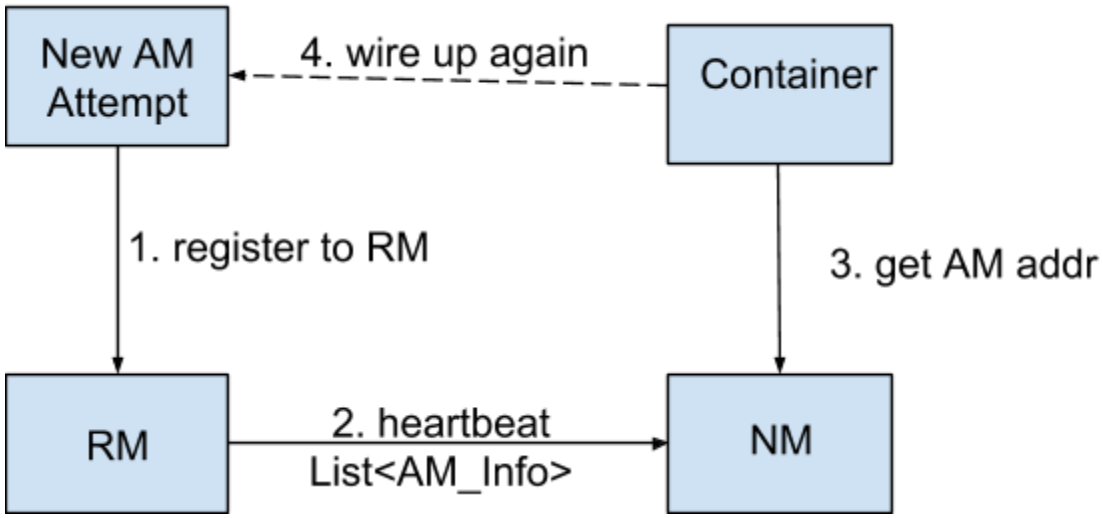
[YARN-4758] AM Discovery Service for YARN Container

Background

Tracked in [YARN-1489], work-preserving during AM restart get supported in YARN and most works has been done. However, there is a missing piece of work for containers to discover address of AM after AM get restarted. This is critical for YARN apps as for most of them, like: MapReduce, Tez or Spark, running containers need to communicate with AM and keep updating progress before it get finished.
Taking MR as an example, to support AM restart with work preserving (https://issues.apache.org/jira/browse/MAPREDUCE-6608) - that said, when AM get failed for some reason, the inflight tasks will keep running/pending and wait for new AM attempt comes back to continue. Specifically saying, running tasks should know where the new AM attempt get launched so TaskUmbilicalProtocol (RPC protocol between MR's tasks and AM) can get retry between clients and new server. The same requirement should apply on other YARN applications to achieve more resilient from AM failure.

AM Discovery Flow

Based on requirements above, we are proposing AM address discovery service via RM-NM heartbeat and new RPC protocol between container and NM.



For details:

1. When AM for new attempt get started, it will register its address-info (single or multiple entries) to RM.

2. When receiving heartbeat from NM, RM will look at all the running containers on the node (*getContainersStatuses()),* find the AM addresses of the corresponding applications and pull them back into *NodeHeartbeatResponse*.

3. NM caches the AMs' address info in a map <*application_id, am_address_info*>, and removes this info when application completes (finished or killed).

4. Container will fetch AM notification from the local NM where it running via a Container-NM RPC request (Wrapped).

More Details

Message Format between NM and RM

Message should at least include necessary info, like: application_id, attempt_id and address info (host + port). To support multiple entry which could be more generic for other address discovery, adding a name for this address is necessary also. To combine with requirement for other service discovery, like: collector service discovery (please refer discussion in https://issues.apache.org/jira/browse/YARN-3359), we may need service launched timestamp info, so RM can do arbitration when race condition happens among duplicated services. The message format is as following:

```
message AddressInfoProto {
  optional string name = 1;
  optional ApplicationIdProto app_id = 2;
  optional int32 attempt_id = 3;
  optional string host = 4;
  optional int32 port = 5;
  optional int64 timestamp = 6;
}
```

Heartbeat between RM and NM

From *NodeHeartbeatRequest*, RM knows all running containers by *getContainersStatuses()* via *NodeStatus.* In the meanwhile, RM can track all AMs' reported address info in RMAppAttempt. Obviously, a list of *AddressInfoProto* will get encoded in *NodeHeartbeatResponse* which include all addresses (AMs or other services) that NM need to know.

An optimization for AM discovery case is: to only report address for AMs that get restarted before as containers should already know when AM first time get launched.

Container fetch Address Info (AM) from NM

Container can get address info from NM, via a new RPC (named as ContainerUmbilicalProtocol) between container and NM. If container haven't fetched updated address or fetched address doesn't work, it will keep retry until fetching correct addresses. Because this protocol (between container and NM) could be used by Application, it could be more user friendly if we wrap it with a new NM client library NMContainerClient which could retry automatically if no address for given service name is available.

Security

Most communication channels are secured already, like: RM-NM or AM-RM. However, a new added communication channel: container - NM, will need to implemented as secured. The design and implementation will discuss in a separated JIRA.

Failed Over/Restart Cases

RM restart/HA case
When RM get failed over or restart, all running AMs' attempt will re-register again to RM that make RMAppAttempt always get updated even after RM failed over/restart cases.

AM failed and restart case

This is the target scenario to address: If AM get failed and restart, the new AM attempt will register to RM, and RM send updated AM address info to related NMs and finally deliver to each containers.

NM restart case
If AM restart happen during NM down, nothing will get missed. After coming back, NM will heartbeat with RM and get latest AM address info.

How YARN Application (i.e.MapReduce) can use this

We need to add a new YARN client API, called: NMContainerClient, for application's task to access AM information directly. This API will not expose implementation details, like: how to fetch AM info with RPC call to NM
The initiative API provided is quit simple:

// get address info (for latest AM, collector, etc.)
InetSocketAddress getAddress(string addressName);

Note: If application AM need to expose multiple address entries (for different protocol), make sure there are on different *addressName*.