
Title: [YARN-3813] Support Application timeout feature in YARN

Authors: Nijel S F, Rohith Sharma K S

Last modified: Jul 09 2015

1.1 Preamble

Many a time, it is needed to restrict the application to complete within a specified time. Like in below case

One requirement is to show the CDR statistics of last few minutes, say for every 5 minutes. The same Job will run continuously with different dataset. So job will be started every 5 minutes. The estimate time for this task is 2 minutes or lesser time. If the application is not completing in the given time the output is not useful.

1.2 Problem statement

Using the current APIs exposed, user need to track the status with the time elapsed and give a kill command to the application.

This is specific to application client logic. Similar requirement can be there with multiple applications and each one need to have the code embedded in application client.

1.3 Requirements

- Users should be able to set a optional timeout value at the time of application submission.
- RM Kill the applications which elapsed the specified time.
- Show the time out metric in JMX.

1.4 Proposal

YARN-3813 suggest to provide a service to track the submitted applications and invoke stop/kill if it elapsed the specified timeout value

- ✓ Add a new configuration to specify the application timeout value
(yarn.application.max.liveness)
 - ✓ Add a new auxiliary service [RMAppTimeOutService] to track the running applications and invoke the kill action.
 - ✓ Add a flag in RMApp to identify the timed out application. This is for metric purpose.
-

1.5 Implementation Details

1. Configuration changes

Add a new parameter, `yarn.application.max.liveness` to `ApplicationSubmissionContext`

2. New auxillary service : `RMApTimeOutService`

Responsibility is to track the running application. Simple logic

```
//if job is running and the time elapsed kill
if ((RMApState == SUBMITTED/ACCEPTED/RUNNING) &&
    && (currentTime - app.getSubmitTime()) >= timeout

    LOG.warn("Job timed ou t" + appId.toString());

    new RMApEvent(appId, RMApEventType.KILL));
app.setAppTimedOut(true);
```

3. Proto changes

Proto changes required to add a new parameter,
`yarn.application.max.liveness` to the `ApplicationSubmissionContext`
