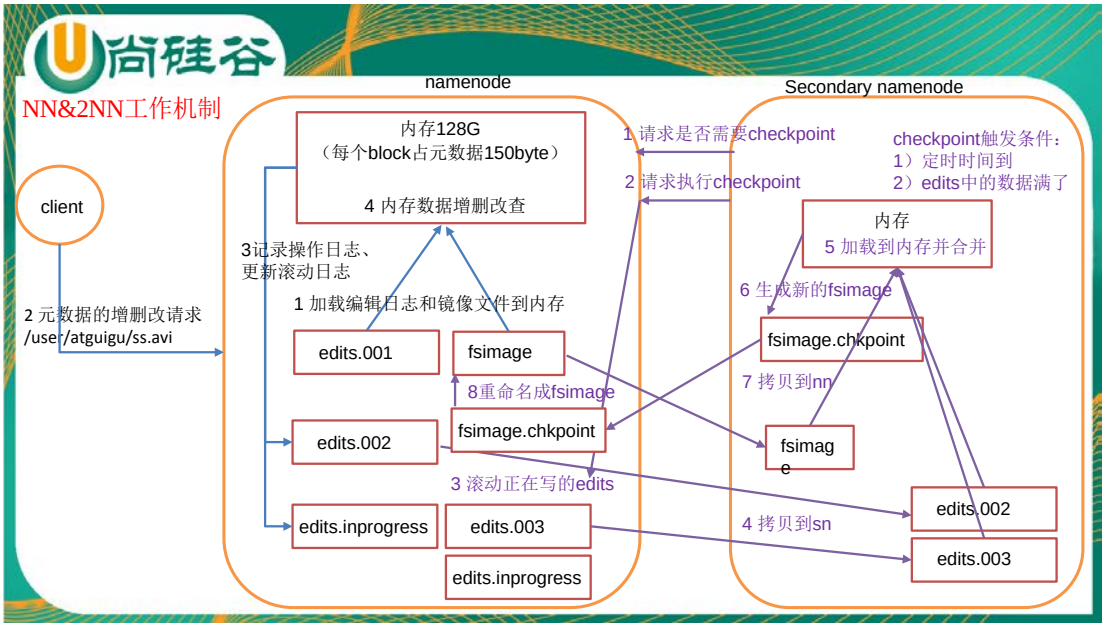
五 NameNode 和 SecondaryNameNode

5.1 NN 和 2NN 工作机制



1）第一阶段：NameNode 启动

（1）第一次启动 NameNode 格式化后，创建 fsimage 和 edits 文件。如果不是第一次启动，直接加载编辑日志和镜像文件到内存。

（2）客户端对元数据进行增删改的请求。

（3）NameNode 记录操作日志，更新滚动日志。

（4）NameNode 在内存中对数据进行增删改查。

2）第二阶段：Secondary NameNode 工作

（1）Secondary NameNode 询问 NameNode 是否需要 checkpoint。直接带回 NameNode 是否检查结果。

（2）Secondary NameNode 请求执行 checkpoint。

（3）NameNode 滚动正在写的 edits 日志。

（4）将滚动前的编辑日志和镜像文件拷贝到 Secondary NameNode。

（5）Secondary NameNode 加载编辑日志和镜像文件到内存，并合并。

（6）生成新的镜像文件 fsimage.chkpoint。

（7）拷贝 fsimage.chkpoint 到 NameNode。

（8）NameNode 将 fsimage.chkpoint 重新命名成 fsimage。

5.2 Fsimage 和 Edits 解析

1）概念

namenode 被格式化之后，将在/opt/module/hadoop-2.7.2/data/tmp/dfs/name/current 目录中产生如下文件

edits_0000000000000000000
fsimage_0000000000000000000.md5
seen_txid
VERSION

（1）Fsimage 文件：HDFS 文件系统元数据的一个永久性的检查点，其中包含 HDFS 文件系统的所有目录和文件 idnode 的序列化信息。

（2）Edits 文件：存放 HDFS 文件系统的所有更新操作的路径，文件系统客户端执行的所有写操作首先会被记录到 edits 文件中。

（3）seen_txid 文件保存的是一个数字，就是最后一个 edits_的数字

（4）每次 NameNode 启动的时候都会将 fsimage 文件读入内存，并从 00001 开始到 seen_txid 中记录的数字依次执行每个 edits 里面的更新操作，保证内存中的元数据信息是最新的、同步的，可以看成 NameNode 启动的时候就将 fsimage 和 edits 文件进行了合并。

2）oiv 查看 fsimage 文件

（1）查看 oiv 和 oev 命令

```
[atguigu@hadoop102 current]$ hdfs
oiv                  apply the offline fsimage viewer to an fsimage
oev                  apply the offline edits viewer to an edits file
```

（2）基本语法

```
hdfs oiv -p 文件类型 -i 镜像文件 -o 转换后文件输出路径
```

（3）案例实操

```
[atguigu@hadoop102 current]$ pwd
/opt/module/hadoop-2.7.2/data/tmp/dfs/name/current
[atguigu@hadoop102 current]$ hdfs oiv -p XML -i fsimage_0000000000000000025 -o
/opt/module/hadoop-2.7.2/fsimage.xml
[atguigu@hadoop102 current]$ cat /opt/module/hadoop-2.7.2/fsimage.xml
```

将显示的 xml 文件内容拷贝到 eclipse 中创建的 xml 文件中，并格式化。部分显示

结果如下。

```
<inode>
    <id>16386</id>
    <type>DIRECTORY</type>
    <name>user</name>
    <mtime>1512722284477</mtime>
    <permission>atguigu:supergroup:rwxr-xr-x</permission>
    <nsquota>-1</nsquota>
    <dsquota>-1</dsquota>
</inode>
<inode>
    <id>16387</id>
    <type>DIRECTORY</type>
    <name>atguigu</name>
    <mtime>1512790549080</mtime>
    <permission>atguigu:supergroup:rwxr-xr-x</permission>
    <nsquota>-1</nsquota>
    <dsquota>-1</dsquota>
</inode>
<inode>
    <id>16389</id>
    <type>FILE</type>
    <name>wc.input</name>
    <replication>3</replication>
    <mtime>1512722322219</mtime>
    <atime>1512722321610</atime>
    <perferredBlockSize>134217728</perferredBlockSize>
    <permission>atguigu:supergroup:rw-r--r--</permission>
    <blocks>
        <block>
            <id>1073741825</id>
            <genstamp>1001</genstamp>
            <numBytes>59</numBytes>
        </block>
    </blocks>
</inode>
```

3）oev 查看 edits 文件

（1）基本语法

hdfs oev -p 文件类型 -i 编辑日志 -o 转换后文件输出路径

（2）案例实操

[atguigu@hadoop102 current]$ hdfs oev -p XML -i

edits_0000000000000000012-0000000000000000013 -o /opt/module/hadoop-2.7.2/edits.xml

[atguigu@hadoop102 current]$ cat /opt/module/hadoop-2.7.2/edits.xml

将显示的 xml 文件内容拷贝到 eclipse 中创建的 xml 文件中，并格式化。显示结果如下。

```
<?xml version="1.0" encoding="UTF-8"?>
<EDITS>
    <EDITS_VERSION>-63</EDITS_VERSION>
    <RECORD>
        <OPCODE>OP_START_LOG_SEGMENT</OPCODE>
        <DATA>
            <TXID>129</TXID>
        </DATA>
    </RECORD>
    <RECORD>
        <OPCODE>OP_ADD</OPCODE>
        <DATA>
            <TXID>130</TXID>
            <LENGTH>0</LENGTH>
            <INODEID>16407</INODEID>
            <PATH>/hello7.txt</PATH>
            <REPLICATION>2</REPLICATION>
            <MTIME>1512943607866</MTIME>
            <ATIME>1512943607866</ATIME>
            <BLOCKSIZE>134217728</BLOCKSIZE>

    <CLIENT_NAME>DFSClient_NONMAPREDUCE_-1544295051_1</CLIENT_
NAME>
            <CLIENT_MACHINE>192.168.1.5</CLIENT_MACHINE>
            <OVERWRITE>true</OVERWRITE>
            <PERMISSION_STATUS>
                <USERNAME>atguigu</USERNAME>
                <GROUPNAME>supergroup</GROUPNAME>
                <MODE>420</MODE>
            </PERMISSION_STATUS>

    <RPC_CLIENTID>908eafd4-9aec-4288-96f1-e8011d181561</RPC_CLIENTID>
            <RPC_CALLID>0</RPC_CALLID>
        </DATA>
    </RECORD>
    <RECORD>
        <OPCODE>OP_ALLOCATE_BLOCK_ID</OPCODE>
        <DATA>
```

```
            <TXID>131</TXID>
            <BLOCK_ID>1073741839</BLOCK_ID>
        </DATA>
    </RECORD>
    <RECORD>
        <OPCODE>OP_SET_GENSTAMP_V2</OPCODE>
        <DATA>
            <TXID>132</TXID>
            <GENSTAMPV2>1016</GENSTAMPV2>
        </DATA>
    </RECORD>
    <RECORD>
        <OPCODE>OP_ADD_BLOCK</OPCODE>
        <DATA>
            <TXID>133</TXID>
            <PATH>/hello7.txt</PATH>
            <BLOCK>
                <BLOCK_ID>1073741839</BLOCK_ID>
                <NUM_BYTES>0</NUM_BYTES>
                <GENSTAMP>1016</GENSTAMP>
            </BLOCK>
            <RPC_CLIENTID></RPC_CLIENTID>
            <RPC_CALLID>-2</RPC_CALLID>
        </DATA>
    </RECORD>
    <RECORD>
        <OPCODE>OP_CLOSE</OPCODE>
        <DATA>
            <TXID>134</TXID>
            <LENGTH>0</LENGTH>
            <INODEID>0</INODEID>
            <PATH>/hello7.txt</PATH>
            <REPLICATION>2</REPLICATION>
            <MTIME>1512943608761</MTIME>
            <ATIME>1512943607866</ATIME>
            <BLOCKSIZE>134217728</BLOCKSIZE>
            <CLIENT_NAME></CLIENT_NAME>
            <CLIENT_MACHINE></CLIENT_MACHINE>
            <OVERWRITE>false</OVERWRITE>
            <BLOCK>
                <BLOCK_ID>1073741839</BLOCK_ID>
                <NUM_BYTES>25</NUM_BYTES>
                <GENSTAMP>1016</GENSTAMP>
            </BLOCK>
```

```
                <PERMISSION_STATUS>
                    <USERNAME>atguigu</USERNAME>
                    <GROUPNAME>supergroup</GROUPNAME>
                    <MODE>420</MODE>
                </PERMISSION_STATUS>
            </DATA>
        </RECORD>
</EDITS>
```

5.3 checkpoint 时间设置

（1）通常情况下，SecondaryNameNode 每隔一小时执行一次。

[hdfs-default.xml]

```
<property>
  <name>dfs.namenode.checkpoint.period</name>
  <value>3600</value>
</property>
```

（2）一分钟检查一次操作次数，当操作次数达到 1 百万时，SecondaryNameNode 执行一次。

```
<property>
  <name>dfs.namenode.checkpoint.txns</name>
  <value>1000000</value>
  <description>操作动作次数</description>
</property>

<property>
  <name>dfs.namenode.checkpoint.check.period</name>
  <value>60</value>
  <description> 1 分钟检查一次操作次数</description>
</property>
```

5.4 NameNode 故障处理

NameNode 故障后，可以采用如下两种方法恢复数据。

方法一：将 SecondaryNameNode 中数据拷贝到 NameNode 存储数据的目录；

1）kill -9 namenode 进程

2）删除 NameNode 存储的数据（/opt/module/hadoop-2.7.2/data/tmp/dfs/name）

[atguigu@hadoop102 hadoop-2.7.2]$ rm -rf /opt/module/hadoop-2.7.2/data/tmp/dfs/name/*

3）拷贝 SecondaryNameNode 中数据到原 NameNode 存储数据目录

[atguigu@hadoop102 dfs]$ scp -r atguigu@hadoop104:/opt/module/hadoop-2.7.2/data/tmp/dfs/namesecondary/* ./name/

4）重新启动 namenode

[atguigu@hadoop102 hadoop-2.7.2]$ sbin/hadoop-daemon.sh start namenode

方法二：使用 -importCheckpoint 选项启动 NameNode 守护进程，从而将 SecondaryNameNode 中数据拷贝到 NameNode 目录中。

1）修改 hdfs-site.xml 中的

```
<property>
  <name>dfs.namenode.checkpoint.period</name>
  <value>120</value>
</property>

<property>
  <name>dfs.namenode.name.dir</name>
  <value>/opt/module/hadoop-2.7.2/data/tmp/dfs/name</value>
</property>
```

2）kill -9 namenode 进程

3）删除 NameNode 存储的数据（/opt/module/hadoop-2.7.2/data/tmp/dfs/name）

[atguigu@hadoop102 hadoop-2.7.2]$ rm -rf /opt/module/hadoop-2.7.2/data/tmp/dfs/name/*

4）如果 SecondaryNameNode 不和 NameNode 在一个主机节点上，需要将 SecondaryNameNode 存储数据的目录拷贝到 NameNode 存储数据的平级目录，并删除 in_use.lock 文件。

```
[atguigu@hadoop102 dfs]$ scp -r atguigu@hadoop104:/opt/module/hadoop-2.7.2/data/tmp/dfs/namesecondary ./

[atguigu@hadoop102 namesecondary]$ rm -rf in_use.lock

[atguigu@hadoop102 dfs]$ pwd
/opt/module/hadoop-2.7.2/data/tmp/dfs

[atguigu@hadoop102 dfs]$ ls
data  name  namesecondary
```

5）导入检查点数据（等待一会 ctrl+c 结束掉）

[atguigu@hadoop102 hadoop-2.7.2]$ bin/hdfs namenode -importCheckpoint

6）启动 namenode

[atguigu@hadoop102 hadoop-2.7.2]$ sbin/hadoop-daemon.sh start namenode

5.5 集群安全模式

1）概述

NameNode 启动时，首先将映像文件（fsimage）载入内存，并执行编辑日志（edits）中的各项操作。一旦在内存中成功建立文件系统元数据的映像，则创建一个新的 fsimage 文件和一个空的编辑日志。此时，NameNode 开始监听 DataNode 请求。但是此刻，NameNode 运行在安全模式，即 NameNode 的文件系统对于客户端来说是只读的。

系统中的数据块的位置并不是由 NameNode 维护的，而是以块列表的形式存储在 DataNode 中。在系统的正常操作期间，NameNode 会在内存中保留所有块位置的映射信息。在安全模式下，各个 DataNode 会向 NameNode 发送最新的块列表信息，NameNode 了解到足够多的块位置信息之后，即可高效运行文件系统。

如果满足“最小副本条件”，NameNode 会在 30 秒钟之后就退出安全模式。所谓的最小副本条件指的是在整个文件系统中 99.9%的块满足最小副本级别（默认值：dfs.replication.min=1）。在启动一个刚刚格式化的 HDFS 集群时，因为系统中还没有任何块，所以 NameNode 不会进入安全模式。

2）基本语法

集群处于安全模式，不能执行重要操作（写操作）。集群启动完成后，自动退出安全模式。

（1）bin/hdfs dfsadmin -safemode get （功能描述：查看安全模式状态）

（2）bin/hdfs dfsadmin -safemode enter （功能描述：进入安全模式状态）

（3）bin/hdfs dfsadmin -safemode leave （功能描述：离开安全模式状态）

（4）bin/hdfs dfsadmin -safemode wait （功能描述：等待安全模式状态）

3）案例

模拟等待安全模式

（1）先进入安全模式

[atguigu@hadoop102 hadoop-2.7.2]$ bin/hdfs dfsadmin -safemode enter

（2）执行下面的脚本

编辑一个脚本

```
#!/bin/bash
```

```
bin/hdfs dfsadmin -safemode wait

bin/hdfs dfs -put ~/hello.txt /root/hello.txt
```

（3）再打开一个窗口，执行

[atguigu@hadoop102 hadoop-2.7.2]$ bin/hdfs dfsadmin -safemode leave

5.6 NameNode 多目录配置

1）NameNode 的本地目录可以配置成多个，且每个目录存放内容相同，增加了可靠性。

2）具体配置如下：

（1）在 hdfs-site.xml 文件中增加如下内容

```
<property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///${hadoop.tmp.dir}/dfs/name1,file:///${hadoop.tmp.dir}/dfs/name2</v
        alue>
</property>
```

（2）停止集群，删除 data 和 logs 中所有数据。

[atguigu@hadoop102 hadoop-2.7.2]$ rm -rf data/ logs/

[atguigu@hadoop103 hadoop-2.7.2]$ rm -rf data/ logs/

[atguigu@hadoop104 hadoop-2.7.2]$ rm -rf data/ logs/

（3）格式化集群并启动。

[atguigu@hadoop102 hadoop-2.7.2]$ bin/hdfs namenode –format

[atguigu@hadoop102 hadoop-2.7.2]$ sbin/start-dfs.sh

（4）查看结果

[atguigu@hadoop102 dfs]$ ll

总用量 12

drwx------. 3 atguigu atguigu 4096 12 月 11 08:03 data

drwxrwxr-x. 3 atguigu atguigu 4096 12 月 11 08:03 name1

drwxrwxr-x. 3 atguigu atguigu 4096 12 月 11 08:03 name2