RMAvailability

Resource Manager Restart

Resource Manager Restart:

The key set of data structures for the Resource Manager is the following:

1. All the Applications that the resource manager is aware of and the following are the states that it maintains per application
    - Application Master Information (state, and other RM submission context)
    - Containers allocated to an application
2. For each NodeManager in the System, it tracks:
    - Node id, an id allocated during registration of the NodeManager which it uses to do status reports and identify itself to the RM
    - The Total Resource that the NodeManager has. This is the resource the NodeManager registers with and notifies the RM that it has X amount of resource
    - The Available/Used resource for the node manager. This is tracked for new allocations in the system which can only be done if there is enough available resources in the nodemanager.
    - Containers running on a NodeManager.

The RM needs to be able to re construct the above set of information on a restart to be able to function correctly.

Given the above we can get away with just persisting the following information:

1. Application Id, state of the application (RUNNING/PENDING etc), Application Master Information – This involves persisting the latest state, application master information
2. The containers that have been allocated per application – This means we have to persist every container allocation in the system
3. The Hostname to NodeID map and the capacity of each node – This means we have to persist once when a NodeManager registers and remove it when a NodeManager is expired
    - **If we make NodeID as hashes of hostname:port, then we don't need that map. Capacity of the nodes can be obtained from the heartbeat, no? -vinodkv**

Note that we are not persisting the nodemanager to container map. This map can be reconstructed with the application to containers map. Each container in the system has information on which nodemanager it belongs to. So it becomes easy to create the nodemanager to container map.
– **May be instead of trusting the AM, we should obtain the container map from the nodemanager when it registers back. -vinodkv**

Also, the available/used information can be derived from to (iii).

For persistence we are using ZK.

A Resource Manager is considered to have restarted and functional once all the states mentioned ResourceManager (RM) Failure:

On the failure of the ResourceManager, the containers in the system keep running. There are 2 entities in the framework that interact with the RM – the application master(AM) and nodemanagers (NM).

On a RM failure, the AM and NM keep running. They both wait for the RM to come up.

The NM buffers its update and so does the AM.

Once the RM restarts, the AM will have to synchronize with the RM on what containers it has. So, there will be an api to allow the AM to resync with the RM on a RM restart. This sync api prevents any loss of containers that the AM might not have been notified of in case of a RM failure.

The NodeManagers continue to send status reports as soon as the RM re starts.

More documentation coming soon.