

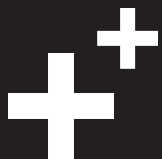
Linux 大数据

NSD HADOOP

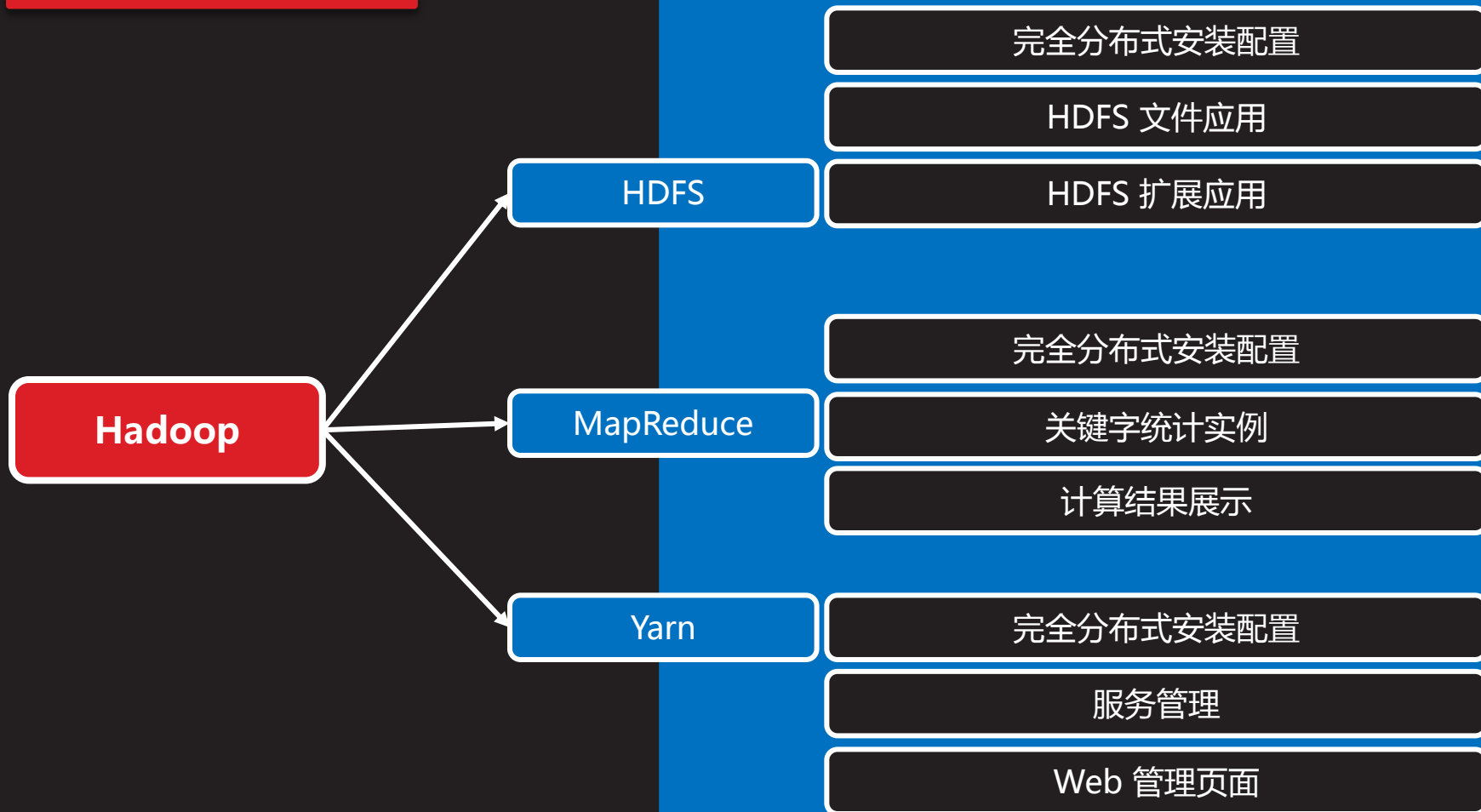
DAY02

内容

上午	09:00 ~ 09:30	课程回顾
	09:30 ~ 10:20	HADOOP 完全分布式
	10:30 ~ 11:20	HDFS 分布式文件系统
	11:30 ~ 12:00	HDFS 扩展应用
下午	14:00 ~ 14:50	Mapreduce 与 Yarn
	15:00 ~ 15:50	
	16:10 ~ 17:00	Hadoop 分析实例
	17:10 ~ 18:00	总结和答疑



大数据



Hadoop 完全分布式

完全分布式

- Hadoop最大的优势就是分布式集群计算，所以在生产环境下都是搭建的最后一种模式：完全分布模式

- 技术准备
- 系统规划
- 搭建
- 测试
- 上线使用



完全分布式

- 系统规划

主机	角色	软件
192.168.4.10 master	NameNode SecondaryNameNode ResourceManager	HDFS YARN
192.168.4.11 node1	DataNode NodeManager	HDFS YARN
192.168.4.12 node2	DataNode NodeManager	HDFS YARN
192.168.4.13 node3	DataNode NodeManager	HDFS YARN



搭建完全分布式

- 1、安装操作系统

- 注意：只开启必要的服务，关闭其他无关的系统服务，系统最小化，服务最小化
- 注意：关闭系统防火墙
- 注意：关闭 selinux
- iptables-save
- sestatus



搭建完全分布式

- 2、在所有系统上安装 JAVA 环境和调试工具 jps
 - 注意：保证所有机器系统版本及 java 版本的一致性
 - 注意：保证所有安装路径的一致性
 - java -version
 - jps



搭建完全分布式

- 3、配置 主机名 和 ip 对应关系 /etc/hosts
 - 注意：所有主机的配置都需要修改
 - 注意：master 要能 ping 通所有主机
 - 注意：node 要能 ping 通 master



完全分布式

- 4、配置 SSH 信任关系

- 注意：不能出现要求输入 yes 的情况，每台机器都要能登录成功
- `ssh-keygen -b 2048 -t rsa -N "" -f key`
- `ssh-copy-id -i ./key.pub root@ip.xx.xx.xx`



HDFS 分布式文件系统

完全分布式

- HDFS 完全分布式系统配置
 - 配置 `hadoop-env.sh`
 - 配置 `core-site.xml`
 - 配置 `hdfs-site.xml`



完全分布式

- 5、HDFS 完全分布式配置

- `hadoop-env.sh`

`which java`

`readlink -f $(which java)`

`– JAVA_HOME= ""`



完全分布式

- 5、HDFS 完全分布式配置

- `hadoop-env.sh`

- `HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-
"/etc/hadoop"}`

- 配置 `HADOOP_CONF_DIR` 路径

- `/usr/local/hadoop/etc/hadoop`



完全分布式

- 5、HDFS 完全分布式配置
- core-site.xml

```
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://192.168.4.10:9000</value>
</property>
<property>
  <name>hadoop.tmp.dir</name>
  <value>/var/hadoop</value>
  <description>A base for other temporary
directories.</description>
</property>
```



完全分布式

- 5、HDFS 完全分布式配置
- core-site.xml
 - `<value>hdfs://192.168.4.10:9000</value>`
 - `<value>/var/hadoop</value>`
 - 注意：所有机器上都要创建
`mkdir -p /var/hadoop`



完全分布式

- 5、HDFS 完全分布式配置

- hdfs-site.xml

```
<property>  
    <name>dfs.replication</name>  
    <value>2</value>  
</property>
```



完全分布式

- 5、HDFS 完全分布式配置
- hdfs-site.xml
 - dfs.namenode.http-address
 - dfs.namenode.secondary.http-address
 - dfs.namenode.name.dir
 - dfs.datanode.data.dir
 - dfs.replication
 - dfs.webhdfs.enabled
 - dfs.permissions.enabled



完全分布式

- 5、HDFS 完全分布式配置
 - 配置完成以后，把 hadoop 的文件夹拷贝到所有机器
 - 在 namenode 上执行格式化操作
`./bin/hdfs namenode -format`
 - 在没有报错的情况下启动集群
`./sbin/start-dfs.sh`



完全分布式

- 5、HDFS 完全分布式配置
 - 验证：
 - 启动以后分别在 namenode 和 datanode 执行命令
jps
 - 成功的情况下应该可以看见
 - NameNode
 - SecondaryNode
 - DataNode



HDFS 基本使用

HDFS 使用

- HDFS 基本命令
 - `./bin/hadoop fs -ls /`
 - 对应 shell 命令 `ls /`
 - `./bin/hadoop fs -mkdir /abc`
 - 对应 shell 命令 `mkdir /abc`
 - `./bin/hadoop fs -rmdir /abc`
 - 对应 shell 命令 `rmdir /abc`



HDFS 使用

- HDFS 基本命令
 - `./bin/hadoop fs -touchz /urfile`
 - 对应 shell 命令 `touch /urfile`
 - `./bin/hadoop fs -cat /urfile`
 - 对应 shell 命令 `cat /urfile`
 - `./bin/hadoop fs -rm /urfile`
 - 对应 shell 命令 `rm /urfile`



HDFS 使用

- HDFS 基本命令

- 上传文件

- `./bin/hadoop fs -put localfile /remotefile`

- 下载文件

- `./bin/hadoop fs -get /remotefile`



Yarn 安装与配置

- yarn 配置文件
 - mapred-site.xml
 - yarm-site.xml



Yarn 安装与配置

- yarn 配置文件
 - mapred-site.xml

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



Yarn 安装与配置

- yarn 配置文件
 - yarn-site.xml

```
<property>  
    <name>yarn.resourcemanager.hostname</name>  
    <value>master</value>  
</property>
```



Yarn 安装与配置

- yarn 配置
 - 配置完成以后把配置同步到所有主机
 - 启动 yarn 服务
 - `./sbin/start-yarn.sh`
 - 在所有主机上执行 `jps` , 查看是否启动成功
 - `resourcemanager`
 - `nodemanager`



提交验证



Hadoop 验证

- 创建文件夹
 - `./bin/hadoop fs -mkdir /input`
- 导入要分析的文件
 - `./bin/hadoop fs -put *.txt /input`



Hadoop 验证

- 提交分析作业
 - `./bin/hadoop`
`jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar wordcount /input /output`
- 查看结果
 - `./bin/hadoop fs -cat output/*`



HDFS 进阶应用



NFS 网关

- NFS 网关用途

- 1.用户可以通过操作系统兼容的本地NFSv3客户端来浏览HDFS文件系统
- 2.用户可以从HDFS文件系统下载文档到本地文件系统
- 3.用户可以通过挂载点直接流化数据。支持文件附加，但是不支持随机写
- NFS 网关支持NFSv3和允许HDFS 作为客户端文件系统的一部分被挂载



NFS 网关

- 特性与注意事项
 - 不支持随机写
 - 在非安全模式，运行网关的用户是代理用户
 - 在安全模式时，Kerberos keytab中的用户是代理用户
 - AIX NFS有一些知道的问题，不能让默认的HDFS NFS网关正常工作，如果想在 AIX 访问 NFS 网关需要配置下面的参数

```
<property>
  <name>nfs.aix.compatibility.mode.enabled</name>
  <value>true</value>
</property>
```



NFS 网关

- 特性与注意事项
 - HDFS超级用户是与NameNode进程本身具有相同标识的用户，超级用户可以执行任何操作，因为权限检查永远不会为超级用户失败。

```
<property>
  <name>nfs.superuser</name>
  <value>the_name_of_hdfs_superuser</value>
</property>
```



NFS 网关

- 特性与注意事项

- 如果客户端安装允许访问时间更新，在某些Unix系统上，用户可以通过使用“noatime”安装来禁用访问时间更新

```
<property>
```

```
  <name>dfs.namenode.access.time.precision</name>
```

```
  <value>0</value>
```

```
  <description>The access time for HDFS file is precise upto this value.
```

```
    The default value is 1 hour. Setting a value of 0 disables access times for HDFS.
```

```
  </description>
```

```
</property>
```



NFS 网关

- 特性与注意事项
 - nfs.dump.dir
 - 用户需要更新文件转储目录参数。NFS客户端经常重新安排写操作，顺序的写操作会以随机到达NFS网关。这个目录常用于临时存储无序的写操作。对于每个文件，无序的写操作会在他们积累在内存中超过一定阈值(如。1 mb)被转储。需要确保有足够的空间的目录。例如，如果应用上传10个100M，那么这个转储目录推荐有1GB左右的空间，以便每个文件都发生最坏的情况。只有NFS网关需要在设置该属性后重启。



NFS 网关

- 特性与注意事项
 - `nfs.exports.allowed.hosts`
 - 默认情况下，`export`可以被任何客户端挂载。为了更好的控制访问，可以设置属性。值字符串为机器名和访问策略，通过空格来分割。机器名的格式可以是单一的主机，Java的正则表达式或者IPv4地址。访问权限使用`rw`或`ro`来指定导出目录的读/写或机器只读访问。如果访问策略没被提供，默认为只读的。每个条目使用“;”来分割。



NFS 网关

- 调试与日志排错
 - 在配置 NFS 网关过程中经常会碰到各种各样的错误，如果出现了错误，打开调试日志是一个不错的选择
- log4j.property
 - log4j.logger.org.apache.hadoop.hdfs.nfs=DEBUG
 - log4j.logger.org.apache.hadoop.oncrpc=DEBUG



NFS 网关

- NFS & portmap 相关配置
 - core-site.xml
 - hdfs-site.xml



NFS 网关

- core-site.xml

```
<property>
  <name>hadoop.proxyuser.root.groups</name>
  <value>*</value>
</property>
<property>
  <name>hadoop.proxyuser.root.hosts</name>
  <value>*</value>
</property>
```



NFS 网关

- core-site.xml
 - `hadoop.proxyuser.{nfsuser}.groups`
 - `hadoop.proxyuser.{nfsuser}.hosts`
 - 这里的 `nfsuser` 是你机器上真实运行 `nfsgw` 的用户
 - 在非安全模式，运行nfs网关的用户为代理用户
 - `groups` 为挂载点用户所使用的组
 - `hosts` 为挂载点主机地址



NFS 网关

- hdfs-site.xml
 - nfs.exports.allowed.hosts
 - 设置允许访问 NFS 主机列与权限，默认 “ro”

```
<property>  
  <name>nfs.exports.allowed.hosts</name>  
  <value>* rw</value>  
</property>
```



NFS 网关

- hdfs-site.xml
 - dfs.namenode.access.time.precision
 - 关闭 access time

```
<property>  
  <name>dfs.namenode.access.time.precision</name>  
  <value>3600000</value>  
</property>
```



NFS 网关

- hdfs-site.xml
 - nfs.dump.dir
 - 设置转储目录

```
<property>  
  <name>nfs.dump.dir</name>  
  <value>/tmp/.hdfs-nfs</value>  
</property>
```



NFS 网关

- `hdfs-site.xml`
 - `nfs.rtmax` & `nfs.wtmax`
 - 用户可以像访问本地文件系统的一部分一样访问HDFS，但硬链接和随机写还不支持。对于大文件I/O的优化，可以在mount的时候增加NFS传输的大小（`rsize`和`wsize`）。在默认情况下，NFS网关支持1MB作为最大的传输大小。更大的数据传输大小，需要在`hdfs-site.xml`中设置“`nfs.rtmax`”和“`nfs.wtmax`”



NFS 网关

- hdfs-site.xml
 - nfs.rtmax & nfs.wtmax

```
<property>  
  <name>nfs.rtmax</name>  
  <value>4194304</value>  
</property>  
<property>  
  <name>nfs.wtmax</name>  
  <value>1048576</value>  
</property>
```



NFS 网关

- hdfs-site.xml
 - nfs.port.monitoring.disabled
 - 允许从没有权限的客户端挂载 nfs

```
<property>  
  <name>nfs.port.monitoring.disabled</name>  
  <value>>false</value>  
</property>
```



NFS 网关

- nfs.map

- 系统管理员必须确保在NFS客户端的用户和在HDFS网关主机上的用户有相同的名称和UID。不同主机上创建的用户需要修改UID（例如使用“usermod -u 123 myusername”），在NFS客户端或者NFS网关主机来进行。如果客户端的用户和NFS网关的用户 uid 不能保持一致需要我们配置 nfs.map 的静态映射关系

- nfs.map

```
uid 10 100 # Map the remote UID 10 the local UID 100
gid 11 101 # Map the remote GID 11 to the local GID 101
```



NFS 网关

- hdfs-site.xml 基本配置

```
<property>  
  <name>nfs.exports.allowed.hosts</name>  
  <value>* rw</value>  
</property>  
<property>  
  <name>nfs.dump.dir</name>  
  <value>/tmp/.hdfs-nfs</value>  
</property>
```



NFS 网关

- 启动与挂载
 - 配置完所有的参数以后就可以启动服务了
 - 这里要注意 关闭系统的 portmap 和 nfs 服务
 - 添加用户
 - 首先打开 log4j 的调试日志
 - 重启 hdfs 集群服务
 - 启动 portmap 服务
 - 启动 nfs3 服务



NFS 网关

- 启动与挂载
 - 启动 portmap 服务
`./sbin/hadoop-daemon.sh --script ./bin/hdfs start portmap`
 - 启动 nfs3
`./sbin/hadoop-daemon.sh --script ./bin/hdfs start nfs3`
 - 这里要特别注意：
 - 启动 portmap 需要使用 root 用户
 - 启动 nfs3 需要使用 core-site 里面设置的用户



NFS 网关

- 启动与挂载
 - 挂载 nfs
 - 目前NFS v3仅使用TCP作为传输协议。 不支持NLM，因此需要安装选项“nolock”。 强烈建议使用安装选项“sync”，因为它可以最小化或避免重新排序写入，这将导致更可预测的吞吐量。 未指定同步选项可能会导致上传大文件时出现不可靠的行为
 - 如果必须使用软安装，用户应该给它一个相对较长的超时（至少不小于主机上的默认超时）



NFS 网关

- 启动与挂载
 - 挂载 NFS 参数
 - vers=3
 - proto=tcp
 - nolock
 - noacl
 - noatime
 - sync

```
mount -t nfs
```

```
-o vers=3,proto=tcp,nolock,noacl,noatime,sync ip.xx.xx.xx:/  
/localdir
```



NFS 网关

- 挂载完成之后就可以进行读写测试了
 - 练习：在 hdfs 基础上配置 ntfs 网关



Hadoop 节点管理

- HDFS 增加节点
 - 1. 配置所有hadoop环境，包括主机名、ssh免密码登录、禁用 selinux、iptables、安装 java 环境
 - 2. 把namnode的配置文件复制到配置文件目录下
 - 3. 修改namenode的slaves文件增加该节点
 - 5. 在该节点启动Datanode
`./sbin/hadoop-daemon.sh start datanode`



Hadoop节点管理

- HDFS 节点管理

- 6. 设置同步带宽，并同步数据

- `./bin/hdfs dfsadmin -setBalancerBandwidth 67108864`

- `./sbin/start-balancer.sh -threshold 5`

- 7. 查看集群状态

- `./bin/hdfs dfsadmin -report`



Hadoop 节点管理

- HDFS 删除节点

- 配置NameNode的hdfs-site.xml

- dfs.replication 副本数量

- 增加 dfs.hosts.exclude 配置

- <property>

- <name>dfs.hosts.exclude</name>

- <value>/usr/local/hadoop/etc/hadoop/exclude</value>

- </property>

- 增加 exclude 配置文件，写入要删除的节点 ip

- 更新数据

- ./bin/hdfs dfsadmin -refreshNodes



Hadoop 节点管理

- HDFS 修复节点

- 修复节点比较简单
- 单独配置一台新 datanode
- 启动服务

`./sbin/hadoop-daemon.sh start datanode`

- 数据恢复是自动的
- 我们上线以后会自动恢复数据，如果数据量非常巨大，可能需要一定的时间



Hadoop节点管理

- Yarn 的相关操作
 - 由于在 2.x hadoop 引入了 yarn 框架，对于计算节点的操作已经变得非常简单
 - 增加节点
`sbin/yarn-daemon.sh start nodemanager`
 - 删除节点
`sbin/yarn-daemon.sh stop nodemanager`
 - 查看节点 (Resourcemanager)
`./bin/yarn node -list`



Hadoop节点管理

- HDFS 还有很多其他的应用方式，比如 native-hdfs，有兴趣的可以自行测试
 - 用到的软件依赖
 - cmake，fuse-devel
 - protobuf
 - protobuf-c
 - native-hdfs-fuse



课程知识点总结

- Hadoop 完全分布式安装的配置
 - java 环境
 - ssh key 认证
 - hosts 配置
 - hadoop-env.sh
 - core-site.xml
 - hdfs-site.xml
 - mapred-site.xml
 - yarn-site.xml
 - slaves



课程知识点总结

- Hadoop NFS网关
 - hadoop-env.sh
 - core-site.xml
 - hdfs-site.xml
 - mount 参数



课程知识点总结

- Hadoop 节点管理
 - datanode 的增加 删除 修复
 - nodemanager 的增加 删除 修复



总结答疑
