**搭建分布式的HDFS文件系统**

**基础知识：**

HDFS用于支持海量数据在普通机器上的存储，提供高可靠性，容错性，安全性，向上层提供文件抽象，名称空间，用户可以像使用普通文件系统一样使用HDFS。

HDFS的架构中最重要的是namenode，namenode负责管理整个文件系统的名称空间，涉及到文件名的操作比如删除，创建都需要访问namenode。在HDFS中，文件在写入时先被划分成block，block是等大的，除了最后一个。namenode记录了从文件名到block的映射，因此读取，写入数据亦与namenode相关。**一般要求namenode进程单独运行在一台非常可靠的机器上**，这台机器的内存应该足够大，因为文件系统的名称空间完全存放在内存里，从而可存放的文件数，目录结构的复杂性都依赖于namenode所在主机的内存大小。

Secondarynamenode进程提供了一个关于namenode的冷备份，在namenode挂掉的情况下，secondarynamenode可在一段时间内替代namenode，一般而言，secondarynamenode应该具备和namenode相同的内存大小。最新版本的hadoop貌似支持多个namenode，这样处于待命状态的namenode就相当于原来的热备份，可靠性更高。

Datanode是文件内容存放的实际位置，文件划分出来的block被分布式地存储在各台主机的datanode上，默认情况下，每个block有三个备份，这三个备份按照一定的策略分配到各个datanode上（通过一种称为机架感知的机制）。Datanode定期向namenode发送heartbeat和blocktable，告诉namenode自己的状态，namenode一旦发现datanode挂掉，启动相应的复制数据块等机制，保持block的备份数目。

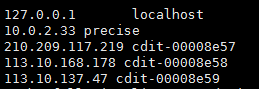
以上是HDFS最基本的内容，更多重要的内容比如namenode的**eidtlog**和**fsimage**，client对文件的读写过程，文件的流式写入，请参看其他资料。

目前仍然是四台主机，SRP01~SRP04，其中SRP01同时作为namenode，secondarynamenode，datanode，其余几台均作为datanode，这样的分配其实不可取，不过由于是实验性质的，我们也就先这样安排，以后再修改。

**步骤一：**用wget下载工具从hadoop官网上下载hadoop-2.6.0，并解压到/opt目录下，注意，所有主机的hadoop位置应该一致，这跟spark的要求一样；

**步骤二：**配置从master到slaves的ssh免密码登录，首先在master上生成公钥，然后发送待各个slave节点上，将该公钥加入授权。具体的步骤请参看spark的配置文档，由于我们已经在spark配置过了，这里不用重新配置；

**步骤三：**配置各个主机上的/etc/hosts文件，在文件中添加主机名到IP地址的映射，比如在第一台主机SRP01上，内容如下：

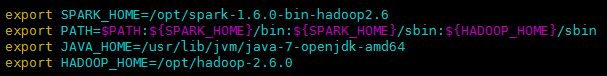


**必须留意SRP02~SRP03使用的是它们各自的外网IP，而SRP01（主机名precise）使用的是该主机的内网IP**！这里非常重要，尤其对于申请回来的云主机，这些云主机通常是由物理主机虚拟出来的，云主机的虚拟网卡通过桥接到物理机的物理网卡上连接网络，**云主机可以访问其他机器的公网IP，却无法访问到自身的公网IP**。比如，命令ping 210.209.117.148（SRP01的公网IP）是无法ping通的，但是ping 10.0.2.33或者ping precise是可行的。

在最开始配置HDFS的时候，由于没有意识到这一点，浪费了非常多的时间，网上也没有对这些问题的解答。在每次通过hadoop根目录下的sbin/start-dfs.sh脚本启动HDFS后，用jps查看java进程，只能看到datanode，而namenode和secondarynamenode都没有启动成功。但是日志文件也没有提示任何错误，这一点让人非常抓狂，基本上无法排错。

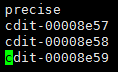
目前对HDFS的工作过程尚不是很清楚，在我配置的时候，每台机器上我都对应配置了该机的/etc/hosts文件，但也有人说只要配置好master上的就行了，master上的/etc/hosts不必与slaves上同步。

**步骤四：**在/etc/profile下添加HADOOP\_HOME（或HADOOP\_PREFIX）环境变量，然后在PATH变量后追加${HADOOP\_HOME}/sbin，以便今后启停HDFS。（最后，如果是要使用hadoop的mapreduce的话，尽量配置好JAVA\_HOME变量，并且在hadoop根目录下的etc/hadoop/hadoop-env.sh添加JAVA\_HOME变量，由于我们用spark，这部分可不必配置）如下图：



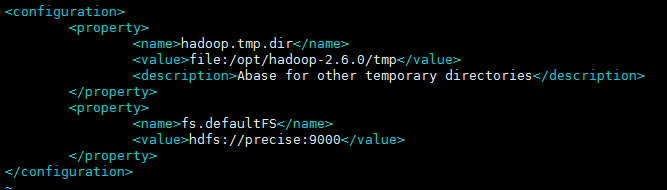
**步骤五：**对HDFS的配置文件进行更改：

**第一个文件：**hadoop根目录下的etc/hadoop/slaves，添加所有作为datanode的主机名，如下图：



这里把precise也作为datanode了，其实precise最好只作为namenode，只运行单个进程，这些今后再做更改。关于这个文件是不是需要在各台机器上同步，目前还不清楚，但我们在所有的机器上都做了相同的配置。

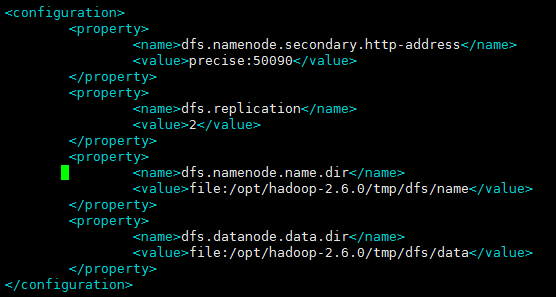
**第二个文件：**hadoop根目录下的etc/hadoop/core-site.xml（所有的机器），修改如下：



这里面有一个非常重要的内容，hadoop.tmp.dir定义了hadoop临时文件的存放位置，这个位置非常重要，默认情况下它就是liunx的/tmp目录，若我们使用默认配置，今后势必造成极大的困扰，因为/tmp会定期清理，造成数据丢失，很多问题都是由于/tmp下的内容被清理掉引起的，需要特别注意。

fs.defaultFS定义了文件系统的所在位置。

**第三个文件：**hadoop根目录下的etc/hadoop/hdfs-site.xml（所有的机器）



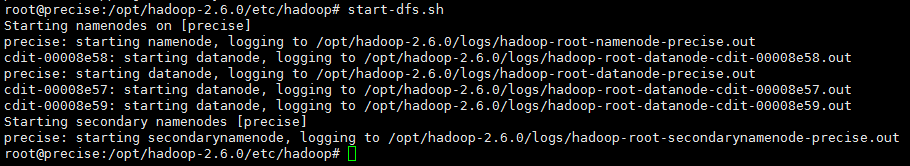
Dfs.namenode.secondary.http-address定义了secondarynamenode的位置，dfs.replication配置每个块的备份的数目，默认为3，dfs.namenode.name.dir和dfs.namenode.data.dir配置namenode和datanode的缓存数据的存放位置。

**步骤六：**在master上格式化namenode并启动HDFS，输入命令：

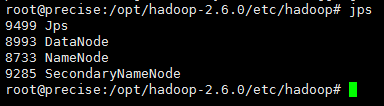
bin/hdfs namenode -format

格式化只需进行一次，格式化会清除namenode的所有数据。

然后运行脚本start-dfs.sh，输出如下：



**步骤七：**输入jps，可以看到master上的java进程已经运行起来了：



我们也可以通过WebUI查看HDFS的情况，在浏览器地址栏输入http://210.209.117.148:50070即可，WebUI还提供了浏览目录等方便的功能，至此，HDFS已经完成基本搭建，使用和更近一步的配置请参考其他文档。

