HBASE-22041

xiaoheipangzi

2019-03-12

Contents

Chapter 1

Root issue HBASE-22041

1.1 Summary

[k8s] The crashed node exists in onlineServer forever, and if it holds the meta data, master will start up hang.

1.2 Description

while master fresh boot, we crash (kill- 9) the RS who hold meta. we find that the master startup fails and print thounds of logs like:

```
1  2019-03-13 01:09:54,896 WARN [RSProcedureDispatcher-pool4-t1] procedure.
       RSProcedureDispatcher: request to server hadoop14,16020,1552410583724 failed due to
       java.net.ConnectException: Call to hadoop14/172.16.1.131:16020 failed on connection
       exception: org.apache.hbase.thirdparty.io.netty.channel.AbstractChannel$
       AnnotatedConnectException: syscall:getsockopt(..) failed: Connection refused:
       hadoop14/172.16.1.131:16020, try=0, retrying...
2  2019-03-13 01:09:55,004 WARN [RSProcedureDispatcher-pool4-t2] procedure.
       RSProcedureDispatcher: request to server hadoop14,16020,1552410583724 failed due to
       org.apache.hadoop.hbase.ipc.FailedServerException: Call to hadoop14
       /172.16.1.131:16020 failed on local exception: org.apache.hadoop.hbase.ipc.
       FailedServerException: This server is in the failed servers list: hadoop14
       /172.16.1.131:16020, try=1, retrying...
3  2019-03-13 01:09:55,114 WARN [RSProcedureDispatcher-pool4-t3] procedure.
       RSProcedureDispatcher: request to server hadoop14,16020,1552410583724 failed due to
       org.apache.hadoop.hbase.ipc.FailedServerException: Call to hadoop14
       /172.16.1.131:16020 failed on local exception: org.apache.hadoop.hbase.ipc.
       FailedServerException: This server is in the failed servers list: hadoop14
       /172.16.1.131:16020, try=2, retrying...
4  2019-03-13 01:09:55,219 WARN [RSProcedureDispatcher-pool4-t4] procedure.
       RSProcedureDispatcher: request to server hadoop14,16020,1552410583724 failed due to
       org.apache.hadoop.hbase.ipc.FailedServerException: Call to hadoop14
       /172.16.1.131:16020 failed on local exception: org.apache.hadoop.hbase.ipc.
       FailedServerException: This server is in the failed servers list: hadoop14
       /172.16.1.131:16020, try=3, retrying...
5  2019-03-13 01:09:55,324 WARN [RSProcedureDispatcher-pool4-t5] procedure.
       RSProcedureDispatcher: request to server hadoop14,16020,1552410583724 failed due to
       org.apache.hadoop.hbase.ipc.FailedServerException: Call to hadoop14
       /172.16.1.131:16020 failed on local exception: org.apache.hadoop.hbase.ipc.
       FailedServerException: This server is in the failed servers list: hadoop14
       /172.16.1.131:16020, try=4, retrying...
6  2019-03-13 01:09:55,428 WARN [RSProcedureDispatcher-pool4-t6] procedure.
       RSProcedureDispatcher: request to server hadoop14,16020,1552410583724 failed due to
       org.apache.hadoop.hbase.ipc.FailedServerException: Call to hadoop14
       /172.16.1.131:16020 failed on local exception: org.apache.hadoop.hbase.ipc.
```

```
          FailedServerException: This server is in the failed servers list: hadoop14
          /172.16.1.131:16020, try=5, retrying...
7  2019-03-13 01:09:55,533 WARN [RSProcedureDispatcher-pool4-t7] procedure.
          RSProcedureDispatcher: request to server hadoop14,16020,1552410583724 failed due to
          org.apache.hadoop.hbase.ipc.FailedServerException: Call to hadoop14
          /172.16.1.131:16020 failed on local exception: org.apache.hadoop.hbase.ipc.
          FailedServerException: This server is in the failed servers list: hadoop14
          /172.16.1.131:16020, try=6, retrying...
8  2019-03-13 01:09:55,638 WARN [RSProcedureDispatcher-pool4-t8] procedure.
          RSProcedureDispatcher: request to server hadoop14,16020,1552410583724 failed due to
          org.apache.hadoop.hbase.ipc.FailedServerException: Call to hadoop14
          /172.16.1.131:16020 failed on local exception: org.apache.hadoop.hbase.ipc.
          FailedServerException: This server is in the failed servers list: hadoop14
          /172.16.1.131:16020, try=7, retrying...
9  2019-03-13 01:09:55,755 WARN [RSProcedureDispatcher-pool4-t9] procedure.
          RSProcedureDispatcher: request to server hadoop14,16020,1552410583724 failed due to
          org.apache.hadoop.hbase.ipc.FailedServerException: Call to hadoop14
          /172.16.1.131:16020 failed on local exception: org.apache.hadoop.hbase.ipc.
          FailedServerException: This server is in the failed servers list: hadoop14
          /172.16.1.131:16020, try=8, retrying...
```

1.3   Attachments

1. bug.zip

2. hbasemaster.log

3. normal.zip

1.4   Comments

1. **xiaoheipangzi:** We have four machine, HMaster is hadoop11, the RegionServers are :hadoop12,hadoop13,hadoop14

   When we start master, who is on a busy machine. So thethe RegionServerTracker#refresh become slow. BeforeRegionServerTracker#refresh detect the hadoop14 join the cluster, hadoop14 crash, but hadoop14 is aleardy added to ServerManager#onlineSever(see below log#3). So even hadoop14 crashes, it exist in onlineServer for ever.

```
1
2
3
4  1 master.ServerManager: Registering regionserver=hadoop14,16020,1552410583724
5
6   2 master.ServerManager: Registering regionserver=hadoop12,16020,1552410578454
7
8   3 master.ServerManager: Registering regionserver=hadoop13,16020,1552410566504
9
10   4 zookeeper.MetaTableLocator: Setting hbase:meta (replicaId=0) location in
         ZooKeeper as hadoop14,16020,1552410583724
11
12
13
14   5 master.RegionServerTracker: RegionServer ephemeral node created, adding [
         hadoop12,16020,1552410578454]
15
```

```
16   6 master.RegionServerTracker: RegionServer ephemeral node created, adding [
         hadoop13,16020,1552410566504]
17
18   7 procedure.RSProcedureDispatcher: request to server hadoop14,16020,1552410583724
         failed due to java.net.ConnectException: Call to hadoop14/172.16.1.131:16020
         failed on connection exception: org.apache.hbase.thirdparty.io.netty.channel.
         AbstractChannel$AnnotatedConnectException: syscall:getsockopt(..) failed:
         Connection refused: hadoop14/172.16.1.131:16020, try=0, retrying...
```

This can cause the master hangs and print log forever, if hadoop14 hold the meta table.

Seeorg.apache.hadoop.hbase.master.procedure.RSProcedureDispatcher.ExecuteProceduresRemoteCall.run

```
1
2   1 try {
3
4   2   sendRequest(getServerName(), request.build());
5
6   3 } catch (IOException e) {
7
8   4   e = unwrapException(e);
9
10  5   // TODO: In the future some operation may want to bail out early.
11
12  6   // TODO: How many times should we retry (use numberOfAttemptsSoFar)
13
14  7   if (!scheduleForRetry(e)) {
15
16  8     remoteCallFailed(procedureEnv, e);
17
18  9   }
19
20  10 }
```

master will sendReust to hadoop14 and fails, so it will callscheduleForRetry to retry , and
inscheduleForRetry , master will check whether hadoop14 is inonlineServers , if is, retry,
hence master will retry forever and print thousands of logs like log #7.

2. **xiaoheipangzi:** [~Apache9] and [~allan163]

Could please see this issue and give somesuggestion to fix it?

4

My personal option is give athreshold for numberOfAttemptsSoFar(like the TODO in above code).

Thanks.

3. **stack:** Does the Master not get notice of hadoop14 expiring so it can run a ServerCrash-Procedure for it? Thanks.

4. **stack:** Looking in log, this looks to be HBase 3.0.0-SNAPSHOT .... master branch build?

5. **timoha:** We are seeing the same issue on 2.2.4 running in kubernetes. The issue appears to be do to with the fact that address of failed regionserver keeps getting readded to failed servers list when RSProcedureDispatcher.sendRequest is called.

Here's with TRACE logging enabled:

```
2020-05-18 17:52:19,643 TRACE [RSProcedureDispatcher-pool3-t127] procedure.
    RSProcedureDispatcher: Building request with operations count=1

2020-05-18 17:52:19,644 DEBUG [RSProcedureDispatcher-pool3-t127] ipc.
    AbstractRpcClient: Not trying to connect to regionserver-1.hbase.hbase.svc.
    cluster.local/10.128.9.13:16020 this server is in the failed servers list

2020-05-18 17:52:19,644 TRACE [RSProcedureDispatcher-pool3-t127] ipc.
    AbstractRpcClient: Call: ExecuteProcedures, callTime: 0ms

2020-05-18 17:52:19,644 DEBUG [RSProcedureDispatcher-pool3-t127] procedure.
    RSProcedureDispatcher: request to regionserver-1.hbase.hbase.svc.cluster.local
    ,16020,1589824187906 failed, try=1474

org.apache.hadoop.hbase.ipc.FailedServerException: Call to regionserver-1.hbase.
    hbase.svc.cluster.local/10.128.9.13:16020 failed on local exception: org.apache
    .hadoop.hbase.ipc.FailedServerException: This server is in the failed servers
    list: regionserver-1.hbase.hbase.svc.cluster.local/10.128.9.13:16020

 at sun.reflect.GeneratedConstructorAccessor13.newInstance(Unknown Source)

 at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(
     DelegatingConstructorAccessorImpl.java:45)

 at java.lang.reflect.Constructor.newInstance(Constructor.java:423)

 at org.apache.hadoop.hbase.ipc.IPCUtil.wrapException(IPCUtil.java:220)

 at org.apache.hadoop.hbase.ipc.AbstractRpcClient.onCallFinished(AbstractRpcClient.
     java:392)

 at org.apache.hadoop.hbase.ipc.AbstractRpcClient.access$100(AbstractRpcClient.java
     :97)

 at org.apache.hadoop.hbase.ipc.AbstractRpcClient$3.run(AbstractRpcClient.java:423)

 at org.apache.hadoop.hbase.ipc.AbstractRpcClient$3.run(AbstractRpcClient.java:419)
```

```
28   at org.apache.hadoop.hbase.ipc.Call.callComplete(Call.java:117)
29
30   at org.apache.hadoop.hbase.ipc.Call.setException(Call.java:132)
31
32   at org.apache.hadoop.hbase.ipc.AbstractRpcClient.callMethod(AbstractRpcClient.java
         :436)
33
34   at org.apache.hadoop.hbase.ipc.AbstractRpcClient.callBlockingMethod(
         AbstractRpcClient.java:330)
35
36   at org.apache.hadoop.hbase.ipc.AbstractRpcClient.access$200(AbstractRpcClient.java
         :97)
37
38   at org.apache.hadoop.hbase.ipc.AbstractRpcClient$BlockingRpcChannelImplementation.
         callBlockingMethod(AbstractRpcClient.java:585)
39
40   at org.apache.hadoop.hbase.shaded.protobuf.generated.AdminProtos$AdminService$
         BlockingStub.executeProcedures(AdminProtos.java:31006)
41
42   at org.apache.hadoop.hbase.master.procedure.RSProcedureDispatcher$
         ExecuteProceduresRemoteCall.sendRequest(RSProcedureDispatcher.java:349)
43
44   at org.apache.hadoop.hbase.master.procedure.RSProcedureDispatcher$
         ExecuteProceduresRemoteCall.run(RSProcedureDispatcher.java:314)
45
46   at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
47
48   at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
49
50   at java.lang.Thread.run(Thread.java:748)
51
52 Caused by: org.apache.hadoop.hbase.ipc.FailedServerException: This server is in the
         failed servers list: regionserver-1.hbase.hbase.svc.cluster.local
         /10.128.9.13:16020
53
54   at org.apache.hadoop.hbase.ipc.AbstractRpcClient.getConnection(AbstractRpcClient.
         java:354)
55
56   at org.apache.hadoop.hbase.ipc.AbstractRpcClient.callMethod(AbstractRpcClient.java
         :433)
57
58   ... 9 more
59
60 2020-05-18 17:52:19,644 WARN [RSProcedureDispatcher-pool3-t127] procedure.
         RSProcedureDispatcher: request to server regionserver-1.hbase.hbase.svc.cluster
         .local,16020,1589824187906 failed due to org.apache.hadoop.hbase.ipc.
         FailedServerException: Call to regionserver-1.hbase.hbase.svc.cluster.local
         /10.128.9.13:16020 failed on local exception: org.apache.hadoop.hbase.ipc.
         FailedServerException: This server is in the failed servers list: regionserver
         -1.hbase.hbase.svc.cluster.local/10.128.9.13:16020, try=1474, retrying...
```

In our case it doesn't recover automatically and we have to restart hbase master to get out of this issue.

6. **timoha:** Here's the error that shows that the address is getting readded to failed servers:

```
1
2 2020-05-18 17:52:20,249 TRACE [RSProcedureDispatcher-pool3-t133] procedure.
         RSProcedureDispatcher: Building request with operations count=1
3
4 2020-05-18 17:52:20,249 TRACE [RSProcedureDispatcher-pool3-t133] ipc.
         NettyRpcConnection: Connecting to regionserver-1.hbase.hbase.svc.cluster.local
```

6

```
          /10.128.9.13:16020
 5

 6  2020-05-18 17:52:20,254 TRACE [RS-EventLoopGroup-1-1] ipc.AbstractRpcClient: Call:
          ExecuteProcedures, callTime: 5ms

 7

 8  2020-05-18 17:52:20,254 DEBUG [RS-EventLoopGroup-1-1] ipc.FailedServers: Added
          failed server with address regionserver-1.hbase.hbase.svc.cluster.local
          /10.128.9.13:16020 to list caused by org.apache.hbase.thirdparty.io.netty.
          channel.AbstractChannel$AnnotatedConnectException: finishConnect(..) failed: No
           route to host: regionserver-1.hbase.hbase.svc.cluster.local/10.128.9.13:16020

 9

10  2020-05-18 17:52:20,254 DEBUG [RSProcedureDispatcher-pool3-t133] procedure.
          RSProcedureDispatcher: request to regionserver-1.hbase.hbase.svc.cluster.local
          ,16020,1589824187906 failed, try=1480

11

12  2020-05-18 17:52:20,255 WARN [RSProcedureDispatcher-pool3-t133] procedure.
          RSProcedureDispatcher: request to server regionserver-1.hbase.hbase.svc.cluster
          .local,16020,1589824187906 failed due to java.net.ConnectException: Call to
          regionserver-1.hbase.hbase.svc.cluster.local/10.128.9.13:16020 failed on
          connection exception: org.apache.hbase.thirdparty.io.netty.channel.
          AbstractChannel$AnnotatedConnectException: finishConnect(..) failed: No route
          to host: regionserver-1.hbase.hbase.svc.cluster.local/10.128.9.13:16020, try
          =1480, retrying...

13

14   ... 10 more

15

16  Caused by: java.net.ConnectException: finishConnect(..) failed: No route to host

17

18   ... 6 more

19

20   at org.apache.hbase.thirdparty.io.netty.channel.epoll.AbstractEpollChannel$
          AbstractEpollUnsafe.finishConnect(AbstractEpollChannel.java:644)

21

22   at org.apache.hbase.thirdparty.io.netty.channel.epoll.AbstractEpollChannel$
          AbstractEpollUnsafe.doFinishConnect(AbstractEpollChannel.java:667)

23

24   at org.apache.hbase.thirdparty.io.netty.channel.unix.Socket.finishConnect(Socket.
          java:269)

25

26   at org.apache.hbase.thirdparty.io.netty.channel.unix.Errors.throwConnectException(
          Errors.java:112)

27

28  Caused by: org.apache.hbase.thirdparty.io.netty.channel.AbstractChannel$
          AnnotatedConnectException: finishConnect(..) failed: No route to host:
          regionserver-1.hbase.hbase.svc.cluster.local/10.128.9.13:16020

29

30   at java.lang.Thread.run(Thread.java:748)

31

32   at org.apache.hbase.thirdparty.io.netty.util.concurrent.FastThreadLocalRunnable.
           run(FastThreadLocalRunnable.java:30)

33

34   at org.apache.hbase.thirdparty.io.netty.util.concurrent.SingleThreadEventExecutor$
          5.run(SingleThreadEventExecutor.java:905)

35

36   at org.apache.hbase.thirdparty.io.netty.channel.epoll.EpollEventLoop.run(
          EpollEventLoop.java:328)

37

38   at org.apache.hbase.thirdparty.io.netty.channel.epoll.EpollEventLoop.processReady(
          EpollEventLoop.java:417)

39

40   at org.apache.hbase.thirdparty.io.netty.channel.epoll.AbstractEpollChannel$
          AbstractEpollUnsafe.epollOutReady(AbstractEpollChannel.java:524)

41

42   at org.apache.hbase.thirdparty.io.netty.channel.epoll.AbstractEpollChannel$
          AbstractEpollUnsafe.finishConnect(AbstractEpollChannel.java:650)

43

44   at org.apache.hbase.thirdparty.io.netty.channel.epoll.AbstractEpollChannel$
```

```
         AbstractEpollUnsafe.fulfillConnectPromise(AbstractEpollChannel.java:631)
45
46   at org.apache.hbase.thirdparty.io.netty.util.concurrent.DefaultPromise.tryFailure(
         DefaultPromise.java:114)
47
48   at org.apache.hbase.thirdparty.io.netty.util.concurrent.DefaultPromise.setFailure0
         (DefaultPromise.java:533)
49
50   at org.apache.hbase.thirdparty.io.netty.util.concurrent.DefaultPromise.setValue0(
         DefaultPromise.java:540)
51
52   at org.apache.hbase.thirdparty.io.netty.util.concurrent.DefaultPromise.
         notifyListeners(DefaultPromise.java:415)
53
54   at org.apache.hbase.thirdparty.io.netty.util.concurrent.DefaultPromise.
         notifyListenersNow(DefaultPromise.java:474)
55
56   at org.apache.hbase.thirdparty.io.netty.util.concurrent.DefaultPromise.
         notifyListeners0(DefaultPromise.java:495)
57
58   at org.apache.hbase.thirdparty.io.netty.util.concurrent.DefaultPromise.
         notifyListener0(DefaultPromise.java:502)
59
60   at org.apache.hadoop.hbase.ipc.NettyRpcConnection$3.operationComplete(
         NettyRpcConnection.java:261)
61
62   at org.apache.hadoop.hbase.ipc.NettyRpcConnection$3.operationComplete(
         NettyRpcConnection.java:267)
63
64   at org.apache.hadoop.hbase.ipc.NettyRpcConnection.access$500(NettyRpcConnection.
         java:71)
65
66   at org.apache.hadoop.hbase.ipc.NettyRpcConnection.failInit(NettyRpcConnection.java
         :179)
67
68   at org.apache.hbase.thirdparty.io.netty.channel.DefaultChannelPipeline.
         fireUserEventTriggered(DefaultChannelPipeline.java:924)
69
70   at org.apache.hbase.thirdparty.io.netty.channel.AbstractChannelHandlerContext.
         invokeUserEventTriggered(AbstractChannelHandlerContext.java:312)
71
72   at org.apache.hbase.thirdparty.io.netty.channel.AbstractChannelHandlerContext.
         invokeUserEventTriggered(AbstractChannelHandlerContext.java:326)
73
74   at org.apache.hbase.thirdparty.io.netty.channel.DefaultChannelPipeline$HeadContext
         .userEventTriggered(DefaultChannelPipeline.java:1426)
75
76   at org.apache.hbase.thirdparty.io.netty.channel.AbstractChannelHandlerContext.
         fireUserEventTriggered(AbstractChannelHandlerContext.java:304)
77
78   at org.apache.hbase.thirdparty.io.netty.channel.AbstractChannelHandlerContext.
         invokeUserEventTriggered(AbstractChannelHandlerContext.java:312)
79
80   at org.apache.hbase.thirdparty.io.netty.channel.AbstractChannelHandlerContext.
         invokeUserEventTriggered(AbstractChannelHandlerContext.java:326)
81
82   at org.apache.hadoop.hbase.ipc.BufferCallBeforeInitHandler.userEventTriggered(
         BufferCallBeforeInitHandler.java:92)
83
84   at org.apache.hadoop.hbase.ipc.Call.setException(Call.java:132)
85
86   at org.apache.hadoop.hbase.ipc.Call.callComplete(Call.java:117)
87
88   at org.apache.hadoop.hbase.ipc.AbstractRpcClient$3.run(AbstractRpcClient.java:419)
89
90   at org.apache.hadoop.hbase.ipc.AbstractRpcClient$3.run(AbstractRpcClient.java:423)
91
```

```
92   at org.apache.hadoop.hbase.ipc.AbstractRpcClient.access$100(AbstractRpcClient.java
        :97)

93

94   at org.apache.hadoop.hbase.ipc.AbstractRpcClient.onCallFinished(AbstractRpcClient.
        java:392)

95

96   at org.apache.hadoop.hbase.ipc.IPCUtil.wrapException(IPCUtil.java:177)

97

98  java.net.ConnectException: Call to regionserver-1.hbase.hbase.svc.cluster.local
        /10.128.9.13:16020 failed on connection exception: org.apache.hbase.thirdparty.
        io.netty.channel.AbstractChannel$AnnotatedConnectException: finishConnect(..)
        failed: No route to host: regionserver-1.hbase.hbase.svc.cluster.local
        /10.128.9.13:16020
```

These errors happen every 2 seconds (which is the default expiry of failed servers)

7. **timoha:** Just reproduced again and I'm seeing ServerCrashProcedure being stuck for the regionserver that it's trying to reconnect to with state=WAITING:SERVER_CRASH_FINISH.

And, ServerCrashProcedure is waiting for TransitRegionStateProcedure procedure with state=WAITING:REGION_

And, TransitRegionStateProcedure is waiting for OpenRegionProcedure procedure with state=RUNNABLE

Regions in transition are in OPENNING state for the regionserver that exists.

If I'm understanding the logs correctly, it's trying to connect to old IP address for the restarted regionserver. In kubernetes when pod is restarted it gets a new IP address and preserves hostname (if it's a statefulset). So, there's some assumption somewhere in HBase that IP address doesn't change or it caches the IP address resolution. In this particular case it looks like it's trying to correctly assign regions to the online regionserver but still uses old IP address.

8. **stack:** Thanks for the detail [~timoha]. Haven't looked at code yet.

We first report it as being in the failed server list and then we start doing 'No route to host'. It starts after the container comes back w/ new IP (looks the same though across textboxes)? What are the dns timeouts on this host? (networkaddress.cache.ttl). We should give up if 'no route to host' for sure.

9. **timoha:** > It starts after the container comes back w/ new IP (looks the same though across textboxes)?

Right, the pod starts with new IP address, but for some reason master is still trying to reach old IP address

> What are the dns timeouts on this host?

We set -Dsun.net.inetaddr.ttl=10 option and it seems to work for other places. Is it necessary to set "networkaddress.cache.tt" option as well?

10. **stack:** Can I have more log [~timoha]? The "Here's the error that shows that the address is getting readded to failed servers" seems to be the hbase async dfs client. Looking at the first exception, tracing, an hbase ServerName is passed down, and on RegionServer RPC Channel creation, we use it -- host and port -- to create the InetSocketAddress (AsyncConnection-Impl#createAddr line #427) which is passed to the AbstractRPCChannel on construction and hardcoded as the client 'address'. On 'No Route To Host', we should throw away the Channel instance instance and create it anew, or catch this and allow addr change during sessions.

11. **timoha:** Attached entire hbasemaster log (hbasemaster.log) with TRACE enabled right before trying to reproduce the issue.

The time I've triggered the issue was "Thu May 21 17:28:42 UTC 2020". And the topology looked like so:

```
hbasemaster-0 10.128.25.30

hbasemaster-1 10.128.6.51

regionserver-0 10.128.53.53

regionserver-1 10.128.9.37

regionserver-2 10.128.14.39
```

They way I trigger the issue is by picking a regionserver with 0 regions (because it was restarted recently), triggering "balancer" and killing the regionserver during the execution of balancer. In this case the regionserver I killed was regionserver-2. Here's how topology

looked like after regionserver 2 came back up:

```
hbasemaster-0 10.128.25.30

hbasemaster-1 10.128.6.51

regionserver-0 10.128.53.53

regionserver-1 10.128.9.37

regionserver-2 10.128.14.40
```

You can see that regionserver-2 came back up with IP 10.128.14.40, but hbasemaster still tries to contact 10.128.14.39

12. **stack:** I wonder if we changed the AbstractRPCChannel so that rather than cache an address (InetSocketAddress), instead we just cached the remote ServerName as per https://kubernetes.io/docs/tutorial application/basic-stateful-set/. The Pod will come back w/ same 'name' but may have a different IP ("This is why it is important not to configure other applications to connect to Pods in a StatefulSet by IP address."). I think this is what we are doing when we cache an ISA. Let me see.... (Thanks for the logs [~timoha])

Oh, I bet HDFS gets confused too... (but maybe not -- IIRC, it creates' a name to use referring to the DN...) Let me check logs.

13. **timoha:** > Oh, I bet HDFS gets confused too... (but maybe not âĂŞ IIRC, it creates' a name to use referring to the DN...) Let me check logs.

So for hadoop we route clients by hostnames (dfs.client.use.datanode.hostname) and provision a k8s service per datanode which results in a stable IP per datanode. That's a workaround to the bug (https://issues.apache.org/jira/browse/HDFS-15250), which I don't think was properly addressed there. Otherwise, we could have just relied on hostnames of the pods without needing a service.

(During the pod restart, its hostname is also removed from DNS resulting in Unresolved-HostnameException for clients).

The most ideal for hbase on k8s would be to not cache any IPs (stateless connections) and not rely on hostnames (kinda like kafka brokers) but that's probably not easy to change.

14. **zhangduo:** When I implemented an in house RPC framework about ten years ago, my solution was to create an unresolved ISA as the rpc connections key, and once we want to connect to the remote peer, we recreate a resolved one.

15. **stack:** Try changing the HDFS-15250 patch so it makes an IOE of UnresolvedAddressException before rethrow? (yeah, what is there doesn't look like a 'fix').

Let me see if can undo caching of ISA (but not do resolve per rpc).

16. **stack:** The TRACE-level logs really help. Thanks [~timoha]

It looks like sun.net.inetaddr.ttl == networkaddress.cache.ttl.

If set to 10, that might be too long looking at the log.

I see this in the log around the time that regionserver-2 came back up:

```
2020-05-21 17:28:55,010 DEBUG [RpcServer.default.FPBQ.Fifo.handler=48,queue=3,port
    =16000] master.ServerManager: STARTUP: Server regionserver-2.hbase.hbase.svc.
    cluster.local,16020,1590082132059 came back up, removed it from the dead
    servers list
```

What above says is that the old server instance, regionserver-2.hbase.hbase.svc.cluster.local,16020,1590081637367, has been replaced in the Master by the regionserver-2.hbase.hbase.svc.cluster.local,16020,1590082132059 (see the differing timestamps).

A little later in the log I see this:

```
2  2020-05-21 17:28:55,745 DEBUG [RSProcedureDispatcher-pool3-t31] master.
       ServerManager: New admin connection to regionserver-2.hbase.hbase.svc.cluster.
       local,16020,1590082132059
3
4  2020-05-21 17:28:55,745 TRACE [RSProcedureDispatcher-pool3-t31] ipc.
       NettyRpcConnection: Connecting to regionserver-2.hbase.hbase.svc.cluster.local
       /10.128.14.39:16020
5
6  ...
```

i.e. the ServerManager is processing the 'new' server part of which is setting up a new connection.

Logging is a little sparse here but code looks like it should do right thing...... The Server-Manager asks to get the admin instance which ends up in here....

```
1
2    @Override
3
4   public AdminProtos.AdminService.BlockingInterface getAdmin(ServerName serverName)
5
6       throws IOException {
7
8     checkClosed();
9
10    if (isDeadServer(serverName)) {
11
12      throw new RegionServerStoppedException(serverName + " is dead.");
13
14    }
15
16    String key = getStubKey(AdminProtos.AdminService.BlockingInterface.class.
           getName(), serverName,
17
18      this.hostnamesCanChange);
19
20    return (AdminProtos.AdminService.BlockingInterface) computeIfAbsentEx(stubs,
           key, () -> {
21
22      BlockingRpcChannel channel =
23
24          this.rpcClient.createBlockingRpcChannel(serverName, user, rpcTimeout);
25
26      return AdminProtos.AdminService.newBlockingStub(channel);
27
28    });
29
30  }
```

13

The call to getStubKey should be doing a new lookup (hostnamesCanChange defaults true -- see HBASE-14544) but it is happening at....~17:28:55,745.

The new server is registered at the Master at ~17:28:55,010.

When did K8S register in DNS the new pod?

What happens if you run w/ -Dsun.net.inetaddr.ttl=1 instead of 10?

17. **timoha:** > When did K8S register in DNS the new pod?

Given the eventually consistent nature of k8s, it's possible that the mapping in DNS is updated after the new regionserver pod has already started. Unfortunately, I can't verify if that's the case as DNS isn't managed by us so I can't add extra logging there. I think for the sake of argument we can assume that the DNS mapping is inconsistent. Although, that could be the case on any infra as DNS can be inconsistent due to caching in multiple places (systemd-resolved, intermediate DNS servers, java dns cache, etc) or just operators being slow to update them.

> What happens if you run w/ -Dsun.net.inetaddr.ttl=1 instead of 10?

I was able to reproduce this issue with ttl=1 as well as ttl=0 (which I guess means no caching).

18. **stack:** bq. I was able to reproduce this issue with ttl=1 as well as ttl=0 (which I guess means no caching).

Ouch.

Any evidence that java process ever notices the DNS update?

Next I think would be looking at throwing away the Connection after N retries. We keep trying for ever.... waiting on an SCP. If can create a new Connection and keep going w/o disrupting any other ongoing RPCs., that sounds like way to go here. Thanks [~timoha]