HDFS番外1 机架感知

背景

分布式集权通常会包含非常多的机器，由于机架槽位和交换机网口的限制，通常大型的分布式集群都会跨好几个机架（`rack`），然而，这么多个机架共同组成一个分布式集群，一般机架间的网络通信速度会小于机架内部的网络通信，并且机架之间的机器的网络通信通常受到上层交换机之间网络宽带的限制。

副本存放策略

一般我们都知道`HDFS`对数据文件的分布式存放是按照分块`block`存放的，然后每个`block`都会有3个副本（默认值），一般这三个副本存放策略为：

- 第一个block：存放在和`Client`所在的`node`里面，如果`Client`不在`node`里面，就随机选取
- 第二个副本放置在第一个节点不同的机架中的node
- 第三个副本放在第一个副本节点同一个机架不同节点上

这里`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，用来作为选取的策略。

配置

> 一般Hadoop的机架感知没有启动，所以可能会出现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。这样的浪费

打开机架感知，在`namenode`所在的机器`hadoop-site.xml`中

```
<property>
  <name>topology.script.file.name</name>
  <value>/path/to/RackAware.py</value>
</property
```

接收参数为一个脚本：
输入为：某台datanode机器的ip地址
输出为：该ip地址对应的datanode所在的rack，例如："/rack"

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：

```
#!/usr/bin/python
#-*-coding:UTF-8 -*-
import sys

rack = {"hadoopnode-176.tj":"rack1",
        "hadoopnode-178.tj":"rack1",
```

```
        "hadoopnode-179.tj":"rack1",
        "hadoopnode-180.tj":"rack1",
        "hadoopnode-186.tj":"rack2",
        "hadoopnode-187.tj":"rack2",
        "hadoopnode-188.tj":"rack2",
        "hadoopnode-190.tj":"rack2",
        "192.168.1.15":"rack1",
        "192.168.1.17":"rack1",
        "192.168.1.18":"rack1",
        "192.168.1.19":"rack1",
        "192.168.1.25":"rack2",
        "192.168.1.26":"rack2",
        "192.168.1.27":"rack2",
        "192.168.1.29":"rack2",
        }


if __name__=="__main__":
    print "/" + rack.get(sys.argv[1],"rack0")
```

由于没有找到确切的文档说明 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。

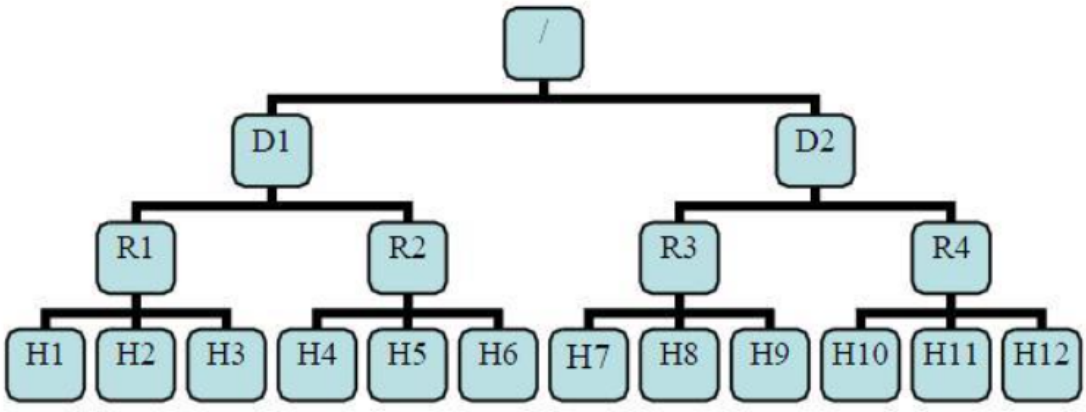
执行命令：`chmod +x RackAware.py`

重启namenode,如果配置成功，namenode启动日志中会输出：

```
2011-12-21 14:28:44,495 INFO org.apache.hadoop.net.NetworkTopology: Adding a new node: /rack1/192.168.1.15:50010
```

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。

```
distance(/D1/R1/H1,/D1/R1/H1)=0  相同的datanode
distance(/D1/R1/H1,/D1/R1/H2)=2  同一rack下的不同datanode
distance(/D1/R1/H1,/D1/R1/H4)=4  同一IDC下的不同datanode
distance(/D1/R1/H1,/D2/R3/H7)=6  不同IDC下的datanode
```