What’s the suspend & resume feature for Hadoop

As we known, AM first asks for resource needs in terms of total number of unfinished task containers in each heartbeat communication with AMS; Hence the number decreases as containers are finished.

AMS brings the requests to ResourceScheduler which will do the actual resource allocation based on the specific scheduler implementation (like FairScheduler). Then resource scheduler responds with the resource allocated by nodes, or empty resource due to resource contention among different jobs within the cluster. If the resource empty response is returned to AM, AM will ask for zero resource in next heartbeats until resources are allocated to AM by resource scheduler upon AMS requesting for specific amount of resources.

When target jobs inside Hadoop are suspended, those already allocated and running task containers will continue to run until their completion or active preemption by other ways. But no more new containers would be allocated to the target jobs.

In contrast, when suspended jobs are put into resume mode, they will continue to run from the previous job progress and have new task containers allocated to complete the rest of the jobs.

Why is the suspend & resume feature necessary for Hadoop

In a multi-application cluster environment, jobs running inside Hadoop YARN may be of lower-priority than jobs running outside Hadoop YARN (like HBase). To give way to other higher-priority jobs inside or outside Hadoop YARN, a user or some cluster-level resource scheduling service should be able to suspend and/or resume some particular jobs within Hadoop YARN.

How to implement Hadoop Job Suspend & Resume REST API

Its implementation architecture is shown in below diagram. Except RMWebService and AMS components in the diagram, all components or code logic block are newly created for suspend and resume feature.

Suspend

Resume

Processing  
Job

SuspendAllocating

RMWebServices

REST API

Resource Scheduler

Suspend/  
resume?

1. Keep a copy of Ask   
2.set the value of NumContainers requested as zero

Restore Ask object with the backup copy



**First, for suspend function in Hadoop**

In order to suspend Hadoop jobs, AMS will overwrite the response to AM to be empty. But before that, AMS first keeps a copy of the last resource request (in the form of total container number) Besides, AMS will not forward requests to ResourceScheduler.

Suspend steps are as following when we suspend a Hadoop job:

1. Put (application\_id , null) into appToAskMap;
2. Put (application\_id , true) into appSuspendStatusMap;
3. ApplicationMasterService checks whether the target job should be suspended  [appSuspendStatusMap.contains(application\_id) && appSuspendStatusMap.get(application\_id) is true] ;
4. Check whether the size of lastask list is 0 or not:  
           if 0 : backup the object of “ask” into lastask , put (application\_id , lastask) into appToAskMap ,and then set the value of NumContainers as 0 in current ask;  
           else : do nothing for “ask”.

If ResourceScheduler checks “ask” to find no request for resources, don't allocate any other new containers for the running application;  
  
  
**Second, for resume function in Hadoop**

In order to resume, AMS will take back the backup resource request and forward it to Resource Scheduler to help allocate resources. In turn, Resource Scheduler returns the resource allocation and AMS will respond accordingly to AM with new resource allocation information.

Resume step is as following when we resume a Hadoop job:

1. Put (application\_id , false) into appSuspendStatusMap;
2. ApplicationMasterService checks whether the job should be resumed [appSuspendStatusMap.contains(application\_id) && appSuspendStatusMap.get(application\_id) is false];
3. Check whether the size of lastask list is 0 or not:  
           if  0   : update nothing for ask ;  
           else : take back from the backup resource request "lastask" into "ask" and forward “ask” to ResourceScheduler to help allocate resources for the application,and then remove the key {application\_id} from appToAskMap.