Non-exclusive Node Partition Design

Authors: Wangda Tan, Mayank Bansal, Ram Venkatesh, Vinod Kumar Vavilapalli with inputs from Arun C Murthy, Jian He.
Last Modified: Feb 24, 2015

Use cases

Story of Partition

(1) Before Hadoop 2.6

Users could provision clusters by capacity and use preemption to enforce capacity guarantees.

For example, there are 4 departments (A, B, C, D) in a company, and each of them has a separate 10-nodes cluster. Clusters are separated, in another word, no one can use idle resources of other clusters.

To make better elasticity, A/B/C/D put their nodes together, build a big cluster, each of them has ¼ minimum capacity of the cluster, and one department can get more than ¼ cluster resource if there're any idle resources.

Preemption policy will enforce their minimum capacity (¼): if an under-satisfied department wants to get more resources, but the cluster is busy, preemption policy will reclaim resources from departments that are over-satisfied.

(2) The Hadoop 2.6 story with node labels

Node labels enable static partitioning of clusters by restricting specific workloads to specific nodes in the cluster.

For example, some day, A/B want to run some extra-large memory analytics applications, so A spends $40k and B spends $60k purchase a 10 node extra-large memory nodes, and make them managed by a the same YARN resource manager. A get 40% minimum capacity and B get 60% minimum capacity of the new nodes.

The 10 extra-large memory nodes partition is named “extra-large-memory”. Only applications from A/B that explicitly ask for “extra-large-memory” resource can get resources from the new partition. Of course, C/D CANNOT use any idle resource of the new partition

(3) This proposal

Expand node label definitions to enable sharing of nodes with labels when they are idle.

From the use case above, to make resources could be better utilized, the “extra-large-memory” partition can be shared to other applications don’t explicitly ask for “extra-large-memory” when the new partition has idle resources.

Application from A/B that explicitly asks for resources on the “extra-large-memory” partition will get preference to allocate on these 10 new nodes.

If the labeled-partition has idle resources, rules for sharing are:

- Since only A/B spent money on the new partition, idle resources will be first allocated to applications from A/B even if they don’t explicitly ask for “extra-large-memory”.
- If there are still idle resources in the labeled-partition, C/D can use it.

Preemption rules will enforce above rules of share in the labeled partition

- Application explicitly asks for “extra-large-memory” from A/B can preempt any other applications
- Application that DOESN’T explicitly ask for “extra-large-memory” from A/B can only preempt applications from C/D
- Applications from C/D CANNOT preempt any other applications from A/B in the labeled partition.

Requirements

Definitions

- Regular apps: apps without any specific requirements to run on specific partitions
- Special apps: apps requiring resources on non-DEFAULT partitions
- Partitioned nodes: Nodes belong to a non-DEFAULT partition

Requirements

- (P0) A node can belong to at most one partition. All nodes belong to a DEFAULT partition unless overridden.
- (P0) Regular apps should first be attempted to be scheduled on nodes in the DEFAULT partition
- (P0) Regular apps can only get partitioned nodes when their queues' non-labeled usage above minimum-configured capacity.
- (P0) Partitioned nodes should run containers from regular apps **ONLY when the node has idle resources**
- (P0) Regular apps need to be preempted from partitioned nodes if any special apps (whose queues are still under their partition limits) come back for new resources
- (P0) When a partitioned node becomes idle ("Idle" here means it has some free resource, and no special apps under resource-limits ask for resource, and reserved resource on the node should be considered), it should first be shared with queues that can access the partition
    - e.g. Queue A/B can access partition:hbase, C cannot. When a node has partition=hbase becomes idle, the node should first run non-partitioned apps from queue A/B before run apps from C.
- (P0) Since regular applications that run on labeled nodes can be evicted when apps that require labels are submitted, regular applications must preferentially be scheduled on nodes that don't have labels.
- (P1) Since containers of regular apps allocated on partitioned nodes will be likely to be preempted, we should have a way to disable such allocation
    - Maybe add a "preemptible" flag to resource request is one solution: RM will **try best** not allocate resource will likely to be preempted for such resource request.

API Changes Proposal (CLI/REST/Java API, etc.)

RM Admin CLI:

```
// add command is as same as before
-addToClusterNodeLabels a,b,c

// Set type of label
// When a node with sharable label, and it's idle, it can be shared with other queues.
// Preemption policy will reclaim resource when labeled queue comes back and asks for
// new resources.
-setNodeLabelsAttributes a.type=exclusive|shareable
```

RM cluster CLI:

It should be able to print info of each node-label.

YarnClient Java API:

In addition to getClusterNodeLabels, add a getClusterNodeLabelInfos, returns node-infos

ResourceManagerAdministrationProtocol Java API

Add one API:
setNodeLabelsAttributes(List<<String label_attribute, String value>> kvs) // now the only label_attribute is <label>.idle-timeout

Capacity Scheduler Related Changes:

Placement
Since regular applications that run on labeled nodes can be evicted when apps that require labels are submitted, regular applications must preferentially be scheduled on nodes that don't have labels.

Headroom
Since regular application could possibly get some resource from labeled partitions, we need to respect "relaxed" resource when computing headroom

User-limit
User-limit should consider configured-minimum only, so we only need consider configured-minimum capacity for both labeled/unlabeled partitions.

Preemption
Preemption policy should respect following rules:

- Labeled apps get the preference on Labeled nodes
- If there is no ask for labeled resources we can assign those nodes to regular apps
- If there is any future ask for those resources, we will preempt the regular apps and give them back to labeled apps.