配置SecondaryNameNode

## 一、SecondaryNameNode概念：

光从字面上来理解，很容易让一些初学者先入为主：SecondaryNameNode(snn)就是NameNode(nn)的热备进程。其实不是。ssn是HDFS架构中的一个组成部分，但是经常由于名字而被人误解它真正的用途，其实它真正的用途，是用来保存namenode中对HDFS metadata的信息的备份，并减少namenode重启的时间。对于hadoop进程中，要配置好并正确的使用snn，还是需要做一些工作的。hadoop的默认配置中让snn进程默认运行在了namenode的那台机器上，但是这样的话，如果这台机器出错，宕机，对恢复HDFS文件系统是很大的灾难，更好的方式是：将snn的进程配置在另外一台机器上运行。

在hadoop中，namenode负责对HDFS的metadata的持久化存储，并且处理来自客户端的对HDFS的各种操作的交互反馈。为了保证交互速度，HDFS文件系统的metadata是被load到namenode机器的内存中的，并且会将内存中的这些数据保存到磁盘进行持久化存储。为了保证这个持久化过程不会成为HDFS操作的瓶颈，hadoop采取的方式是：没有对任何一次的当前文件系统的snapshot进行持久化，对HDFS最近一段时间的操作list会被保存到namenode中的一个叫Editlog的文件中去。当重启namenode时，除了load fslmage意外，还会对这个Editlog文件中记录的HDFS操作进行replay，以恢复HDFS重启之前的最终状态。

而SecondaryNameNode，会周期性的将Editlog中记录的对HDFS的操作合并到一个checkpoint中，然后清空Editlog。所以namenode的重启就会Load最新的一个checkpoint，并replay Editlog中记录的hdfs操作，由于Editlog中记录的是从上一次checkpoint以后到现在的操作列表，所以就会比较小。如果没有snn的这个周期性的合并过程，那么当每次重启namenode的时候，就会花费很长的时间。而这样周期性的合并就能减少重启的时间。同时也能保证HDFS系统的完整性。这就是SecondaryNameNode所做的事情。所以snn并不能分担namenode上对HDFS交互性操作的压力。尽管如此，当namenode机器宕机或者namenode进程出问题时，namenode的daemon进程可以通过人工的方式从snn上拷贝一份metadata来恢复HDFS文件系统。

至于为什么要将snn进程运行在一台非NameNode的机器上，这主要出于两点考虑：

1、可扩展性：创建一个新的HDFS的snapshot需要将namenode中load到内存的metadata信息全部拷贝一遍，这样的操作需要的内存和namenode占用的内存一样，由于分配给namenode进程的内存其实是对HDFS文件系统的限制，如果分布式文件系统非常的大，那么namenode那台机器的内存就可能会被namenode进程全部占据。

2、容错性：当snn创建一个checkpoint的时候，它会将checkpoint拷贝成metadata的几个拷贝。将这个操作运行到另外一台机器，还可以提供分布式文件系统的容错性。

SECONDARYNAMENODE工作原理

### 日志与镜像的定期合并总共分五步：

1. SecondaryNameNode通知NameNode准备提交edits文件，此时主节点产生edits.new
2. SecondaryNameNode通过http get方式获取NameNode的fsimage与edits文件（在SecondaryNameNode的current同级目录下可见到 temp.check-point或者previous-checkpoint目录，这些目录中存储着从namenode拷贝来的镜像文件）
3. SecondaryNameNode开始合并获取的上述两个文件，产生一个新的fsimage文件fsimage.ckpt
4. SecondaryNameNode用http post方式发送fsimage.ckpt至NameNode
5. NameNode将fsimage.ckpt与edits.new文件分别重命名为fsimage与edits，然后更新fstime，整个checkpoint过程到此结束。 在新版本的hadoop中（hadoop0.21.0）,SecondaryNameNode两个作用被两个节点替换， checkpoint node与backup node.         SecondaryNameNode备份由三个参数控制fs.checkpoint.period控制周期，fs.checkpoint.size控制日志文件超过多少大小时合并， dfs.http.address表示http地址，这个参数在SecondaryNameNode为单独节点时需要设置。

## 二、配置将SeconddaryNameNode运行在另外一台机器上

DFS的一次运行实例是通过在namenode机器上的$HADOOP\_HOME/bin/start-dfs.sh(或者start-all.sh)脚本来启动的。这个脚本会在运行该脚本的机器上启动namenode进程，而slaves机器上都会启动DataNode进程，slave机器的列表保存在conf/slaves文件中，一行一台机器。并且会在另外一台机器上启动一个snn进程，这台机器由conf/masters文件指定。所以，这里需要严格注意，conf/masters文件中指定的机器，并不是说jobtracker或者namenode进程要运行在这台机器上，因为这些进程是运行在launch bin/start-dfs.sh或者bin/start-mapred.sh(start-all.sh)的机器上的。所以，master这个文件名是非常的令人混淆的，应该叫做secondaries会比较合适。然后通过以下步骤：

1、将所有想要运行secondarynamenode进程的机器写到masters文件中，一行一台。（这一步没操作也可能运行）

2、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：

[复制代码](javascript:void(0);)

1 <property>

2 <name>dfs.secondary.http.address</name>

3 <value>**192.168**.**1.152**:**50090**</value>

# 如果secondarynamenode为多个话可以设置为0.0.0.0:50090

4 </property>

5 <property>

6 <name>dfs.http.address</name>

7 <value>**192.168**.**1.151**:**50070**</value>

8 </property>

[复制代码](javascript:void(0);)

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。

[复制代码](javascript:void(0);)

1 <property>

2 <name>fs.checkpoint.period</name>

3 <value>**3600**</value>

4 <description>

The number of seconds between two periodic checkpoints.

</description>

5 </property>

6 <property>

7 <name>fs.checkpoint.size</name>

8 <value>**67108864**</value>

9 <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>

10 </property>

11 <property>

12 <name>fs.checkpoint.dir</name>

13 <value>/app/user/hdfs/namesecondary</value>

14 <description>Determines where on the local filesystem the DFS secondary namenode should store the temporary images to merge.If this is a comma-delimited list of directories then the image is replicated in all of the directories for redundancy.</description>

15 </property>

[复制代码](javascript:void(0);)

配置完成之后，我们需要检查一次是否成功。我们可以通过查看运行secondarynamenode的机器上文件目录来确定是否成功配置。

## **三、恢复**

1、配置完成后，如何恢复。首先我们kill掉namenode进程，然后将hadoop.tmp.dir目录下的数据删除掉。制造master挂掉情况。

2、在配置参数dfs.name.dir指定的位置建立一个空文件夹；把检查点目录的位置赋值给配置参数fs.checkpoint.dir；启动namenode，并加上-importCheckpoint。

3、启动namenode的时候采用hadoop namenode -importCheckpoint.

## 四、Secondarynamenode的启动和停止

启动：

bin/hadoop-daemons.sh start secondarynamenode

停止：

   bin/hadoop-daemons.sh stop secondarynamenode

## 五、总结

1、配置SecondNamenode

配置hdfs-site.xml

创建masters文件，添加SecondNamenode的IP

SecondNamenode端口是50090

2、SecondNamenode作用

提高速度：edits只记录增删等操作，经过配置的时间间隔，把edits和fsimage文件发到SecondNamenode上合并，减少磁盘的IO

3、Namenode和SecondNamenode如何管理元数据

元数据存放在Namenode磁盘的fsimage文件中，不包含block位置

启动后加载到内存中，Datanode向Namenode汇报block位置信息，存在内存中

HDFS的增、删操作记录到Namenode磁盘的edits文件中

经过配置的时间间隔，把edits和fsimage文件发到SecondNamenode上合并

合并完把fsimage传回namenode节点