

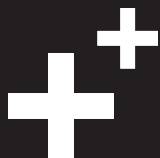
Linux 大数据

NSD HADOOP

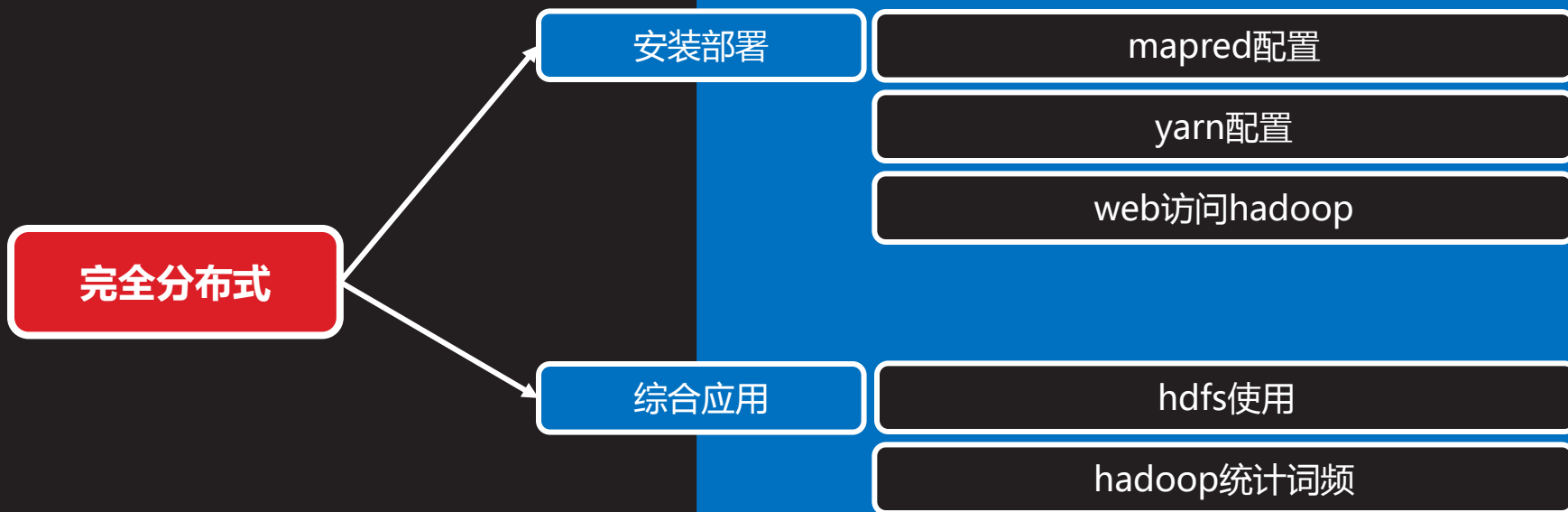
DAY02

内容

上午	09:00 ~ 09:30	课程回顾
	09:30 ~ 10:20	完全分布式
	10:30 ~ 11:20	
	11:30 ~ 12:00	
下午	14:00 ~ 14:50	节点管理
	15:00 ~ 15:50	NFS网关
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑



完全分布式



安装部署



安装部署

- Hadoop 三大核心组件
 - 分布式文件系统
 - HDFS 已经部署完毕
 - 分布式计算框架
 - mapreduce
 - 集群资源管理
 - yarn



安装部署

- 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



安装部署

- 分布式计算框架 mapred-site.xml
 - 改名
 - FROM: mapred-site.xml.template
 - To: mapred-site.xml
 - 资源管理类
 - mapreduce.framework.name



安装部署

- 分布式计算框架 mapred-site.xml

- 只支持 local 和 yarn 两种
- 单机使用 local
- 集群使用 yarn

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



安装部署

- 资源管理 yarn-site.xml
 - resourcemanager 地址
 - yarn.resourcemanager.hostname
 - nodemanager 使用哪个计算框架
 - yarn.nodemanager.aux-services
 - mapreduce_shuffle 是我们使用计算框架的名称



安装部署

- 资源管理 yarn-site.xml
 - yarn-site.xml 配置

```
<property>
  <name>yarn.resourcemanager.hostname</name>
  <value>nn01</value>
</property>
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
```



安装部署

- 启动服务

- /usr/local/hadoop/sbin/start-yarn.sh

- 验证服务

- /usr/local/hadoop/bin/yarn node -list

```
[root@nn01 hadoop]# ./bin/yarn node -list
```

```
... ..
```

node2:33486	RUNNING	node2:8042	0
node1:35816	RUNNING	node1:8042	0
node3:40941	RUNNING	node3:8042	0



web访问hadoop

- 使用 web 访问hadoop
 - namenode web页面(nn01)
 - <http://192.168.1.10:50070/>
 - secondary namenode web 页面(nn01)
 - <http://192.168.1.10:50090/>
 - datanode web 页面(node1,node2,node3)
 - <http://192.168.1.11:50075/>



web访问hadoop

- 使用 web 访问hadoop
 - resourcemanager web页面(nn01)
 - <http://192.168.1.10:8088/>
 - nodemanager web 页面(node1,node2,node3)
 - <http://192.168.1.11:8042/>



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`



hadoop词频统计

- 1、在集群文件系统里创建文件夹
- 2、上传要分析的文件到目录中
- 3、分析上传文件
- 4、展示结果



Hadoop 验证

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

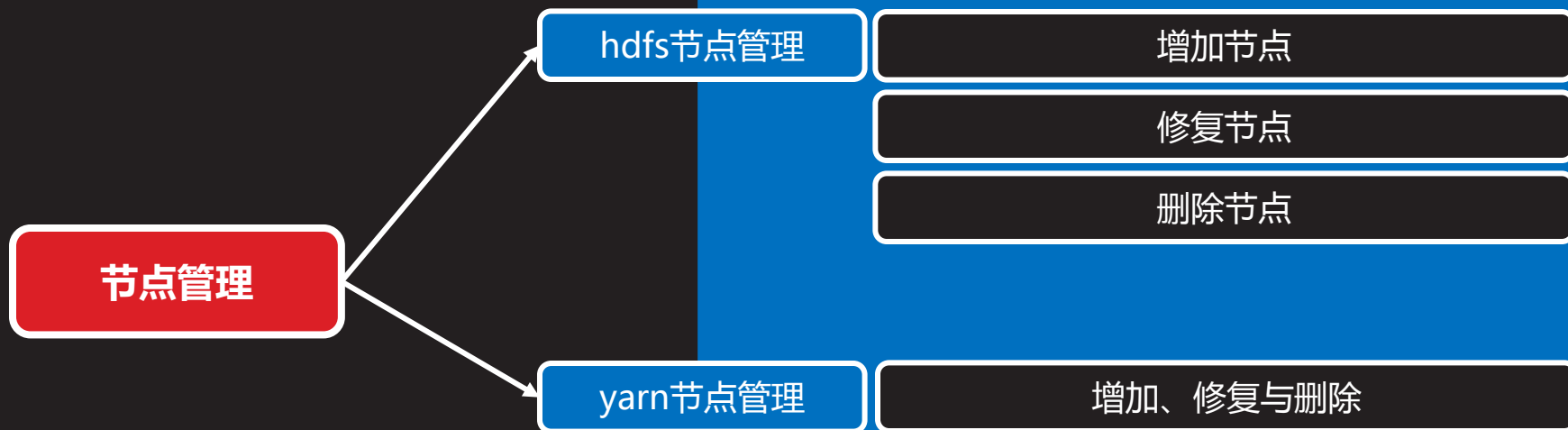


Hadoop 验证

- 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/*`



节点管理



HDFS节点管理



Hadoop 节点管理

- HDFS 增加节点
 - 1、启动一个新的系统，禁用 selinux、禁用 firewalld
 - 2、设置 ssh 免密码登录
 - 3、在所有节点增加新节点的主机信息 /etc/hosts
 - 4、安装 java openjdk 运行环境
 - 5、拷贝namenode的 /usr/local/hadoop 到本机
 - 6、修改namenode的slaves文件增加该节点
 - 7、在该节点启动Datanode
`./sbin/hadoop-daemon.sh start datanode`



Hadoop节点管理

- HDFS 节点管理

- 增加节点 续

- 8、设置同步带宽，并同步数据

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

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

- 9、查看集群状态

- `./bin/hdfs dfsadmin -report`



Hadoop节点管理

- HDFS 修复节点

- 修复节点比较简单，与增加节点基本一致
- 需要注意新节点的 ip 与 主机名 要与损坏节点一致
- 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

- HDFS 删除节点

- 配置NameNode的 hdfs-site.xml

- 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 主机名

- 更新数据

- ./bin/hdfs dfsadmin -refreshNodes



Hadoop 节点管理

- HDFS 删除节点状态
 - 查看状态 `./bin/hdfs dfsadmin -report`
 - Normal 正常状态
 - Decommissioned in Program 数据正在迁移
 - Decommissioned 数据迁移完成
- 注意：只有当状态变成 Decommissioned 才能 down 机下线



yarn节点管理



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`

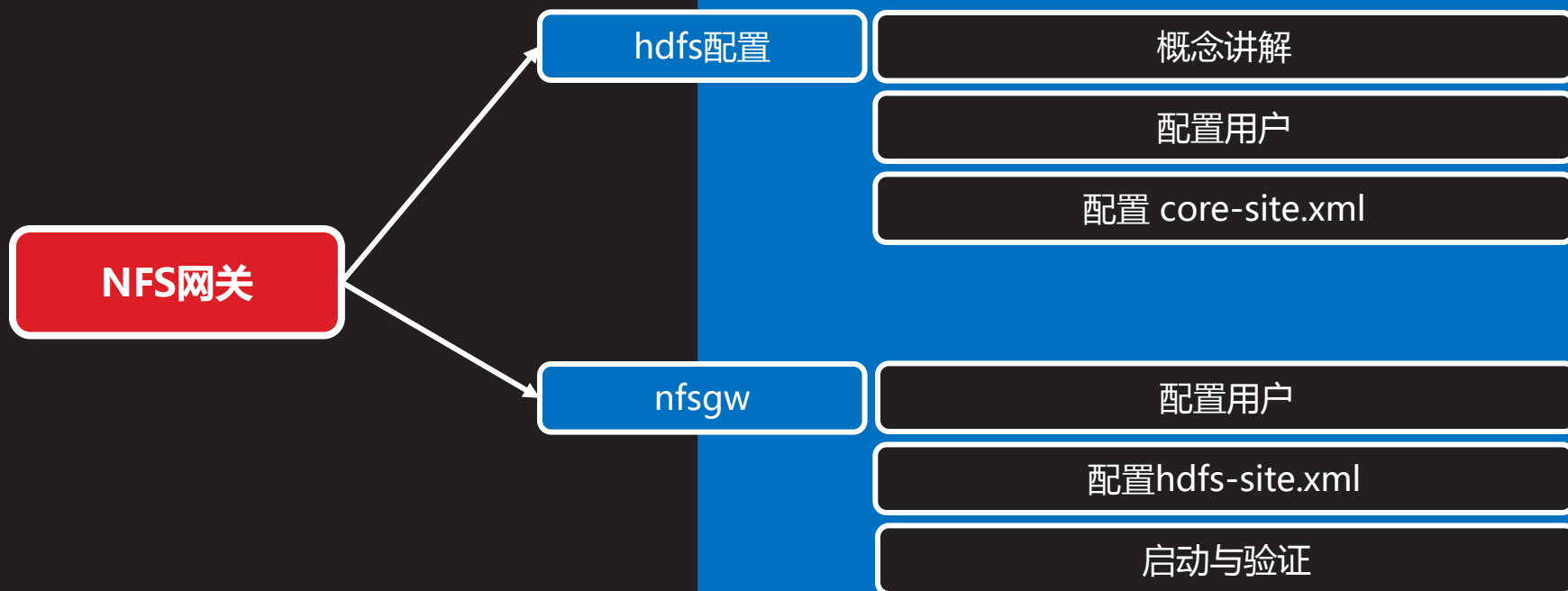


Hadoop节点管理

- yarn 的系统环境配置与 hdfs 的基础环境配置是相同的，这里就不重复列出了
- 由于 yarn 不包含数据，所以在增加删除修复节点的时候比较简单，hdfs 要注意数据安全



NFS网关



HDFS配置



NFS 网关

- NFS 网关用途
 - 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
 - 用户可以从HDFS文件系统下载文档到本地文件系统
 - 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
 - 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。
 - 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



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

- 配置代理用户
 - 在 namenode 和 nfsgw 上添加代理用户
 - 代理用户的 uid gid 用户名 必须完全相同
 - 如果因特殊原因客户端的用户和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

- 核心配置 core-site.xml
 - `hadoop.proxyuser.{代理用户}.groups`
 - `hadoop.proxyuser.{代理用户}.hosts`
 - 这里的 {代理用户} 是你机器上真实运行 nfs3 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

- 核心配置 core-site.xml

```
... ..  
    <property>  
        <name>hadoop.proxyuser.nsd1802.groups</name>  
        <value>*</value>  
    </property>  
    <property>  
        <name>hadoop.proxyuser.nsd1802.hosts</name>  
        <value>*</value>  
    </property>  
... ..
```



NFS网关

- 配置步骤
 - 停止集群所有服务
 - `./sbin/stop-all.sh`
 - 分发配置文件到所有主机
 - 启动 hdfs
 - `./sbin/start-dfs.sh`



NFSGW配置



NFS 网关

- 配置步骤
 - 1、启动一个新的系统，禁用 selinux、禁用 firewalld
 - 2、卸载 rpcbind、nfs-utils
 - 3、配置 /etc/hosts，添加所有 namenode 和 datanode 的主机名与 ip 对应关系
 - 4、安装 java openjdk 的运行环境
 - 5、同步 namenode 的 /usr/local/hadoop 到本机
 - 6、配置 hdfs-site.xml
 - 7、启动服务



NFS 网关

- 配置文件 `hdfs-site.xml`
- `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

- hdfs-site.xml 配置 续

- nfs.exports.allowed.hosts

- 配置 * rw

... ..

```
<property>
```

```
  <name>nfs.exports.allowed.hosts</name>
```

```
  <value>* rw</value>
```

```
</property>
```

... ..



NFS 网关

- hdfs-site.xml 配置 续
- 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

- 配置文件 hdfs-site.xml
 - nfs.dump.dir

... ..

```
<property>
  <name>nfs.dump.dir</name>
  <value>/var/nfstmp</value>
</property>
```

... ..

- 配置完该属性后要创建 /var/nfstmp 文件夹


```
mkdir /var/nfstmp
```
- 并且把该文件夹的属组改成 代理用户



NFS 网关

• 启动与挂载

- 设置 /usr/local/hadoop/logs 权限，为代理用户赋予读写执行的权限

```
setfacl -m user:proxyuser:rwx /usr/local/hadoop/logs
```

- 使用 root 用户启动 portmap 服务

```
./sbin/hadoop-daemon.sh --script ./bin/hdfs start portmap
```

- 使用代理用户启动 nfs3

```
./sbin/hadoop-daemon.sh --script ./bin/hdfs start nfs3
```



NFS 网关

- 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 里面设置的代理用户
 - 必须先启动 portmap 之后再启动 nfs3
 - 如果 portmap 重启了，在重启之后 nfs3 也需要重启



NFS 网关

- 启动与挂载
 - 目前NFS 只能使用v3版本
 - vers=3
 - 仅使用TCP作为传输协议。
 - proto=tcp
 - 不支持NLM
 - nolock
 - 禁用 access time 的时间更新
 - noatime



NFS 网关

• 启动与挂载

- 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

- 启动一台机器安装 nfs-utils

```
yum install nfs-utils
```

- 挂载 nfs

```
mount -t nfs -o vers=3,proto=tcp,noatime,nolock,sync \
192.168.1.15:/ /mnt/
```



总结答疑
