**Morning：**

1. Hadoop概述

**-Hadoop常用组件**

• HDFS：Hadoop分布式文件系统

• Mapreduce：分布式计算框架(映射+规约：指令分发+结果合并)

• Yarn：集群资源管理系统

• Zookeeper(分布式协作服务)

• Hbase(分布式列存数据库)

• Hive(基于Hadoop的数据仓库)

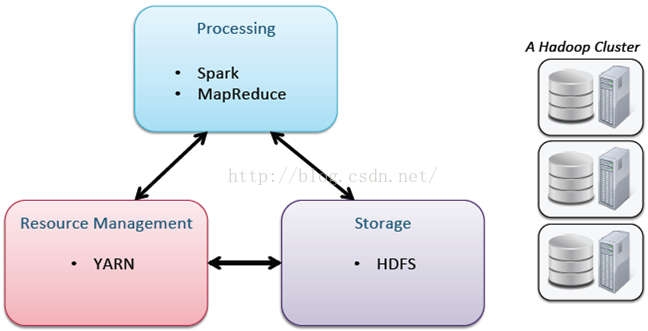
• Sqoop(数据同步工具)

• Pig(基于Hadoop的数据流系统)

• Mahout(数据挖掘算法库)

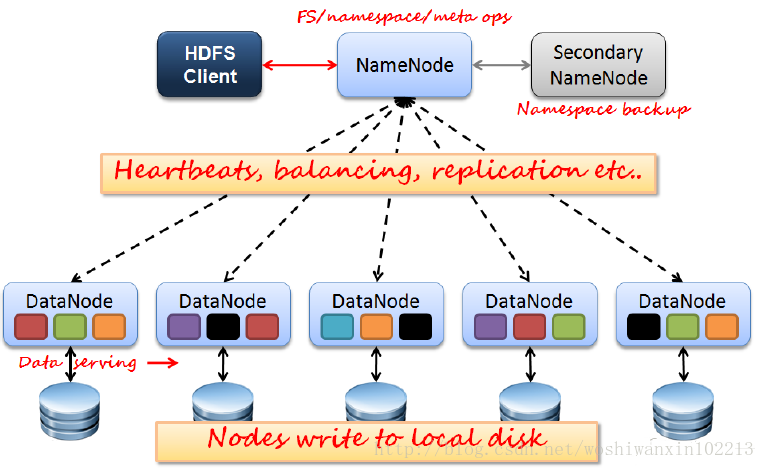
• Flume(日志收集工具)

**-三大核心组件关系图**



1. HDFS

**-HDFS结构**



**工作原理：**

1. Client切分文件，请求写入
2. NameNode决定存储节点、备份策略，返回给Client
3. Client与DataNode交互，执行写入/读取
4. DataNode存储完毕后，汇总信息给NameNode
5. Secondary NameNode，记录数据变更日志，定期把fsimage与日志合并；推送给NameNode。

**-角色与概念**NameNode：

-Master节点，管理HDFS的名称空间和数据块映射信息(存到哪个节点)

-配置副本策略(决定存多少份)，处理所有客户端请求。

Secondary NameNode：

-定期合并fsimage(映射信息)和fsedits(数据变更日志)

-紧急情况下，可辅助恢复NameNode(不是热备)

DataNode：

-数据存储节点；汇报存储信息给NameNode(存完之后告诉NameNode)

Client：

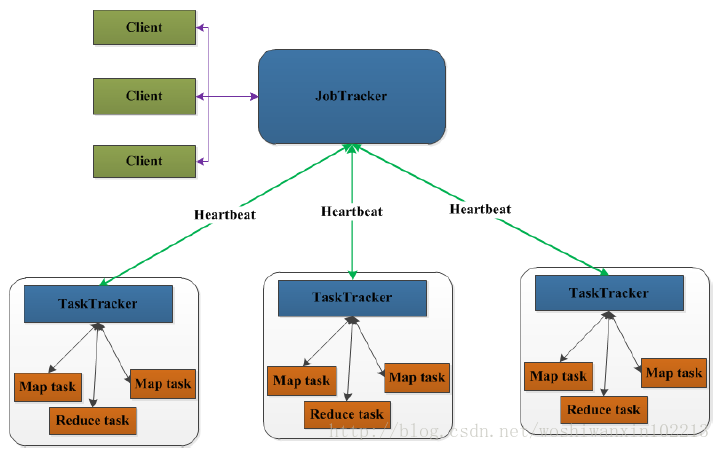
-切分文件(默认每块128M)，访问HDFS(读写)

-与NameNode交互，获取文件位置信息

-与DataNode交互，读取和写入数据。

3、Mapreduce

**-Mapreduce结构**

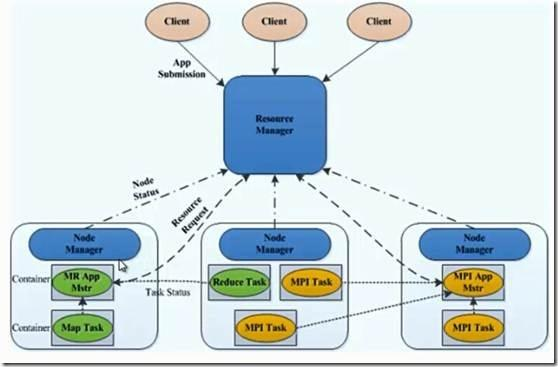


JobTacker：将任务分解，分派给TaskTracker

TaskTracker：运行MapTack和ReduceTask，汇报任务状态给JobTaker

4、Yarn

-Yarn结构



YARN 的核心思想：

将 JobTracker 和 TaskTacker 进行分离,它由下面几大构成组件:

- ResourceManager 一个全局的资源管理器

- NodeManager 每个节点(RM)代理

- ApplicationMaster 表示每个应用

- 每一个ApplicationMaster有多个Container在NodeManager上运行

**Afternoon：**

1、单机模式Hadoop

**-准备工作**

虚拟机nn01：2G内存、2CPU、20G硬盘，IP：192.168.1.10

对域名强依赖(正反解析)

**-单机模式**

1）依赖包安装

获取软件：http://hadoop.apache.org

[root@nn01 ~]# yum -y install java-1.8.0-openjdk-devel

#安装java环境、jps工具

2）hadoop安装

[root@nn01 ~]# tar -xf hadoop-2.7.6.tar.gz

[root@nn01 ~]# mv hadoop-2.7.6 /usr/local/hadoop

3）环境配置

[root@nn01 ~]# rpm -ql java-1.8.0-openjdk

/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.../jre/...

[root@nn01 ~]# cd /usr/local/hadoop

[root@nn01 hadoop]# vim ./etc/hadooop/hadoop-env.sh

export JAVA\_HOME="/usr/lib/jvm/java-1.8.0-openjdk-.../jre"

export HADOOP\_CONF\_DIR=${..-"/usr/local/hadoop/etc/hadoop"}

[root@nn01 hadoop]# ./bin/hadoop version

1. 测试：统计词频

[root@nn01 hadoop]# ./bin/hadoop jar \

>./share/hadoop/mapreduce/hadoop-map..-ex..-2.7.6.jar \

> wordcount README.txt xx

#bin/hadoop是shell脚本，jar选项可以执行jar包

#hadoop-mapreduce-examples.jar支持多种统计分析方法

#本例实现将README.txt统计词频，输出到当前的xx目录

**-伪分布式**

安装与配置与完全分布式类似，区别是所有角色安装在一台机器。

xml文件配置格式：

<property>

<name>关键字</name>

<value>变量值</value>

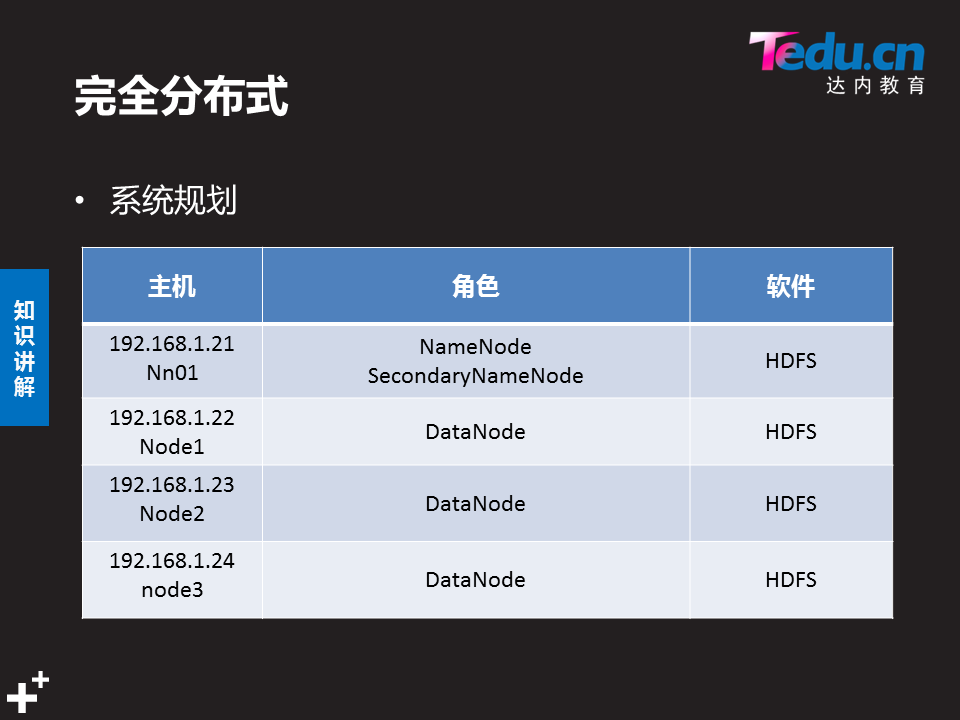
<description> 描述 </description>

</property>

参考官方手册：https://hadoop.apache.org/docs/

2、完全分布式Hadoop

**步骤1：系统规划**



**步骤2：基础环境准备**

-新建3台虚拟机(node1~3)：2G内存、2CPU、20G硬盘

-安装java-1.8.0-openjdk-devel

-配置/etc/hosts

**步骤3：配置ssh信任关系**

[root@nn01 ~]# ssh-keygen ; ssh-copy-id xxx

#给所有node包括本机发送key

[root@nn01 ~]# vim /etc/ssh/ssh\_config

StrictHostKeyChecking no

#取消keychecking询问

**步骤4：环境配置文件hadoop-env.sh**

JAVA\_HOME、HADOOP\_CONF\_DIR

**步骤5：核心配置文件core-site.xml**

[root@nn01 hadoop]# vim core-site.xml

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://nn01:9000</value>

</property>

#声明使用的文件系统为hdfs，nn01为NameNode节点地址

<property>

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

<value>/var/hadoop</value>

</property>

#所有数据的根目录，决定hdfs的总容量

</configuration>

**步骤6：HDFS配置文件**

[root@nn01 hadoop]# vim hdfs-site.xml

<configuration>

<property>

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

<value>nn01:50070</value>

</property>

#定义Namenode节点地址

<property>

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

<value>nn01:50090</value>

</property>

#定义Secondary Namenode节点地址

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

#定义备份策略

</configuration>

**步骤7：节点配置文件slaves**

[root@nn01 hadoop]# vim slaves

node1

node2

node3

#写DataNode的节点名称

**步骤8：HDFS完全分布式配置**

1）所有主机创建/tmp/hadoop，同步hadoop目录

[root@nn01 ~]# for i in node{1..3}

> do

> ssh $i /tmp/hadoop

> rsync -avz --delete /usr/local/hadoop $i:/usr/local/

> done

1. 在nn01执行格式化、启动集群

[root@nn01 hadoop]# ./bin/hdfs namenode -format

[root@nn01 hadoop]# ./sbin/start-dfs.sh

3）JPS验证角色

[root@nn01 ~]# jps

11395 Jps

11275 SecondaryNameNode

11086 NameNode

[root@node? ~]# jps

10438 Jps

10364 DataNode

1. 节点验证

[root@nn01 hadoop]# ./bin/hdfs dfsadmin -report

... ...

Live datanodes (3):

**步骤：Day06续**

**-排错思路**

1）检查3个配置文件是否正确(域名、端口)

2）检查防火墙是否卸载、/etc/hosts是否配置正确

3）查看/usr/local/hadoop/logs/xx.log，检查是NameNode还是DataNode的问题，只查看非”INFO”的日志行即可。