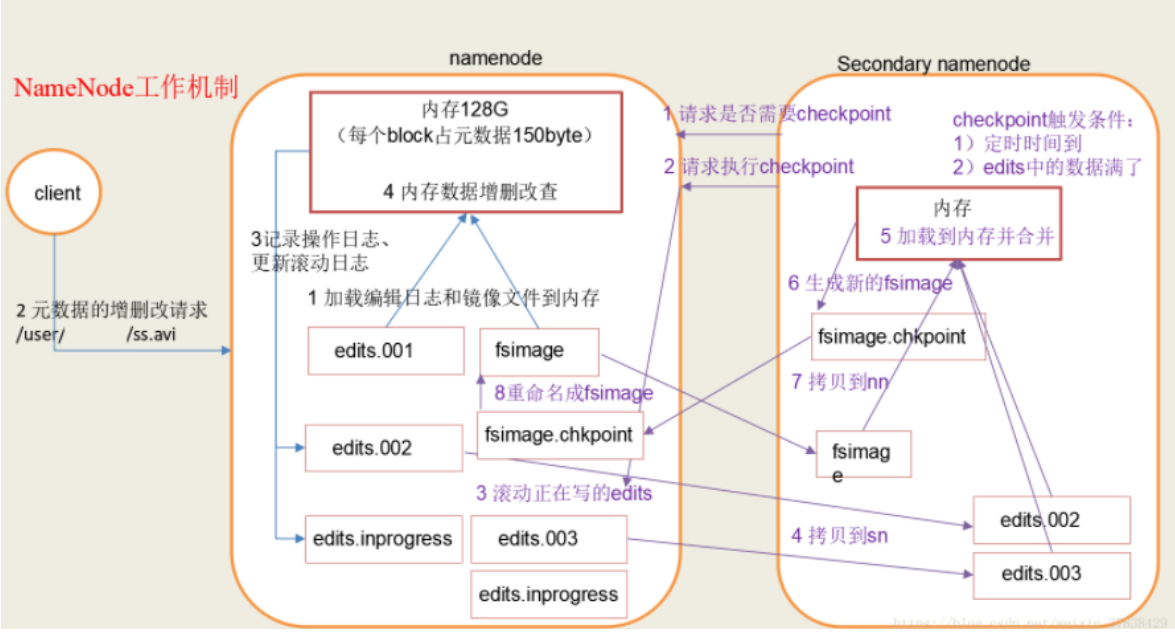
HDFS5 SecondaryNameNode

> SecondaryNameNode在HDFS中扮演着**辅助**的作用,负责辅助NameNode管理工作。由于editlog文件很大所有，集群再次启动时会花费较长时间。**为了加快集群的启动时间，所以使用secondarynameNode辅助NameNode合并Fsimage，editlog。**

工作机制

> 讨论到工作机制的时候，必须要重新查看一下namenode的工作了



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添删改的请求
（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查改
```

第二阶段：Secondary NameNode工作

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

NN和2NN工作机制详解：

namenode启动时，先滚动edits并生成一个空的edits.inprogress，然后加载edits和fsimage到内存中，此时namenode内存就持有最新的元数据信息。client开始对namenode发送元数据的增删改查的请求，这些请求的操作首先会被记录的edits.inprogress中（查询元数据的操作不会被记录在edits中，因为查询操作不会更改元数据信息），如果此时namenode挂掉，重启后会从edits中读取元数据的信息。然后，namenode会在内存中执行元数据的增删改查的操作。

由于edits中记录的操作会越来越多，edits文件会越来越大，导致namenode在启动加载edits时会很慢，所以需要对edits和fsimage进行合并（所谓合并，就是将edits和fsimage加载到内存中，照着edits中的操作一步步执行，最终形成新的fsimage）。secondarynamenode的作用就是帮助namenode进行edits和fsimage的合并工作。

secondarynamenode首先会询问namenode是否需要checkpoint（触发checkpoint需要满足两个条件中的任意一个，定时时间到和edits中数据写满了）。直接带回namenode是否检查结果。secondarynamenode执行checkpoint操作，首先会让namenode滚动edits并生成一个空的edits.inprogress，滚动edits的目的是给edits打个标记，以后所有新的操作都写入edits.inprogress，其他未合并的edits和fsimage会拷贝到secondarynamenode的本地，然后将拷贝的edits和fsimage加载到内存中进行合并，生成fsimage.chkpoint，然后将fsimage.chkpoint拷贝给namenode，重命名为fsimage后替换掉原来的fsimage。namenode在启动时就只需要加载之前未合并的edits和fsimage即可，因为合并过的edits中的元数据信息已经被记录在fsimage中。

checkpoint时间设置

- 通常情况下，`SecondaryNameNode`每隔一个小时执行一次

  `hdfs-default.xml`

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

- 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 >
  ```

总结

> 前提：不使用SNN,日志文件会很大，日志大会导致集群恢复到上次关机前的状态花费很长时间，这个时候集群处于安全模式，集群不可用。长时间处于安全模式。

> 目标：加快集群二次启动的速度。（减少集群二次启动的时间）
> SecondaryNamenode**周期性复制**NameNode的FSIMAGE 和edits到本机（SecondaryNamenode本机），将两个文件进行合并，最终生成全新的Fsimage,将最新的Fsimage发送回Namenode 。
>
> 意义：**辅助NameNode合并Fsimage Edits.减小了日志的大小，加快了集群的二次启动速度。**
> 附:SecondaryNamenode自己独立部署在有一个节点上。此节点的配置要与NameNode相同(至少,有条件也可以大于NameNode的配置)。