**机架感知到底是咋回事？**

**一、机架感知是什么？**

告诉Hadoop集群中哪台机器属于哪个机架

**二、那么怎么告诉呢？**

Hadoop对机架的感知并非是自适应的，亦即，hadoop集群分辨某台slave机器是属于哪个rack并非是智能感知的，而是需要hadoop的管理者人为的告知hadoop哪台机器属于哪个rack，这样在hadoop的namenode启动初始化时，会将这些机器与rack的对应信息保存在内存中，用来作为对接下来所有的HDFS的写块操作分配datanode列表时（比如3个block对应三台datanode）的选择datanode策略，尽量将三个副本分布到不同的rack。

**三、什么情况下会涉及到机架感知？**

Hadoop集群规模很大的情况下。

**四、机架感知需要考虑的情况（权衡可靠性、可用性、带宽消耗）**

（1）不同节点之间的通信能够尽量发生在同一个机架之内，而不是跨机架

（2）为了提高容错能力，名称节点会尽可能把数据块的副本放到多个机架上。

**五、通过什么方式能够告知hadoop namenode哪些slaves机器属于哪个rack？以下是配置步骤。**  
  
    默认情况下，hadoop的机架感知是没有被启用的。所以，在通常情况下，hadoop集群的HDFS在选机器的时候，是随机选择的，也就是说，很有可能在写数据时，hadoop将第一块数据block1写到了rack1上，然后随机的选择下将block2写入到了rack2下，此时两个rack之间产生了数据传输的流量，再接下来，在随机的情况下，又将block3重新又写回了rack1，此时，两个rack之间又产生了一次数据流量。在job处理的数据量非常的大，或者往hadoop推送的数据量非常大的时候，这种情况会造成rack之间的网络流量成倍的上升，成为性能的瓶颈，进而影响作业的性能以至于整个集群的服务。  
    要将hadoop机架感知的功能启用，配置非常简单，在namenode所在机器的hdfs-site.xml配置文件中配置一个选项：

1.<property>

2.  <name> net.topology.script.file.name </name>

3.  <value>/path/to/RackAware.py</value>

4.</property>

这个配置选项的value指定为一个可执行程序，通常为一个脚本，该脚本接受一个参数，输出一个值。接受的参数通常为某台datanode机器的ip地址，而输出的值通常为该ip地址对应的datanode所在的rack，例如”/rack1”。Namenode启动时，会判断该配置选项是否为空，如果非空，则表示已经用机架感知的配置，此时namenode会根据配置寻找该脚本，并在接收到每一个datanode的heartbeat时，将该datanode的ip地址作为参数传给该脚本运行，并将得到的输出作为该datanode所属的机架，保存到内存的一个map中。  
  
    至于脚本的编写，就需要将真实的网络拓朴和机架信息了解清楚后，通过该脚本能够将机器的ip地址正确的映射到相应的机架上去。一个简单的实现如下：

1.#!/usr/bin/python

2.#-\*-coding:UTF-8 -\*-

3.import sys

4.

5.rack = {"hadoopnode-176.tj":"rack1",

6.        "hadoopnode-178.tj":"rack1",

7.        "hadoopnode-179.tj":"rack1",

8.        "hadoopnode-180.tj":"rack1",

9.        "hadoopnode-186.tj":"rack2",

10.        "hadoopnode-187.tj":"rack2",

11.        "hadoopnode-188.tj":"rack2",

12.        "hadoopnode-190.tj":"rack2",

13.        "192.168.1.15":"rack1",

14.        "192.168.1.17":"rack1",

15.        "192.168.1.18":"rack1",

16.        "192.168.1.19":"rack1",

17.        "192.168.1.25":"rack2",

18.        "192.168.1.26":"rack2",

19.        "192.168.1.27":"rack2",

20.        "192.168.1.29":"rack2",

21.        }

22.

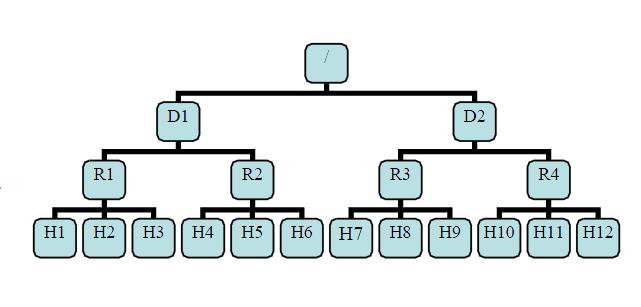
23.

24.if \_\_name\_\_=="\_\_main\_\_":

25.    print "/" + rack.get(sys.argv[1],"rack0")

由于没有确切的文档说明 到底是主机名还是ip地址会被传入到脚本，所以在脚本中最好兼容主机名和ip地址，如果机房架构比较复杂的话，脚本可以返回如：/dc1/rack1 类似的字符串。  
  
执行命令：chmod +x RackAware.py  
  
重启namenode,如果配置成功，namenode启动日志中会输出：

**INFO org.apache.hadoop.net.NetworkTopology: Adding a new node: /rack1/192.168.1.15:50010**

**六、网络拓扑机器之间的距离**  
这里基于一个网络拓扑案例，介绍在复杂的网络拓扑中hadoop集群每台机器之间的距离  
  
   
  
  
有了机架感知，NameNode就可以画出上图所示的datanode网络拓扑图。D1,R1都是交换机，最底层是datanode。则H1的rackid=/D1/R1/H1，H1的parent是R1，R1的是D1。这些rackid信息可以通过topology.script.file.name配置。有了这些rackid信息就可以计算出任意两台datanode之间的距离。

1.distance(/D1/R1/H1,/D1/R1/H1)=0  相同的datanode

2.distance(/D1/R1/H1,/D1/R1/H2)=2  同一rack下的不同datanode

3.distance(/D1/R1/H1,/D1/R2/H4)=4  同一IDC(互联网数据中心（机房）)下的不同datanode

4.distance(/D1/R1/H1,/D2/R3/H7)=6  不同IDC下的datanode