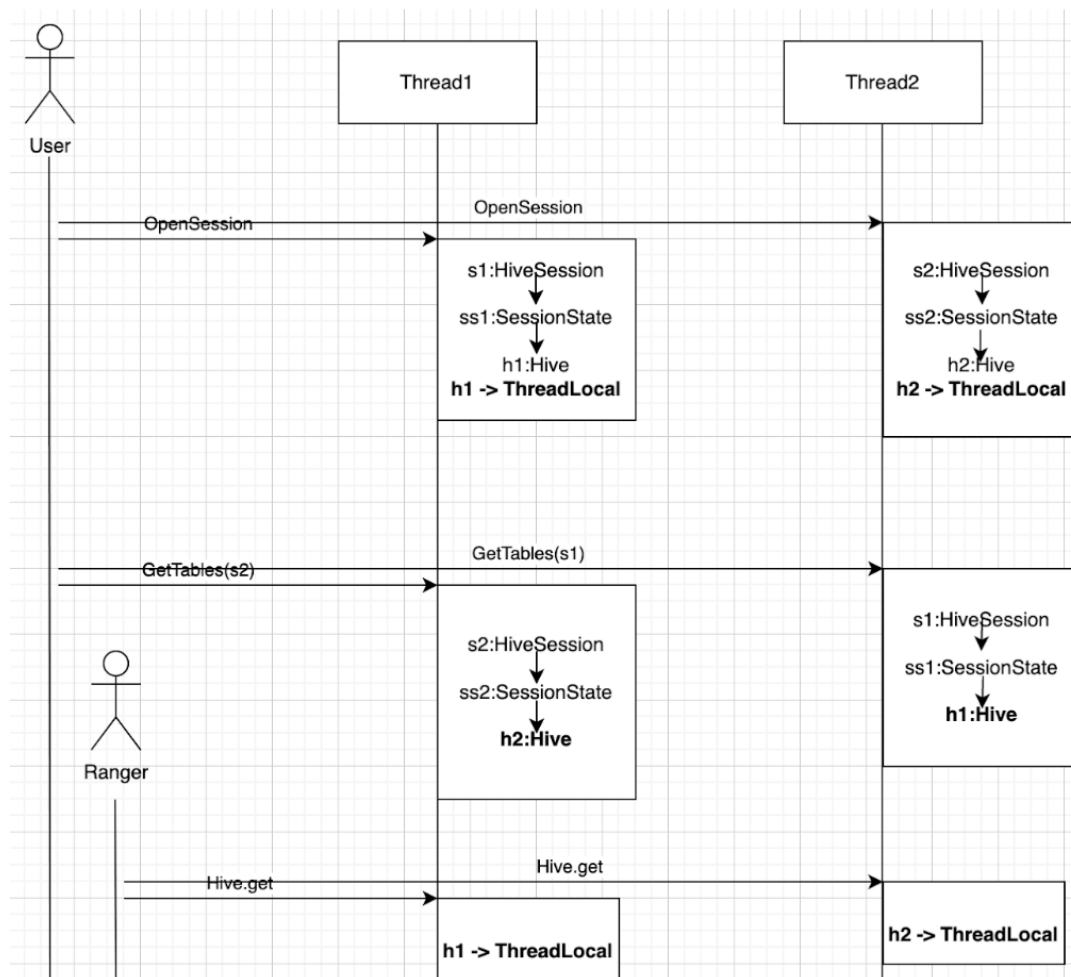


Process

1. In response to opening a session, the server handler will create a HiveSession with an unique session id. In HiveSession, we create the SessionState and a hive instance, the hive instance is stored in the shared ThreadLocal for future use as well;
2. When the handler switches to process the request from other session, first it will get the HiveSession by the session id, then use the hive from HiveSession to communicate with the HMS, it should be noted that the same hive instance in the ThreadLocal can be referenced by other handlers;
3. Ranger if any invokes Hive.get() to retrieve the hive, another handler might be using the same instance.

Let's take an example, for a GetTables request:



As the picture shows, In step 2, Thread1 or Thread2 starts to process the GetTables request from session s2 and s1 respectively, but their thread-local hives don't get updated, so there is a

chance that Thread1 holds the resource h2 waiting for h1, and Thread2 holds the resource h1 waiting for h2. In such a situation, the deadlock can happen.

You can repro the case by testing TestHS2SessionHive without HIVE-27201.