**大数据学院**

**2018年7月3日**



**[第一单元 大数据概论和分布式介绍]**

Zookeeper标准化课件设计

**《zookeeper》**

**第七单元 大数据概论与分布式介绍**

1. **本单元教学目标**
2. 知识目标：

**·** 了解zookeeper的作用；

**·** 理解掌握zookeeper架构；

**·**  掌握zookeeper的工作流程;

**·**  了解zookeeper的三种部署方式；

**·** 掌握总结的zookeeper问题。

1. 能力目标：

**·** 能够自行配置运行zookeeper单机模式；

* 能够自行配置运行zookeeper伪分布模式；

**·** 能够熟练应用命令操作zookeeper；

**·** 初步掌握java程序连接zookeeper，操作zookeeper；

1. **课时分配**

****

1. **第一课时：大数据概论**

本课时通过对大数据和分布式的宏观介绍，使学生初步建立大数据知识体系，使得学生能为学习以后的课程打下基础。

1. 知识目标：

**·** 掌握zookeeper的架构；

**·** 掌握理解zookeeper的节点；

**·** 掌握理解zookeeper会话；

**·** 掌握理解zookeeper监听。

**·** 掌握理解zookeeper的特点；

**·** 掌握理解zookeeper的运行流程；

1. 能力目标：

**·** 能够画出zookeeper的架构；

**·** 能够描述zookeeper的工作场景；

1. **第二课时： 详解分布式和Hadoop**
2. 知识目标：

**·** 了解zookeeper的3种部署方式；

**· 了解zookeeper安装部署的前提**；

* 掌握zookeeper的配置要领；
* 掌握单机与伪分布zookeeper的启动停止的方式；

1. 能力目标：

**· 能够搭建运行单机模式**；

**·** 能够搭建运行伪分布模式

**·** 能够熟练配置运行伪分布模式

1. **第三课时： 命令与程序例子**
2. 知识目标：

**·** 掌握zookeeper查看节点；

**·** 掌握zookeeper创建节点；

**·** 掌握zookeeper删除节点；

**·** 掌握zookeeper获取节点内容；

**·** 掌握 java程序连接zookeeper；

**·** 掌握3种节点（持久，临时，顺序）的创建方式；

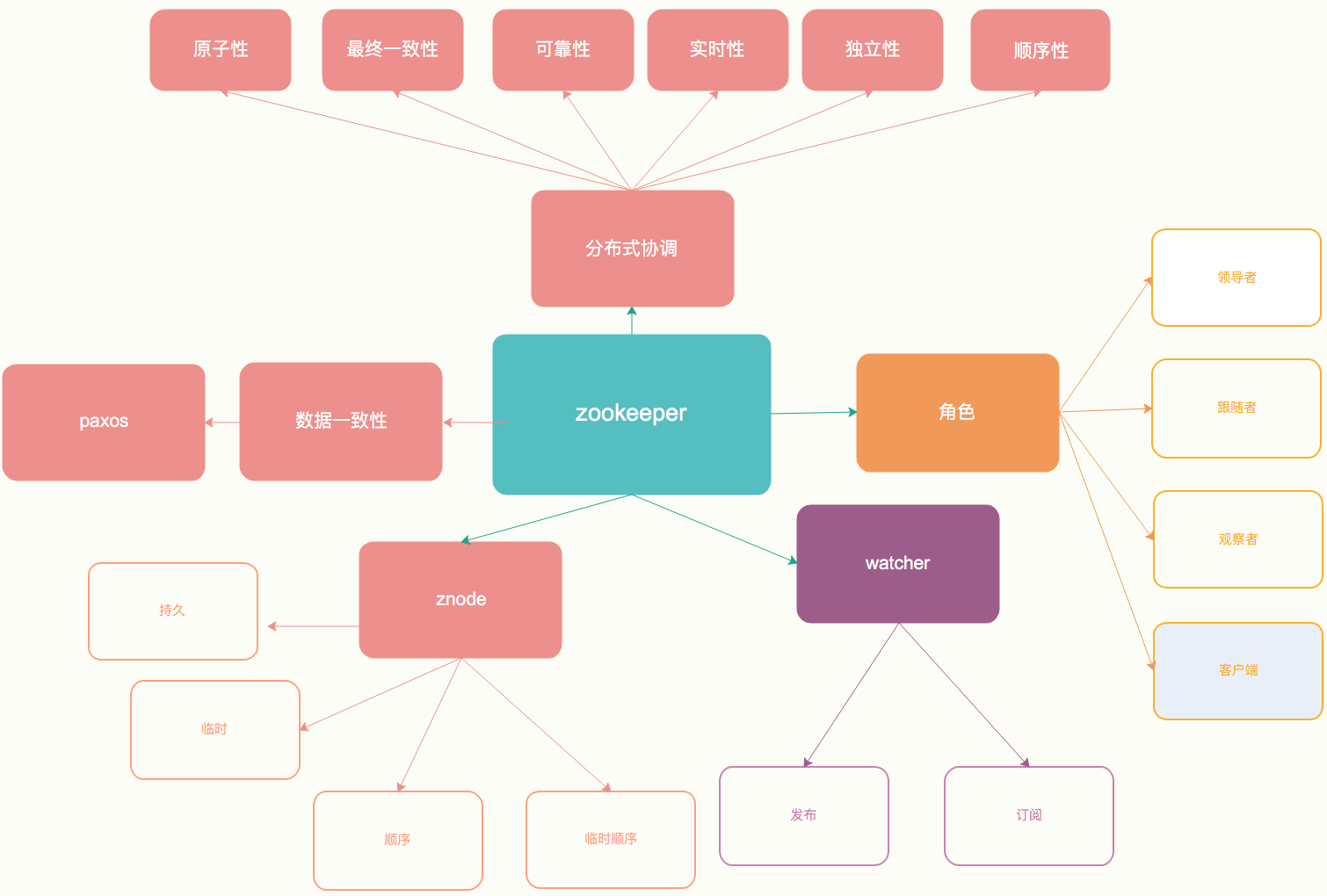
**·** 掌握 java程序连接简单操作zookeeper；

1. 能力目标：

**·** 能够用zookeeper命令熟练操作zookeeper存储的节点；

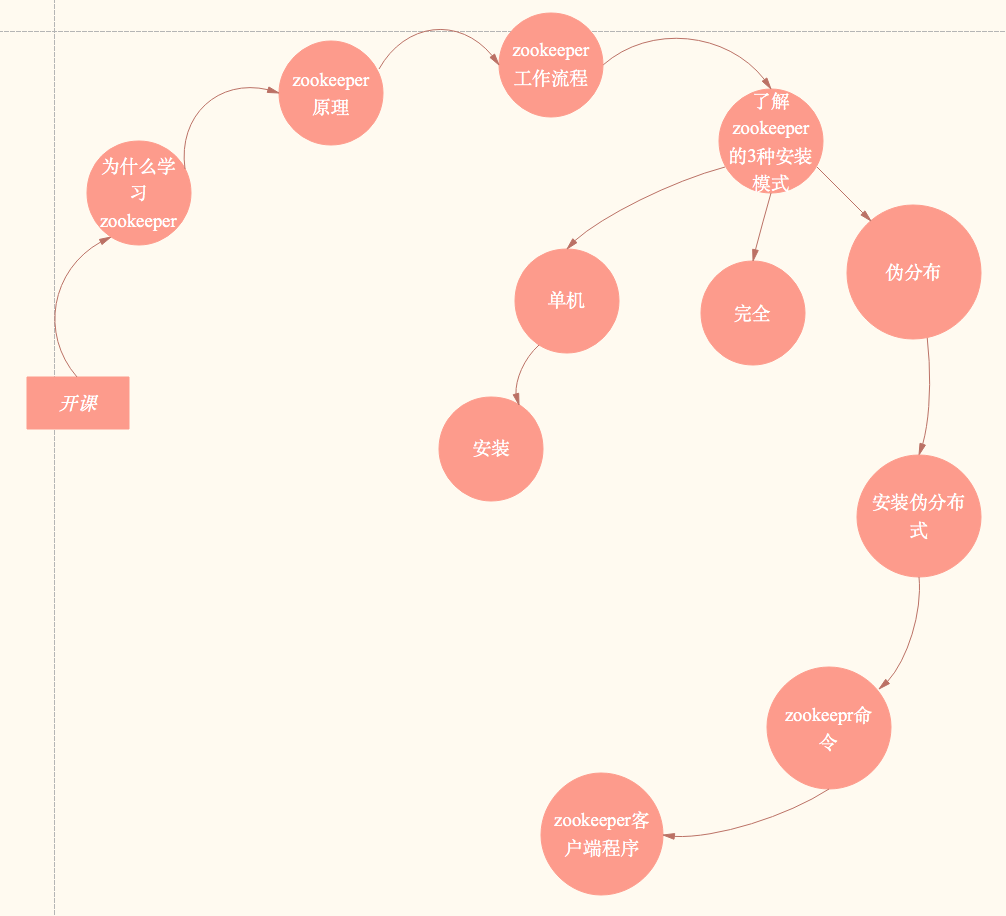
**· 能用java连接操作zookeeper的节点**

**三、本单元概念图**



**四、教学过程脚本设计—第25课时**

1. **脚本设计思路**



1. **脚本切片设计思路**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **内容** | **小节目标** | **设计亮点** | **时间** |
| **1** | **课堂礼仪** | 集中学生的注意力，喊出士气，以饱满的精神状态投入本节课的学习 | 整齐、声音洪亮、铿锵有力，提升精气神！ | 30” |
| **2** | **为什么需要zookeeper?** | 了解分布式要实现哪些，怎么实现 | 分布式的实现与怎么实现 |  |
| **3** | **Zookeeper是什么?** | 了解什么是zookeeper | 分布式协调服务，针对大型分布式系统的可靠协调系统 |  |
| **4** | **Zookeeper能做什么** | 了解zookeeper能做什么？ | 命名服务，配置管理，集群管理等等，有很多 |  |
| **5** | **Zookeeper架构** | 了解zookeeper的架构 | 是一种客户端-服务端架构 | 3' |
| **6** | **Zookeeper的文件系统** | 了解zookeeper的文件系统的树形结构 | Zookeeper的节点成为znode, zookeeper的文件系统是树形结构 |  |
| **7** | **znode的类型** | 掌握3种类型 | 持久,顺序,临时 |  |
| **8** | **会话（session）** | 了解会话的机制 | 客户端连接到服务器后，将建立会话，并为客户端分配会话id |  |
| **9** | **监视(watches)** | 了解监视的机制 | 监视程序将通知发送到已经注册客户端的任何znode更改 | 2'30" |
| **10** | **Zookeeper工作流程** | 了解工作流程 | 读数据与存储数据的 | 4'30" |
| **11** | **Zookeeper 节点数量要求** | 了解zookeeper节点要求 | 小于等于255与奇数 |  |
| **12** | **安装zookeeper的三种种安装方式** | 了解安装方式 | 单机，伪分布，完全分布式 |  |
| **13** | **单机安装方式的步骤** | 步骤概览 | 整体了解安装步骤 |  |
| **14** | **确认安装配置linux环境** | Linux的选择 | Linux的选择，讲课用centos6 |  |
| **15** | **单机安装zookeeper前奏** | 确认安装部署的操作系统与java环境 | 确认操作系统：  Linux:支持开发与部署  Windows:仅用作开发  Mac os:仅用作开发  确认java:  Java -version |  |
| **16** | **获取安装压缩包** | zookeeper-3.4.10.jar | 获取与解压zookeeper压缩包 | 5' |
| **17** | **修改配置文件** | 修改配置文件名  修改配置文件  dataDir=/tmp/zookeeper  clientPort=2181 | 修改配置文件：数据存储目录与客户端连接的端口 | 5' |
| **18** | **启动** | 启动步骤  查看是否启动 | zkServer.sh start  zkServer.sh status  jps | 2'30" |
| **19** | **进入命令演示** | zkCli.sh  演示简单命令 | ls /  create /username xiaozhang |  |
| **20** | **退出命令行与停止zookeeper** | ctrl+c | Ctrl+c  zkServer.sh stop |  |
| **21** | **重温Zookeeper伪分布式** | 重新温习什么是伪分布式 | 多个zookeeper节点在一个服务器里启动 |  |
| **22** | **伪分布式的步骤** | 了解zookeeper的安装步骤 | 了解zookeeper的安装步骤 |  |
| **23** | **伪分布式要求** | 伪分布式的要求 | 数据存储目录与端口的要求 |  |
| **24** | **伪分布式配置文件配置** | 配置 | 数据存储目录，端口，zookeeper之间的通信配置 |  |
| **25** | **伪分布式启动与退出** | 启动zookeeper | 启动命令，查看进程 |  |
| **26** | **Zookeeper命令操作** | Zookeeper的命令的作用与操作 | 查看，创建节点等命令的作用 |  |
| **27** | **编写Zookeeper的客户端程序例子** | 客户端程序来操作zookeeper | 连接zookeeper,操作zookeeper |  |
| **28** | **知识点总结** | Zookeeper所学知识点总结 | Zookeeper所学知识点总结 |  |

1. **脚本切片一：**[**课堂礼仪**](file:///D:\llq\八维\工作\教学标准化计划\切片(总共14片)\切片1.mp4)

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **课堂礼仪** | 使学生注意力集中并进入上课状态 |  |  |
| **讲师：**上课！①  **班长：**起立！  **讲师：**同学们好！  **学生：**老师好！  **班长：**我们的信念是：  **全体：**知识^很简单，学习^很快乐，潜力^无极限！  **班长：**我们的口号是：  **全体：**辉煌八维大数据，引领时代高科技！人工智能显神奇，高薪就业创佳绩！  **讲师：**请坐！② | 1. 进入教室前整理着装，提前30秒面带自信的笑容的走进教室。，给学生打招呼。当上课铃声响起，喊出“上课”，声音高于授课声音。 2. 细节要求：学生发型整齐、脸部整洁、面带微笑或者礼仪笑（八颗牙齿），严禁大笑，举止合理，避免指手画脚。   IMG_256IMG_256   1. 学生礼仪内容：学习信念，学院口号，班级口号。   讲师授课过程中要面带微笑。   1. 插入合理手势，避免课堂枯燥。   ①动作表情：上课起立手势  ②动作表情：全体请坐手势 | **0:30** |

1. **脚本切片二**：**为什么学zookeeper**

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **为什么Zookeeper** | **了解为什么学习zookeeper** |  |  |
| 老师：问学生  分布式需要具备什么特点？  老师：  可靠性：单个或几个系统（或服务器主机）失败不会使整个系统失败。  可伸缩性：分布式的性能可以随时增加  配置：动态更新集群环境里的配置  老师：如何做到可靠性，可伸缩性，透明度那？  Zookeeper就可以帮助分布式系统做到。 | **举例**  **Hbase利用zookeeper协调hbase的**hmaser  **节点的高可用**  **hadoop利用zookeeper实现namenode节点高可用** | 3' |

1. **脚本切片三：**zookeeper是什么

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **Zookeeper是什么** | **Zukeeper是什么** |  |  |
| 老师：  官方说法：  Zookeeper是一个高性能，分布式的，开源分布式应用协调服务。它提供了简单原始的功能，分布式应用可以基于它实现更高级的服务，比如同步，配置管理，集群管理，名空间。它被设计为易于编程，使用文件系统目录树作为数据模型 |  | **0:30** |

1. **脚本切片四：zookeeper能做什么**

|  |  |  |  |
| --- | --- | --- | --- |
| **模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **Zookeeper能做什么** | 可以用来做什么 |  |  |
| 讲师：  命名服务：按名称标识集群中的节点。类似与DNS,但是通常只用于服务器节点。  配置管理：连接服务器节点的最新配置信息  集群管理：实时加入/离开集群中的节点和节点状态  领导选举：选择节点作为协调目的的领导者  锁定：在修改数据时锁定它，防止冲突  高度可靠的数据注册表：一个或几个点点的关闭不影响集群的可用性 |  | 2' |

1. **脚本切片五：zookeeper架构**

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **Zookeeper架构** | **了解zooleeper的架构** |  |  |
| 讲师：zookeeper是一种“客户端-服务端”架构  Client(客户端)：客户端是分布式应用程序中的一个节点，它从服务器访问信息，对于特定的时间间隔，每个客户端都会向服务器发送一条消息，以让服务器知道自己是活的，同样服务器在客户端连接是发送确认，如果连接的服务器没有响应，则客户端会自动将信息重定向到另一台服务器  Server(服务器端)：在多个zookeeper中的一个节点，提供所有的服务给客户。向客户端发出确认信息，通知服务器是否处于活动状态。  Leader（领导者）:多个zookeeper节点只能有一个leader节点，领导者在zookeeper启动时确认。  Flower(追随者):除了一个leader节点,剩下的都是flower节点 | 4.png | 3' |

1. **脚本切片六：**zookeeper的文件系统

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **zookeeper文件系统** | **Zookeeper的文件系统的认识** |  |  |
| **老师：**  **Zookeeper本身也是可以存数据的，比如通过存储的数据来协调集群，这个时候存储的就是服务器的信息。**  **老师：问学生**  **Zookeeper中存储的数据有没有规则那?还是随机的将数据存到zookeeper?**  **老师:zookeeper中的文件系统（定义了组织文件的方式）是一种层级式的树形结构。Zookeeper中的节点被称为znode。每个znode由一个名称标识并用斜线“/”来分割。** |  | 4'30" |

1. **脚本切片七：**znode的类型

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **znode的类型** | **了解持久，顺序，临时3种节点** |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **老师：持久，顺序，临时**  **持久：即使在创建了特定znode的客户端断开连接后，持久性znode仍处于活动状态，默认情况下，除非另行指定，否则所有的znode都是永久性的。**  **临时性的znode:客户端在，临时znode就在，当客户端断开链接，临时的znode会被删除**  **顺序znode :顺序znodes可以是持久的或临时的，节点后会跟随一个10位数字的顺序号。** |  | 4'30" |

1. **脚本切片八：**会话（session）

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **会话** | **了解会话(session)** |  |  |
| 老师：  客户端连接到服务器后，将建立会话，并为客户端分配会话id.服务器端在指定的时间内没收到客户端的心跳信息（会话超时），那么服务端将会判断客户端已经掉线。 |  | 4'30" |

1. **脚本切片九：**监视(watches)

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
|  | 了解监视 |  |  |
| **监视** | 老师：  1：监视程序是一个简单的机制，客户端可以获取有关zookeeper更改的通知。  2：通知什么时候发送：  Znode发生改变时  监视仅会被触发  3：触发次数  只会触发一次。如果客户端需要再次通知，则需要通过另一个读操作来完成。 |  |  |

1. **脚本切片十：**Zookeeper工作流程

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **Zookeeper工作流程** | **安装解压hadoop** |  |  |
| 教师：  1:zookeeper启动  2：zookeeper等待客户端连接  3：客户端连接到zookeeper的任何一个节点  4：连接的节点就会为客户端分配一个回话id,并向客户端确认  5：客户端定期向节点发送心跳  6：读取：客户端向具有znode路径的节点发送“读请求”，并通过所读节点的数据库中获取请求的znode返回该节点  7：存储：客户端连接的服务器会将存储请求转发给领导者，然后领导者向所有的追随者重新发出写请求。如果大多数节点写入成功，写入就算成功。否则写入失败。 |  | 4'30" |

1. **脚本切片十一：**zookeper节点数量要求

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **Zookeeper节点数量要求** | **了解节点数量的要求** |  |  |
| 教师：  1：节点数量要求必须为基数。  2：生产环境与测试环境一般大于等于3个。  开发环境一般一个就可以了。  3：节点数量最多255,但是节点数量不易过多。因为要存储数据的话，zookeeper领导要同步所存储的数据给跟随节点。影响效率。 |  | 4'30" |

1. **脚本切片十二：**安装zookeeper的三种种安装方式

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
|  | Zookeeper的三种安装方式 |  |  |
| **安装zookeeper的三种安装方式** | 单机：一个zookeeper进程  伪分布：多个zookeeper进程在一个机器里  完全：多个zookeeper进程分布到多台机器（容器）里 |  | 5' |

1. **脚本切片十三：**单机安装方式的步骤

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **单机安装方式的步骤** | **格式化命令使用** |  |  |
| 确认配置安装linux环境  确认是否安装java  获取安装zookeeper压缩包  安装配置zookeeper  启动 |  | 5' |

1. **脚本切片十四：**确认安装配置linux环境

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **确认安装配置linux环境** | 确认安装配置linux环境 |  |  |
| 讲师：安装配置linux环境请参照hadoop安装配置部分。如果已经配置过并无问题，则无需进行此步 |  | 3' |

1. **脚本切片十五：**确认安装配置java环境

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **确认安装配置java环境** | 确认安装配置java |  |  |
| 讲师：因为zookeeper依赖于java,所以要先确认安装java环境  Java -version 查看java版本  如果配置java ,上传jdk压缩包，解压配置java |  | 3' |

1. **脚本切片十六：**获取安装zookeeper压缩包

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **获取安装zookeeper压缩包** | 如何获取安装zookeeper |  |  |
| 讲师：  1：下载  <http://zookeeper.apache.org/>  2：上传到指定linux目录/opt  3：解压安装zookeeper  tar -zxvfzookeeper-3.4.10.jar |  | 3' |

1. **脚本切片十七：**配置zookeeper环境变量

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **配置zookeeper环境变量** | 如何获取安装zookeeper |  |  |
| 讲师：配置环境变量在/etc/profile里配置，将配置增加到profile文件的末尾  vi /etc/profile  export ZOOKEEPER\_HOME=/opt/zookeeper-3.4.10  export PATH=$PATH:$ZOOKEEPER\_HOME/bin  配置完wq保存退出，执行如下命令  source /etc/profile |  | 3' |

1. **脚本切片十八：**配置zookeeper配置文件

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **配置zookeeper配置文件** | 如何配置zookeeper |  |  |
| 讲师：  1:cd 进入zookeeper的解压目录里的conf目录  cd /opt/zookeeper-3.4.10/conf  2:拷贝配置文件（因为zookeeper的配置文件名默认是zoo.cfg）  cpzoo\_sample.cfgzoo.cfg  3:vi编辑配置zoo.cfg  vi zoo.cfg  修改数据保存的目录：  dataDir=/opt/zookeeper （这个目录用户必须在linux里存在，可以自定义任意目录）  指定客户端访问服务端的端口：  clientPort=2181（2181是默认端口，这个端口可以随意指定）  4:wq保存退出 |  | 3' |

1. **脚本切片十九：**启动

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **启动** | 启动zookeeper |  |  |
| 讲师：  如果未配置环境变量path  1：进入zookeeper命令启动目录  cd /opt/zookeeper-3.4.10/bin/  2:启动  ./zkServer.sh start  ./zkServer.sh status  Jps  如果配置了环境变量path,不用进入bin目录  zkServer.sh start  zkServer.sh status  Jps |  | 3' |

1. **脚本切片二十：**进入命令演示

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **进入命令演示** | 进入命令行，演示简单命令 |  |  |
| 讲师：  zkCli.sh  3:退出命令行  Ctrl+c  4:停止zookeeper  zkServer.sh stop | ls /  create /username xiaoming  ls / | 3' |

1. **脚本切片二十一：**重温伪分布式

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **重温伪分布式** | 重温什么是伪分布式 |  |  |
| 讲师：  在一台机器里启动多个zookeeper | 注意：  多个zookeeper的数据目录不能公用  多个zookeeper的端口不能冲突 | 3' |

1. **脚本切片二十二：**安装伪分布式的步骤

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **安装伪分布式的步骤** | 了解伪分布式的步骤 |  |  |
| 老师：步骤如下  确认配置安装linux环境 (同单机部署方式，略)  确认是否安装java(同单机部署方式，略)  获取安装zookeeper压缩包(同单机部署方式，略)  复制zookeeper配置文件并配置  分别启动3个zookeeper |  | 3' |

1. **脚本切片二十三：**复制zookeeper配置文件并配置

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **复制zookeeper配置文件并配置** | 配置伪分布 |  |  |
| 老师：要在伪分布式环境中启动几个zookeeper,就复制几份zookeeper配置文件（必须是基数个节点所以要复制基数份）  3份，5份。咱们复制3份  复制3份配置文件命名如下：  zoo1.cfg 第一个节点  zoo2.cfg 第二个节点  zoo3.cfg 第三个节点  老师：1：而在同一个服务器里启动3个zookeeper ,要求在同一个服务器里每个zookeeper的端口与数据目录不能一致，所以必须修改。2：三个zookeeper要互相知道其它zookeeper的存在，所以要在每个zookeeper里增加所有zookeeper的ip与互相通信的端口  第一个zookeeper的配置  vi打开第一个zoo1.cfg  dataDir与clientPort修改如下：  dataDir=/opt/zookeeper1  clientPort=2181  在配置文件最后增加如下配置  server.1=hadoop:2888:3888  server.2=hadoop:2889:3889  server.3=hadoop:2890:3890  server.1，server.2,server.3代表3台zookeeper进行分布式配置。  1,2,3 这3个数字代表zookeeper的id编号。  Hadoop 表示与3个zookeeper进行通信的ip地址  2888 是leader与flower进行通信的端口号  3888 是选举端口号  接下来先给第一个zookeeper配置编号。配置方法如下：  定位到zoo1.cfg里的dataDir目录。在里面建立myid文件。并在里面输入1.  cd /opt/zookeeper1  vimyid在里面输入1并保存退出。  Myid里的数字不能超过255  第二个zookeeper的配置  vi打开第一个zoo2.cfg  dataDir与clientPort修改如下：  dataDir=/opt/zookeeper2  clientPort=2182  在配置文件最后增加如下配置  server.1=hadoop:2888:3888  server.2=hadoop:2889:3889  server.3=hadoop:2890:3890  在目录/opt/zookeeper2建立myid,内容是2  同样的方法配置第三个zookeeper |  | 3' |

1. **脚本切片二十四：**伪分布式的启动退出

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **伪分布式的启动和退出** | 启动方法 |  |  |
| 进入启动目录  cd /opt/zookeeper-3.4.10/bin  ./zkServer.sh start zoo1.cfg 启动第一个zookeeper  ./zkServer.sh start zoo2.cfg 启动第一个zookeeper  ./zkServer.sh start zoo3.cfg 启动第一个zookeeper  Jps查看zookeeper的3个进程是否在  停止  ./zkServer.sh stop zoo3.cfg停止指定的第3个zookeeper |  | 3' |

1. **脚本切片二十五：**Zookeeper命令操作

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **Zookeeper命令操作** | Zookeeper的基本命令操作 |  |  |
| ls /:列出“/”下面的其它的节点  create -e /name1xiaozhang 创建临时节点，用quit命令退出或结束会话节点消失  create -s /name2xiaowang创建顺序节点  creae -e -s /name3 xiaoli创建临时数据节点  get /name 通过名字获取节点  rmr /name 删除指定名字的节点 |  | 3' |

1. **脚本切片二十六：**编写Zookeeper的客户端程序例子

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **编写zookeeper客户端程序的例子** | 理解基本的操作zookeeper的客户端程序，如何连接zookeeper |  |  |
| 定义zookeeper客户端对象：  ZooKeeperclient **= new**ZooKeeper("172.16.245.5:2181"/\*链接字符串\*/, 1000, **null**) | **package** com;  **import**org.apache.zookeeper.CreateMode;  **import**org.apache.zookeeper.ZooDefs.Ids;  **import**org.apache.zookeeper.ZooKeeper;  **publicclass**MyClient {  **private**ZooKeeperclient = **null**;  **publicstaticvoid** main(String[] args) {  **try** {  // 实例化MyClient对象  MyClientclient = **new**MyClient();  // 调用自定义connectZK方法，连接zookeeper  client.connectZK();  // 调用自定义add方法向zookeeper添加一个节点  client.add();  //向节点下添加子节点  client.addchile();  //获取指定节点的内容  client.getData();  } **catch** (Exception e) {  e.printStackTrace();  }  }  /\*\*  \* 连接ZK  \*/  **publicvoid**connectZK() **throws** Exception {  // 连接字符串，可以指定多个,中间用逗号分开  String connectString = "172.16.245.5:2181";  // session超时时间  **int**sessionTimeout = 1000;  //连接zookeeper  client = **new**ZooKeeper(connectString/\*链接字符串\*/, sessionTimeout, **null**);  }  /\*\*  \* 新增节点name  \*  \* **@throws** Exception  \*/  **publicvoid**add() **throws** Exception {  String create = client.create("/name", "zhangsan".getBytes(), Ids.***OPEN\_ACL\_UNSAFE***, CreateMode.***PERSISTENT***);// PERSISTENT持久化节点  System.***out***.println(create);  }  /\*\*  \* 在name节点下创建sex  \*  \* **@throws** Exception  \*/  **publicvoid**addchile() **throws** Exception {  String create = client.create("/name/sex"/\*指定路径\*/, "man".getBytes()/\*指定路径上存储的值\*/, Ids.***OPEN\_ACL\_UNSAFE***, CreateMode.***PERSISTENT***/\*持久化节点\*/);  System.***out***.println(create);  }  /\*\*  \* 获取数据，同时注册监听，获取数据的时候Stat直接给null，获取所有版本  \*  \* **@throws** Exception  \*/  **publicvoid**getData() **throws** Exception {    **byte**[] data = client.getData("/name/sex"/\*指定路径\*/, **true**, **null**);  String db = **new** String(data);//将获取的数据转换为字符串  System.***out***.println(db);//打印字符串  }  } | 3' |

1. **脚本切片二十七：知识点总结**

|  |  |  |  |
| --- | --- | --- | --- |
| **课堂模块** | **教学内容/台词** | **教学标准/素材** | **时长** |
| **面试问题** | 使通过面试式提问把课堂知识贯彻。提高学生的面试能力。 |  |  |
| 描述一下zookeeper能做什么？  描述一下zookeeper是什么？  zookeeper的节点的名字叫什么？  zookeeper的节点的类型都有什么？  描述顺序节点的特点？  描述持久节点的特点？  描述临时节点的特点？  描述一下zookeeper的会话？  描述一下zookeeper的监听？  描述一下zookeeper的工作流程？  Zookeeper的默认端口?  Zookeeper的集群的节点数量最多是多少?  Zookeeper启动的命令是？  Zookeeper 指定启动文件启动是？  停止指定的zookeeper命令?  查看zookeeper的进程的命令是？  进入zookeeper的命令行的命令是？  命令行查看根下有什么节点的命令是？  创建节点的命令是？  Zookeeper的状态查看命令是？  详细查看zookeeper进程的命令是？  zookeeper单机与伪分布模式有什么区别？ | 获取指定的节点的命令是？  创建持久节点的命令是？  创建临时节点的命令是？  创建顺序节点的命令是？  Zookeeper的节点被称为？  如何创建临时顺序节点？ | **5:30** |