- **Sunil G and Prabhu Joseph** with inputs from **Varun Vasudev & Jian He**

Shuffle Metrics: Design

Currently, poor networks can cause a slowdown in job performance which can be hard to analyze. We can use the shuffle service to expose network speeds across nodes and racks to help debug slow nodes in a cluster.

The existing shuffle service gives various informations such as total number of bytes sent and total connections etc. And the shuffle service can be easily extended to support per host shuffle metrics for per-host shuffle speed metrics.

Requirement

Using existing shuffle module of Nodemanager, one can do the following.

- Calculate metrics such as shuffle bytes sent/received and its time taken per host level. Speed could be measured as bytes per milli seconds.
- Store this metric in a time series based data structure per host to get network speed indications (of all other hosts) over a period of time
- These metrics could be exposed via JMX, REST etc.

Design

A high level design approach.

Step 1) Identify per-host shuffle speed metrics

NodeManager's shuffle service module will be able to identify the host machines from which shuffle requests are coming. We could identify number of bytes sent per host level and its time taken for sending that data. From these two metrics, per-host level shuffle speed can be calculated for any specific time duration.

This means that each NM (shuffle module) will store shuffle speed metric against all the other nodes (NM's). However this 1 to n-1 ("n" number of nodes in cluster) storage will only give aggregate shuffle speed of host NM against all the other nodes. It is still possible that one connection to a specific NM became bad, and aggregate shuffle speed will not immediately reflect this dip in speed (it may take more iterations to bring down average speed down). Thus we also need few last connection's data.

Details of this store is updated in next section.

Others Considerations:

Later, user could submit filters (different node id's) and shuffle metrics could be fetched for these configured nodes from a given host. This could help to locate a suspicious node faster by verifying its network speed.

Step 2) Store per-host shuffle metrics by using Hadoop Metrics system

Admin could make meaningful derivations from shuffle speed metrics when it's compared against past data such as "last 10 connections shuffle speed" against a specific host. This can easily detect a dip in network speed between two nodes.

Level-1 Metrics

For example given a host, it will be receiving shuffle connection requests from other nodes in the cluster. For each of these nodes, there will be a metrics entry to store the shuffle speed metrics from host machine to this target node.
Hadoop Metric System's Histogram could be used here. ContainerMetrics is already making use of this to get various percentiles. Mostly we are looking here is for rolling average (mean) in general with max and min.

Level-2 Metrics

There can be multiple connection requests from a single node itself, hence we will be able to collect few shuffle metrics samples across last N number of connections. And this can give us a trend of network speed against that node from this host.

Few assumptions

- Shuffle speed metric unit will be in Bytes per MilliSeconds.
- From each NodeManager, we can get shuffle speed metrics against all other nodes
    - Total average speed

- Average speed for last 10 connections
- Max/Min speed

Note

- Since each NM will store metric against each other nodes, we are going to have N x N storage size for this new metrics. Memory test will be done to verify the mem usage.