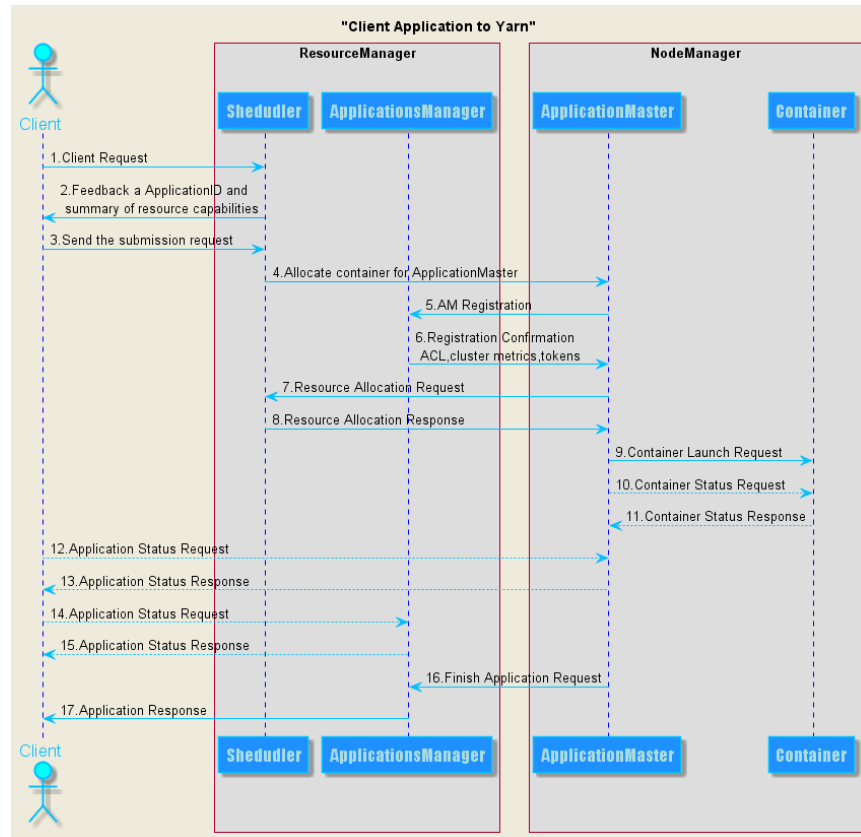


Contents

1	Yarn WorkFlow	2
2	Modification	2
2.1	AddApplication (34)	2
2.1.1	reject the steaming application without tags	2
2.2	AssignedContainer (48)	3
2.2.1	insulting the streaming task and mapreduce task	3
2.3	ReloadAllocationConfiguration	6
2.3.1	a thread reload the allocation periodically	6
2.3.2	get configuration form ddw-api	7
3	Associated Book	7
	Contributing	

1 Yarn WorkFlow



2 Modification

2.1 AddApplication (34)

2.1.1 reject the steaming application without tags

- modified code

```
//*****DDW modification(DDW )*****//
if(queue.getQueueName().startsWith("root.stream_")){
    Set<String> tags = rmApp.getApplicationTags();
    if(CollectionUtils.isEmpty(tags)){// spark.yarn.tags
        String msg = "[DDW] The streaming task require the 'spark.yarn.tags' when
        LOG.info(msg);
```

```

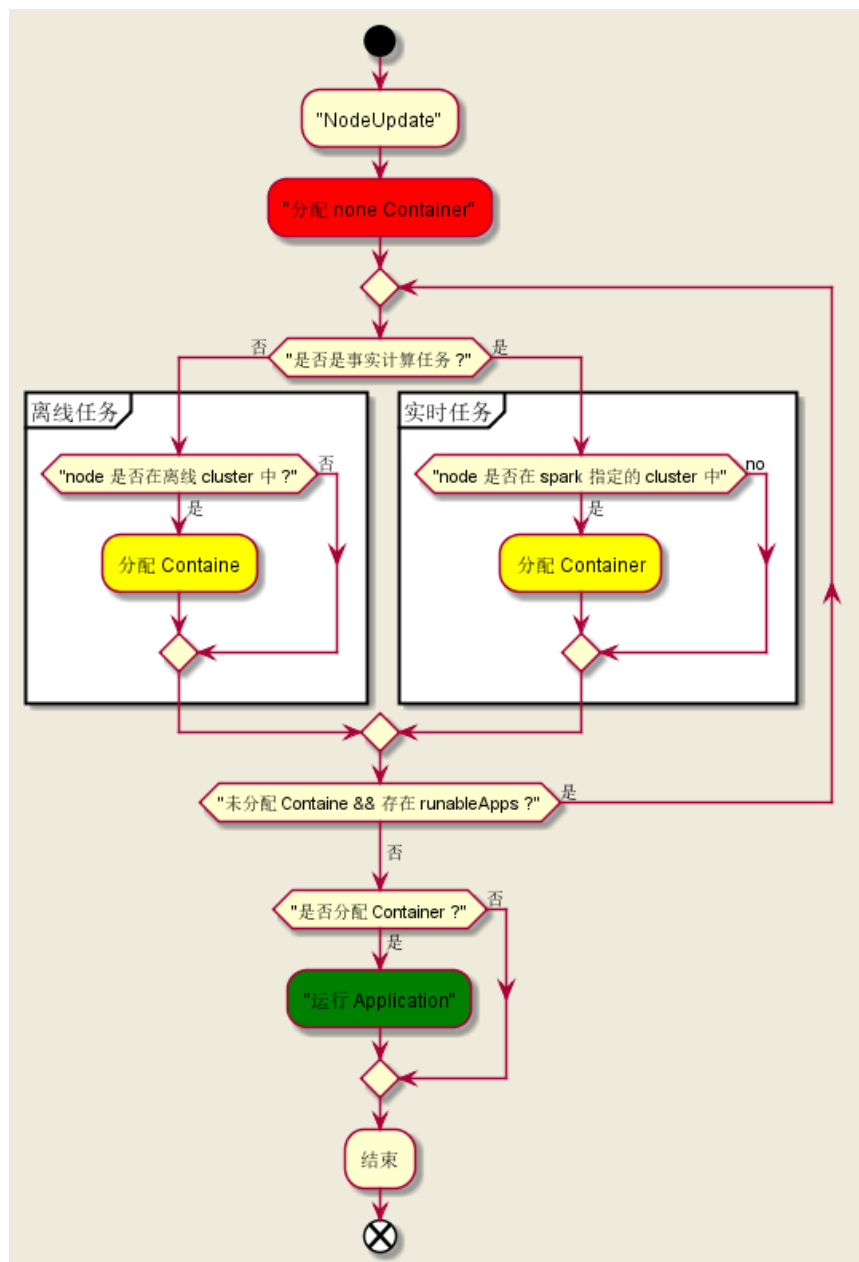
        rmContext.getDispatcher().getEventHandler()
            .handle(new RMAppeRejectedEvent(applicationId, msg));
        return;
    }else {
        for(String tag:tags){
            if(StringUtils.isEmpty(tag)){// spark.yarn.tags
                String msg = "[DDW] The 'spark.yarn.tags' have an empty tag when s
                LOG.info(msg);
                rmContext.getDispatcher().getEventHandler()
                    .handle(new RMAppeRejectedEvent(applicationId, msg));
                return;
            }
            if(!tag.startsWith("signature#")){//
                boolean isInNodes=false;
                for(NodeId nodeId:nodes.keySet()){
                    if(nodeId.getHost().equals(tag)){
                        isInNodes=true;
                        break;
                    }
                }
                if(!isInNodes) {
                    String msg = "[DDW] The 'spark.yarn.tags' have a host that is
                    LOG.info(msg);
                    rmContext.getDispatcher().getEventHandler()
                        .handle(new RMAppeRejectedEvent(applicationId, msg));
                    return;
                }
            }
        }
    }
}
//*****DDW modification(DDW )*****//

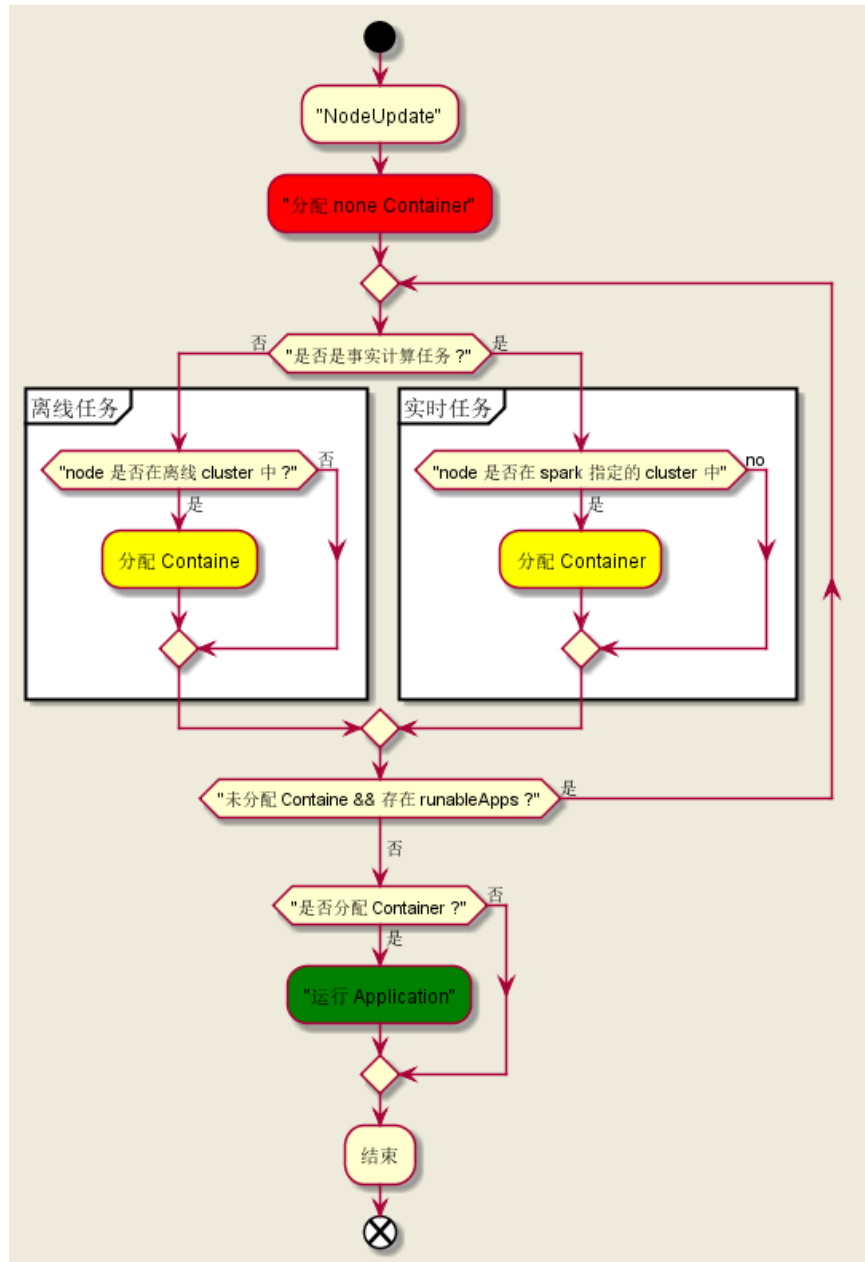
```

2.2 AssignedContainer (48)

2.2.1 insulting the streaming task and mapreduce task

- workflow





- modified code

```
//*****DDW modification(DDW )*****//
```

```
Map<String, Set<String>> groupMap = scheduler.getAllocationConfiguration().getGroupMap
```

```

for (FSAppAttempt sched : runnableApps) {

    if (sched.getQueueName().startsWith("root.stream_")) { // node
        RMApp rmApp = sched.getRMApp();
        Set<String> tags = rmApp.getApplicationTags();
        if (CollectionUtils.isEmpty(tags) || !tags.contains(node.getNodeName())) {
            continue;
        }
    } else { //
        if (!groupMap.get("MAPREDUCE").contains(node.getNodeName())) {
            continue;
        }
    }
}
}
//*****DDW modification(DDW )*****//

```

2.3 ReloadAllocationConfiguration

2.3.1 a thread reload the allocation periodically

- modified code

```

reloadThread = new Thread() {
    @Override
    public void run() {
        while (running) {
            //*****DDW modification(DDW )*****//
            try {
                reloadAllocations();
            } catch (Exception ex) {
                if (!lastReloadAttemptFailed) {
                    LOG.error("Failed to reload fair scheduler config file - " +
                        "will use existing allocations.", ex);
                }
                lastReloadAttemptFailed = true;
            }
            //*****DDW modification(DDW )*****//
            try {
                Thread.sleep(reloadIntervalMs);
            } catch (InterruptedException ex) {

```

```

        LOG.info(
            "Interrupted while waiting to reload alloc configuration"
        )
    }
}
};
reloadThread.setName("AllocationFileReloader");
reloadThread.setDaemon(true);

```

2.3.2 get configuration form ddw-api

- modified code

```

//*****DDW modification(DDW )*****//
    Map<String, Set<String>> groupMap = getGroupMapFromHttp();
    if(MapUtils.isEmpty(groupMap)|| CollectionUtils.isEmpty(groupMap.get("MAPREDUCE")))
        throw new ParserConfigurationException("[DDW] can not get the group configuration")
    //1.ResourceManager
    //2.ResourceManager reload
}

AllocationConfiguration info = new AllocationConfiguration(minQueueResources,
    maxQueueResources, queueMaxApps, userMaxApps, queueWeights,
    queueMaxAMShares, userMaxAppsDefault, queueMaxAppsDefault,
    queueMaxResourcesDefault, queueMaxAMShareDefault, queuePolicies,
    defaultSchedPolicy, minSharePreemptionTimeouts,
    fairSharePreemptionTimeouts, fairSharePreemptionThresholds, queueAcls,
    newPlacementPolicy, configuredQueues, nonPreemptableQueues, groupMap);
//*****DDW modification(DDW )*****//

```

3 Associated Book