# Hadoop平台介绍

1、Hadoop概述

Hadoop实现了一个[分布式文件系统](https://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F%E6%96%87%E4%BB%B6%E7%B3%BB%E7%BB%9F)（Hadoop Distributed File System），简称HDFS。HDFS有高[容错性](https://baike.baidu.com/item/%E5%AE%B9%E9%94%99%E6%80%A7)的特点，并且设计用来部署在低廉的（low-cost）硬件上；而且它提供高吞吐量（high throughput）来访问[应用程序](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F)的数据，适合那些有着超大数据集（large data set）的应用程序。HDFS放宽了（relax）POSIX的要求，可以以流的形式访问（streaming access）文件系统中的数据。

Hadoop的框架最核心的设计就是：HDFS和MapReduce。HDFS为海量的数据提供了存储，则MapReduce为海量的数据提供了计算。

2、Hadoop的作用

在本系统中，对于大量采集到的数据，采用hadoop框架的核心设计：HDFS，

并为Hadoop生态圈中其他的软件（如 **Hbase**等）作底层支持。

[HDFS](https://baike.baidu.com/item/HDFS)就像一个传统的分级文件系统。可以创建、删除、移动或[重命名](https://baike.baidu.com/item/%E9%87%8D%E5%91%BD%E5%90%8D)文件，等等。但是 HDFS 的架构是基于一组特定的节点构建的（参见图 1），这是由它自身的特点决定的。这些节点包括 NameNode（仅一个），它在 HDFS 内部提供元数据服务；DataNode，它为 HDFS 提供存储块。

存储在 HDFS 中的文件被分成块，然后将这些块复制到多个计算机中（DataNode）。这与传统的 RAID 架构大不相同。块的大小（通常为 64MB）和复制的块数量在创建文件时由客户机决定。NameNode 可以控制所有文件操作。HDFS 内部的所有通信都基于标准的 [TCP/IP](https://baike.baidu.com/item/TCP%2FIP) 协议。

3、系统采用的Hadoop安装配置

版本：hadoop-2.8.2

相关配置文件：

core-site.xml

<configuration>  
<!-- 指定hadoop运行时产生文件的存储路径 -->  
 <property>  
 <name>hadoop.tmp.dir</name>  
 <value>/usr/hdfs/tmp</value>  
 </property>  
 <property>  
 <name>fs.default.name</name>  
 <value>hdfs://master:9000</value>  
 </property>  
</configuration>

hdfs-site.xml

<configuration>  
 <!-- 设置namenode存放的路径 -->  
 <property>  
 <name>dfs.namenode.name.dir</name>  
 <value>/usr/hdfs/name</value>  
 </property>  
 <!-- 设置hdfs副本数量 -->  
 <property>  
 <name>dfs.replication</name>  
 <value>2</value>  
 </property>  
 <!-- 设置datanode存放的路径 -->  
 <property>  
 <name>dfs.datanode.data.dir</name>  
 <value>/usr/hdfs/data</value>  
 </property>  
 <!-- 设置hdfs web端打开-->  
 <property>  
 <name>dfs.webhdfs.enabled</name>  
 <value>true</value>  
</property>

</configuration>

mapred-site.xml

<configuration>  
 <property>  
 <name>mapreduce.framework.name</name>  
 <value>yarn</value>  
 </property>  
</configuration>

yarn-site.xml

configuration>  
<!-- Site specific YARN configuration properties -->  
<property>  
 <name>yarn.resourcemanager.address</name>  
 <value>master:18040</value>  
 </property>  
 <property>  
 <name>yarn.resourcemanager.scheduler.address</name>  
 <value>master:18030</value>  
 </property>  
 <property>  
 <name>yarn.resourcemanager.webapp.address</name>  
 <value>master:8088</value>  
 </property>  
 <property>  
 <name>yarn.resourcemanager.resource-tracker.address</name>  
 <value>master:18025</value>  
 </property>  
 <property>  
 <name>yarn.resourcemanager.admin.address</name>  
 <value>master:18141</value>  
 </property>  
 <property>  
 <name>yarn.nodemanager.aux-services</name>  
 <value>mapreduce.shuffle</value>  
 </property>  
 <property>  
 <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>  
 <value>org.apache.hadoop.mapred.ShuffleHandler</value>  
 </property>  
</configuration>

slaves

slave1  
slave2

各服务器节点IP

59.110.159.69 master

39.106.26.245 slave1

39.106.68.2 slave2