

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

Authors: Nijel S F, Rohith Sharma K S with Input from Vinod, Jian He, Varun Vasudev and Sunil G.

Last modified: 28th Nov, 2016

1. Preamble
2. Problem statement
3. Requirements
4. Proposal
5. Design
6. Update Timeout for Application
7. Sequence Diagram for App Attributes update

1 Preamble

Many a time, it is needed to restrict the application to complete within a specified time. 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.

2 Problem statement

Using the existing APIs exposed in Yarn, 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.

3 Requirements

- Users should be able to set optional timeout value at the time of application submission.
- Users should be able to update timeout value after application is submitted.

4 Proposal

Provide a monitoring service to track the submitted applications with user given timeout. Invoke kill if application elapsed the specified timeout value. Sometimes user would like to update timeout, so provide a API to update timeout value.

5. Design

Application timeouts can be many. Introduce enum ApplicationTimeoutTypes with LIFETIME i.e application start monitoring from submitted time. New timeout type can be added here to support in future.

```

public enum ApplicationTimeoutType {

    /**
     * <p>
     * Timeout imposed on overall application life time. It includes actual
     * run-time plus non-runtime. Non-runtime delays are time elapsed by scheduler
     * to allocate container, time taken to store in RMStateStore and etc.
     * </p>
     * If this is set, then timeout monitoring start from application submission
     * time.
     * /
     LIFETIME;
}

```

ApplicationSubmissionContext can have API to get/set timeout values. User can set multiple timeout values at the time of submission. The time value is in seconds.

```

/**
 * Get <code>ApplicationTimeouts</code> of the application. Timeout value is
 * in seconds.
 * @return all <code>ApplicationTimeouts</code> of the application.
 */
@Public
@Unstable
public abstract Map<ApplicationTimeoutType, Long> getApplicationTimeouts();

/**
 * Set the <code>ApplicationTimeouts</code> for the application in seconds.
 * All pre-existing Map entries are cleared before adding the new Map.
 * <p>
 * <b>Note:</b> If application timeout value is less than or equal to zero
 * then application submission will throw an exception.
 * </p>
 * @param applicationTimeouts <code>ApplicationTimeouts</code>s for the
 * application
 */
@Public
@Unstable
public abstract void setApplicationTimeouts(
    Map<ApplicationTimeoutType, Long> applicationTimeouts);

```

Once application is submitted with timeout, RM server start monitoring this application.

6. Update Timeout for Application

Users are allowed to update timeout of an application. If application is not monitoring earlier, then RM start monitoring application from NOW. **Note:** All the YarnClient-->ResourceManager communication happens in ISO8601 yyyy-MM-dd'T'HH:mm:ss.SSSZ format. This way, YarnClient-->RM communication is Idempotent.

1. YarnClient API

API Name	Idempotent
updateApplicationTimeouts(UpdateApplicationTimeoutsRequest request)	Yes

2. CLI interface

```
./yarn application --appId $appId --updateLifetime $value
```

3. REST interface

Feature	Operation	Path	Description
Timeout	PUT	/apps/{appid}/timeouts/\$type	One ApplicationTimeoutType can update using this URI
	GET	/apps/{appid}/timeouts/ /apps/{appid}/timeout/\$type	<ol style="list-style-type: none"> 1. Return collections of timeout value 2. Return single timeout value

Sample REST output

```
{
  "timeouts": {
    "timeout": [
      {
        "type": "LIFETIME",
        "expiryTime": "UNLIMITED",
        "remainingTimeInSeconds": -1
      }
    ]
  }
}
```

7. Sequence Diagram for App Attributes update.

