[YARN-4576] Enhancement for Tracking Blacklist in AM Launching

Before YARN-2005, YARN blacklist mechanism is to track the bad nodes by AM: If AM tried to launch containers on a specific node get failed for several times, AM will blacklist this node in future resource asking. This mechanism works only for normal containers. From our observation on behaviors of several clusters: if this problematic node launch AM failed, then RM could pick up this problematic node to launch next AM attempts again and again that cause application failure in case other functional nodes are busy. In many cases, the customized healthy checker script cannot be sensitive enough to aware these node as unavaialbe/unhealthy when several containers get launched failed.

In YARN-2005, we can have a BlacklistManager to track nodes in blacklist in each RMapp. Those nodes who launching AM attempt failed before will get blacklisted for specific application. To get rid of potential risks that all nodes being blacklisted by BlacklistManager, a new property ***disable-failure-threshold*** is involved to stop adding more nodes into blacklist if hit certain ratio already.

There are already some enhancements for this AM blacklist mechanism: YARN-4284 is to address the more wider case (in case AM container is not completed successfully or preempted) for AM container get launched failure and YARN-4389 tries to make blacklist configuration settings available for updating by application itself to address app's specific requirement. However, there are still several gaps to address more scenarios:

Global Blacklist

We need a global blacklist in addition to each app's blacklist to track AM container failures in global affection. That means we need to differentiate the non-succeed ContainerExitStatus reasoning from NM or more related to App.

The reason we need to differentiate here is: AM could get more chance to fail if other AM get failed before if reason belongs to global scope. A quick example is: in a busy cluster, all nodes are busy except two problematic (like: with disk failures) nodes: node A and node B, app1 already submit and get failed in two AM attempts on node A and node B. app2 and other apps should wait for other busy nodes rather than waste AM attempts on these two problematic nodes.

Some typical examples for failure affection scope are as below:

DISKS_FAILED: it belongs to global affection as the disk failure could happen on AM of other application.

KILLED_EXCEEDED_VMEM: it belongs to per app affection as other application AM could use less virtual memory than failed AM.

KILLED_BY_RESOURCEMANAGER: this happens during NM resync with RM when RM restart without work preserving. This shouldn't count into global/per app affection as the failure is from RM.

The proposed container failure scope list is as below:

Global: {"***DISKS_FAILED***"}

- We need to add this node with such failure to global blacklist, so all new launching for AM attempt will get rid of this node.

Per App: {"***KILLED_EXCEEDED_VMEM***", "***KILLED_EXCEEDED_PMEM***", "***INVALID***", "***ABORTED***"}

- We need to put this node with such failure to per app blacklist, so this app's new AM attempt will get rid of this node.

Innocent: {"***SUCCESS***", "***PREEMPT***", "***KILLED_BY_RESOURCEMANAGER***", "***KILLED_AFTER_APP_COMPLETION***"}

- This container exit status are expected or not caused by NM failure. We should ignore and do nothing for these exit status.

No possible: {***KILLED_BY_APPMASTER***}

- This container exit status is not expected and we should log a WARN message for advanced trouble shooting..

Time Window for Blacklist

If AM container failure is recognized as global impact instead of app specific issue, we should consider the blacklist is not a permanent thing but with a specific time window for more chance in case NM issue is get fixed/recovered later. We need a monitor/timer thread to watch all nodes in blacklist. In general, the time window only suit for global blacklist. However, in some case (like we already hit ***disable-failure-threshold***), that means we should remove this nodes from per app blacklist as well or we could be short of good nodes.

Pluggable Blacklist Policy

We could have user defined black list policies to address more possible cases and scenarios. Like different users could prefer different rules (not only strict/loose, but also some requirements other than container failure) to blacklist a node for launching AM. It would be nice that we can make blacklist policy pluggable for user to extend, just like: block placement policy we have in HDFS.

Feature of Whitelist

For some test scenario or addressing some corner cases, we could consider to have a whitelist mechanism for AM launching. It is just like resource request mechanism in YARN that we can make the whitelist a strict or loose rule through configuration.

Other Cases

NM state change (restart, decommission/recommission, resource update, etc.) should affect nodes in blacklist. We may consider to track the reason of container failures in node blacklist so we can check if NM state change can help in this case or not (like resource change on NM won't help for disk failure, etc.).