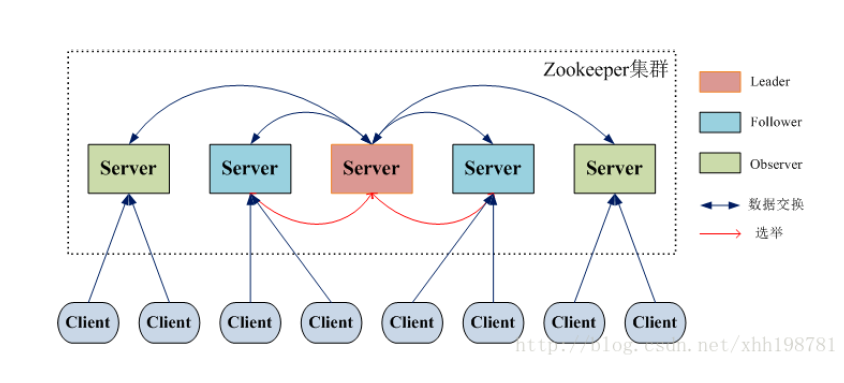
**必须提前看的文章：https://www.douban.com/note/208430424/**

安装zk集群的时候，server.A=B：C：D：其中A 是一个数字，表示这个是第几号服务器；B 是这个服务器的ip地址；C 表示的是这个服务器与集群中的Leader 服务器交换信息的端口；D 表示的是万一集群中的Leader 服务器挂了，需要一个端口来重新进行选举，选出一个新的Leader，而这个端口就是用来执行选举时服务器相互通信的端口。如果是伪集群的配置方式，由于B 都是一样，所以不同的Zookeeper 实例通信端口号不能一样，所以要给它们分配不同的端口号



Zookeeper架构图

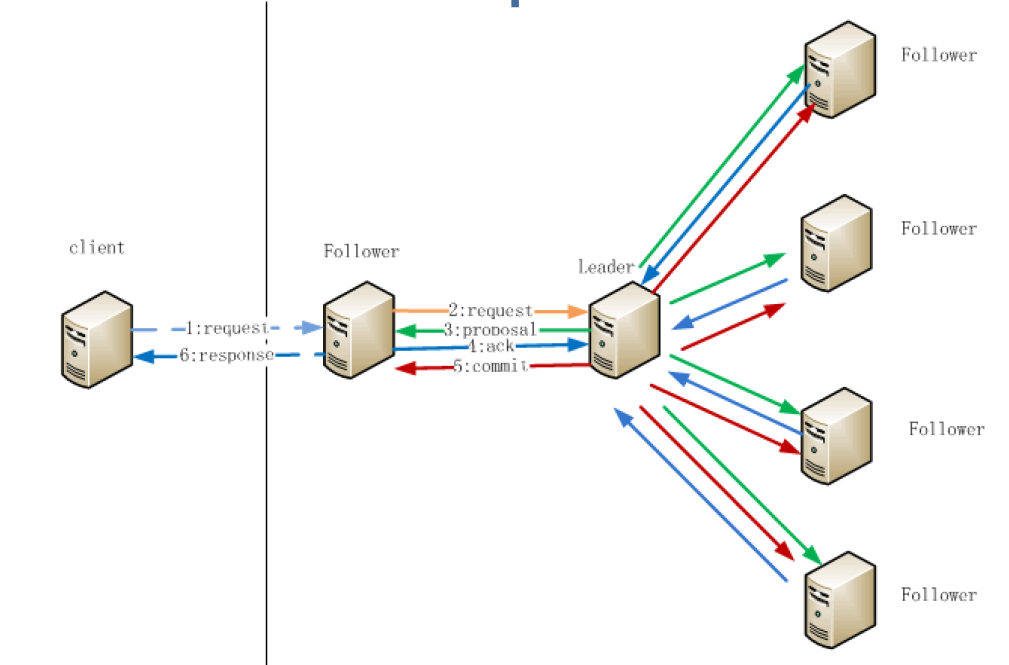
* Leader，负责进行投票的发起和决议，更新系统状态。
* Learner，包括follower和observer。
* Follower用于接收客户端请求并向客户端返回结果，在选举过程中参与投票。
* Observer可以接受客户端连接，将写请求转发给leader，但observer不参与投票过程，只同步leader的状态，observer的目的是为了扩展系统，提高读取速度。

（为什么需要Observer呢？为什么不直接增加Follower呢？如果我们为了实现吞吐量，增加了大量的Follower，那么当每次一次update提交的时候，都需要超过一半的follower进行投票，需要大量的时间，如果我们不增加Follower，仅仅增加Observer，那么在增加了吞吐量的同时，在update的时候，由于Follower并没有增加，所以进行投票的时候会很快）

* Client，请求发起方。
* Proposal，请求
* Zxid，为了保证事务的顺序一致性，zookeeper采用了递增的事务id号（zxid）来标识事务。所有的提议（proposal）都在被提出的时候加上了zxid。实现中zxid是一个64位的数字，它高32位是epoch用来标识leader关系是否改变，每次一个leader被选出来，它都会有一个新的epoch，标识当前属于那个leader的统治时期。低32位用于递增计数。

举例：

* 如果一个客户端1（第一个client）查询数据，那么他会想Observer发起请求，Observer读取自己已经备份的文件并返回客户端需要的数据，这个时候并不能保证数据是最新的，客户端可以通过sync方法进行说明需要最新的数据，这个时候，Observer会打电话给leader来进行数据的同步（为什么会出现数据不通过呢？例如：由于网络的原因，leader发起更新指令，但是observer还没有收到。），当同步更新完成后返回更新的最新的结果给client。
* 如果客户端1（第一个client）需要更新（增，删，改）数据，那么他会把该请求（proposal）发送给leader，为什么需要发送给leader（保证proposal被按照顺序处理），当leader收到该请求以后，leader会把该请求（proposal）广播给所有的followers，followers收到该proposal以后，会进行发送ack给leader来表明时候同意该proposal，如果leader收到了超过一半的ack同意消息，就会给所有的followers发送commit来说明该proposal生效，请更新到最新消息。如果在收到超过一半的ack不同意消息，就会给所有的followers发送commit来说明该proposal不生效，observer这个时候需要进行数据的同步，下图具体说明了该情况。



数据的更新操作（增，删，改）

Zk 奇数的原因：

由于投票需要半数以上，举例：我们有8台服务器，那么半数以上至少是5台，他允许3台服务器宕机，如果我们有7台服务器，那么半数以上至少是是4台，他允许3台服务器宕机，所以8台和7台服务器他的容错是一样的。

Zk启动的状态来进一步说明zk的选举机制：

* 如果我们有3台zk集群，现在我们启动第一台服务器，然后观察zk的status，发现出现error，因为集群有3台服务器，现在只启动一台服务器，半数以上至少需要2台才能保证zk集群正常运行，所以现在zk虽然进程启动起来了，但是他并没有被分配角色，所以error。他这个时候不断的想第二台和第三台服务器发送选举请求，但是都失败。
* 接着我们启动第二台服务器，然后查看第二台和第一台服务器的status，就会发现错误消失了，一般都是第二台是leader，因为在第二台启动的时候，第二台机器也进行发送我要成为leader的请求，由于这个时候zid是最大的，所以一般都会成功，所以第二台成为了leader，这个仅仅是自己实验的结果，自己对结果的理解。
* 接着启动第三太服务器，自然回事follower。

