

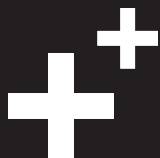
# 大型架构及配置技术

**NSD ARCHITECTURE**

**DAY06**

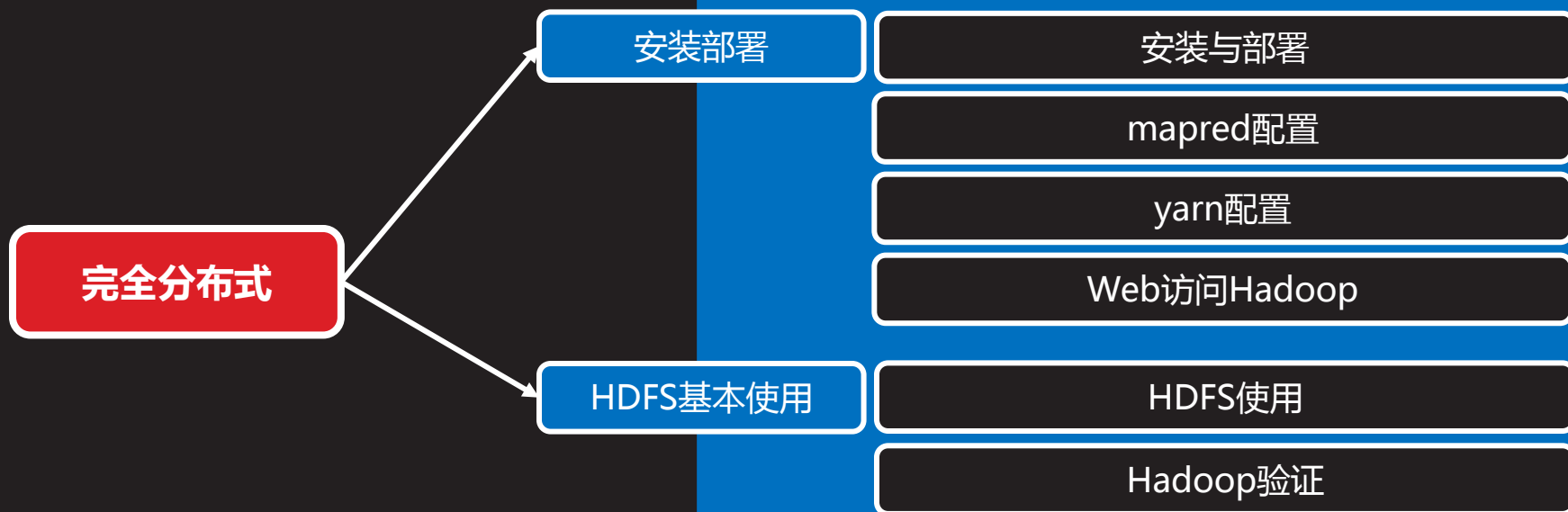
# 内容

上午	09:00 ~ 09:30	作业讲解和回顾
	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 ~ 17:10	
	17:20 ~ 18:00	总结和答疑



# 完全分布式

---



# 安装部署

---

# 安装与部署

- 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



# 安装与部署（续1）

- 系统规划

主机	角色	软件
192.168.1.21 master	NameNode SecondaryNameNode ResourceManager	HDFS YARN
192.168.1.22 node1	DataNode NodeManager	HDFS YARN
192.168.1.23 node2	DataNode NodeManager	HDFS YARN
192.168.1.24 node3	DataNode NodeManager	HDFS YARN



# mapred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



# mapred部署（续1）

- 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>
```





# yarn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 计算框架的名称  
`mapreduce_shuffle`



# yarn部署（续1）

- 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>
```



# yarn部署（续2）

- 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://192.168.1.21:50070/>
  - secondary namenode web 页面(nn01)  
# <http://192.168.1.21:50090/>
  - datanode web 页面(node1,node2,node3)  
# <http://192.168.1.22:50075/>



# Web访问Hadoop ( 续1 )

- 使用Web访问Hadoop
  - resourcemanager web页面(nn01)  
# <http://192.168.1.21:8088/>
  - nodemanager web页面(node1,node2,node3)  
# <http://192.168.1.22:8042/>



# 案例1：安装与部署

1. 对mapred和yarn文件进行配置
2. 验证访问Hadoop



# 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（续1）

- 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（续2）

- HDFS基本命令

- 上传文件

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

- 下载文件

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



## 案例2：Hadoop词频统计

1. 在集群文件系统中创建文件夹
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 ( 续1 )

- 提交分析作业

```
# ./bin/Hadoop
```

```
# jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-  
2.7.3.jar wordcount /input /output
```

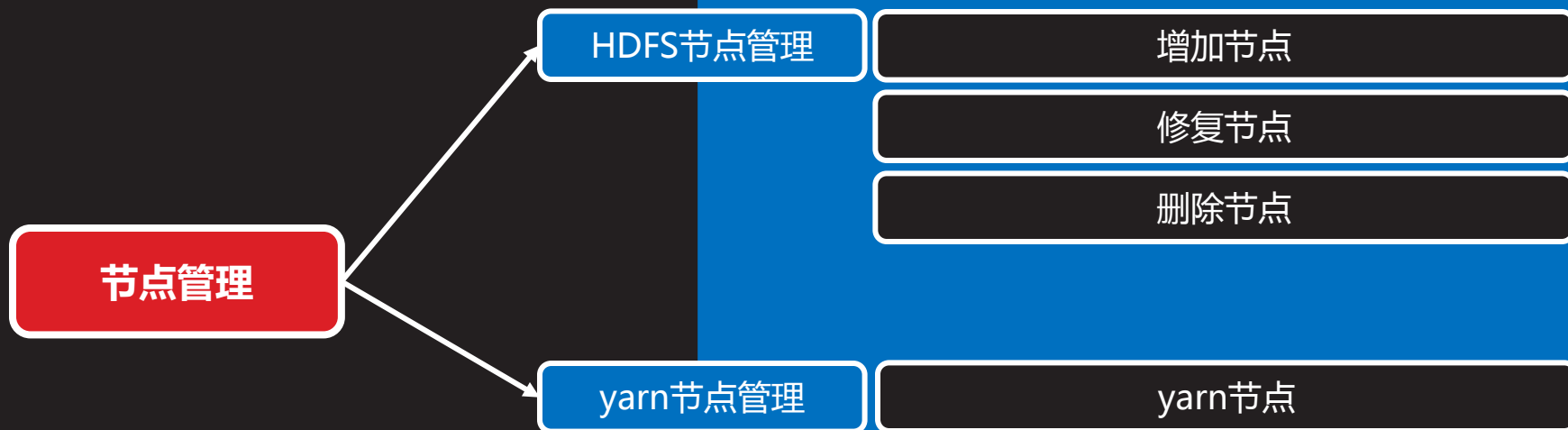
- 查看结果

```
# ./bin/hadoop fs -cat output/*
```



# 节点管理

---



# HDFS节点管理



# 增加节点

- HDFS增加结点
  - 启动一个新的系统，禁用Selinux、禁用firewalld
  - 设置SSH免密码登录
  - 在所有节点修改/etc/hosts，增加新节点的主机信息
  - 安装java运行环境（java-1.8.0-openjdk-devel）
  - 拷贝NameNode的/usr/local/hadoop到本机
  - 修改NameNode的slaves文件增加该节点
  - 在该节点启动DataNode
    - ./sbin/hadoop-daemon.sh start datanode





# 增加节点（续1）

- HDFS节点管理
  - 设置同步带宽，并同步数据

```
# ./bin/hdfs dfsadmin -setBalancerBandwidth 67108864  
# ./sbin/start-balancer.sh
```

- 查看集群状态

```
# ./bin/hdfs dfsadmin -report
```



# 修复节点

- HDFS修复节点
  - 修复节点比较简单，与增加节点基本一致
  - 注意：新节点的ip和主机名要与损坏节点的一致
  - 启动服务

```
# ./sbin/hadoop-daemon.sh start datanode
```
  - 数据恢复是自动的
  - 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



# 删除节点

- 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
```



# 删除节点（续1）

- 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 : 仅当状态变成Decommissioned才能down机下线



## 案例3：节点管理

1. 增加一个新的节点
2. 查看状态
3. 删除节点



# yarn节点管理

---

# yarn节点

- yarn的相关操作
  - 由于Hadoop在2.x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
```



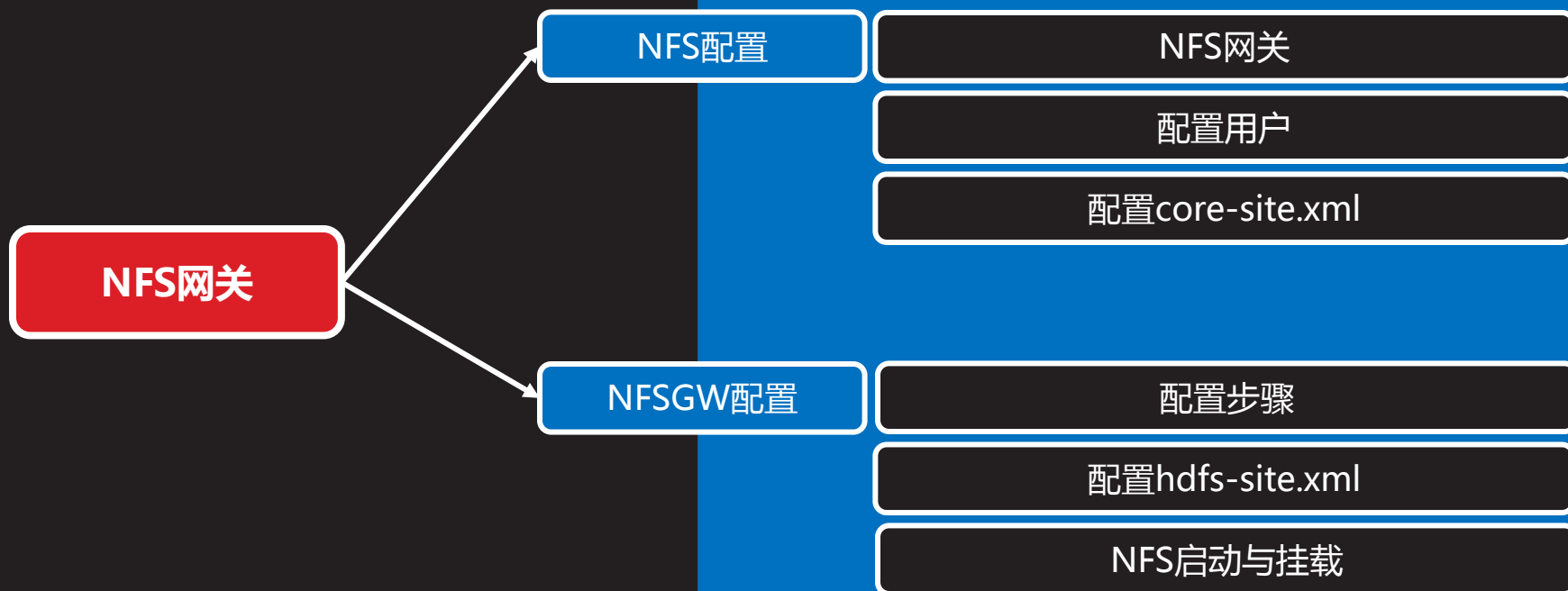
## yarn节点（续1）

- yarn的系统环境配置与HDFS的基础环境配置是相同的，这里不再重复列出
- 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



# NFS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（续1）

- 特性
  - 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认为超级用户失败
- 注意事项
  - 在非安全模式下，运行网关进程的用户是代理用户
  - 在安全模式下，Kerberos keytab中的用户是代理用户



# NFS网关（续2）

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



# 配置用户

- 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
```



# 配置core-site.xml

- 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}是主机上真实运行的nfs3的用户
- 在非安全模式下，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



# 配置core-site.xml ( 续1 )

- 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>  
... ..
```





## 配置core-site.xml ( 续2 )

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



# NFSGW配置



# 配置步骤

- 配置步骤
  - 启动一个新的系统，禁用Selinux、禁用firewalld
  - 卸载rpcbind、nfs-utils
  - 配置/etc/hosts，添加所有NameNode和DataNode的主机名与ip对应关系
  - 安装JAVA运行环境（java-1.8.0-openjdk-devel）
  - 同步NameNode的/usr/local/hadoop到本机
  - 配置hdfs-site.xml
  - 启动服务



# 配置hdfs-site.xml

- 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。值和字符串对应机器名和访问策略，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
  - 使用rw或ro可以指定导出目录的读写或只读权限。  
如果访问策略没被提供，默认为只读。每个条目使用";"来分割



# 配置hdfs-site.xml ( 续1 )

- hdfs-site.xml配置
  - nfs.exports.allowed.hosts
  - 配置 \* rw

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



## 配置hdfs-site.xml ( 续2 )

- 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，1M)时被转储。需要确保有足够的空间的目录
  - 如：应用上传10个100M，那么这个转储目录推荐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



## 配置hdfs-site.xml ( 续3 )

- 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启动与挂载

- 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启动与挂载（续1）

- 警告
  - 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启动与挂载（续2）

- 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启动与挂载（续3）

- 启动与挂载
  - 强烈建议使用安装选项sync，它可以最小化避免重排序写入造成不可预测的吞吐量，未指定同步选项可能会导致上传大文件时出现不可靠行为
  - 启动一台机器并安装nfs-utils
 

```
# yum install nfs-utils
```
  - 挂载nfs
 

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



## 案例4：NFS配置

1. 创建代理用户
2. 启动一个新系统，禁用Selinux和firewalld
3. 配置NFSWG
4. 启动服务
5. 挂载NFS并实现开机自启



# 总结和答疑

---

总结和答疑

NFS挂载

问题现象

故障分析及排除

# NFS挂载



# 问题现象

- NFS挂载失败



# 故障分析及排除

- 原因分析
  - nfs3不是代理用户启动或启动的顺序出错
- 解决方案
  - 使用代理用户启动nfs3
  - 先用root用户启动portmap再用代理用户启动nfs3

