在学习zookeeper之前，需要了解一些分布式的基础知识。

一、分布式基础知识

1、分布式环境的特点

分布性：系统部署在不同的主机上，甚至这些主机有可能在不同的城市，同时分布情况也会随时变动

对等性：分布式系统中的计算机没有主从之分，既没有控制整个系统的主机，也没有被控制的从机，组成分布式系统的所有计算机节点都是对等的

并发性：程序运行过程中，并发性操作是很常见的。比如同一个分布式系统中的多个节点，同时访问一个共享资源。数据库、分布式存储

无序性：进程之间的消息通信，会出现顺序不一致问题

2、分布式环境下面临的问题

网络通信：进程之间的消息通信，会出现顺序不一致问题

网络分区（脑裂）：当网络发生异常导致分布式系统中部分节点之间的网络延时不断增大，最终导致组成分布式架构的所有节点，只有部分节点能够正常通信

三态：在分布式架构里面，除了成功、失败、超时

节点故障：

分布式事务：ACID(原子性、一致性、隔离性、持久性)

3、中心化和去中心化

4、经典的CAP/BASE理论

CAP

C（一致性 Consistency）: 所有节点上的数据，时刻保持一致

可用性（Availability）：每个请求都能够收到一个响应，无论响应成功或者失败

分区容错 （Partition-tolerance）：表示系统出现脑裂以后，可能导致某些server与集群中的其他机器失去联系

CP / AP

CAP理论仅适用于原子读写的Nosql场景，不适用于数据库系统

BASE

基于CAP理论，CAP理论并不适用于数据库事务（因为更新一些错误的数据而导致数据出现紊乱，无论什么样的数据库高可用方案都是

徒劳），虽然XA事务可以保证数据库在分布式系统下的ACID特性，但是会带来性能方面的影响；

eBay尝试了一种完全不同的套路，放宽了对事务ACID的要求。提出了BASE理论

Basically available ： 数据库采用分片模式， 把100W的用户数据分布在5个实例上。如果破坏了其中一个实例，仍然可以保证

80%的用户可用

soft-state： 在基于client-server模式的系统中，server端是否有状态，决定了系统是否具备良好的水平扩展、负载均衡、故障恢复等特性。

Server端承诺会维护client端状态数据，这个状态仅仅维持一小段时间, 这段时间以后，server端就会丢弃这个状态，恢复正常状态

Eventually consistent：数据的最终一致性

二、zookeeper初识

1、zookeeper是什么

zookeeper是一个开源的分布式协调服务，是由雅虎创建的，基于google chubby。

分布式数据一致性的解决方案

2、zookeeper的作用

数据的发布/订阅（配置中心:disconf） 、 负载均衡（dubbo利用了zookeeper机制实现负载均衡）、分布式协调/通知、集群管理 、命名服务、

Master选举(kafka、hadoop、hbase)、分布式队列、分布式锁

3、zookeeper的特性

顺序一致性：从同一个客户端发起的事务请求，最终会严格按照顺序被应用到zookeeper中

原子性：所有事务的请求的处理结果在整个集群中的所有机器上都是一致的，也就是说，要么整个集群中所有机器都成功应用了某一事务，要么都不应用。

可靠性：一旦服务器应用了模一个事务数据，并且对客户端做了响应，那么这个数据在整个集群中一定是同步的并且保留下来的

实时性：一旦一个事务被成功应用，客户端就能够立即从服务端读取到事务变更后的最新数据；（zookeeper仅仅保证在一定时间内，近实时）

单一视图：无论客户端连接的是那个ZooKeeper服务，其看到的服务端数据模型都是一致的

4、zookeeper安装

（一）单机配置

1、下载zookeeper下载zookeeper的安装包 http://apache.fayea.com/zookeeper/stable/zookeeper-3.4.10.tar.gz

2、解压

cd /usr/local

tar -zxvf zookeeper-3.4.10.tar.gz

ls

3、修改配置

cd 到 ZK_HOME/conf , copy一份zoo.cfg

cp zoo_sample.cfg zoo.cfg

cd zookeeper-3.4.10/conf

cp zoo_sample.cfg zoo.cfg

4、启动

cd 到 ZK_HOME

sh bin/zkServer.sh start

5、可以通过命令行界面工具来操作，进入方式如下

cd 到 ZK_HOME

sh zkCli.sh -server ip:port

（二）集群配置

提前准备至少三台centos7的机器，或者虚拟机，他们的IP分别如下

192.168.190.101

192.168.190.102

192.168.190.105

1、安装

在每一台的虚拟机系统中都按照单机配置去安装完成。

2、修改配置文件

在192.168.190.101台虚拟机中进行相关配置

进入cd 到 ZK_HOME目录

vim conf/zoo.cfg

添加如下内容：

server.1=192.168.190.101:2888:3881

server.2=192.168.190.102:2888:3881

server.3=192.168.190.105:2888:3881

2181是客户端连接zookeeper服务的端口，这是一个TCP port。 2888是zookeeper的端口 3881是leader选举的端口

server.id=host:port:port

id的取值范围： 1~255； 用id来标识该机器在集群中的机器序号

2881是zookeeper的端口；不能与服务端口相同，否则会出现端口冲突

3881表示leader选举的端口

其他每台都这样操作

3、添加myid文件

cd /tmp/zookeeper下

vim myid 写入1

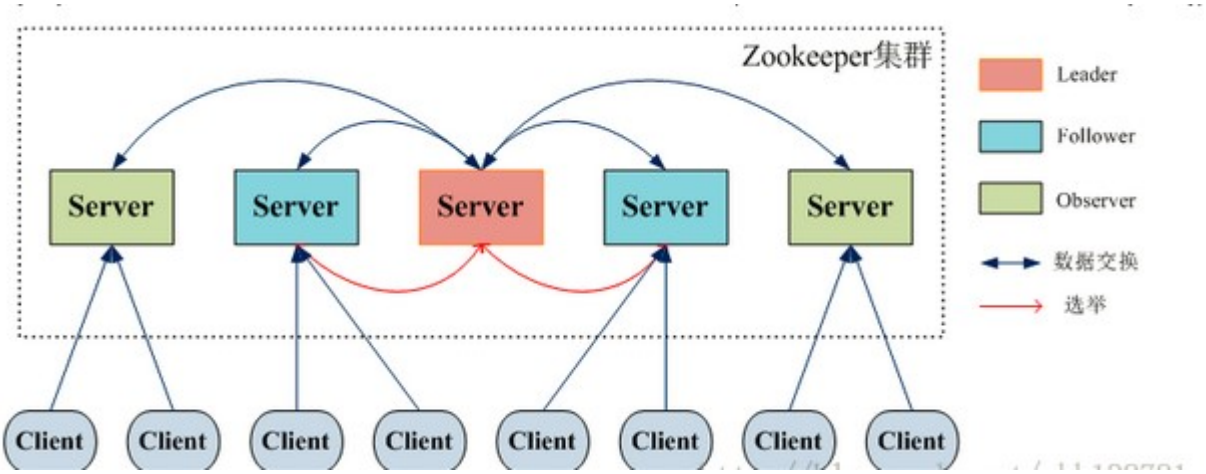
每台服务器对应的myid和server后面的数字有关;以此类推

192.168.190.101 myid 1

192.168.190.102 myid 2

192.168.190.105 myid 3

分别启动三台服务器，bin下面有错误日志,zookeeper.out. /usr/local/zookeeper/zookeeper-3.4.8/bin./zkServer.sh start./zkServer.sh status #查看zookeeper的启动状态 报错链接异常，有可能是防火墙问题， systemctl stop firewalld.service #停止firewallsystemctl disable firewalld.service #禁止firewall开机启动 firewall-cmd --state #查看默认防火墙状态（关闭后显示notrunning，开启后显示running） 5、zookeeper的三种角色Leader:接受所有Follower的提案请求并统一协调发起提案的投票，负责与所有的Follower进行内部的数据交换(同步); □ （1）服务请求的唯一调度和处理者，保证集群事务处理的顺序性 （2）集群内部各服务的调度者 Follower: 直接为客户端服务并参与提案的投票，同时与Leader进行数据交换(同步); □ （1）处理客户端非事务请求，转发事务请求给Leader服务 （2）参与事务请求Proposal的投票 （3）参与Leader选举的投票 Observer: 直接为客户端服务但并不参与提案的投票，同时也与Leader进行数据交换(同步)，因此，Observer可以在不影响写性能的情况下，提升 读的性能



6、observer的作用及配置方法

增加一台计算机：ip为192.168.190.104,安装配置，同时在zoo.cfg文件中

增加 peerType = observer

然后在每台的计算机中的{zoookeeper_home}/conf/zoof下都添加如下配置

server.1=192.168.190.101:2888:3881

server.2=192.168.190.102:2888:3881

server.3=192.168.190.105:2888:3881

server.4=192.168.190.104:2888:3881:observer

192.168.190.101配置

```
#集群配置
server.1=192.168.190.101:2888:3881
server.2=192.168.190.102:2888:3881
server.3=192.168.190.105:2888:3881
server.4=192.168.190.104:2888:3881:observer
```

192.168.190.102配置

```
server.1=192.168.190.101:2888:3881
server.2=192.168.190.102:2888:3881
server.3=192.168.190.105:2888:3881
server.4=192.168.190.104:2888:3881:observer
~
```

192.168.190.105配置

```
#autopurge.purgeInterval=1
server.1=192.168.190.101:2888:3881
server.2=192.168.190.102:2888:3881
server.3=192.168.190.105:2888:3881
server.4=192.168.190.104:2888:3881:observer
~
```

192.168.190.104配置

```
peerType=observer


server.1=192.168.190.101:2888:3881
server.2=192.168.190.102:2888:3881
server.3=192.168.190.105:2888:3881
server.4=192.168.190.104:2888:3881:observer #这里是指定observer机器
```

启动

7.伪集群模式

server.1=192.168.190.101:2888:3881

server.2=192.168.190.101:2889:3882

server.3=192.168.190.101:2890:3883