

HDFS Federation

Key Terminologies

➤ **Namespace**

Namespace manages directories, files and blocks.

Consists of directories, files and blocks.

It supports all the namespace related file system operations such as create, delete, modify and list files and directories.

simply we can say Namespace means file system hierarchy structure.

➤ **Block Pool**

Block pool is nothing but set of blocks belonging to a specific Namespace

➤ **Namespace Volume**

Namespace volume is nothing but namespace along with its block pool.

Limitations of Current HDFS(prior federation)

- ✓ The namespace is not scalable like DataNodes. Hence, we can have only that number of DataNodes in the cluster that a single NameNode can handle.
- ✓ The performance of the entire Hadoop System depends on the throughput of the NameNode. Therefore, entire performance of all the HDFS operations depends on how many tasks the NameNode can handle at a particular time.
- ✓ The NameNode stores the entire namespace in RAM for fast access. This leads to limitations in terms of memory size i.e. The number of namespace objects (files and blocks) that a single namespace server can cope up with.
- ✓ There is **no isolation** among tenant organization that are using the cluster.

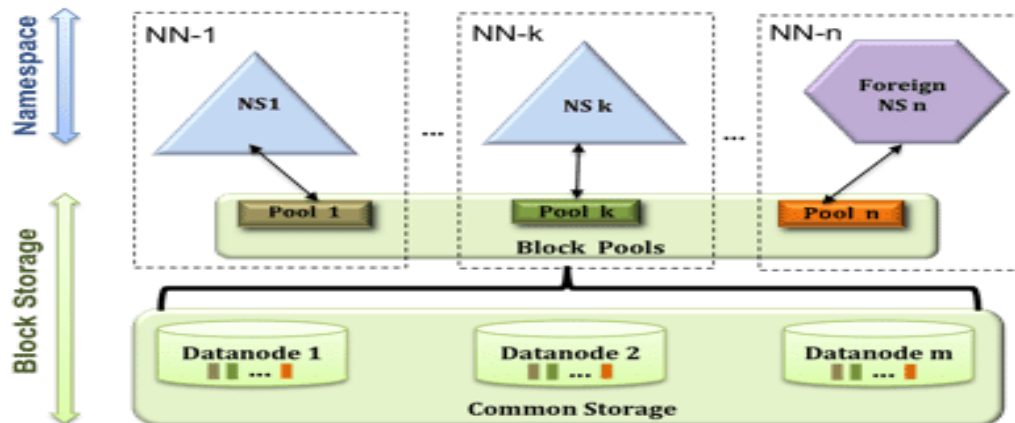
Federation

The prior HDFS architecture allows only a single namespace for the entire cluster. In that configuration, a single Namenode manages the namespace. HDFS Federation addresses this limitation by adding support for multiple Namenodes/namespaces to HDFS.

In order to scale the name service horizontally, federation uses multiple independent Namenodes/namespaces. The Namenodes are federated; the Namenodes are independent

and do not require coordination with each other. The Datanodes are used as common storage for blocks by all the Namenodes. Each Datanode registers with all the Namenodes in the cluster. Datanodes send periodic heartbeats and block reports. They also handle commands from the Namenodes.

Federation configuration is **backward compatible** and allows existing single Namenode configurations to work without any change



A Block Pool is a set of blocks that belong to a single namespace. Datanodes store blocks for all the block pools in the cluster. Each Block Pool is managed independently. This allows a namespace to generate Block IDs for new blocks without the need for coordination with the other namespaces. A Namenode failure does not prevent the Datanode from serving other Namenodes in the cluster.

A ClusterID identifier is used to identify all the nodes in the cluster. When a Namenode is formatted, this identifier is either provided or auto generated. This ID should be used for formatting the other Namenodes into the cluster.

we can get the cluster id from namenode ui or get it from namenode.name.dir loaction.

In this location we have a file called VERSION it contains all the information like cluster and blockpool id etc..

```
[hduser@namenode current]$ pwd
/usr/local/hadoop_store/hdfs/namenode/current
[hduser@namenode current]$ ls -lrt
total 1040
-rw-rw-r--. 1 hduser hduser      207 Apr 26 02:09 VERSION
-rw-rw-r--. 1 hduser hduser     353 Apr 26 02:09 fsimage_00000000000000000000
-rw-rw-r--. 1 hduser hduser      62 Apr 26 02:09 fsimage_00000000000000000000.md5
-rw-rw-r--. 1 hduser hduser       2 Apr 26 02:11 seen_txid
-rw-rw-r--. 1 hduser hduser 1048576 Apr 26 02:20 edits_inprogress_00000000000000000001
[hduser@namenode current]$ cat VERSION
#Fri Apr 26 02:09:16 PDT 2019
namespaceID=1946297944
clusterID=CID-384797d3-94d5-45ff-ad3a-2e324d119d3e
cTime=0
storageType=NAME_NODE
blockpoolID=BP-360120198-192.168.112.161-1556269756449
layoutVersion=-63
[hduser@namenode current]$
```

Overview 'namenode:9000' (active)

Started:	Fri Apr 26 02:11:03 PDT 2019
Version:	2.7.3, rbaa91f7c6bc9cb92be5982de4719c1c8af91ccff
Compiled:	2016-08-18T01:41Z by root from branch-2.7.3
Cluster ID:	CID-384797d3-94d5-45ff-ad3a-2e324d119d3e
Block Pool ID:	BP-360120198-192.168.112.161-1556269756449

⤵

Key concepts from Federation

- ✓ In HDFS Federation Architecture, we have horizontal scalability of name service. Therefore, we have multiple NameNodes which are federated, i.e. Independent from each other.
- ✓ The DataNodes are present at the bottom i.e. Underlying storage layer.
- ✓ Each DataNode registers with all the NameNodes in the cluster.
- ✓ The DataNodes transmit periodic heartbeats, block reports and handles commands from the NameNodes.

Benefits of HDFS Federation

❖ Isolation

In HDFS federation different categories of application and users can be isolated to different namespaces by using many namenodes.

❖ Namespace Scalability

In federation many namenodes horizontally scales up in the filesystem namespace.

❖ Performance

File system throughput is not limited by a single Namenode

We can improve Read/write operation throughput by adding more namenodes.

ViewFs

The View File System (ViewFs) provides a way to manage multiple Hadoop file system namespaces (or namespace volumes). It is particularly useful for clusters having multiple namenodes, and hence multiple namespaces, in HDFS Federation. ViewFs is analogous to *client side mount tables* in some Unix/Linux systems. ViewFs can be used to create personalized namespace views and also per-cluster common views.