Disk drive as a resource in YARN

By Swapnil Daingade, Santosh Marella and Yuliya Feldman

Problem Statement

Currently, the number of disks present on a node is not considered a factor while scheduling containers on that node. Containers are scheduled based only on memory and cpu cores available. All of these compete for I/O bandwidth to a relatively smaller number of disks present. The multiplexing of I/O across containers can lead to slower overall progress and sub-optimal resource utilization as containers starved for I/O bandwidth hold on to other resources like cpu and memory. This problem can be solved by considering the number of disks (spindles) present on a node as a resource and including it in deciding how many containers can be concurrently run on that node.

Proposed Solution

We provide the administrator a knob to control the number of containers that can be run on the node depending on the number of disks in addition to cpu cores, memory present on the node. Currently a NodeManager advertises its cpu and memory to the ResourceManager by specifying the following properties **yarn.nodemanager.resource.cpu-vcores** and **yarn.nodemanager.resource.memory-mb**. We add a new property for disk.

```
<property>
   <description>Number of spindles that can be allocated for containers.</description>
   <name>yarn.nodemanager.resource.io-spindles</name>
   <value><int value></value>
</property>
```

We modify the ResourceRequest class to include disk so that the ResourceManager and schedulers can use them while scheduling containers on nodes.

One of the common types of applications run on a YARN cluster are map-reduce applications. The resources that are consumed by different types of containers while running a map-reduce application on YARN are defined by the following properties.

Container Type	CPU	Memory
MRAppMaster	`yarn.app.mapreduce.am.resource.cpu-vcores`	`yarn.app.mapreduce.am.resource.mb`
Mapper	`mapreduce.map.cpu.vcores`	`mapreduce.map.memory.mb`
Reducer	`mapreduce.reduce.cpu.vcores`	`mapreduce.reduce.memory.mb`

We add two new properties for disk. Disk consumption for a map task is defined by **`mapreduce.map.disk.`** Disk consumption for a reduce task is defined by **`mapreduce.reduce.disk.`** Disk consumption for an instance of MRAppMaster is considered to be zero. The default resource consumption values for containers when running a map-reduce YARN application are defined as

Container Type	CPU core	Memory	Disk
MRAppMaster	1	2 GB	0.0
Mapper	1	1 GB	0.5 (or 2 mappers for every disk)
Reducer	1	3 GB	1.33 (or 3 reducers for every 4 disks)

Users are free to override the above mentioned properties depending on how I/O intensive their workloads are.

Implementation Details

Protobuf changes to yarn_protos.proto

Added disk as a double type.

```
message ResourceProto {
  optional int32 memory = 1;
  optional int32 virtual_cores = 2;
  optional double disks = 20;
}
```

Changes to org.apache.hadoop.yarn.api.records.Resource

Add new methods for Resource creation that includes disk

```
public static Resource newInstance(int memory, int vCores, double disks);
public abstract double getDisks();
public abstract void setDisks(double disks);
```

Changes to org.apache.hadoop.mapreduce.MRJobConfig

Add defaults for disk consumed by map and reduce containers
public static final String MAP_DISK = "mapreduce.map.disk";
public static final double DEFAULT_MAP_DISK = 0.5;
public static final String REDUCE_DISK = "mapreduce.reduce.disk";
public static final double DEFAULT_REDUCE_DISK = 1.33;

Changes to org.apache.hadoop.yarn.conf.YarnConfiguration

Add defaults for disk used by ResourceManager and NodeManager
public static final String RM_SCHEDULER_MINIMUM_ALLOCATION_DISKS =
  YARN_PREFIX + "scheduler.minimum-allocation-disks";
public static final double DEFAULT_RM_SCHEDULER_MINIMUM_ALLOCATION_DISKS = 0.0;

public static final String RM_SCHEDULER_MAXIMUM_ALLOCATION_DISKS =
  YARN_PREFIX + "scheduler.maximum-allocation-disks";
public static final double DEFAULT_RM_SCHEDULER_MAXIMUM_ALLOCATION_DISKS = 4.0;

/** Number of disks which can be allocated for containers.*/
public static final String NM_DISKS = NM_PREFIX + "resource.io-spindles";
public static final double DEFAULT_NM_DISKS = 0;

Changes to Schedulers

- Make changes to FairScheduler, CapacityScheduler and FifoScheduler to accomodate disk as a resource.
- Make changes to org.apache.hadoop.yarn.util.resource.Resources to do resource arithmetic with support for disks.
- Add new resource calculator called **DiskBasedResourceCalculator** derived from DefaultResourceCalculator that takes disks into consideration. This is to be the default resource calculator for Capacity Scheduler. Modify the following methods to accommodate disks.

public class DiskBasedResourceCalculator extends DefaultResourceCalculator {
    public Resource divideAndCeil(Resource numerator, int denominator);
    public Resource normalize(Resource r, Resource minimumResource, Resource maximumResource, Resource stepFactor);
    public Resource roundUp(Resource r, Resource stepFactor);
    public Resource roundDown(Resource r, Resource stepFactor);
    public Resource multiplyAndNormalizeUp(Resource r, double by, Resource stepFactor);

```
        public Resource multiplyAndNormalizeDown(Resource r, double by, Resource
        stepFactor);
}
```

- Add new resource calculator called **DiskBasedDominantResourceCalculator** derived from DominantResourceCalculator that takes disk into consideration while scheduling based on dominant resources. Modify the following methods to accommodate disks.

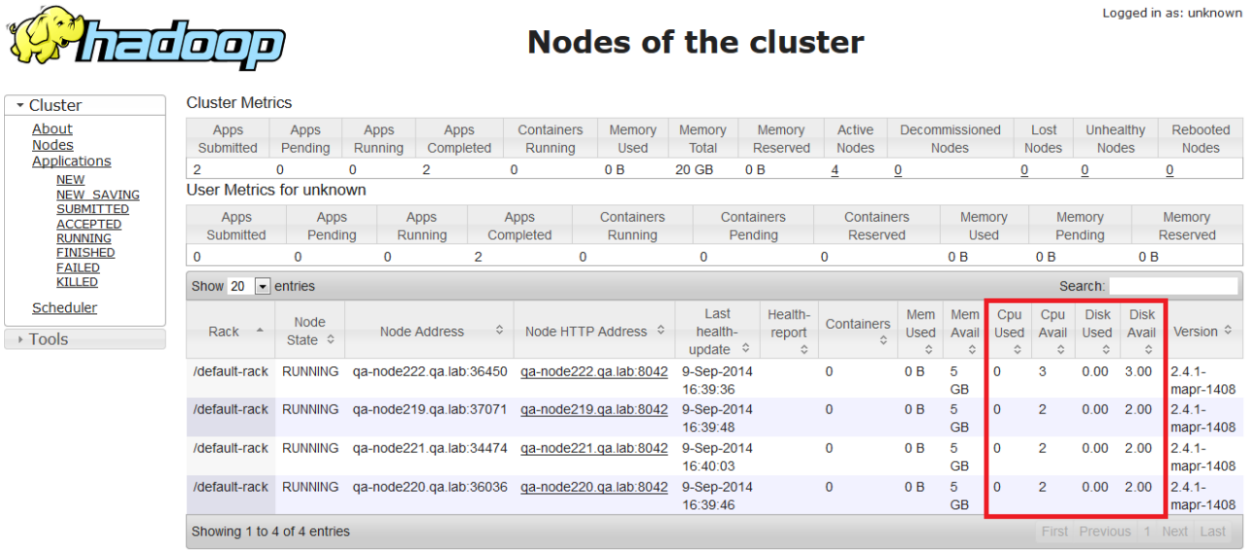
```
public class DiskBasedDominantResourceCalculator extends DominantResourceCalculator {
        protected float getResourceAsValue(Resource clusterResource, Resource resource,
        boolean dominant)
        public int computeAvailableContainers(Resource available, Resource required)
        public float ratio(Resource a, Resource b);
        public Resource divideAndCeil(Resource numerator, int denominator)
        public Resource normalize(Resource r, Resource minimumResource, Resource
        maximumResource, Resource stepFactor)
        public Resource roundUp(Resource r, Resource stepFactor)
        public Resource roundDown(Resource r, Resource stepFactor)
        public Resource multiplyAndNormalizeUp(Resource r, double by, Resource
stepFactor)
        public Resource multiplyAndNormalizeDown(Resource r, double by, Resource
        stepFactor)
}
```

- Add support for disk in queue allocations file for Fair Scheduler configuration.

```
<queue name="label_12">
        <minResources>10240 mb, 4 vcores, 2 disks</minResources>
        <maxResources>19456 mb,10 vcores, 5 disks</maxResources>
        <maxRunningApps>20</maxRunningApps>
        <weight>3.0</weight>
        <schedulingPolicy>fair</schedulingPolicy>
        <queue name="label_12_OR">
                <minResources>1024 mb,1 vcores, 1 disks</minResources>
        </queue>
        <queue name="label_12_TEST">
                <minResources>1024 mb,1 vcores, 1 disks</minResources>
        </queue>
</queue>
```

UI Changes

Make changes to the ResourceManager UI including the Nodes page and the scheduler pages to include disk.



Logged in as: unknown

Nodes of the cluster

- Cluster
  - About
  - Nodes
  - Applications
    - NEW
    - NEW_SAVING
    - SUBMITTED
    - ACCEPTED
    - RUNNING
    - FINISHED
    - FAILED
    - KILLED
  - Scheduler
- Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
2	0	0	2	0	0 B	20 GB	0 B	4	0	0	0	0

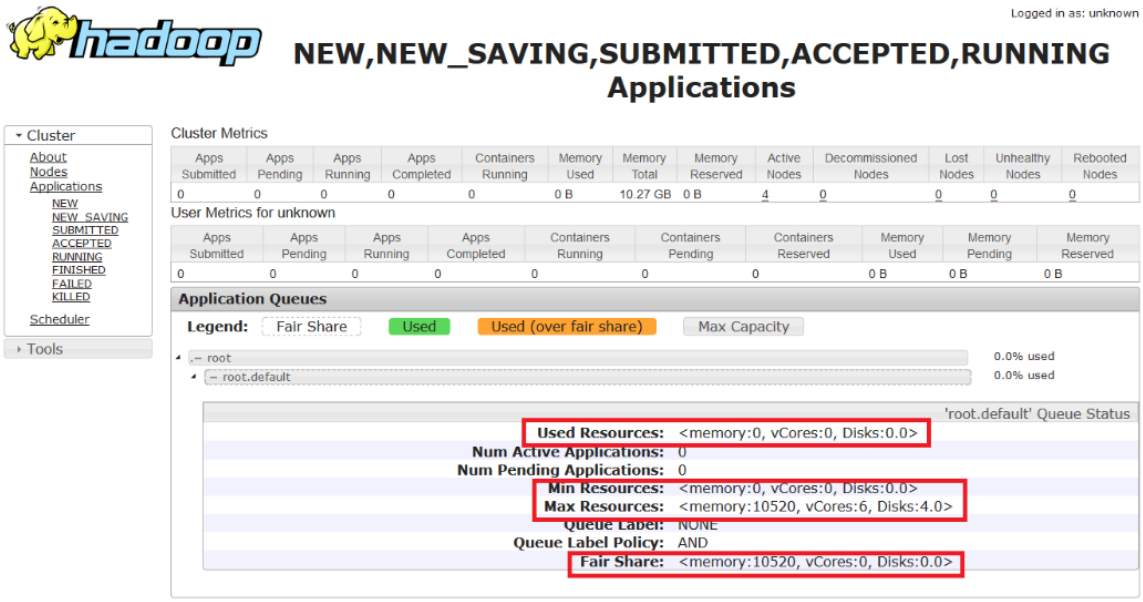
User Metrics for unknown

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Containers Reserved	Memory Used	Memory Pending	Memory Reserved
0	0	0	2	0	0	0	0 B	0 B	0 B

Show 20 entries Search:

Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Mem Used	Mem Avail	Cpu Used	Cpu Avail	Disk Used	Disk Avail	Version
/default-rack	RUNNING	qa-node222.qa.lab:36450	qa-node222.qa.lab:8042	9-Sep-2014 16:39:36		0	0 B	5 GB	0	3	0.00	3.00	2.4.1-mapr-1408
/default-rack	RUNNING	qa-node219.qa.lab:37071	qa-node219.qa.lab:8042	9-Sep-2014 16:39:48		0	0 B	5 GB	0	2	0.00	2.00	2.4.1-mapr-1408
/default-rack	RUNNING	qa-node221.qa.lab:34474	qa-node221.qa.lab:8042	9-Sep-2014 16:40:03		0	0 B	5 GB	0	2	0.00	2.00	2.4.1-mapr-1408
/default-rack	RUNNING	qa-node220.qa.lab:36036	qa-node220.qa.lab:8042	9-Sep-2014 16:39:46		0	0 B	5 GB	0	2	0.00	2.00	2.4.1-mapr-1408

Showing 1 to 4 of 4 entries First Previous 1 Next Last



Logged in as: unknown

NEW,NEW_SAVING,SUBMITTED,ACCEPTED,RUNNING Applications

- Cluster
  - About
  - Nodes
  - Applications
    - NEW
    - NEW_SAVING
    - SUBMITTED
    - ACCEPTED
    - RUNNING
    - FINISHED
    - FAILED
    - KILLED
  - Scheduler
- Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
0	0	0	0	0	0 B	10.27 GB	0 B	4	0	0	0	0

User Metrics for unknown

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Containers Reserved	Memory Used	Memory Pending	Memory Reserved
0	0	0	0	0	0	0	0 B	0 B	0 B

Application Queues

Legend: Fair Share | Used | Used (over fair share) | Max Capacity

- root 0.0% used
  - root.default 0.0% used

'root.default' Queue Status

Used Resources:	<memory:0, vCores:0, Disks:0.0>
Num Active Applications:	0
Num Pending Applications:	0
Min Resources:	<memory:0, vCores:0, Disks:0.0>
Max Resources:	<memory:10520, vCores:6, Disks:4.0>
Queue Label:	NONE
Queue Label Policy:	AND
Fair Share:	<memory:10520, vCores:0, Disks:0.0>