

Scheduler ActivityManager - Test Report

Success Test cases:

1. Query /scheduler/activities and check the response with no NM

Command: scheduler/activities

REST O/P:

```
{  
  "diagnostic": "No node manager running in the cluster"  
}
```

2. Query /scheduler/activities and check the response with few NMs before registration.

Command: scheduler/activities

REST O/P:

```
{  
  "diagnostic": "waiting for next allocation"  
}
```

3. Allocation for an application is done and app has finished or (No apps is running)

Command: scheduler/activities?nodeId=localhost:25006

REST o/p:

```
{  
  "nodeId": "localhost:25006"  
  "timeStamp": "1469792611186"  
  "allocations": {  
    "finalAllocationState": "SKIPPED"  
    "root": {  
      "name": "root"  
      "priority": "-1"  
      "allocationState": "SKIPPED"  
      "Diagnostic": "do not need more resource"  
    }  
  }  
}
```

Comments

- a. I think Diagnostic message could be improved. "Applications does not need more resource"
- b. For node activity, "**priority**": "-1" does not make sense. Could we hide the same?
- c. **timeStamp** is not meaningful ("*timeStamp*": "1469792611186"). Its could be date and time or relative to previous activity.

4. Allocation for an application is done and app is running with pending requests.

Command: scheduler/activities?nodeId=localhost:25006

REST o/p:

```
{
  "nodeId": "localhost:25006"
  "timeStamp": "1469793414915"
  "allocations": {
    "allocatedContainerId": "container_1469792969489_0001_01_000001"
    "finalAllocationState": "ALLOCATED"
    "root": {
      "name": "root"
      "priority": "-1"
      "allocationState": "ACCEPTED"
      "children": {
        "name": "default"
        "priority": "-1"
        "allocationState": "ACCEPTED"
        "children": {
          "name": "application_1469792969489_0001"
          "priority": "0"
          "allocationState": "ACCEPTED"
          "children": {
            "name": "container_1469792969489_0001_01_000001"
            "priority": "0"
            "allocationState": "ALLOCATED"
          }
        }
      }
    }
  }
}
```

Comments:

- a. *finalAllocationState* is for application. Could we say *finalAppAllocationState*.
- b. In queue level, is "*allocationState*" meaningful?
- c. As mentioned earlier, priority could be hidden in places where its -1.
- d. As an improvement, its better to give pending resource requests per app after allocation
- e.

5. Allocation for an application is done and app is running. Cluster is full.

Command: scheduler/activities?nodeId=localhost:25006

REST o/p:

```
{
  "nodeId": "localhost:25006"
  "diagnostic": "do not have available resources"
}
```

6. Allocation for an application is done and app is running. Second app is awaiting due to AM resource percentage.

Command: scheduler/app-activities?appId=application_1469802642816_0004

REST o/p:

```
{
  "applicationId": "application_1469802642816_0004"
  "diagnostic": "waiting for display"
  "timeStamp": "1469803850533"
}
```

Comments:

It is not desired o/p. It should am-resource not available. Please help to check the same.

7. Allocation for an application is done and app is running. Second app was awaiting due to AM resource percentage. Once app1 is done, app2 has got resource.

Command: scheduler/app-activities?appId=application_1469802642816_0004

REST o/p:

```
{
  "applicationId": "application_1469802642816_0004"
  "allocations": {
    "nodeId": "localhost:25006"
    "queueName": "default"
    "priority": "0"
    "allocatedContainerId": "container_1469802642816_0004_01_000001"
    "allocationState": "ACCEPTED"
    "timeStamp": "1469803970106"
    "allocationAttempt": {
      "name": "container_1469802642816_0004_01_000001"
      "priority": "0"
      "allocationState": "ALLOCATED"
    }
  }
}
```

Comments:

1. State of *application_1469802642816_0004* after allocation will be *RUNNING*. Could we propose something here that what can be target state since container is allocated (failed or running)
2. Could we also print *node_label* here.

Negative Test cases:

1. Query is done for invalid node.

Command: cluster/scheduler/activities?nodeId=localhost:25009

REST o/p

```
{
  "nodeId": "localhost:25009"
  "diagnostic": "Cannot find node manager with given node id"
}
```

2. App query is done for invalid app

Command: cluster/scheduler/app-activities?appId=applicatmb076

REST o/p:

```
{
  "RemoteException": {
    "exception": "IllegalArgumentException"
    "message": "Invalid ApplicationId prefix: applicatmb076. The valid ApplicationId should start with prefix application"
    "javaClassName": "java.lang.IllegalArgumentException"
  }-
}
```