HADOOP CLUSTERS: AN INTRODUCTION

Hadoop clusters 101

In talking about Hadoop clusters, first we need to define two terms: **cluster** and **node**. A cluster is a collection of nodes. A node is a process running on a virtual or physical machine or in a container. We say process because a code would be running other programs beside Hadoop.

When Hadoop is not running in cluster mode, it is said to be running in **local mode**. That would be suitable for, say, installing Hadoop on one machine just to learn it. When you run Hadoop in local node it writes data to the local file system instead of HDFS (Hadoop Distributed File System).

Hadoop is a master-slave model, with one master (albeit with an optional High Availability hot standby) coordinating the role of many slaves. Yarn is the resource manager that coordinates what task runs where, keeping in mind available CPU, memory, network bandwidth, and storage.

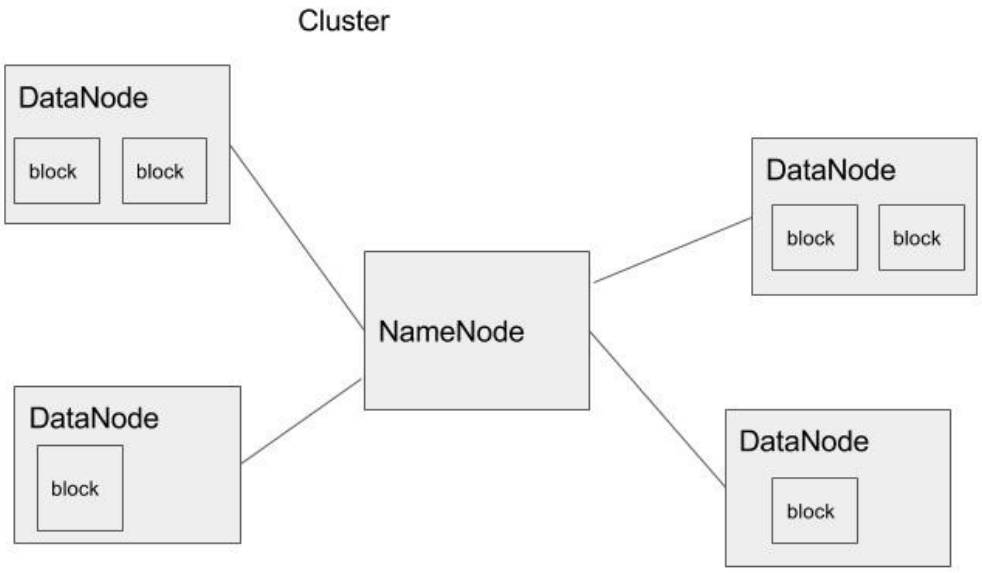
One can scale out a Hadoop cluster, which means add more nodes. Hadoop is said to be **linearly scalable**. That means for every node you add you get a corresponding boost in throughput. More generally if you have n nodes then adding 1 mode give you (1/n) additional computing power. That type of distributed computing is a major shift from the days of using a single server where when you add memory and CPUs it produces only a marginal increase in throughout.

(This article is part of our Hadoop Guide. Use the right-hand menu to navigate.)

Datanode and Namenode

The NameNode is the Hadoop master. It consults with DataNodes in the cluster when copying data or running mapReduce operations. It is this design that lets a user copy a very large file onto a Hadoop mount point like /data. Files copied to /data exist as blocks on different DataNodes in the cluster. The collection of DataNodes is what we call the HDFS.

This basic idea is illustrated below.



Yarn

Apache Yarn is a part of Hadoop that can also be used outside of Hadoop as a standalone resource manager. NodeManager takes instructions from the Yarn scheduler to decide which node should run which task. Yarn consists of two pieces: ResourceManager and NodeManager. The NodeManager reports to the ResourceManager CPU, memory, disk, and network usage so that the ResourceManager can decide where to direct new tasks. The ResourceManager does this with the Scheduler and ApplicationsManager.

Adding nodes to the cluster

Adding nodes to a Hadoop cluster is as easy as copying the server name to $HADOOP_HOME/conf/slaves file then starting the DataNode daemon on the new node.

Communicating between nodes

When you install Hadoop, you enable ssh and create ssh keys for the Hadoop user. This lets Hadoop communicate between the nodes by using RCP (remote procedure call) without having to enter a password. Formally this abstraction on top of the TCP protocol is called Client Protocol and the

DataNode Protocol. The DataNodes send a heartbeat to the NameNode to let it know that they are still working.

Hadoop nodes configuration

Hadoop configuration is fairly easy in that you do the configuration on the master and then copy that and the Hadoop software directly onto the data nodes without needed to maintain a different configuration on each.

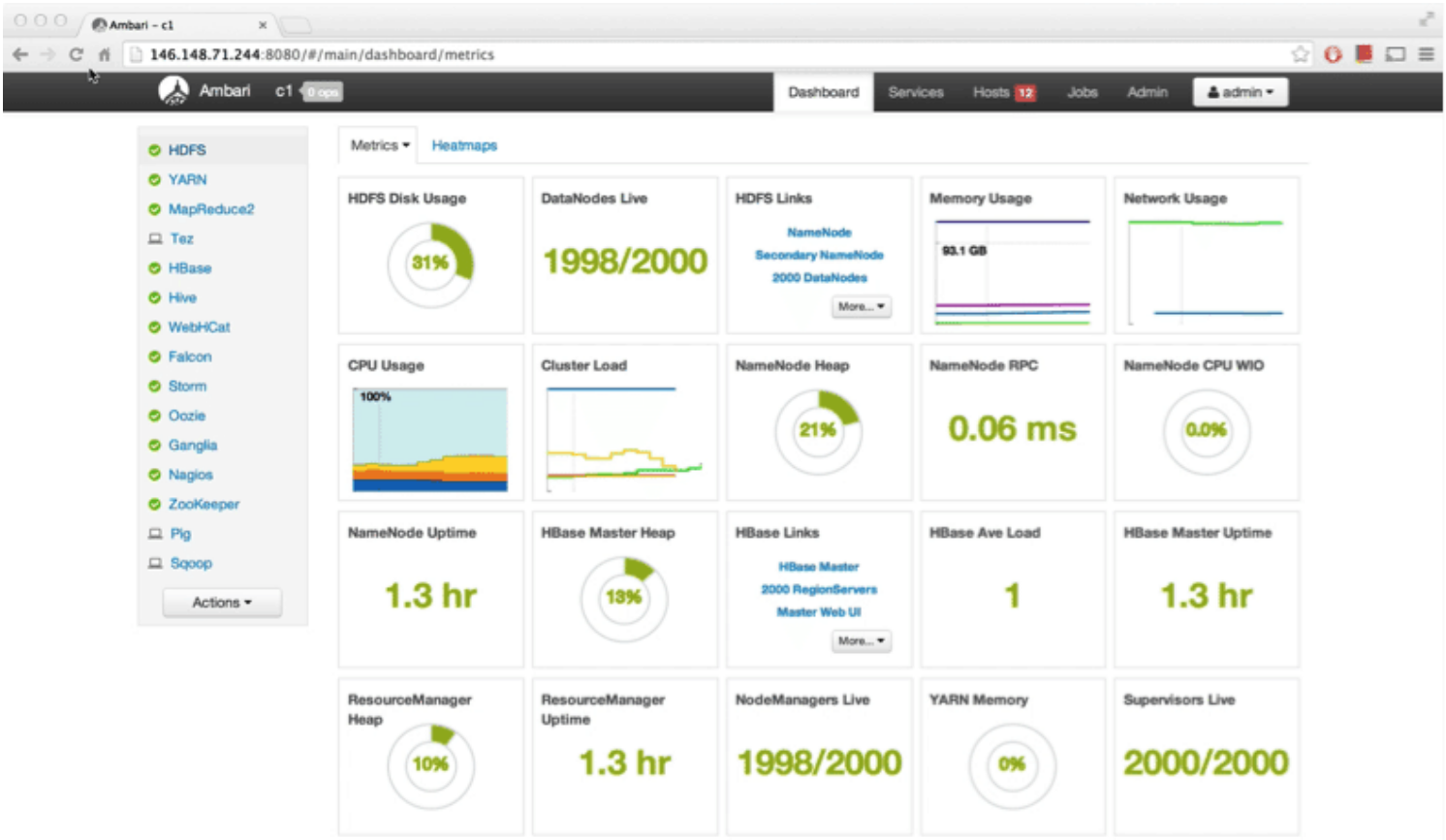
The main Hadoop configuration files are core-site.xml and hdfs-site.xml. This is where you set the port number where Hadoop files can be reached, the replication factor (i.e, the number of replicates or number of copies of data blocks to keep),the location of the FSImage (keeps track of changes to the data files), etc. You can also configure authentication there to put security into the Hadoop cluster,which by default has none.

Cluster management

Hadoop has a command line interface as well an API. But there is no real tool for orchestration (meaning managing, including monitoring) and installing new machines.

There are some options for that. One is Apache Ambari. It is used and promoted by certain Hadoop clouds like Hortonworks.

Here is a view of the Ambari dashboard from HortonWorks:



As you can see it has a lot of metrics and tools not offered by the basic, rather simple, Hadoop and Yarn web interfaces.

It exposes its services as REST web APIs. So other vendors have added it to their operations

platform, like the Microsoft Systems Center and Teradata.

With Ambari instead of typing stop-dfs.sh on each data node you can use the rolling restarts feature to reboot each machine when you want to implement some kind of change. As you can image if you have more than a handful of machines then doing this from the command line would be time consuming.

Ambari also helps to manage more than one cluster at the same time. And to build out each you can use the Ambari Blueprint wizard to layout where you want NameNodes, DataNodes, and provide configuration details. This is also useful as you can build development or test clusters and automate the build of those. You only run the wizard one time and then it saves it as an API so that you can script building new clusters.