HiveServer2 Interactive High Availability

HiveServer2 Interactive (HSI) is the single point of failure in the LLAP cluster. When HSI process/node fails, all the queries submitted by client (beeline, jdbc, odbc etc.) are lost as all sessions maintained by HSI for client connections will be destroyed and clients will no longer be able to connect to HSI until it is restarted. This failure will also close all pre-warmed AMs that were initially opened by HSI (HIVE-17386 registers AMs with Zookeeper but still cleans up AMs on HSI shutdown). This document intends to provide High Availability (HA) feature to HSI by adding redundancy in the form of Active/Passive standby configuration where only one HSI will be in Active mode and one or more HSI instances can be in passive standby mode ready to takeover on Active HSI failure. Currently we do not plan to support Active/Active configuration as it is not trivial to coordinate resources for workload manager and other stateful information between multiple active HSI instances.

Active/Passive configuration
In the Active/Passive standby configuration, all HSI instances registers itself under a Zookeeper namespace but only one will be elected leader (via leader latch recipe provided by curator). The elected leader is then responsible for launching/prewarming tez AM pool for first time. Before launching/prewarming tez AM pool, the elected leader will check paths under AM namespace registry to see if AMs already exists (left over by previous leader), if so it will use those AMs instead of creating new ones (assuming DAGs that were running earlier are killed by the leader that revoked recently). To connect to the active leader HSI instance, clients will use the dynamic service discovery (preferred approach over direct connections as failover can switch HSI hosts) for getting node information about active instance.

Dynamic service discovery
Client will use the following JDBC uri for connection to active HSI

jdbc:hive2://<zooker_quorum>;serviceDiscoveryMode=zooKeeperHA;zooKeeperNamespace=hiveServer2-interactive

When clients use service discovery uri to connect to HSI, hive connection will iterate through list of HSI instances registered under ZK namespace and identify the elected leader. The node information (hostname:port) of the elected leader will then be returned to client for connecting to Active HSI instance. If active HSI loses its leadership for some reason (long GC pause resulting in session/connection timeout or network split), the leadership will be revoked and all client connections will be closed resulting in current in-flight queries to be killed (check Leader Revocation section for alternate choices).

In order for old clients to connect with new server, only leader will publish host:port information in HS2 (passive instances can put host:port in a different structure for debugging or displaying in web endpoints).

Leader Election
HSI will make use of curator's leader election recipe (creates a leader latch path) to elect Active HSI instance. Active leader will hold on to leader latch path until its leadership has to be revoked. Elected leader will create AM sessions from scratch (first time only) or create sessions from ZK namespace where previous AMs are already registered. Apart from session pool creation, the elected leader will also initialize workload management state (create/acquire ducks) before serving subsequent queries.

Leadership Revocation
In the event of network split, as long as ZK's majority quorum is maintained on the side of the leader, current elected leader will hold on to its leadership and serve client queries. If ZK quorum majority falls on the side of passive instances, current leader will lose connectivity to ZK (releasing latch path) and a new leader will be elected from one of the passive standby instances. If there is 1:1 split of ZK quorum and HSI instances, no leader will be elected and existing leader will revoke its leadership, all in-flight queries will be killed and clients will no longer be able to connect to HSI which is the worst case (check Failover Handling section for alternate choices). If ZK quorum cannot be established, a leader cannot be elected. Hence a manual intervention will be required to establish ZK quorum (healing network or add instances).

If all passive standby instances goes away, manual intervention is required to restart HSI. HSI can also warn users (also via healthcheck endpoints) when HA is configured and there are no passive standby HSI instances.
When current active HSI instance loses connectivity with ZK, it has to revoke its leadership (immediately or after timeout with connection retries). Leader revocation will revoke all the ducks, closing all the session handles and kill in-flight and queued queries (queries has to be

killed to give away resources/ducks to new active instance). Leader revocation however will not close tez AMs that are already registered with ZK.

Alternate, options to avoid killing all in-flight queries

- Revoke leadership after a timeout. This will delay new leader to be elected to give more time for in-flight queries or timeout expires after which leadership is revoked immediately. Release ducks and AMs as and when queries are completed for new leader to takeover.
- Revoke all ducks and hope for best (this will make all queries run in speculation mode and are subjected to heavy pre-emption on a busy cluster). This will still hold-on to AMs that are running the queries.

Failover Handling

Automatic failover is done via ZK watches + new leader election. In a scenario where active HSI instance goes down, leader latch will be released which passive instances figures out via watch notifications. The leader latch will then be acquired by one of the passive standby instance. This passive instance will also wait (or async?) for acquiring the ducks and session pool recreation before serving queries to client. During automatic failover, clients will wait (with some retry policy) to get node information of new active HSI instance.
Admins will also be able to manually failover via hive cli (active-to-standby and standby-to-active transitions).

Debuggability

Expose HSI instances information via web endpoint which will provide information about all HSI instances, AM pool information etc.
Provide health check endpoint to warn users/admins when a manual intervention would be required (only one active-no passive, 1:1 split where no active can be elected etc.)
Hive CLI for manual failover and inspection of HSI instances.