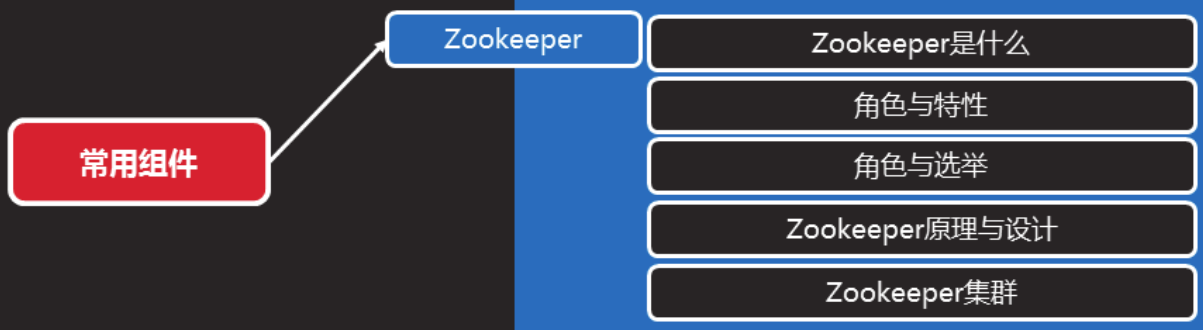
ARCHITECTURE DAY07

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	常用组件
	10:30 ~ 11:20	Kafka集群
	11:30 ~ 12:00	
下午	14:00 ~ 14:50	Hadoop高可用
	15:00 ~ 15:50	
	16:10 ~ 17:10	
	17:20 ~ 18:00	总结和答疑

常用组件



Zookeeper

Zookeeper是什么

知识讲解

- Zookeeper是什么
  - Zookeeper是一个开源的分布式应用程序协调服务
- Zookeeper能做什么
  - Zookeeper是用来保证数据在集群间的事务一致性

Zookeeper是什么（续1）

知识讲解

- Zookeeper应用场景
  - 集群分布式锁
  - 集群统一命名服务
  - 分布式协调服务

角色与选举

知识讲解

- Zookeeper角色与选举
  - 服务在启动的时候是没有角色的（LOOKING）
  - 角色是通过选举产生的
  - 选举产生一个Leader，剩下的是Follower
- 选举Leader原则
  - 集群中超过半数机器投票选择Leader
  - 假如集群中拥有n台服务器，那么Leader必须得到n/2+1台服务器的投票

角色与选举（续1）

Tedu.cn 达内教育

知识讲解

- Zookeeper角色与选举
  - 如果Leader死亡，重新选举Leader
  - 如果死亡的机器数量达到一半，则集群挂掉
  - 如果无法得到足够的投票数量，就重新发起投票，如果参与投票的机器不足n/2+1，则集群停止工作
  - 如果Follower死亡过多，剩余机器不足n/2+1，则集群也会停止工作
  - Observer不计算在投票总设备数量里面

Zookeeper原理与设计

Tedu.cn 达内教育

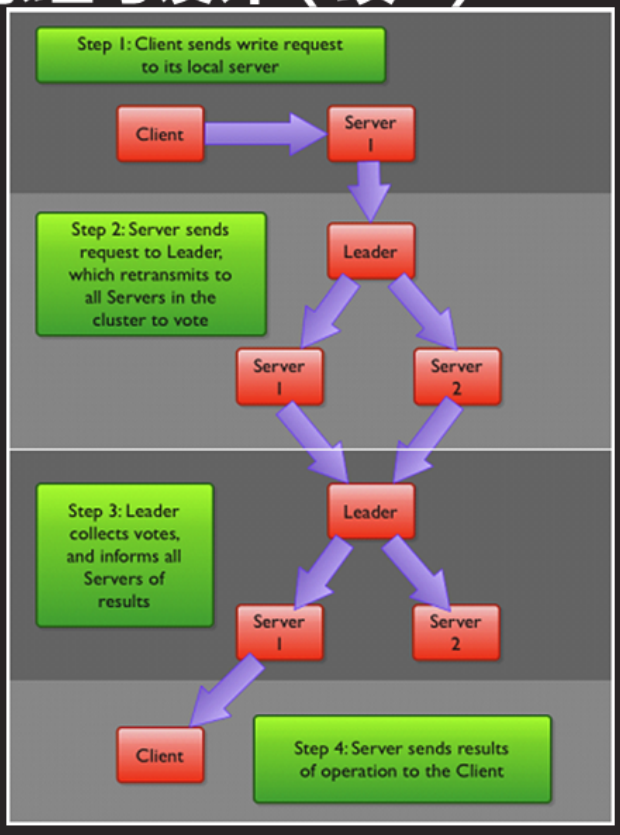
知识讲解

- Zookeeper可伸缩扩展性原理与设计
  - Leader所有写相关操作
  - Follower读操作与响应Leader提议
  - 在Observer出现以前，Zookeeper的伸缩性由Follower来实现，我们可以通过添加Follower节点的数量来保证Zookeeper服务的读性能，但是随着Follower节点数量的增加，Zookeeper服务的写性能受到了影响

Zookeeper原理与设计（续2）

Tedu.cn 达内教育

知识讲解



Zookeeper原理与设计（续4）

Tedu.cn 达内教育

知识讲解

- 续上页
  - 所以，我们不得不在增加Client数量的期望和我们希望保持较好吞吐性能的期望间进行权衡。要打破这一耦合关系，我们引入了不参与投票的服务器Observer。Observer可以接受客户端的连接，并将写请求转发给Leader节点。但Leader节点不会要求Observer参加投票，仅仅在上述第3步那样，和其他服务节点一起得到投票结果

Zookeeper集群（续1）

Tedu.cn 达内教育

知识讲解

- zoo.cfg集群的安装配置
  - 创建datadir指定的目录
    ```
    # mkdir  /tmp/zookeeper
    ```
  - 在目录下创建id对应主机名的myid文件
- 关于myid文件
  - myid文件中只有一个数字
  - 注意：请确保每个server的myid文件中id数字不同
  - server.id中的id与myid中的id必须一致
  - id的范围是1~255

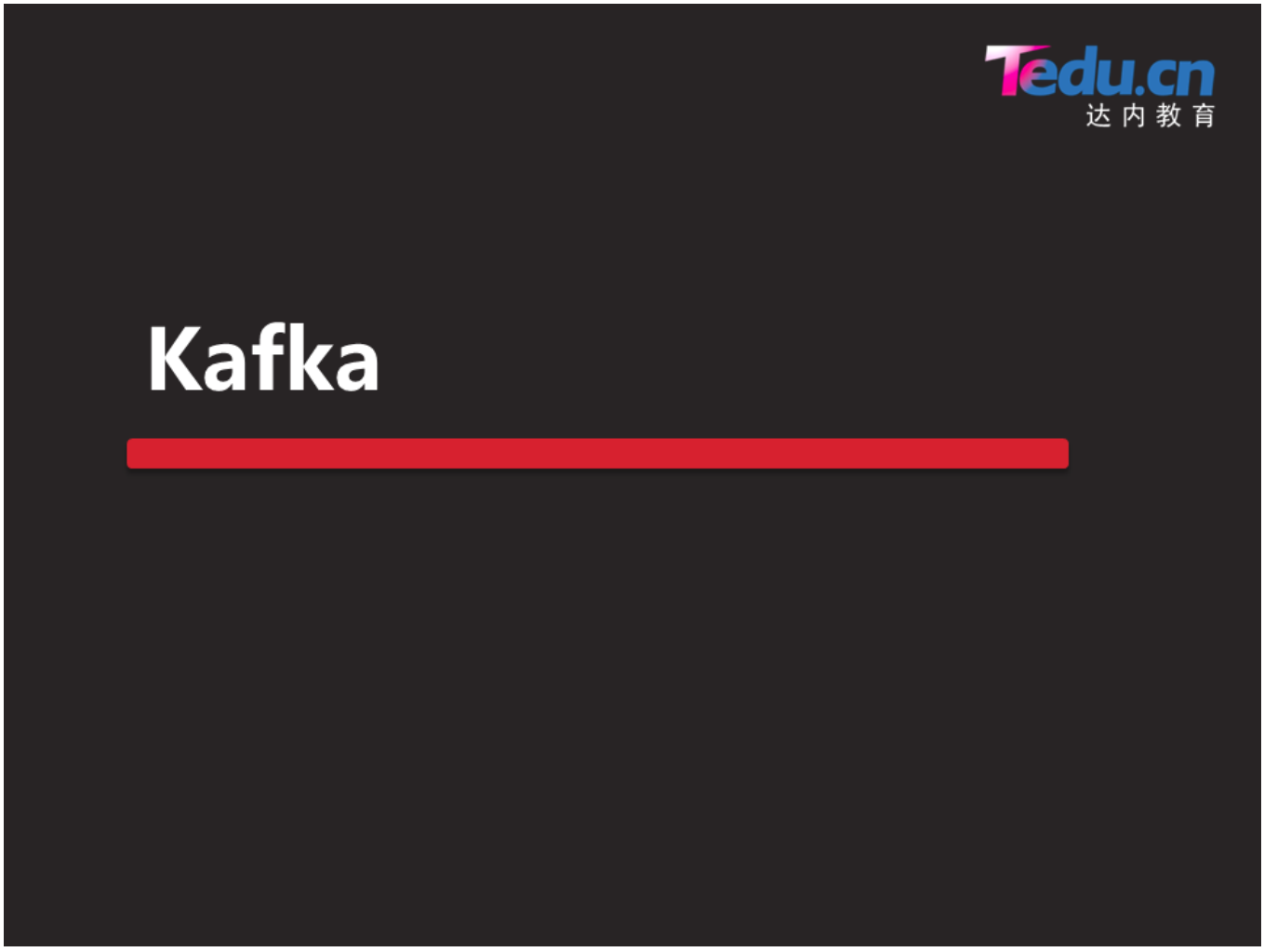
Zookeeper集群（续2）

Tedu.cn 达内教育

知识讲解

- Zookeeper集群的安装配置
  - 启动集群，查看验证（在所有集群节点执行）
    ```
    # /usr/local/zk/bin/zkServer.sh start
    ```
  - 查看角色
    ```
    # /usr/local/zk/bin/zkServer.sh status
    or
    { echo 'stat';yes; }|telnet 192.168.4.10 2181
    ```
  - Zookeeper管理文档
    http://zookeeper.apache.org/doc/r3.4.10/zookeeperAdmin.html


Kafka集群
Kafka集群
Kafka
什么是Kafka
Kafka角色
Kafka集群安装与配置


Tedu.cn
达内教育
Kafka

Kafka角色

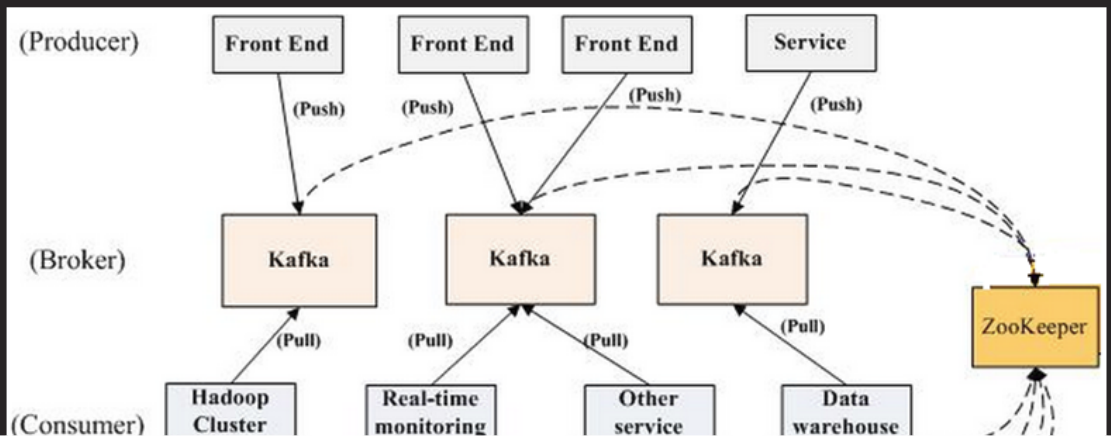
知识讲解

- Kafka角色与集群结构
  - producer：生产者，负责发布消息
  - consumer：消费者，负责读取处理消息
  - topic：消息的类别
  - Parition：每个Topic包含一个或多个Partition
  - Broker：Kafka集群包含一个或多个服务器
- Kafka通过Zookeeper管理集群配置，选举Leader

Kafka集群安装与配置

知识讲解

- Kafka角色与集群结构



Kafka集群安装与配置（续3）

Tedu.cn 达内教育

知识讲解

- Kafka集群的安装配置
  - 在所有主机启动服务

```
# ./bin/kafka-server-start.sh -daemon config/server.properties
```

- 验证
  - jps命令应该能看到Kafka模块
  - netstat应该能看到9092在监听

Kafka集群安装与配置（续4）

Tedu.cn 达内教育

知识讲解

- 集群验证与消息发布
  - 创建一个 topic

```
# ./bin/kafka-topics.sh --create --partitions 2 --replication-factor 2 \
--zookeeper node3:2181 --topic mymsg
```

  - 生产者

```
# ./bin/kafka-console-producer.sh \
--broker-list  master:9092,node1:9092 --topic mymsg
```

  - 消费者

```
# ./bin/kafka-console-consumer.sh \
--bootstrap-server node2:9092,node3:9092 --topic mymsg
```

Hadoop高可用

Hadoop高可用

为什么需要NameNode

知识讲解

- 原因
  - NameNode是HDFS的核心配置，HDFS又是Hadoop核心组件，NameNode在Hadoop集群中至关重要
  - NameNode宕机，将导致集群不可用，如果NameNode数据丢失将导致整个集群的数据丢失，而NameNode的数据的更新又比较频繁，实现NameNode高可用势在必行

解决方案

知识讲解

- 官方提供了两种解决方案
  - HDFS with NFS
  - HDFS with QJM
- 两种方案异同

NFS	QJM
NN	NN
ZK	ZK
ZKFailoverController	ZKFailoverController
NFS	JournalNode

解决方案（续1）

知识讲解

- HA方案对比
  - 都能实现热备
  - 都是一个Active NN和一个Standby NN
  - 都使用Zookeeper和ZKFC来实现自动失效恢复
  - 失效切换都使用Fencin配置的方法来Active NN
  - NFS数据共享变更方案把数据存储在共享存储里，我们还需要考虑NFS的高可用设计
  - QJM不需要共享存储，但需要让每一个DN都知道两个NN的位置，并把块信息和心跳包发送给Active和Standby这两个NN

使用方案

Tedu.cn 达内教育

知识讲解

- 使用原因（QJM）
  - 解决NameNode单点故障问题
  - Hadoop给出了HDFS的高可用HA方案：HDFS通常由两个NameNode组成，一个处于Active状态，另一个处于Standby状态。Active NameNode对外提供服务，比如处理来自客户端的RPC请求，而Standby NameNode则不对外提供服务，仅同步Active NameNode的状态，以便能够在它失败时进行切换

NameNode高可用

Tedu.cn 达内教育

知识讲解

- NameNode高可用架构
  - 为了让Standby Node与Active Node保持同步，这两个Node都与一组称为JNS的互相独立的进程保持通信（Journal Nodes）。当Active Node更新了namespace，它将记录修改日志发送给JNS的多数派。Standby Node将会从JNS中读取这些edits，并持续关注它们对日志的变更
  - Standby Node将日志变更应用在自己的namespace中，当Failover发生时，Standby将会在提升自己为Active之前，确保能够从JNS中读取所有的edits，即在Failover发生之前Standy持有的namespace与Active保持完全同步

NameNode高可用（续1）

Tedu.cn 达内教育

知识讲解

- NameNode高可用架构 续.....
  - NameNode更新很频繁，为了保持主备数据的一致性，为了支持快速Failover，Standby Node持有集群中blocks的最新位置是非常必要的。为了达到这一目的，DataNodes上需要同时配置这两个Namenode的地址，同时和它们都建立心跳连接，并把block位置发送给它们

NameNode高可用（续2）

Tedu.cn 达内教育

知识讲解

- NameNode高可用架构 续.....
  - 任何时刻，只能有一个Active NameNode，否则会导致集群操作混乱，两个NameNode将会有两种不同的数据状态，可能会导致数据丢失或状态异常，这种情况通常称为"split-brain"（脑裂，三节点通讯阻断，即集群中不同的DataNode看到了不同的Active NameNodes）
  - 对于JNS而言，任何时候只允许一个NameNode作为writer；在Failover期间，原来的Standby Node将会接管Active的所有职能，并负责向JNS写入日志记录，这种机制阻止了其他NameNode处于Active状态的问题

NameNode架构图（续1）

Tedu.cn 达内教育

- 系统规划

知识讲解

主机	角色	软件
192.168.1.21	NameNode1	Hadoop
192.168.1.25	NameNode2	Hadoop
192.168.1.22 Node1	DataNode journalNode Zookeeper	HDFS Zookeeper
192.168.1.23 Node2	DataNode journalNode Zookeeper	HDFS Zookeeper
192.168.1.24 Node3	DataNode journalNode Zookeeper	HDFS Zookeeper

core-site配置

Tedu.cn 达内教育

知识讲解

- core-site.xml文件

```
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://mycluster</value>
</property>
<property>
  <name>hadoop.tmp.dir</name>
  <value>/var/hadoop</value>
</property>
<property>
  <name>ha.zookeeper.quorum</name>
  <value>node1:2181,node2:2181,node3:2181</value>
</property>
```