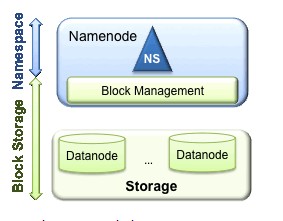
HDFS Federation

hdfs主要包含两层结构：

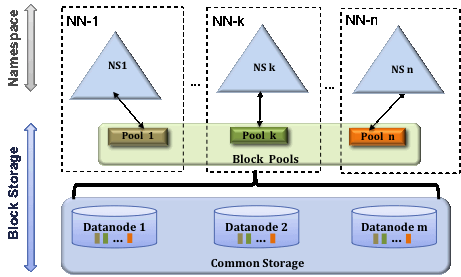


1. NameSpace，管理目录、文件和数据块，支持常见的文件系统操作，如创建、修改、删除文件等
2. Block Storage Service，包含两个部分：

* Block Management，在NameNode中运行，通过心跳维护集群中dn的基本关系；根据心跳中的block reports维护blocks的位置；支持数据块基本操作（创建、删除及管理副本的复制和存放）
* Physical Storage，存储实际的数据块并提供对数据块的读写服务。这部分分别在DN上实现。

在上述的架构中，整个集群中存在一个namespace，而该namespace被仅有的一个Active NN管理。这个架构的最大的问题是HDFS的扩展性，其底层存储（DN）是水平扩展，当集群存储空间不够时，可简单的添加机器进行水平扩展，但namespace只存放在单个NN上，内存中存储这个分布式文件系统的元数据信息，单台集群的服务能力是固定的，这限制了集群中数据块、文件和目录的数目，而且单NN的吞吐量比较低。

为了解决这个问题，HDFS中开发了Federation模式，其原理如下图所示：



Federation中，使用多个NN来使HDFS的Namespace能够水平扩展，NN之间相互独立并且不需要相互协调。每个DN被所有的NN用作公共存储块的地方，每个DN都会向所在集群中所有的NN注册，并且会周期性的发送心跳和块信息报告，同时处理来自NN的指令。

# 1.概念

1. Block Pool

属于单个Namespace的一组Block。DN为每一个Block Pool存储块，可以将DN看做是一个物理概念，而Block Pool是一个重新将Block划分的逻辑概念，同一个DN可以存着属于多个block pool的多个块。

Block Pool由属于同一个Namespace的数据块组成，内部自治，不会与其它Block Pool交互。因此一个NameNode挂掉，不会影响其他NN。

1. NameSpace Volume

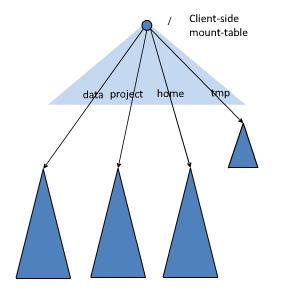
NN的namespace和对应的block pool一起被称为namespace volume，它是管理的基本单位，当一个NN被删除，其所有的datanode上对应的block pool也要相应删除。

1. ClusterID

在HDFS Federation中使用ClusterID来区分集群中的每个节点，当格式化NN时，这个ClusterID会自动生成，也可以手动指定。

1. Multi Namespaces

在集群中多个命名空间，相互之间会进行数据的共享和访问。使用全局唯一的命名空间是解决数据共享和访问的一种方法。可以使用Client Side Mount Table做到数据的共享和访问：



深色三角型代表一个独立的命名空间，浅色代表从客户端角度去访问下方的子命名空间。各个深色的命名空间Mount到浅色的表中，客户可以根据不同的挂载点来访问不同的命名空间，这如同在Linux系统中访问不同的挂载点是一样。推荐使用ViewFs来模拟Unix/Linux系统中的Client side mount table。

# 2.Federation的配置

启用HDFS Federation后，在集群中启动多个可用的namenode，可以根据需要分配这些NN，例如User1使用nn1，User1使用nn2。需要配置的参数：

1. 在hdfs-site.xml配置中，增加dfs.nameservice参数
2. 根据配置的NameServiceID，配置相对应的参数，如下表：

|  |  |
| --- | --- |
| Namenode | dfs.namenode.rpc-address  dfs.namenode.servicerpc-address  dfs.namenode.http-address  dfs.namenode.https-address  dfs.namenode.keytab.file  dfs.namenode.name.dir  dfs.namenode.edits.dir  dfs.namenode.checkpoint.dir  dfs.namenode.checkpoint.edits.dir |
| Secondary Namenode | dfs.namenode.secondary.http-address  dfs.secondary.namenode.keytab.file |
| BackupNode | dfs.namenode.backup.address  dfs.secondary.namenode.keytab.file |

两个NameSpace的Federation的配置示例如下(hdfs-site.xml)：

*<property>*

*<name>dfs.nameservices</name>*

*<value>ns1,ns2</value>*

*</property>*

ns1配置，启动两个NameNode作为HA：

*<property>*

*<name>dfs.namenode.http-address.ns1.nn1</name>*

*<value>cmhhost1.novalocal:50070</value>*

*</property>*

*<property>*

*<name>dfs.namenode.http-address.ns1.nn2</name>*

*<value>cmhhost2.novalocal:50070</value>*

*</property>*

*<property>*

*<name>dfs.namenode.https-address.ns1.nn1</name>*

*<value>cmhhost1.novalocal:50470</value>*

*</property>*

*<property>*

*<name>dfs.namenode.https-address.ns1.nn2</name>*

*<value>cmhhost2.novalocal:50470</value>*

*</property>*

*<property>*

*<name>dfs.namenode.rpc-address.ns1.nn1</name>*

*<value>cmhhost1.novalocal:8020</value>*

*</property>*

*<property>*

*<name>dfs.namenode.rpc-address.ns1.nn2</name>*

*<value>cmhhost2.novalocal:8020</value>*

*</property>*

*<property>*

*<name>dfs.namenode.servicerpc-address.ns1.nn1</name>*

*<value>cmhhost1.novalocal:8070</value>*

*</property>*

*<property>*

*<name>dfs.namenode.servicerpc-address.ns1.nn2</name>*

*<value>cmhhost2.novalocal:8070</value>*

*</property>*

*<property>*

*<name>dfs.namenode.shared.edits.dir</name> <value>qjournal://cmhhost2.novalocal:8485;cmhhost1.novalocal:8485;cmhhost6.novalocal:8485/ns1</value>*

*</property>*

NS2的相关配置，启动两个NameNode作为HA

*<property>*

*<name>dfs.namenode.http-address.ns2.nn3</name>*

*<value>cmhhost3.novalocal:50070</value>*

*</property>*

*<property>*

*<name>dfs.namenode.http-address.ns2.nn4</name>*

*<value>cmhhost6.novalocal:50070</value>*

*</property>*

*<property>*

*<name>dfs.namenode.https-address.ns2.nn3</name>*

*<value>cmhhost3.novalocal:50470</value>*

*</property>*

*<property>*

*<name>dfs.namenode.https-address.ns2.nn4</name>*

*<value>cmhhost6.novalocal:50470</value>*

*</property>*

*<property>*

*<name>dfs.namenode.rpc-address.ns2.nn3</name>*

*<value>cmhhost3.novalocal:8020</value>*

*</property>*

*<property>*

*<name>dfs.namenode.rpc-address.ns2.nn4</name>*

*<value>cmhhost6.novalocal:8020</value>*

*</property>*

*<property>*

*<name>dfs.namenode.servicerpc-address.ns2.nn3</name>*

*<value>cmhhost3.novalocal:8070</value>*

*</property>*

*<property>*

*<name>dfs.namenode.servicerpc-address.ns2.nn4</name>*

*<value>cmhhost6.novalocal:8070</value>*

*</property>*

*<property>*

*<name>dfs.namenode.shared.edits.dir</name> <value>qjournal://cmhhost2.novalocal:8485;cmhhost1.novalocal:8485;cmhhost6.novalocal:8485/ns2</value>*

*</property>*

# 3.ViewFS

可以使用View File System(ViewFS)来管理Hadoop中的多个Namespaces，ViewFS类似于Unix/Linux系统中的client side mount table。上例中的viewFs的配置如下：

*<property>*

*<name>fs.defaultFS</name>*

*<value>viewfs://ns</value>*

*</property>*

*<property>*

*<name>fs.viewfs.mounttable.ns.link./ns1</name>*

*<value>hdfs://ns1/ns1</value>*

*</property>*

*<property>*

*<name>fs.viewfs.mounttable.ns.link./ns2</name>*

*<value>hdfs://ns2/ns2</value>*

*</property>*

上面的配置，将一个全局逻辑目录映射到具体的namenode目录中，如将/ns1目录映射到hdfs://ns1/ns1目录中，将/ns2目录映射到hdfs://ns2/ns2目录下，修改配置后，查看如下：

*$ hdfs dfs -ls /*

*Found 2 items*

*-r-xr-xr-x - hdfs hadoop 0 2017-11-03 10:02 /ns1*

*-r-xr-xr-x - hdfs hadoop 0 2017-11-03 10:02 /ns2*

http://blog.csdn.net/strongerbit/article/details/7013221