“YARN Federation Router” *YARN-3659*

Design Document

Design Contributors

Kishore Chaliparambil

Carlo Curino

Giovanni Matteo Fumarola

Ellen Hui

Subru Krishnan

Sarvesh Sakalanaga

Siavash Soleimanifard

Introduction

YARN applications are submitted to one of the Routers, which in turn applies a routing policy (obtained from the Policy Store), queries the State Store for the sub-cluster URL and redirects the application submission request to the appropriate sub-cluster RM. We call the sub-cluster where the job is started the “home sub-cluster”, and we call all other sub-clusters a job is spanning on “secondary sub-clusters”. The Router exposes the ApplicationClientProtocol to the outside world, transparently hiding the presence of multiple RMs. To achieve this the Router also persists the mapping between the application and its home sub-cluster into the State Store. This allows Routers to be soft-state while supporting user requests cheaply, as any Router can recover this application to home sub-cluster mapping and direct requests to the right RM without broadcasting them. For performance caching and session stickiness might be advisable. At any one time, a job can span across one home sub-cluster and multiple secondary sub-clusters, but the policies we are devising will try to limit the footprint of each job to minimize overhead on the scheduling infrastructure (more in section on scalability/load).

The Router will implement an interceptor pattern to generalize the approach and have only dynamically coupling for Federation.

This provides a placeholder for:
1) throttling misbehaving clients (YARN-1546);
2) mask the access to a YARN Cluster (YARN-5411);
3) mask the access to multiple RMs (YARN-3659).

In this document, we are going to analyze the 4 main APIs for ApplicationClientProtocol.

Get New Application

Yarn Router forwards every requests to the RM.
During this operation there will be no communication with the State Store. The Router will forward the requests to any sub-cluster.

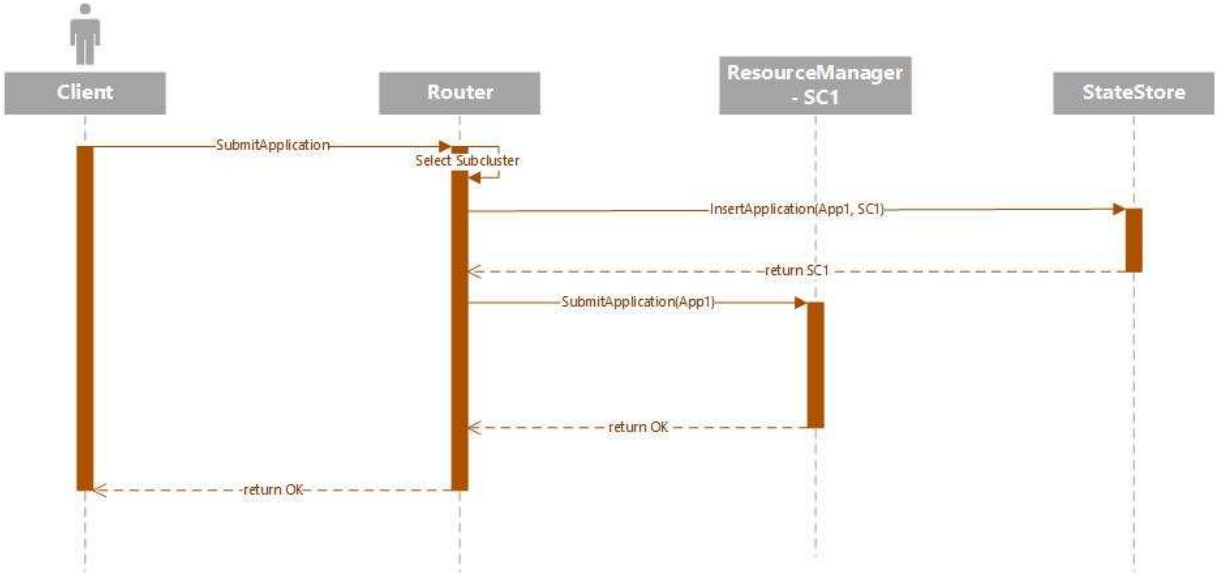
Possible failure:

Client: behavior as YARN.
Router: the client will timeout and resubmit.
ResourceManager: the Router will timeout and contacts another RM.

Submit Application

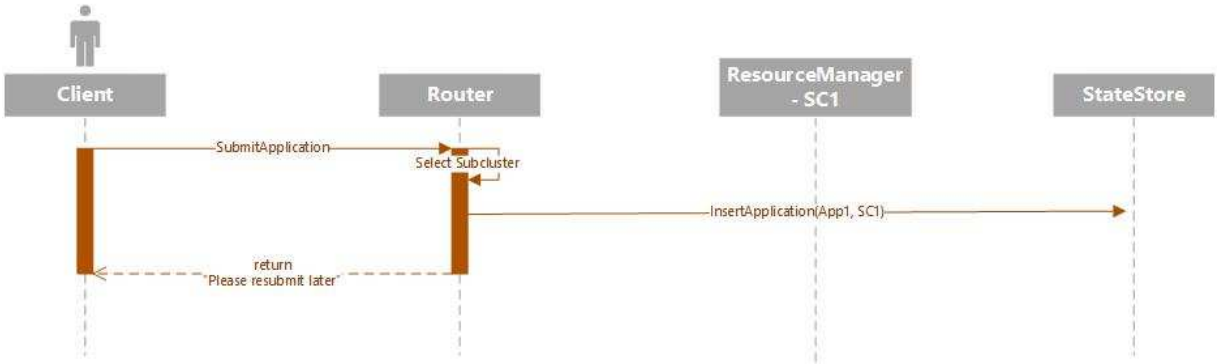
Today, in YARN there are no checks of any applicationId submitted.

Base scenarios:



- The client submits an application to the Router.
- The Router selects one sub-cluster to forward the request.
- The Router inserts a tuple into State Store with the selected sub-cluster (e.g. SC1) and the appId.
- The State Store replies with the selected sub-cluster (e.g. SC1).
- The Router submits the request to the selected sub-cluster.

In case of State Store failure:

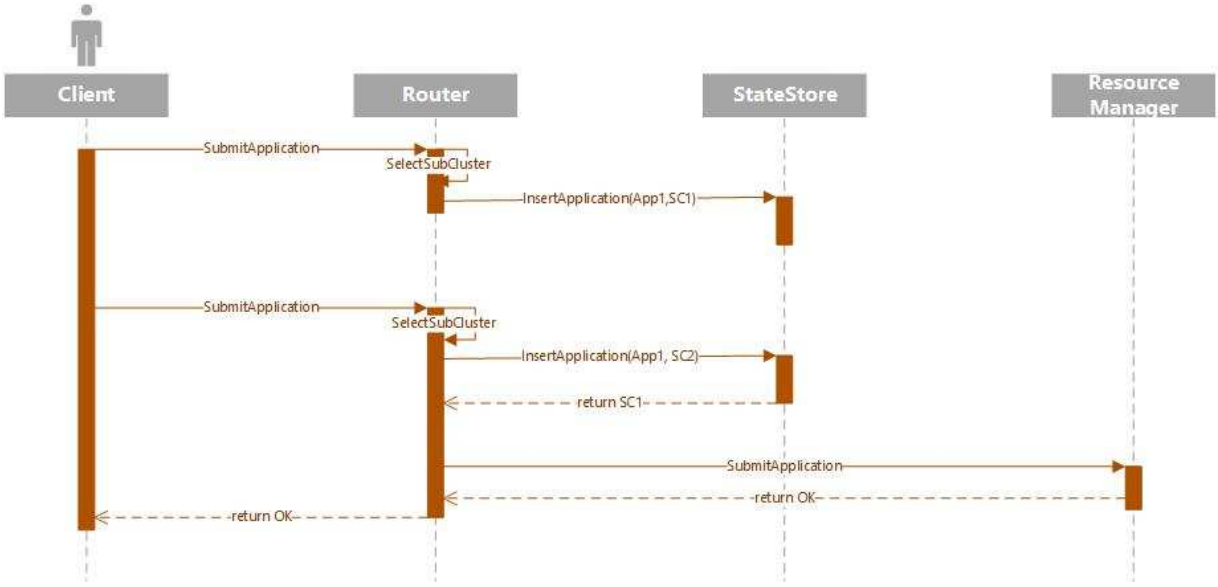


- The client submits an application to the Router.
- The Router selects one sub-cluster to forward the request.
- The Router inserts a tuple into State Store with the selected sub-cluster (e.g. SC1) and the appId.
- The State Store is down, and the Router times out.
- The Router replies to the client with an error message.

If State Store fails after inserting the tuple, there is no change to the normal execution.

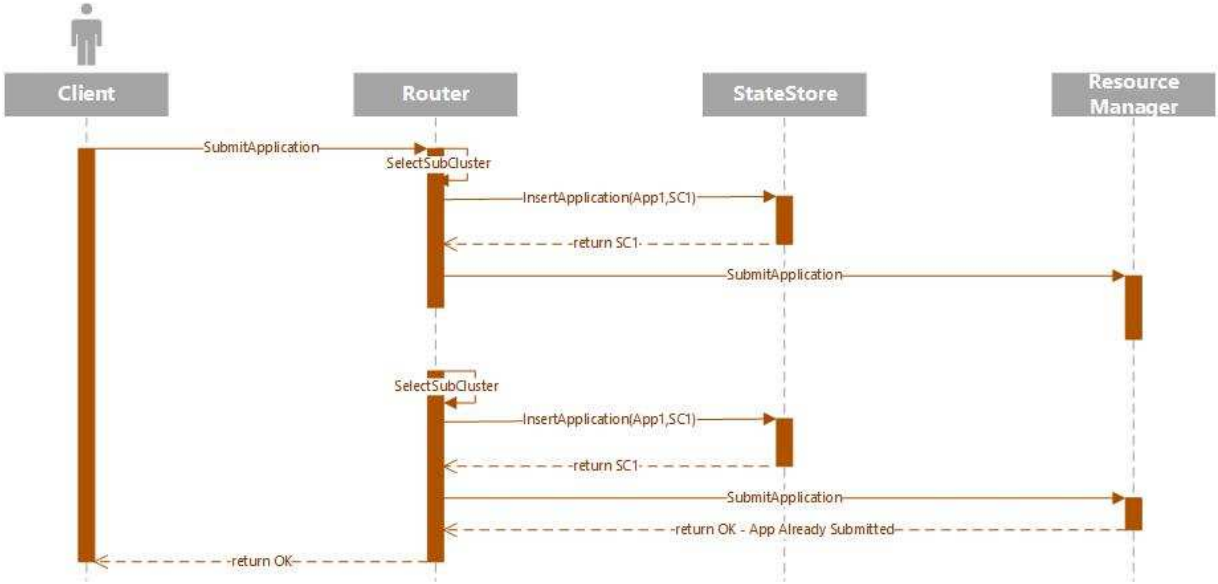
In case of Router failure

Scenario 1 – Crash before submission to the ResourceManager



- The client submits an application to the Router.
- The Router selects one sub-cluster to forward the request.
- The Router inserts a tuple into State Store with the selected sub-cluster (e.g. SC1) and the appId.
- The Router crashes.
- The client timeouts and resubmits the application.
- The Router selects one sub-cluster to forward the request.
- The Router inserts a tuple into State Store with the selected sub-cluster (e.g. SC2) and the appId.
- Because the tuple is already inserted in the State Store, it returns the previous selected sub-cluster.
- The Router submits the request to the selected sub-cluster.

Scenario 2 – Crash after submission to the ResourceManager



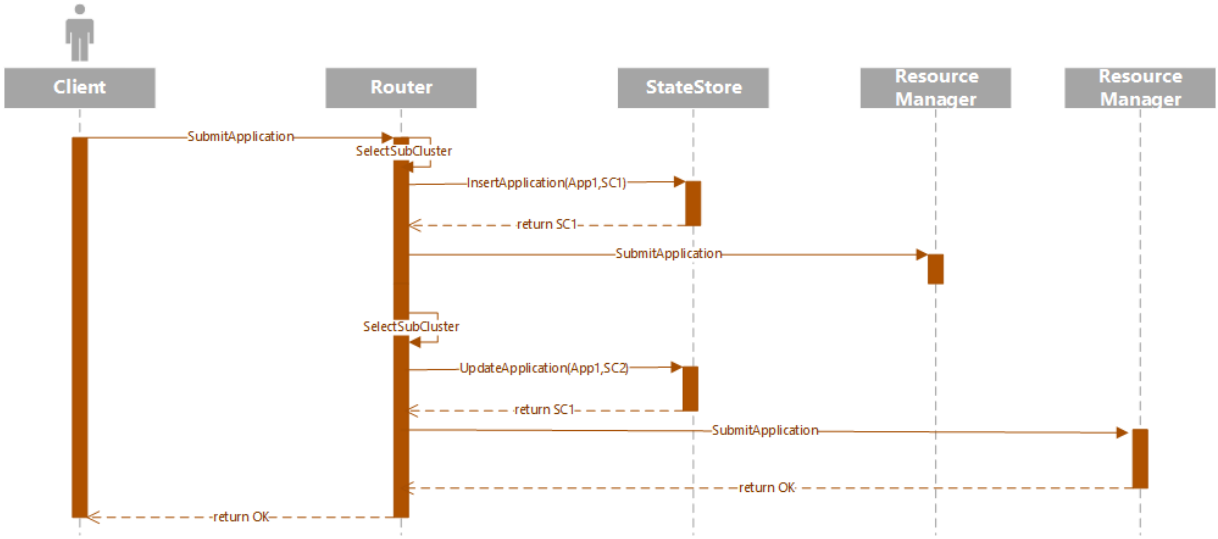
- The client submits an application to the Router.
- The Router selects one sub-cluster to forward the request.
- The Router inserts a tuple into State Store with the selected sub-cluster (e.g. SC1) and the appId.
- The Router submits the request to the selected sub-cluster.
- The Router crashes.
- The client timeouts and resubmit the application.
- The Router selects one sub-cluster to forward the request.
- The Router inserts a tuple into State Store with the selected sub-cluster (e.g. SC1) and the appId.
- The State Store replies with the selected sub-cluster (e.g. SC1).
- The Router submits the request to the selected sub-cluster.

When a client re-submits the same application to the same RM, it does not raise an exception and replies with operation successful message.

In case of Client failure:

There is no change to the normal execution.

In case of ResourceManager failure:



- The client submits an application to the Router.
- The Router selects one sub-cluster to forward the request.
- The Router inserts a tuple into State Store with the selected sub-cluster (e.g. SC1) and the appId.
- The Router submits the request to the selected sub-cluster.
- The entire sub-cluster is down – all the RMs in HA
- The Router times out.
- The Router selects a new sub-cluster to forward the request.
- The Router update a tuple into State Store with the selected sub-cluster (e.g. SC2) and the appId.
- The State Store replies with OK answer.
- The Router submits the request to the selected sub-cluster.

Get App Report/Kill App

The Yarn Router will forward to the respective Yarn RM in which the AM is running.

Client: behavior as YARN.
Router: the client will timeout and resubmit the request.
ResourceManager: the Router will timeout and it will contact RM again up to #retry times.
State Store: the Router will timeout and it will contact State Store again up to #retry times.