**说明：NN:namenode ，SNN：secondarynamenode，DN：datanode**

**Spark的集群搭建**

1. Local（单机模式），只能在一台机器上运行，主要用来做测试
2. Standalone模式：集群模式（hdfs）
3. Spark on yarn模式：生产上用的比较多的集群模式（hdfs，yarn）

# Hadoop的集群搭建及原理

1. Local模式
2. 伪分布式模式
3. 集群模式：Not on yarn , on yarn
4. HA模式

**Hadoop历史版本**

## 1.0

* + - 1. 有hdfs（分布式文件系统）
      2. MapReduce:计算框架、资源管理框架

## 2.0

1. Hdfs（分布式文件系统）
2. 1.0版本的MapReduce做了一个拆分，把计算的归计算，资源管理的归资源管理。计算框架由MapReduce来实现，资源管理由Yarn来实现
3. 因为yarn的存在，所以扩展了HADOOP的应用，比如说spark就是基于YARN来搭建计算平台

## 3.0

1. 本质上和2.0的版本相差不大，只是做了很多优化

# HADOOP的部署

单机版部署，主要是用来做测试（没有HDFS的）

配置要求：

修改hadoop-env.sh中的JAVA\_HOME就行

伪分布式模式：

伪分布，在一台机器上模拟出主节点和工作节点（主节点和工作节点都是由进程来模拟），已经存在了HDFS

前期准备：

节点名称（hosts，hostname）

节点无密码访问

关闭防火墙

配置jdk，hadoop的环境

几个主要文件：

hadoop-env.sh

core-site.xml

hdfs-site.xml

slaves

集群的启动

格式化NN：hadoop namenode -format

启动：start-dfs.sh

集群模式：

非YARN部署模式(真正的集群，NN,DN,SNN)

前期准备：

* + - * 节点取名（hostname）
      * 名字访问各个节点（vi /etc/hosts）
      * 无密码访问（master1到包含master1的所有其他节点）
      * jdk ,hadoop的环境变量的配置（profile）

Hadoop的配置文件

**hadoop-env.sh**

**core-site.xml**

<configuration>

<property>

<name>fs.defaultFS</name> #NN的URL

<value>hdfs://master:9000</value>

</property>

<property>

<name>hadoop.tmp.dir</name> #指定hdfs的存放物理路径

<value>/home/hadoop/tmp</value>

</property>

</configuration>

**hdfs-site.xml**

<configuration>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

#定义主namenode的web访问URL

<property>

<name>dfs.namenode.http-address</name>

<value>master1:50070</value>

</property>

#定义第二个名字节点的访问URL同时也是定义了第二个名字节点所在的主机

<property>

<name>dfs.namenode.secondary.http-address</name>

<value>master2:50090</value>

</property>

<property>

#名字节点数据存放路径，注意，只有在NN节点下，该配置起作用

备注:初始化名字节点也是初始化该路径下的内容

<name>dfs.namenode.name.dir</name>

<value>/home/Hadoop/tmp/name</value>

</property>

<property>

<name>dfs.namenode.data.dir</name>

<value>/home/Hadoop/tmp/data</value>

</property>

</configuration>

**Slaves**

Slave1

Slave2

Slave3

**mapred-site.xml**

<configuration>

#指定资源管理yarn

<property>

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

<value>yarn</value>

</property>

</configuration>

**yarn-site.xml**

<configuration>

#指定yarn的主机

<property>

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

<value>master1</value>

</property>

#指定nodemanager上运行的附属服务，需要配置成mapreduce\_shuffle，才可以运行MapReduce

<property>

<name>yarn.nodename.sux.services</name>

<value>mapreduce\_shuffle</value>

</property>

</configuration>

启动：

第一步格式化NN：hadoop namenode -format

启动：start-dfs.sh(stop-dfs.sh)

ON YARN的部署模式

YARN的讲解：

Yarn和hdfs一样，在运行模式下，都是能够独立的集群

Yarn这个集群是一个主从模式的集群（ResourceManager Node ,NodeManager）

RM:负责集群总体资源的管理和分配

NM:负责某一个（NM所在的）节点的资源的管理和分配，NM一般情况下，会部署到DN所在的节点上

# Spark

运维模式有以下几种

* local模式（单节点模式）
* standalone模式（真正的分布模式）
* spark on yarn 模式

## 单节点模式（local）

* **Local machine**
* 图示

  描述已自动生成

注释：

1. 所有的角色都运行在一台主机上
2. Worker是由线程来进行模拟的
3. 也由集群的管理功能（其实管理的就是线程）
4. 更多的是用来进行测试（spark-shell）

配置的前期准备工作：

JVM中运行，所以要配置jdk

问题：scala的sdk要不要配置？ 不要，因为架包在编译的时候，就已经生成了字节码，这个字节码只需要JVM的支持就可以了

操作系统的配置（名字访问master，防火墙，无密码访问）

1. 解压spark并更名
2. 配置Spark的环境变量（bin， sbin）
3. 运行(spark-shell)

## Standalone(真正的集群，资源管理靠spark本身完成的)

图示

描述已自动生成

大部分是用来做测试（很少拿来生产使用）

## Spark on yarn

1. 有关yarn的粗略概念
2. Spark如何on yarn
3. 如何配置
   1. 配置好节点的操作系统环境
   2. 配置好jdk
   3. 配置hadoop（hdfs, 所以hadoop集群要运行起来； 需要的yarn在hadoop里面，要配置好yarn的环境）
   4. Spark的几个配置文件配置好和yarn相关的内容

Spark-env.sh

Spark-defaults.conf

Log4……