host Identification has changed

This can happen when a user has connected to the VM before or in the past but the VM got reformatted or has changed its configurations in the meantime.During reformatting the host

key changes or is destroyed ,a new host key is generated due to this SSH can't remember the connection.

8.Server closed network connection unexpectedly

This can occur when user IP is blocked,User is using a proxy server or VPN

9.User does not have permissions

When the user is a regular user or does not belong to the sudoers group certain VM's might refuse connection via SSH.

10.Server Idle timeout

Sometimes the user is prompted to input their password but before they respond the server closes connection.This can occur when the IdleTimeout is set to a smaller value due to which it closes connection after a very short duration.

11.VM boot disk is full

When SSH connection is established the guest environment adds the session's public ssh to the authorized_keys file.But if the disk is full connection will fail

12.VM is booting in maintenance mode

When booting in maintenance mode,VM doesn't accept SSH connections.

13.SSH might fail after upgrade VM's kernel

A VM might experience kernel panic after kernel update causing VM to be inaccessible.

Ques 3. A web server connects to a MySQL Database and serves data to users. Users are noticing high latency in response times. How will you debug this scenario to find the cause for slowness?

Please note that

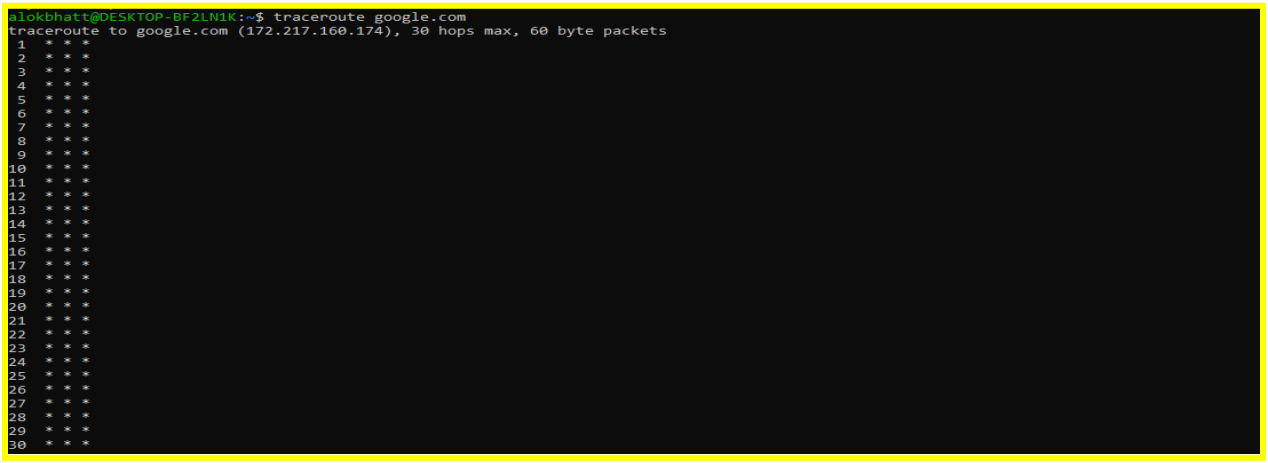
There have been no configuration changes and the issue started occurring recently.

Step 1- Check the time taken to connect to the server

$ curl -s -o /dev/null -w '%{time_connect}\n' www. Google.com

Step 2- If latency is found then using traceroute command check which hop is having high latency.

$traceroute www.google.com



<u>If no found with high latency then other possible reasons for latency might be -</u>

Client has low bandwidth - The server is responding quickly but the response might be slowed down by network bottleneck or it slowed down because client has low bandwidth.

No Configuration Upgradation - As there is no configuration change , it might be possible database is filling up and require more storage as the number of queries are increasing.
As soon as the traffic is increasing it results in slow response for the queries.