HDFS Components

Every HDFS cluster is comprised of one or two NameNodes, and as many DataNodes as your IT budget will allow. What makes a server into a NameNode or a DataNode is an eponymous Java process running as a UNIX daemon on it. What determines whether you are running just one or two NameNodes is your tolerance for HDFS outages. With just one NameNode - if it goes down your whole HDFS deployment is unavailable (even though the DataNodes may be running just fine). With two NameNodes running in an Active/Standby configuration, the Standby can take over in cases where the Active one fails or needs to be brought down for maintenance. This is called a High Availability (HA) configuration and we will cover it in more details later in this chapter.

- **NameNode (one or two per cluster)**
  - Represents a single filesystem namespace rooted at **/**
  - Is the master service of HDFS
  - Determines and maintains how the chunks of data are distributed across the DataNodes
  - Actual data never resides here, only metadata (e.g., maps of where blocks are distributed).
- **DataNode (as many as you want per cluster)**
  - Stores the chunks of data, and is responsible for replicating the chunks across other DataNodes
  - Default number of replicas on most clusters is 3 (but it can be

changed on a per-file basis)
- Default block size on most clusters is 128MB.

A default replication factor of 3 for every block is good for two reasons. First of all, this level of replication means you can afford to have one node failing, and still have fault tolerance with two nodes left. Secondly, it allows for the third copy to be placed on a different rack in the datacenter, thus maximizing the chances that at least one copy of the block will survive all major catastrophic events. As such, a replication factor of 3 for every block can be a good starting point to make sure that your mission-critical applications are up and running in crises situations. But, for an extra level of protection (or to make sure that a copy of a block is available locally on all nodes in a data center - thus maximizing the locality of data), it may be better to have a higher level of replication factor configured. The good news is that, with HDFS, you don't have to decide upfront. It allows the configuration of the replication factor on a per-file or a per-folder basis.

Learn About Verified Certificates

© All Rights Reserved