

# 实验三： Hadoop的部署与常用命令

---

2018/9/25

王晓桐

# Outline

---

- Hadoop简介
- Hadoop项目结构
- Hadoop部署

# Hadoop简介



- 开源的分布式计算平台
- 基于Java语言开发
- 核心是分布式文件系统HDFS ( Hadoop Distributed File System) 和 大规模并行计算编程模型MapReduce

# Hadoop简介——特性

---

Hadoop是一个能够对大量数据进行分布式处理的软件框架，并且是以一种可靠、高效、可伸缩的方式进行处理的，它具有以下几个方面的特性：

- 高可靠性
- 高效性
- 高可扩展性
- 高容错性
- 成本低
- 运行在Linux平台上
- 支持多种编程语言

# Hadoop简介——应用现状

---

- Hadoop凭借其突出的优势，已经在各个领域得到了广泛的应用，而互联网领域是其应用的主阵地
- 2007年，雅虎在Sunnyvale总部建立了M45——一个包含了4000个处理器和1.5PB容量的Hadoop集群系统
- Facebook作为全球知名的社交网站，Hadoop是非常理想的选择，Facebook主要将Hadoop平台用于日志处理、推荐系统和数据仓库等方面
- 国内采用Hadoop的公司主要有百度、淘宝、网易、华为、中国移动等，其中，淘宝的Hadoop集群比较大

# Hadoop简介

---

Hadoop框架中最核心的设计是为海量数据提供存储的HDFS和对数据进行计算的MapReduce

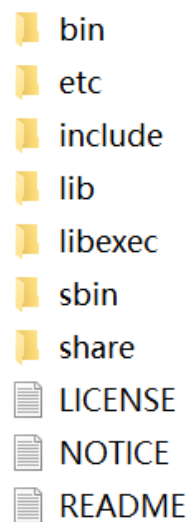
- 从磁盘或者网络读取数据，即I/O密集工作
- 计算数据，即CPU密集工作

Hadoop集群的整体性能取决于CPU、内存、网络以及存储之间的性能平衡，因此运营团队在选择机器配置时要针对不同的工作节点选择合适的硬件类型。一个基本的Hadoop集群中的节点主要有：

- NameNode，负责协调集群中的数据存储
- DataNode，负责存储被拆分的数据块
- JobTracker，负责协调数据计算任务
- TaskTracker，负责执行由JobTracker指派的任务
- SecondaryNameNode，帮助NameNode收集文件系统运行的状态信息

# Hadoop 目录结构

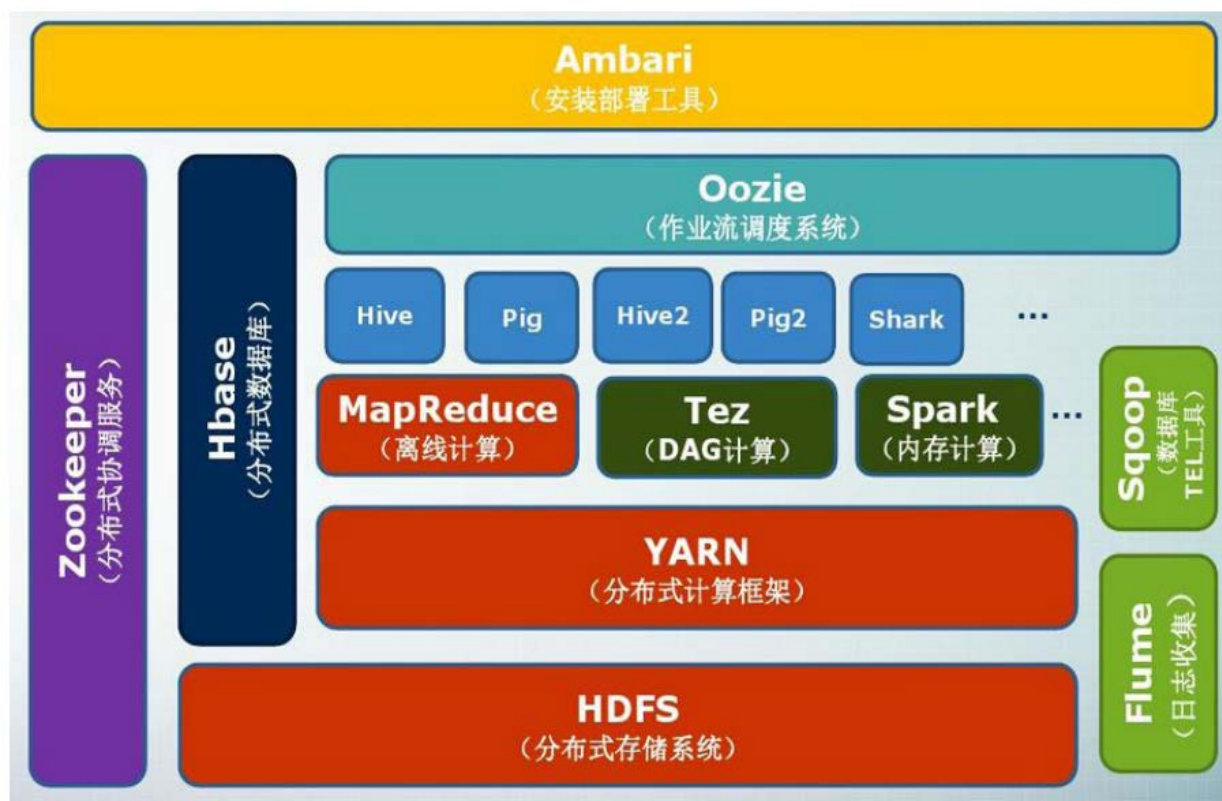
---



1. **bin:** Hadoop最基本的管理脚本和使用脚本的目录，这些脚本是sbin目录下管理脚本的基础实现，用户可以直接使用这些脚本管理和使用Hadoop
2. **etc:** Hadoop配置文件所在的目录，包括core-site.xml、hdfs-site.xml、mapred-site.xml等
3. **include:** 对外提供的编程库头文件（具体动态库和静态库在lib目录中），这些头文件均是用C++定义的，通常用于C++程序访问HDFS或者编写MapReduce程序
4. **lib:** 该目录包含了Hadoop对外提供的编程动态库和静态库，与include目录中的头文件结合使用
5. **libexec:** 各个服务对用的shell配置文件所在的目录，可用于配置日志输出、启动参数（比如JVM参数）等基本信息
6. **sbin:** Hadoop管理脚本所在的目录，主要包含HDFS和YARN中各类服务的启动/关闭脚本
7. **share:** Hadoop各个模块编译后的jar包所在的目录

# Hadoop简介——项目结构

Hadoop项目结构不断丰富发展，已经形成一个丰富的Hadoop生态系统





# Hadoop部署

---

共有三种模式

- 单机

- 默认模式为本地模式，无需进行其他配置即可运行
- 单 Java 进程，方便进行调试，存储采用本地文件系统

- 伪分布式

- 在单节点上以伪分布式的方式运行，Hadoop 进程以分离的 Java 进程来运行
- 节点既作为 NameNode 也作为 DataNode，同时，读取的是 HDFS 中的文件

- 分布式

- 使用多个节点构成集群环境来运行Hadoop

# Hadoop部署

---

Hadoop的基本安装配置步骤

- 创建hadoop用户
- SSH 登录权限设置
- 下载安装
- 单机部署
- 伪分布式部署
- 分布式部署

# 创建hadoop用户

---

如果安装 Ubuntu 的时候不是用的 “hadoop” 用户，那么需要增加一个名为 hadoop 的用户

首先按 **ctrl+alt+t** 打开终端窗口，输入如下命令创建新用户：

```
sudo useradd -m hadoop -s /bin/bash
```

上面这条命令创建了可以登陆的 hadoop 用户，并使用 /bin/bash 作为 shell

接着使用如下命令设置密码，可简单设置为 hadoop，按提示输入两次密码：

```
sudo passwd hadoop
```

可为 hadoop 用户增加管理员权限，方便部署，避免一些对新手来说比较棘手的权限问题：

```
sudo adduser hadoop sudo
```

# SSH登录权限设置

---

## 配置SSH的原因

- Hadoop名称节点（NameNode）需要启动集群中所有机器的Hadoop守护进程，这个过程需要通过SSH登录来实现
- Hadoop并没有提供SSH输入密码登录的形式，因此，为了能够顺利登录每台机器，需要将所有机器配置为名称节点可以无密码登录它们

## SSH本机免密钥登录

- 查看当前/home/hadoop目录下有无.ssh文件夹，若无，则创建mkdir ~/.ssh，修改权限 chmod 700 ~/.ssh
- 执行指令 ssh-keygen -t rsa 生成公钥和私钥 (一路回车)
- 执行 cat ./id\_rsa.pub >> ./authorized\_keys 将密钥加入授权
- 执行指令 ssh localhost 进行测试

# Hadoop 下载

Hadoop官网下载 `hadoop-xxx.tar.gz`，这是编译好的，另一种src则是Hadoop的源码，需要编译才能使用 <https://hadoop.apache.org/releases.html>

官网默认是64位

32位编译版本链接 [https://pan.baidu.com/s/1gLRyK6a\\_szall\\_QUZhrU2g](https://pan.baidu.com/s/1gLRyK6a_szall_QUZhrU2g) 提取码: a85u

## Download

Hadoop is released as source code tarballs with corresponding binary tarballs for convenience. The downloads are distributed via mir using GPG or SHA-256.

Version	Release date	源代码	编译好的
		Source download	Binary download
3.1.1	2018 Aug 8	<a href="#">source (checksum signature)</a>	<a href="#">binary (checksum signature)</a>
3.0.3	2018 May 31	<a href="#">source (checksum signature)</a>	<a href="#">binary (checksum signature)</a>
2.9.1	2018 May 3	<a href="#">source (checksum signature)</a>	<a href="#">binary (checksum signature)</a>
2.7.5	2017 Dec 14	<a href="#">source (checksum signature)</a>	<a href="#">binary (checksum signature)</a>
2.8.3	2017 Dec 12	<a href="#">source (checksum signature)</a>	<a href="#">binary (checksum signature)</a>

# Hadoop单机部署

---

- 选择将 Hadoop 安装至 /usr/local/ 中

```
sudo tar -zxf hadoop-2.7.3.tar.gz -C /usr/local # 解压到/usr/local中
cd /usr/local/
sudo mv ./hadoop-2.7.3/ ./hadoop # 将文件夹名改为hadoop
sudo chown -R hadoop:hadoop ./hadoop # 修改文件权限
```

- Hadoop 解压后即可使用。输入如下命令来检查 Hadoop 是否可用，成功则会显示 Hadoop 版本信息：

```
cd /usr/local/hadoop
./bin/hadoop version
```

Hadoop 默认模式为非分布式模式（本地模式），无需进行其他配置即可运行。

# Hadoop单机部署

操作示例: 执行下面的命令, 会显示所有示例的简介信息, 包括grep/join/word count等

```
cd /usr/local/hadoop
./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar
```

```
xc@ubuntu16g-1:/usr/local/hadoop$ ./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar
An example program must be given as the first argument.
Valid program names are:
  aggregatewordcount: An Aggregate based map/reduce program that counts the words in the input files.
  aggregatewordhist: An Aggregate based map/reduce program that computes the histogram of the words in the input files.
  bbp: A map/reduce program that uses Bailey-Borwein-Plouffe to compute exact digits of Pi.
  dbcount: An example job that count the pageview counts from a database.
  distbbp: A map/reduce program that uses a BBP-type formula to compute exact bits of Pi.
  grep: A map/reduce program that counts the matches of a regex in the input.
  join: A job that effects a join over sorted, equally partitioned datasets
  multifilewc: A job that counts words from several files.
  pentomino: A map/reduce tile laying program to find solutions to pentomino problems.
  pi: A map/reduce program that estimates Pi using a quasi-Monte Carlo method.
  randomtextwriter: A map/reduce program that writes 10GB of random textual data per node.
  randomwriter: A map/reduce program that writes 10GB of random data per node.
  secondarysort: An example defining a secondary sort to the reduce.
  sort: A map/reduce program that sorts the data written by the random writer.
  sudoku: A sudoku solver.
  teragen: Generate data for the terasort
  terasort: Run the terasort
  teravalidate: Checking results of terasort
  wordcount: A map/reduce program that counts the words in the input files.
  wordmean: A map/reduce program that counts the average length of the words in the input files.
  wordmedian: A map/reduce program that counts the median length of the words in the input files.
  wordstandarddeviation: A map/reduce program that counts the standard deviation of the length of the words in the input files.
```

# Hadoop单机部署

---

操作示例: 以grep为例, 从所有文件中筛选出符合正则表达式的单词, 并统计单词次数

```
cd /usr/local/hadoop
mkdir input
cp ./etc/hadoop/*.xml ./input # 将配置文件复制到input目录下
./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar
grep ./input ./output 'dfs[a-z.]+'
cat ./output/* #查看运行结果
```

Hadoop运行程序时, 输出目录不能存在, 否则会报错

Hadoop默认不会覆盖结果文件, 因此, 再次运行上面的实例会提示出错, 如果要再次运行, 需要先将output文件夹删除

```
rm -f ./output
```



# Hadoop伪分布式部署

---

Hadoop 的配置文件位于 `/usr/local/hadoop/etc/hadoop/` 中，伪分布式需要修改2个配置文件 **core-site.xml** 和 **hdfs-site.xml**

配置文件是 xml 格式，每个配置以声明 property 的 name 和 value 的方式来实现

- 修改配置文件：core-site.xml，hdfs-site.xml，mapred-site.xml
- 初始化文件系统 `hadoop namenode -format`
- 启动所有进程 `start-all.sh`
- 访问web界面，查看Hadoop信息
- 运行实例

# Hadoop伪分布式部署

---

修改配置文件 **core-site.xml**

```
<configuration>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>file:/usr/local/hadoop/tmp</value>
    <description>Abase for other temporary directories.</description>
  </property>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

- **hadoop.tmp.dir**表示存放临时数据的目录，即包括**NameNode**的数据，也包括**DataNode**的数据。该路径任意指定，只要实际存在该文件夹即可
- **name**为**fs.defaultFS**的值，表示**hdfs**路径的逻辑名称

# Hadoop伪分布式部署

---

修改配置文件 **hdfs-site.xml**

- **dfs.replication**表示副本的数量，伪分布式要设置为1
- **dfs.namenode.name.dir**表示名称节点的元数据保存目录
- **dfs.datanode.data.dir**表示数据节点的数据保存目录

Hadoop的运行方式取决于配置文件，启动Hadoop时会读取配置文件；如果想从伪分布式模式切换回单机模式，只需要删除**core-site**中的配置项即可

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:/usr/local/hadoop/tmp/dfs/name</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:/usr/local/hadoop/tmp/dfs/data</value>
  </property></configuration>
```

# Hadoop伪分布式部署

---

执行名称节点格式化

```
cd /usr/local/hadoop  
./bin/hdfs namenode -format
```

若执行成功，则会看到has been successfully formatted 和 Exiting with status 0提示信息

若执行失败，则会看到Exiting with status 1

# Hadoop伪分布式部署

---

## 启动Hadoop

```
cd /usr/local/hadoop  
./sbin/start-dfs.sh
```

通过执行jps来判断是否启动成功，若成功，则会出现NameNode，DataNode和Secondary Node的进程

## 关闭Hadoop

```
cd /usr/local/hadoop  
./sbin/stop-dfs.sh
```

下次启动时，无须再进行名称节点的格式化

# Hadoop伪分布式部署

---

启动Hadoop报错 localhost: Error: JAVA\_HOME is not set and could not be found

```
xc@ubuntu16g-1:/usr/local/hadoop$ ./sbin/start-dfs.sh
Starting namenodes on [localhost]
xc@localhost's password:
localhost: Error: JAVA_HOME is not set and could not be found.
xc@localhost's password: [ ]
```

Java环境已经配置好了，终端输入java -version也能显示版本信息。是因为hadoop-env.sh中的Java路径配置出错，改成绝对路径就可以了

# Hadoop伪分布式部署

---

## 使用Web界面查看HDFS信息

在浏览器地址栏中输入<http://localhost:50070>就可以查看名称节点和数据节点信息，还可以在线查看HDFS中的文件

# Hadoop伪分布式部署

---

关于三种Shell命令方式的区别：

- `hadoop fs`

`hadoop fs`适用于任何不同的文件系统，比如本地文件系统和HDFS文件系统

- `hadoop dfs`

`hadoop dfs`只能适用于HDFS文件系统

- `hdfs dfs`

`hdfs dfs`跟`hadoop dfs`的命令作用一样，也只能适用于HDFS文件系统



# Hadoop伪分布式部署

---

操作示例：还使用单机模式中的例子

```
cd /usr/local/hadoop
./bin/hdfs dfs -mkdir -p /user/hadoop # 在HDFS中为hadoop用户创建目录
./bin/hdfs dfs -mkdir input # 在HDFS中创建hadoop用户对应的input目录
./bin/hdfs dfs -put ./etc/hadoop/*.xml input # 将本地文件复制到HDFS中
./bin/hdfs dfs -ls input # 查看HDFS中的文件列表
./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar
grep input output 'dfs[a-z.]+'
./bin/hdfs dfs -cat output/* #查看运行结果
```

# Hadoop分布式部署

---

1. 选取主节点Master (10.11.6.91)，从节点Slave (10.11.6.92/93/94)
2. 所有的节点，不管是Master还是Slave，都部署好Java环境、创建hadoop用户、安装hadoop、安装SSH
3. 所有节点配置SSH免密钥登陆
  - (1) 傻瓜式：具体步骤见《大数据基础编程、实验和案例教程》P69
  - (2) 简化命令：在Master节点上执行下面的命令

```
ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop@10.11.6.92  
ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop@10.11.6.93  
ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop@10.11.6.94
```

# Hadoop分布式部署

---

## 4. 每个节点修改host文件

```
sudo vim /etc/hosts
10.11.6.91 hadoop1
10.11.6.92 hadoop2
10.11.6.93 hadoop3
10.11.6.94 hadoop4
```

## 5. 修改配置文件，具体见《大数据基础编程、实验和案例教程》P70 /usr/local/hadoop/etc/hadoop下

(1) slaves

(2) core-site.xml 设置临时文件的保存路径，注意fs.defaultFS指定了HDFS的访问地址，使用Master的hostip

(3) hdfs-site.xml 设置冗余副本的数量、secondary namenode的地址、以及名称节点、数据节点的数据保存目录

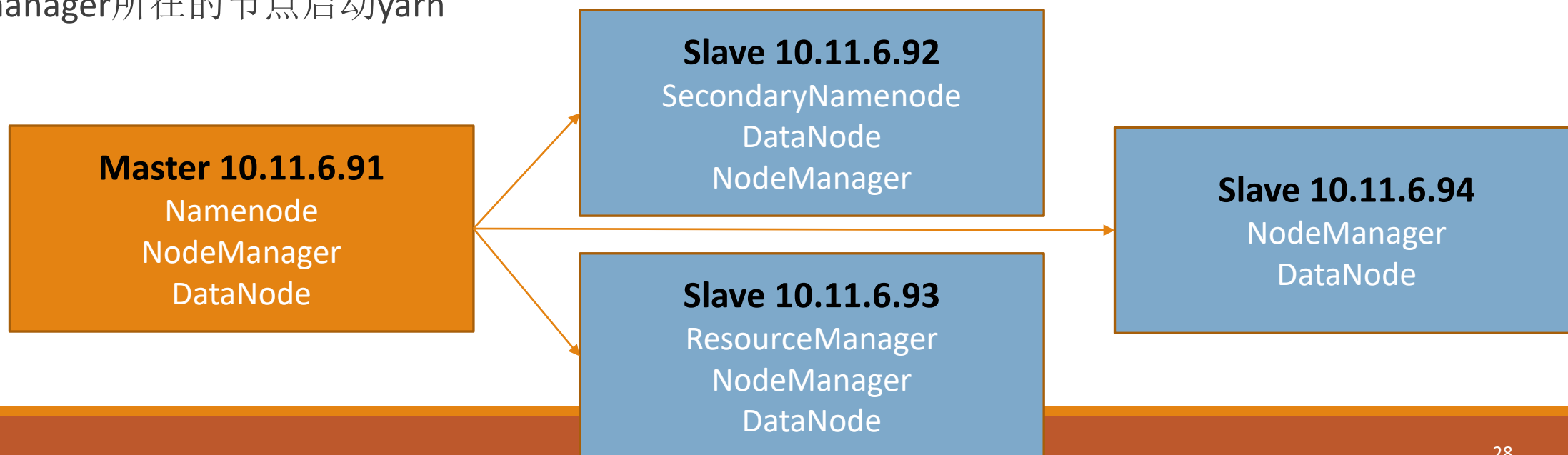
(4) mapred-site.xml

# Hadoop分布式部署

5. 启动hadoop，执行以下命令

```
start-dfs.sh  
start-yarn.sh
```

需要注意的是，如果namenode和resource manager设置的不是同一个节点，则需要在resource manager所在的节点启动yarn



# JPS

```
hadoop@ubuntu16g-2:~$ jps
The program 'jps' can be found in the following packages:
* openjdk-8-jdk-headless
* openjdk-9-jdk-headless
```

可能出现的问题: **-bash: jps: command not found** 或者

原因可能是java环境变量配置得不对

解决方案:

## 1. 先查看java位置

```
which java
```

## 2. 如果位置为/usr/bin/java, 则执行下面的命令删除这个软链接

```
sudo ln -s /usr/lib/jvm/jdk1.8.0_181/bin/java
```