[YARN-4091] Introduce more debug/diagnostics information to detail out scheduler activity

Chen Ge and **Wangda Tan** with inputs from **Sunil Govind**

Problems

YARN scheduler can assign resources to many applications on different nodes in every second. For now, it is hard for users to figure out why some applications are accepted or rejected. We have heard lots of complaints and questions from users like: "Why my application cannot run when cluster has available resources?", "Why other queues/applications can get resource before me?".

Proposal

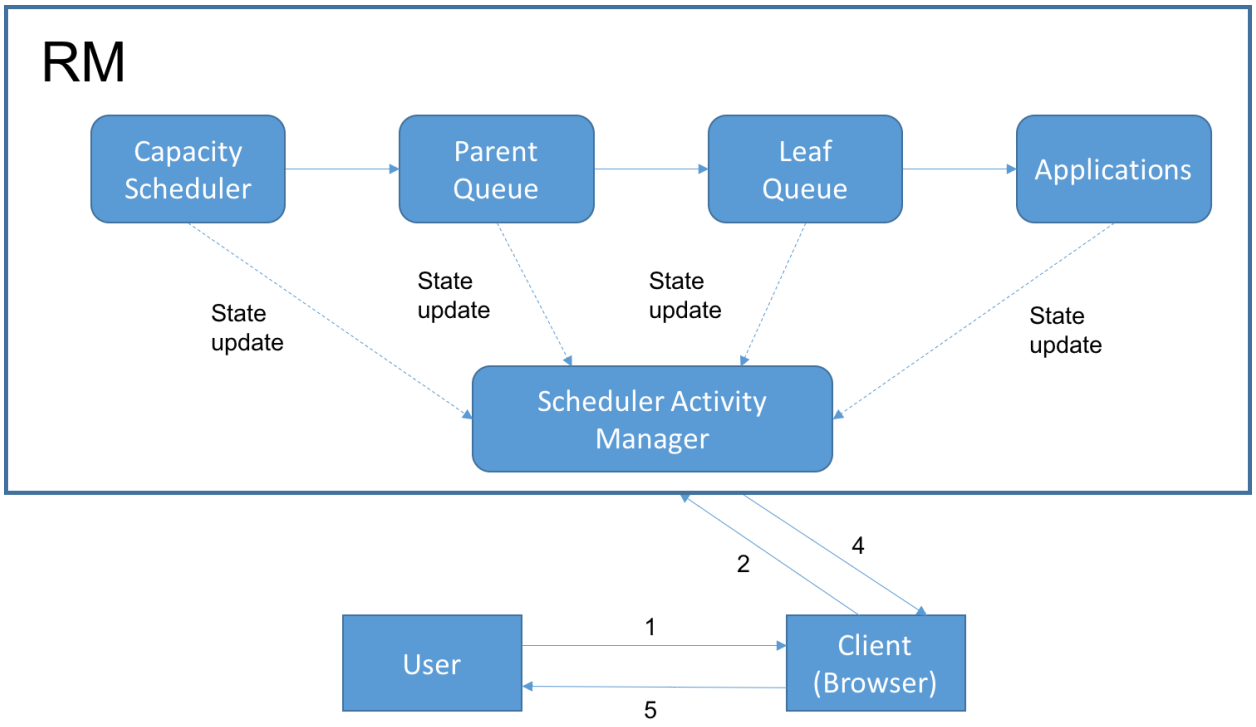
To help with debuggability, we propose a convenient and intuitive way to show how the scheduler makes decisions. We track down every scheduler activities, such as how queues/applications are sorted, the reason why to reject queue/application, etc.

Once user sends a request to check activities, detailed information about each node, queue and application will be displayed on webpage.

Users can set optional parameters for detailed information, like node id, queue name or application id to filter responses from scheduler.

High-level Design

1. User clicks on the YARN UI to check scheduler activities.
2. Client (Browser) sends REST request to RM
3. YARN RM receives and handles the REST request.
4. Client polls RM until REST response available
5. Render scheduler activities on web UI



REST API

http://<rm http address:port>/ws/v1/cluster/scheduler/activities
Optional parameter:

- node-id: Only look at specific node (by default to look at next node's activities)
  - When node-id is not specified, scheduler will give result of next available node.
- queue-name: Only look at activities under a given queue name

```
"Scheduler-activity": {
  "activities": {
    "node": {
       "nodeId": "node:1234"
```

```
available-resources: <3GB, 1VCores>
allocations: [
  "allocation": {
    "container": {
      "containerId": container-1
      "state": "allocated-from-reserved"
    }
    "queue": {
      "queueId": queue-a
      "applications": [
        "application": {
          "applicationId": app_1
          "priority": 5
          "state": "accepted"
        }
      ]
    }
  },

  "allocation": {
    "container": {
      "state": "skipped" // Nothing allocated
    }
    "root": {
      "queueId": queue-root
      "state": "accepted"
      "children": [
        "queue": {
          "queueId": queue-a
          "applications": [
            "application": {
              "applicationId": app_2
              "priority": 3
              "state": "skipped"
              "diagnostic": "Application \"app_2\" does not need more resources"
            }
          ]
        },
        "queue": {
          "queueId": queue-b
          "applications": [
            "application": {
              "applicationId": app_3
              "priority": 4
              "state": "skipped"
              "diagnostic": "Application \"app_3"\ is waiting for node-locality-delay"
            }
          ]
        }
      ]
    }
  },

  "allocation": {
    "container": {
      "containerId": container-5
      "state": "allocated"
    }
    "root": {
      "queueId": queue-root
```

```
              "state": "accepted"
              "children": [
                "queue": {
                  "queueId": queue-a
                  "applications": [
                    "application": {
                      "applicationId": app_6
                      "priority": 3
                      "state": "rejected"
                      "diagnostic": "Application \"app_6"\ over user limit"
                    }
                  ]
                },
                "queue": {
                  "queueId": queue-c
                  "applications": [
                    "application": {
                      "applicationId": app_7
                      "priority": 4
                      "state": "accepted"
                    }
                  ]
                }
              ]
            }
          }
        ]
      }
    }
  }
}
```

Allocation state: allocated/reserved/allocated-from-reserved/skipped

- **Allocated:** successfully allocate a new non-reserved container.
- **Reserved**: reserve the container for future allocation.
- **Allocated-from-reserved**: successfully allocate a new container from an existing reserved container.
- **Skipped:** Nothing allocated.

Queue/application state: accepted/rejected/skipped

- **Accepted**: container is allocated to sub-queues/applications or this queue/application
- **Rejected**: container could not be allocated to sub-queues or this application
  - (Diagnostic examples: over maximum-capacity for this queue OR over user limit OR cannot access this node)
- **Skipped**: Queue or application voluntarily give up to use the resource
  - (Diagnostic examples:doesn't need more resources OR waiting for node-locality-delay)

Things To Discuss

Scheduler activities of single node v.s. scheduler activities of all-nodes

Existing API only returns scheduler's activities of the single node, and it will be set on demand. We deliberately avoid collecting global scheduler's activities, it is not possible to store all scheduler's activities in memory for large clusters.