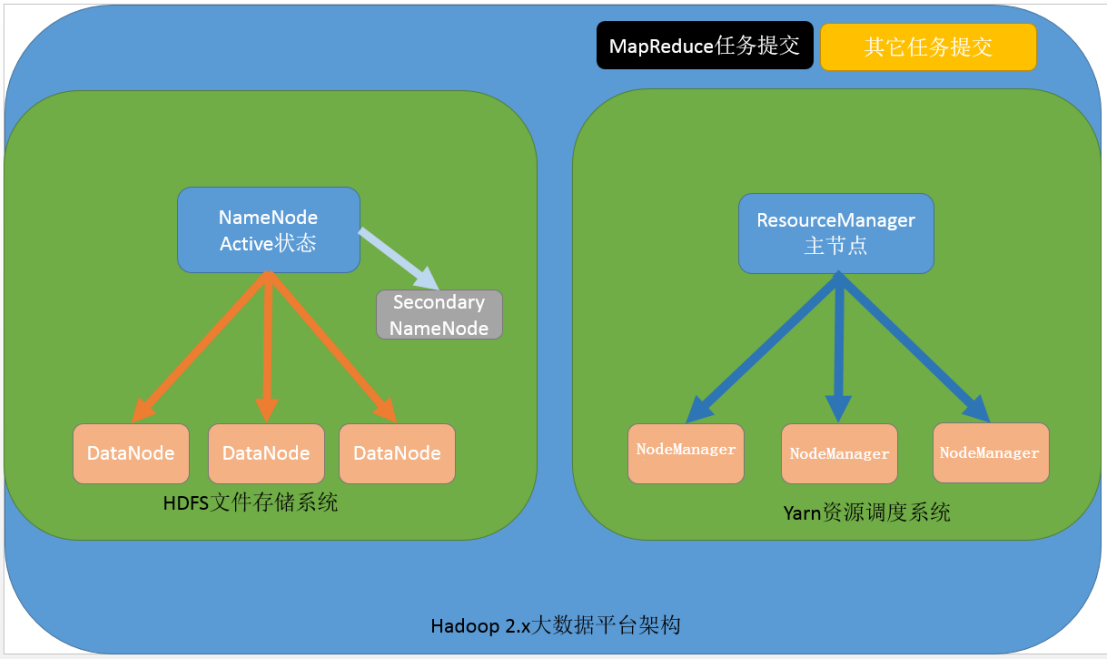
HDFS 的元数据信息 FSimage 以及 edits 和 secondaryNN 的作用

在 hadoop 当中，使用如下架构的时候



也就是 namenode 就一个的时候，所有的元数据信息都保存在了 FsImage 与 Eidts 文件当中，这两个文件就记录了所有的数据的元数据信息，元数据信息的保存目录配置在了 hdfs-site.xml 当中

```
<property>
                <name>dfs.namenode.name.dir</name>
                <value>file:///export/servers/hadoop-2.6.0-
cdh5.14.0/hadoopDatas/namenodeDatas</value>
        </property>
<property>
                <name>dfs.namenode.edits.dir</name>
                <value>file:///export/servers/hadoop-2.6.0-
cdh5.14.0/hadoopDatas/dfs/nn/edits</value>
  </property>
```

1、FSImage 与 edits 详解

客户端对 hdfs 进行写文件时会首先被记录在 edits 文件中。

edits 修改时元数据也会更新。

每次 hdfs 更新时 edits 先更新后客户端才会看到最新信息。

fsimage:是 namenode 中关于元数据的镜像，一般称为检查点。

一般开始时对 namenode 的操作都放在 edits 中，为什么不放在 fsimage 中呢？

因为 fsimage 是 namenode 的完整的镜像，内容很大，如果每次都加载到内存的话生成树状拓扑结构，这是非常耗内存和 CPU。

fsimage 内容包含了 namenode 管理下的所有 datanode 中文件及文件 block 及 block 所在的 datanode 的元数据信息。随着 edits 内容增大，就需要在一定时间点和 fsimage 合并。

合并过程见 SecondaryNameNode 如何辅助管理 FSImage 与 edits

2、FSimage 文件当中的文件信息查看

官方查看文档

http://archive.cloudera.com/cdh5/cdh/5/hadoop-2.6.0-cdh5.14.0/hadoop-project-dist/hadoop-hdfs/HdfsEditsViewer.html

使用命令 hdfs oiv

```
cd /export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/namenodeDatas/current

hdfs oiv -i fsimage_0000000000000000150 -p XML -o qlg.xml
```

```
[root@node01 servers]# hdfs oiv -i fsimage_0000000000000000150 -p XML -o qlg.xml
[root@node01 servers]# ll
total 24
-rw-r--r--  1 root root 2241 Jul 18 07:09 fsimage_0000000000000000150
drwxr-xr-x 11 root root 4096 Jul 17 19:36 hadoop-2.6.0-cdh5.14.0
drwxr-xr-x  8 uucp  143 4096 Jul 12  2017 jdk1.8.0_141
-rw-r--r--  1 root root 8657 Jul 18 07:12 qlg.xml
```

（md5 校验文件，不是真正文件）

3、edits 当中的文件信息查看

官方查看文档

http://archive.cloudera.com/cdh5/cdh/5/hadoop-2.6.0-cdh5.14.0/hadoop-project-dist/hadoop-hdfs/HdfsEditsViewer.html

查看命令 hdfs oev

```
cd /export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/dfs/nn/edits

hdfs oev -i edits_inprogress_0000000000000000001 -o slg.xml -p XML
```

```
[root@node01 servers]# hdfs oev -i edits_inprogress_0000000000000000001 -o slg.x
ml -p XML
[root@node01 servers]# ll
total 1096
-rw-r--r--  1 root root 1048576 Jul 18 09:30 edits_inprogress_000000000000000000
1
drwxr-xr-x 11 root root    4096 Jul 18 08:53 hadoop-2.6.0-cdh5.14.0
drwxr-xr-x  8 uucp  143    4096 Jul 12  2017 jdk1.8.0_141
-rw-r--r--  1 root root   63853 Jul 18 09:32 slg.xml
[root@node01 servers]#
```

4、secondarynameNode 如何辅助管理 FSImage 与 Edits 文件

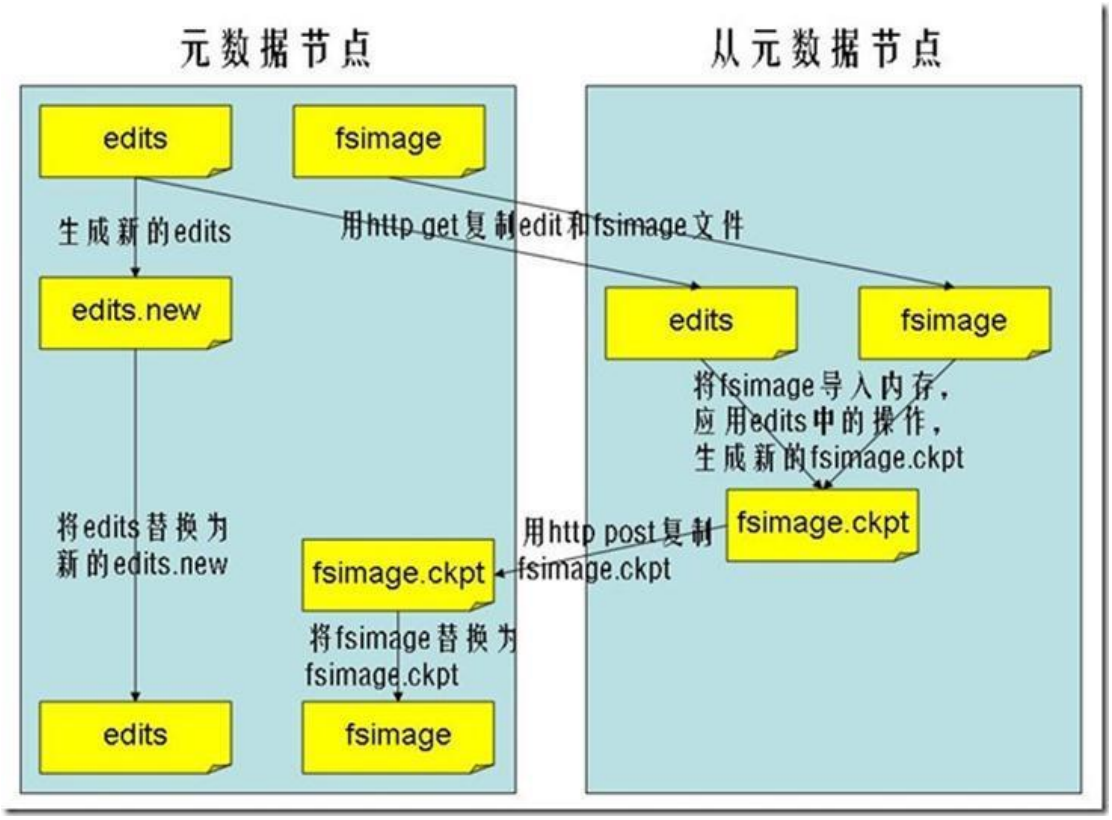
①：secnonaryNN 通知 NameNode 切换 editlog

②：secondaryNN 从 NameNode 中获得 FSImage 和 editlog(通过 http 方式)

③：secondaryNN 将 FSImage 载入内存，然后开始合并 editlog，合并之后成为新的 fsimage

④：secondaryNN 将新的 fsimage 发回给 NameNode

⑤：NameNode 用新的 fsimage 替换旧的 fsimage



- Secondary NameNode 定期合并 fsimage 和 edits 日志，将 edits 日志文件大小控制在一个限度下。
- **配置Secondary NameNode**
  - conf/masters文件指定的为Secondary NameNode节点
  - 修改在masters文件中配置了的机器上的conf/hdfs-site.xml文件，加上如下选项：

```
<property> <name>dfs.http.address</name> <value>namenode.hadoop-host.com:50070</value> </property>
```

  - core-site.xml：这里有2个参数可配置，但一般来说我们不做修改。
fs.checkpoint.period表示多长时间记录一次hdfs的镜像。默认是1小时。
fs.checkpoint.size表示一次记录多大的size，默认64M。

```
<property> <name>fs.checkpoint.period</name> <value>3600</value>
<description>The number of seconds between two periodic checkpoints.
</description> </property>
<property> <name>fs.checkpoint.size</name> <value>67108864</value>
<description>The size of the current edit log (in bytes) that triggers a periodic checkpoint even if the fs.checkpoint.period hasn't expired. </description>
</property>
```

完成合并的是 secondarynamenode，会请求 namenode 停止使用 edits,暂时将新写操作放入一个新的文件中（edits.new)。

secondarynamenode 从 namenode 中通过 http get 获得 edits，因为要和 fsimage 合并，所以也是通过 http get 的方式把 fsimage 加载到内存，然后逐一执行具体对文件系统的操作，与 fsimage 合并，生成新的 fsimage，然后把 fsimage 发送给 namenode，通过 http post 的方式。namenode 从 secondarynamenode 获得了 fsimage 后会把原有的 fsimage 替换为新的 fsimage,把 edits.new 变成 edits。同时会更新 fstime。

hadoop 进入安全模式时需要管理员使用 dfsadmin 的 save namespace 来创建新的检查点。

secondarynamenode 在合并 edits 和 fsimage 时需要消耗的内存和 namenode 差不多，所以一般把 namenode 和 secondarynamenode 放在不同的机器上。

fs.checkpoint.period: 默认是一个小时（3600s)

fs.checkpoint.size: edits 达到一定大小时也会触发合并（默认 64MB)