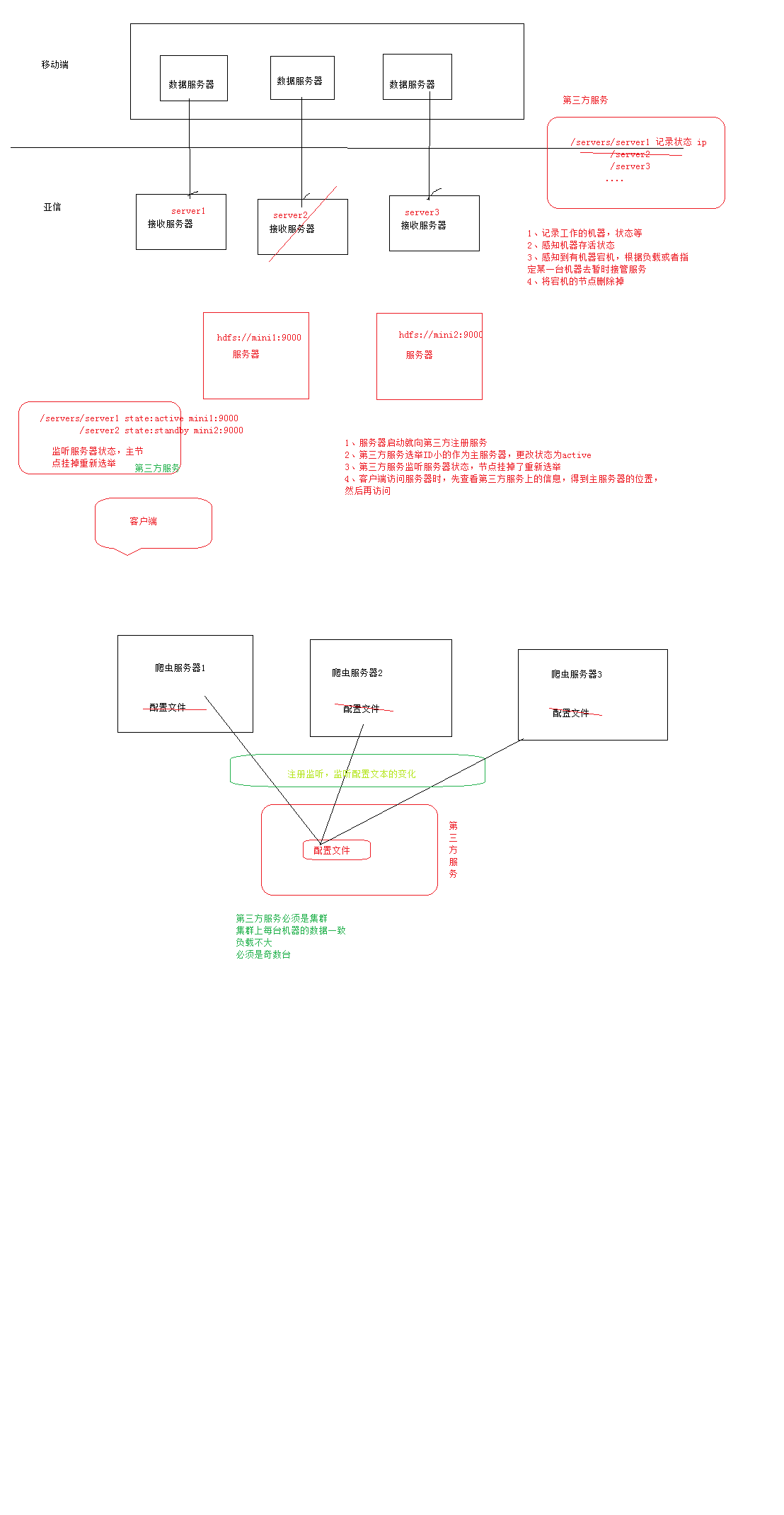
### Zookeeper

1. 概念

是一个分布式协调服务，为用户的分布式程序提供协调服务

* 1. 场景一：节点的动态上下线
  2. 场景二：高可用、主从选举，解决单点故障
  3. 统一配置管理



1. 理解
   1. Zookeeper是为分布式程序服务的
   2. Zk本身就是一个分布式程序（zk只要有半数节点存活就能正常工作，必须是奇数台）
   3. 提供的服务：主从协调、节点的动态上下线、统一的配置管理、统一的名称服务、分布式锁
   4. 真正提供服务的就是：存储数据，提供数据节点监听功能
2. Zookeeper的集群机制

半数机制：集群中半数以上机器存活，集群可用，zk适合存在奇数台上

1. Zk的集群角色分配
   1. Zk的配置文件中没有配置主从
   2. Zk有选举机制，集群启动后会自动选举主节点
   3. 所有数据更新，都是主节点更新数据，然后通知从节点同步
2. Zk的安装
   1. 先安装好JDK
   2. 上传文件至/data
   3. 解包解压缩 tar -zxvf /data/zk.tar.gz -C /apps
   4. mv /apps/zk... /apps/zk-3.4.10
   5. 修改环境变量
   6. 重新source
   7. 修改配置文件
      1. 进入到$ZK\_HOME/conf
      2. 将zoo\_sample.cfg拷贝出一份并改名字为zoo.cfg(conf目录下)
      3. Vi zoo.cfg
3. Zk的结构和命令
   1. 特性：
      1. 一个leader，多个follower
      2. 数据一致性，每个server保存一份数据相同的副本，clien无论连接到哪儿个server数据都是一致的
      3. 分布式读写、更新等需求由leader转发
      4. 更新请求顺序进行
      5. 数据更新原子性，要么成功要么失败
      6. 实时性：一定时间范围内，client能够读到最新数据
   2. 数据结构
      1. 层次化的数据结构
      2. 每个节点在zk中都叫做一个znode,并且有一个唯一的路径标识
      3. 在znode上可以包含数据和子节点
      4. 客户端可以在子节点上设置监视器
4. Zk的命令
   1. zkCli.sh 直接连接本机
   2. Ls 查看结构
   3. Get：获取数据

cZxid = 0x0 节点创建时的zxid

ctime = Thu Jan 01 08:00:00 CST 1970 最后更新时间

mZxid = 0x0 节点创建时的时间戳

mtime = Thu Jan 01 08:00:00 CST 1970 最后更新的时间戳

pZxid = 0x0

cversion = -1 子节点的更新次数

dataVersion = 0 节点数据更新次数

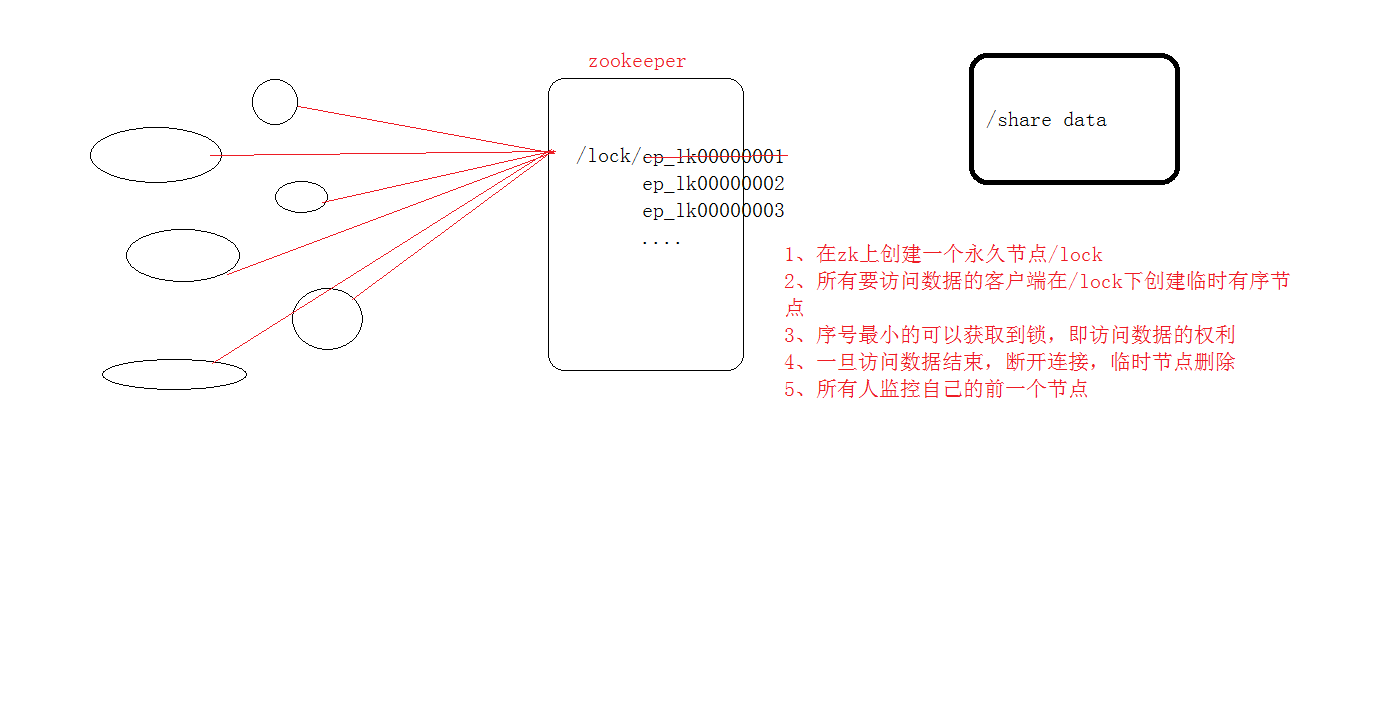
aclVersion = 0

ephemeralOwner = 0x0 如果是永久节点值就是0，如果是临时节点，这个值就不为0

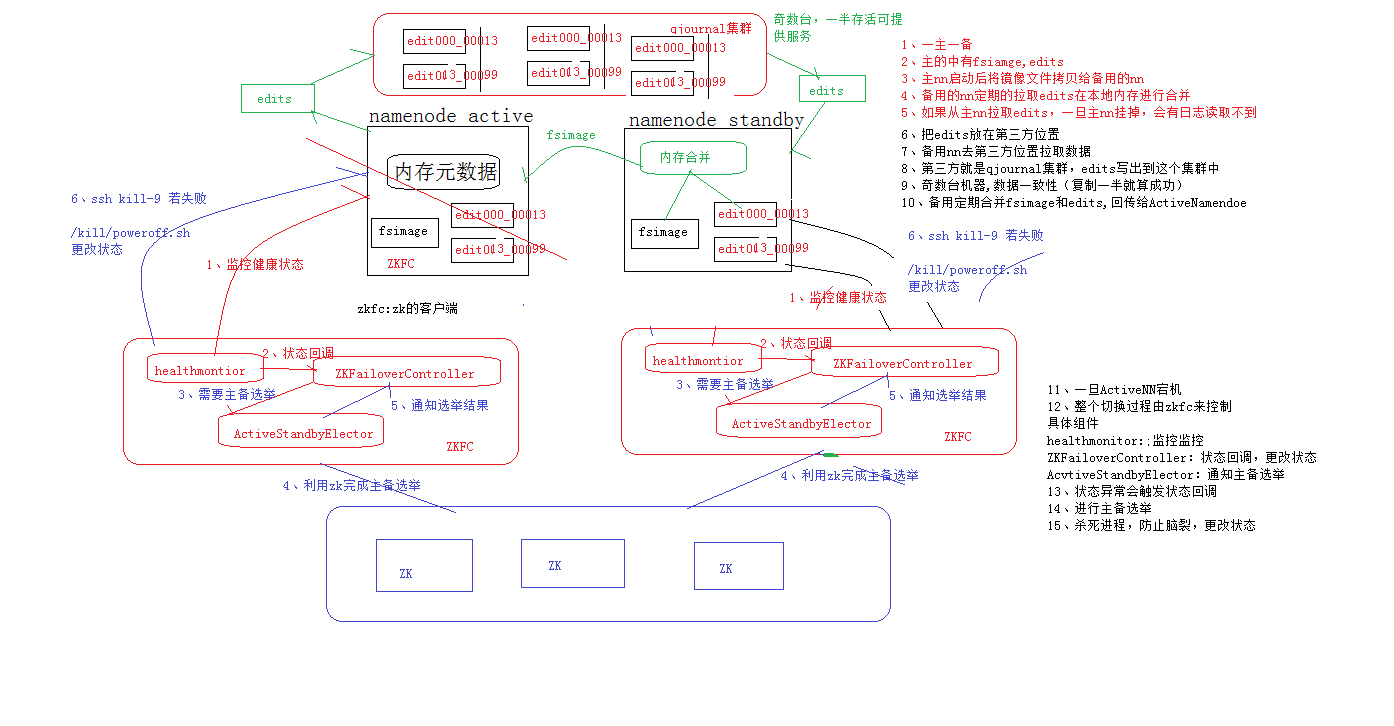
dataLength = 0 数据长度

numChildren = 1 子节点个数

* 1. Help帮助命令
  2. 创建节点 create -s -e
     1. 创建临时节点（可以利用这种临时节点做动态上下线感知）
     2. 临时有序节点（分布式锁）
     3. 永久节点
     4. 永久有序节点
  3. Set 设置节点数据内容
  4. Watch 监听，一旦节点内容改变或者节点下子节点发生变化也能监听的到
     1. 监听内容变化 get /app watch 另一个客户端 set /app/aaa 111
     2. 监听子节点变化 ls /app watch 另一个客户端 create -s -e /app/bbb 222



### HDFS高可用的搭建及原理



1. hadoop的ha机制
   1. 正式引入是2.x版本，之前版本中没有HA机制
   2. 所谓HA就是高可用7\*24小时不中断服务
   3. 实际高可用最关键的是解决单点故障问题
   4. Hadoop的HA严格来说可以分为各个组件的HA，即HDFS的HA，yarn的HA
   5. HDFS的HA详解
      1. 通过双namenode消除单点故障问题
      2. 双namenode协调工作的要点
         1. 元数据管理方式需要改变
            1. 内存中保存一份元数据
            2. Edits日志只有一份，只有active状态的namenode节点可以进行写操作
            3. 两个namenode都可以读取edits
            4. Eidts放在一个共享的位置中（qjournal和NFS）
         2. 需要一个状态管理功能模块
2. 安装规划
   1. Mini1：namenode zkfc datanode zookeeper journalnode
   2. Mini2：namenode zkfc datanode zookeeper journalnode
   3. Mini3： datanode zookeeper journalnode
   4. Mini2要能够ssh免密码登录任何一台机器
3. 安装准备
   1. Hosts、ip、hostname
   2. 防火墙
   3. 时间同步
   4. Jdk
   5. Ssh免密
4. 配置文件的修改
5. 集群启动
   1. 启动zk
   2. 启动journal集群（存日志）
   3. 格式化namenode
   4. 将mini1上的元数据拷贝给mini2
      1. Scp 方式拷贝
      2. 将mini1上的namenode启动起来，然后使用命令将数据同步过来

hdfs namenode -bootstrapStandby

* 1. 格式化zkfc

hdfs zkfc -formatZK

* 1. 启动集群 start-dfs.sh

1. 启动高可用集群之前，必须先启动zk，并且zk能够正常提供服务
2. HDFS高可用shell
   1. NameNode的状态切换管理
      1. hdfs haadmin -getServiceState nn2
      2. hdfs haadmin -transitionToStandby nn1 --forcemanual

### HDFS高可用的Java连接方式

public class HdfsTest {

public static void main(String[] args) throws Exception{

Configuration conf = new Configuration();

//conf.set("fs.defaultFS","hdfs://mini1:9000");

//conf.addResource("core-site.xml");

//conf.addResource("hdfs-site.xml");

//System.setProperty("HADOOP\_USER\_NAME","root");

//FileSystem fs = FileSystem.get(conf);

conf.set("fs.defaultFS","ns1");

conf.set("dfs.nameservices","ns1");

conf.set("dfs.ha.namenodes.ns1","nn1,nn2");

conf.set("dfs.namenode.rpc-address.ns1.nn1","mini1:9000");

conf.set("dfs.namenode.rpc-address.ns1.nn2","mini2:9000");

conf.set("dfs.client.failover.proxy.provider.ns1","org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider");

FileSystem fs = FileSystem.get(new URI("hdfs://ns1"),conf,"root");

fs.mkdirs(new Path("/e"));

}

### 总结

#### 离线分析项目的处理流程

HDFS做存储的

MR做计算的

Yarn做任务管理和资源调度的

Hive：做数据分析的

HBASE:做数据存储的

Flume：数据采集

Sqoop:数据导入导出工具

分布式系统的理解

HDFS:分布式文件系统

高吞吐量、一次写入多次读取、不适合存储小文件

切块（客户端进行切块），默认块大小128M

副本数量默认是3

副本放置策略

安全模式

如何离开安全模式

Namenode的工作职责

元数据

Fsimage文件的查看

Edits文件的查看

Seen\_txid

Checkpoint的过程

Namenode的启动过程

Datanode的工作职责

心跳机制

Datanode下线判断

集群的动态上下线

负载均衡

写数据流程

读数据流程

HDFS的API如何解决权限问题（四种）

HDFS的shell操作

HDFS的api，流式读取数据

HDFS的API 获取块的位置，文件名称，存在于哪儿个主机FileStatus

Zk的应用场景

Zk提供服务的基础：存储数据（每个节点最多存储1M的数据），提供监听

zk监听：监听数据变化，监听节点变化

Zk的节点类型：

Zk的命令

Mapreduce:一个基于yarn的计算系统

Yarn:任务调度和资源管理