## Hadoop概述

### Hadoop创始人

Doug Cutting，斯坦福大学毕业的。

Cutting是Lucence,nutch,hadoop的创始人。

### Hadoop起源

2004年，开发一款搜索引擎，nutch。nutch基于Lucence实现的搜索引擎，能够从互联网上抓取网页数据。抓取来的海量数据的存储问题。但是，这些海量数据都是非结构化数据，不能存在关系型数据库里。如果连数据的存储和管理都解决不了化，就谈不上后续为用户提供搜索服务，包括通过算法去优化检索速度。

正好那时候，Google发表了两篇论文《Google File System》 《Google MapReduce》

《Google File System》简称GFS，是Google公司用于解决海量数据存储的文件系统。

《Google MapReduce》简称MapReduce,是Google的计算框架，基于GFS。

2006年cutting根据《Google File System》设计了Nutch的HDFS,hadoop distributed file system。

Hadoop最开始是nutch的子项目，目的是解决nutch的海量数据存储问题。在nutch 0.8版本之后，Hadoop独立处理，成为一个独立的项目。后来，又根据《Google MapReduce》设计了基于HDFS的uMapRedce计算框架。

2006年之后，cutting带着Hadoop去了雅虎，当时有100多人的团队共同帮cutting完善hadoop。后来yahoo把Hadoop贡献了Apache。所以，现在Hadoop是Apache的顶级项目

### 大数据的特点

1.TB,PB级别的数据

2.非结构化数据

3.快速增长，每天都会有大量数据产生。

### 大数据带来的变化

1.大数据没有产生之前，数据分析，主要要依靠比较精良的算法。算法越严密，结果越精确。但是有了大数据之后，当数据量达到一定规模的时候，数据本身就可以说话了。所以就不需要太依靠算法去分析结果。

2.当数据量达到一定程度后，不需要考虑因果关系，就能得到结果。

所以，大数据是一种新的生产力。

### HDFS设计

单台机器的硬件扩展

纵向扩展的问题，是有硬件瓶颈的，包括成本也会指数型增长。

#### namenode工作职责

1.要知道管理有哪些机器节点，即有哪些datanode。比如ip信息等。

2.要管理文件信息，文件名、文件多大、文件被切成几块、以及每一块的存贮位置信息（存在哪个datanode节点上了），即管理元数据信息。

3.要有一个机制要知道集群里datanode节点的状态变化。可以rpc心跳机制来做到。

4.namenode存在单点故障问题，可以再引入一台管理者节点。

5.datanode挂掉后，可能数据就丢失，文件就不完整了，要有一个备份机制，一般来说，一个文件块，有3个备份，本机一份，同一机架的其他datanode有一份，另外一机器的机器上有一份。

## Hadoop三种搭建模式

1. 单机模式：只支持MapReduce，不支持HDFS。这种模式一般用于调试MapReduce任务用的。
2. 伪分布模式：这种模式即支持MR,也支持HDFS，用多个线程模拟多台真实机器，即模拟真实的分布式环境。
3. 完全分布式：用多个机器或多台虚拟机去搭建集群

## Hadoop伪分布式安装搭建

### 下载地址

<http://hadoop.apache.org/releases.html>

### 关闭防火墙

命令：

service iptables stop

执行该命令可以关闭防火墙，但是如果虚拟机重启的话，防火墙会重新开启。

命令：

chkconfig iptables off

执行该命令可以永久关闭防火墙。

### 配置主机名

命令：

hostname

执行该命令可以查看当前主机的主机名。

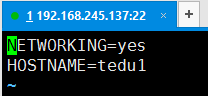
命令：

hostname tedu

执行该命令可以将主机名修改为tedu，但是这种方式是临时修改的方式，当前生效，但当虚拟机重新启动后，主机名会恢复到原来的主机名。

命令：

vim /etc/sysconfig/network

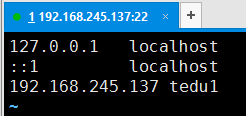


执行以上命令，通过vim打开设置主机名的文件（如上图），将HOSTNAME的属性设置为tedu1。需要注意的是该主机名里不能包含下划线、特殊字符（#$）；但是这种方式重启虚拟机才生效，所以我们将以上两种方式结合使用。

### 配置hosts文件

命令：

vim /etc/hosts



执行该命令，在hosts文件中配置ip 和 主机名的映射，这样配置后我们就可以使用主机名做配置了。好处：ip地址发生变化后，我们不需要去一个一个修改hadoop相关的配置文件，只需在hosts文件中对应的修改即可。

### 配置免密码登陆

命令：

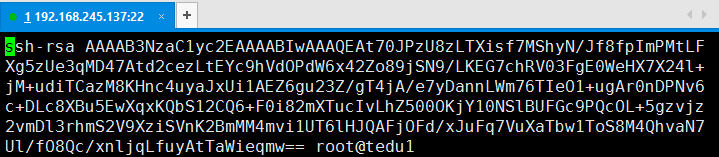
ssh-keygen

执行该命令，一直回车，之后会在虚拟机本地生成一对公钥和私钥，默认生成在.ssh目录下。

命令：

ssh-copy-id root@tedu1

因为我们hadoop伪分布式搭建在一台虚拟机上，所以，我们只需要将公钥拷贝到本地即可。执行该命令后，在.ssh目录下的authorized\_keys文件中会增加出一条已知的公钥信息,如下图：



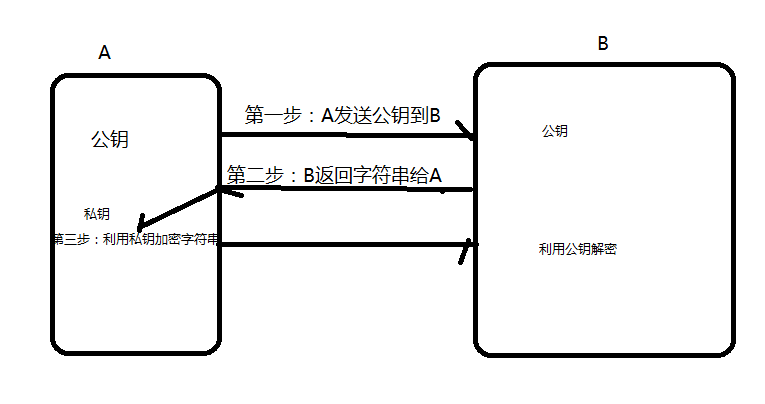
讲解：

Hadoop这台机器自己登陆自己需要输入密码

ssh-keygen 一直回车

RSA 非对称加密算法 2048位，解不开的。

扩展：不存在解不开的密码



### 安装和配置jdk

1. 上传jdk安装包，并解压：

解压命令：

tar –zxvf jdk-8u65-linux-x64.tar.gz -C /home/work/app

1. 配置java环境变量

命令：

vim /etc/profile

编辑/etc/profile文件，并加入以下配置：

export JAVA\_HOME=/home/work/app/jdk1.8.0\_65/

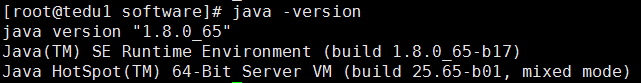
export PATH=$PATH:$JAVA\_HOME/bin:

1. 查看java环境变量配置是否成功

命令：

java –version

执行该命令，可查看安装jdk的版本信息，如下图：

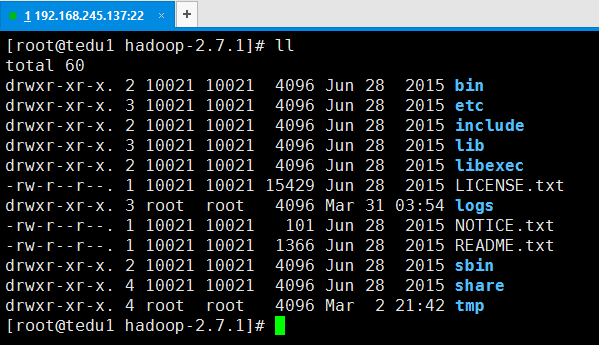


### 上传和解压hadoop安装包

解压命令：

tar -zxvf hadoop-2.7.1\_64bit.tar.gz -C /home/work/app

Hadoop安装完成后，进入hadoop，hadoop目录如下：



其中etc目录为hadoop配置文件所在的目录。

### 配置hadoop-env.sh

命令：

vim hadoop-env.sh

修改如下参数配置：

export JAVA\_HOME=/home/app/jdk1.7.0\_55/

export HADOOP\_CONF\_DIR=/home/work/app/hadoop-2.7.1/etc/hadoop

### 修改core-site.xml

命令：

vim core-site.xml

编辑配置文件core-site.xml,在该文件中添加一下内容:

指定hdfs老大,namenode的地址

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://tedu1:9000</value>

</property>

<!--hadoop临时目录-->

<property>

<name>hadoop.tmp.dir</name>

<value>/home/work/app/hadoop-2.7.1/tmp</value>

</property>

</configuration>

### 修改hdfs-site.xml

命令：

vim hdfs-site.xml

编辑配置文件hdfs-site.xml,在该文件中添加一下内容:

Hdfs存储数据的副本数量 包括自己本身

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<!--配置权限 设置任何用户都可以访问hdfs-->

<property>

<name>dfs.permissions</name>

<value>false</value>

</property>

</configuration>

### 修改mapred-site.xml

Hadoop配置文件中提供的是mapred-site.xml的模板，将mapred-site.xml.template复制一份并将名称修改为mapred-site.xml，编辑mapred-site.xml,修改其中内容为：

<!--指定mapreduce运行在yarn上-->

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>

### 修改yarn-site.xml

<!--指定yarn的老大 resourceManage地址 -->

<configuration>

<property>

<name>yarn.resourcemanager.hostname</name>

<value>tedu1</value>

</property>

<!--指定nodemanager获取数据的方式-- >

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

</configuration>

### 配置slaves文件

指定datanode节点的位置：单机则配置为当前主机

tedu1

### 配置hadoop的环境变量

命令：

vim /etc/profile

在该文件中添加一下内容：

export HADOOP\_HOME=/home/work/a6pp/hadoop-2.7.1/

export PATH=$PATH:$JAVA\_HOME/bin:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin:

命令：

source /etc/profile

执行该命令使修改生效。

### 格式化namenode

命令：

hdfs namenode –format

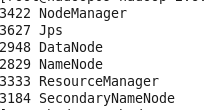
初始化的作用。

### 启动hadoop

进入hadoop安装目录中的sbin目录中，执行./start-all.sh命令启动hadoop。

### 验证hadoop安装是否成功

在linux命令行中输入jps命令：



### 如上图，出现如上进程，则说明hadoop启动成功。

## Hadoop之HDFS

### HDFS概述

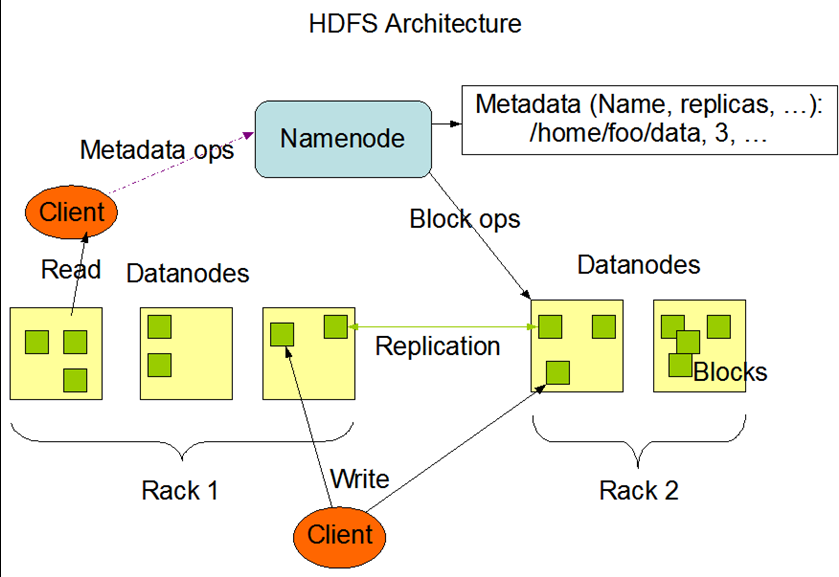


图1 hdfs体系架构图

HDFS是一个主从形式体系结构，对于用户来说，使用HDFS存储文件就像使用传统系统（例如Linux）存储文件一样，甚至提供的一些交互式命令也非常类似于Linux命令，可以通过目录（文件）路径对文件执行操作。区别则在于HDFS是一个分布式的存储结构，主要由五大部分组成，分别为NameNode、DataNode、Client、Rack、Block。

1、 Hadoop集群中有且仅有一个NameNode，但是可以有多个DataNode。

2、 Hadoop存储数据文件到HDFS上时，会将文件切成一定大小的块，以数据块（Block）为单位存储数据。

3、 NameNode节点存储文件的元数据，而文件的内容则以块的形式保存到DataNode节点上。

4、 Block块会有指定数量的副本，这样保证了数据文件的可靠性。

5、 客户端（Client）通过和NameNode、DataNode交互访问HDFS分布式文件存储系统。

具体来看下每个部分的作用：

NameNode

Hadoop集群中只允许有一个正在活动（active）的NameNode节点，该节点管理文件系统的命名空间，维护着文件系统树以及整棵树内所有的文件和目录。

NameNode主要保存了以下元数据内容：

1、 文件的大小、文件的切块数量、创建日期等；

2、 文件和块之间的映射关系

3、 块和DataNode之间的映射关系

需要注意的是，这些数据会在内存和硬盘上都保存一份。这样做的原因是：在硬盘上保存一份可以保证信息的安全性；在内存中保存一份可以使NameNode快速做出响应。

DataNode

Hadoop集群中一般会配置多个DataNode，一个文件具体的内容存储在一个或者多个DataNode节点上，DataNode也会保存一些元数据信息，例如：block id和本地文件的映射关系。DataNode真正执行数据的读写。

rack

机架，分布式的集群通常包含非常多的机器，由于受到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，由多个机架上的机器共同组成一个分布式集群。机架内的机器之间的网络速度通常都会高于跨机架机器之间的网络速度，并且机架之间机器的网络通信通常受到上层交换机间网络带宽的限制。

我们之前说过Block块默认有3个副本，那这3个副本怎么存放呢？

第一个block副本放在和client所在的同一个DataNode里（如果client不在集群范围内，则这第一个node是随机选取的）。

第二个副本放置在与第一个节点不同的机架中的node中（随机选择）。

第三个副本放置在与第一个副本所在节点同一机架的另一个节点上

如果还有更多的副本就随机放在集群的DataNode里。

client

客户端，凡是通过API或指令操作的一端都可以看做是客户端

blockSize

数据块。Hadoop1.0：64MB。Hadoop2.0 ：128MB。

块大小的问题。从大数据处理角度来看，块越大越好。所以从技术的发展，以后的块会越来越大，因为块大，会减少磁盘寻址次数，从而减少寻址时间。

### HDFS指令

1. 创建目录：

hadoop fs -mkdir /test

hadoop fs –mkdir /test1

1. 查看目录：

hadoop fs -ls /

1. 上传文件：

hadoop fs –put README.txt /test

1. 查看文件内容：

hadoop fs –cat /test/README.txt

1. 重命名：

hadoop fs –mv /test /test2

1. 复制：

hadoop fs –cp /test2 /test1

1. 创建空文件：

hadoop fs –touchz /test2/a.txt

1. 移动文件：

hadoop fs –mv /test2/a.txt /test1

1. 下载文件：

hadoop fs –get /test1/README.txt /home

1. 删除文件：

hadoop fs –rmr /test1/a.txt