Generic History ACLs

Motivation

When users get application related information from RM, we will check whether the user has the access to the application before responding hist request. However, the generic history service provided by the timeline server doesn't have the access control now, the user can inquiry the information of others' applications without limitation, which introduce critical security concern for the generic history service user. Therefore, we need to fix this security hole.

Solution

Basically, we need to provide the similar access control of the application history data for timeline server as we did for RM. And one of our goal is to make getting application(s), attempt(s) and container(s) from either RM or TS is transparent to the user. With the regard to the current TS status, we propose to do the following things:

- For generic history service, the best practice is to check the application ACLs only now, while at RM side, we check both the application and the queue ACLs to determine whether the user has the access to the application. The problem is that the generic history service only store the application information, but not the queue information. Once we store the latter one, we can use it the reconstruct the queue at the time when the application was running on a YARN cluster, and can check user's queue access as well. Another argument is when the application is finished, it is actually removed from the queue. It seems that the application should be no longer related to the queue at all. Therefore, we don't need to allow the queue admin to get access to the application any more, unless it is set to be on the application view_app list as well. Hence let's split this and focus on the application ACLs only.
- We need to persist ACLs information into the timeline server together with the the application entity.
    - We only need to take care of VIEW_APP ACLs, because the application cannot be modified after it's done.
- We need to provide the consistent ACLs check for CLI, web UI and web services. Even RM doesn't do this properly (only RM CLI works correctly with ACLs). To do that, we need to plugin a ApplicationACLsManager inside the ApplicationHistoryManager, to check users' access before returning the reports from the 7 getter methods.

Relationship with Timeline data ACLs

Timeline data ACLs is used to control the access of the timeline data. In the scenario of the generic history service, RM writes the data, and it should allow timeline server (its login user) to read the data. Generic history data is the payload of the timeline data, and the generic history data ACLs are part of it and are stored into the timeline store. Once the timeline server (the builtin generic history service) reads the generic history data from the timeline store, and uses the stored generic history ACLs to judge whether the user has the access the cooked application/attempt/container report(s).

Implementation

Here’re some implementation details we need to take care:

- We may want to reuse ApplicationACLsManager in the timeline server.