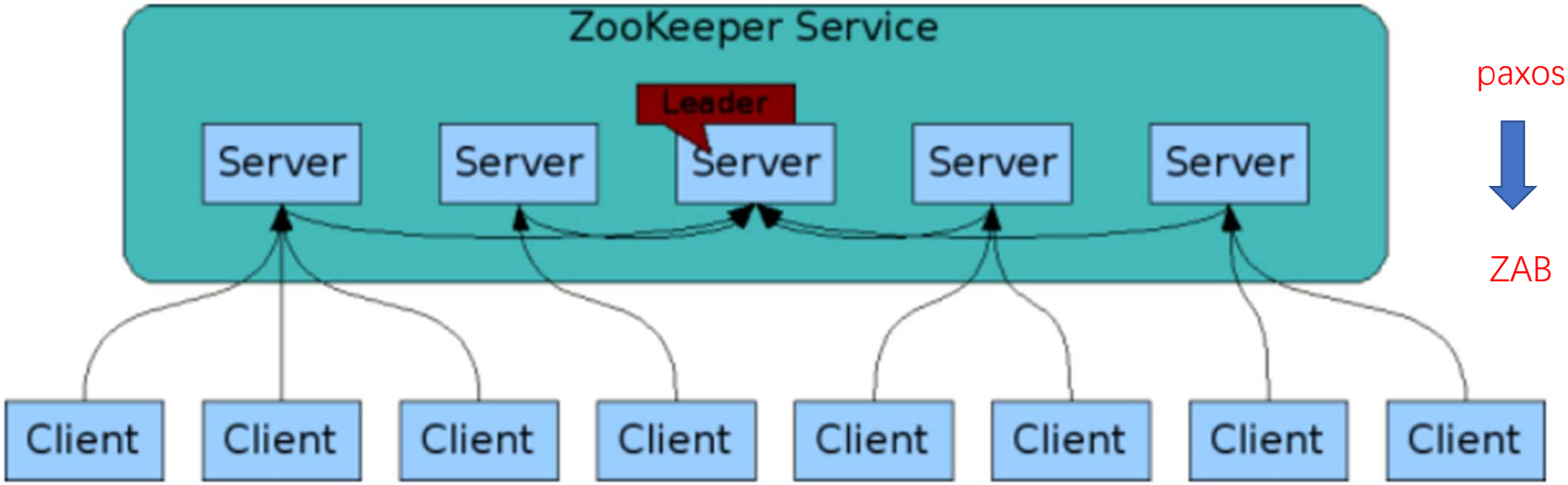
ZooKeeper

ZooKeeper, a service for coordinating processes of distributed applications.

https://zookeeper.apache.org/

To summarize, in this paper our main contributions are:

- **Coordination kernel**: We propose a wait-free coordination service with relaxed consistency guarantees for use in distributed systems. In particular, we describe our design and implementation of a coordination kernel, which we have used in many critical applications to implement various coordination techniques.
- **Coordination recipes**: We show how ZooKeeper can be used to build higher level coordination primitives, even blocking and strongly consistent primitives, that are often used in distributed applications.
- **Experience with Coordination**: We share some of the ways that we use ZooKeeper and evaluate its performance.



paxos

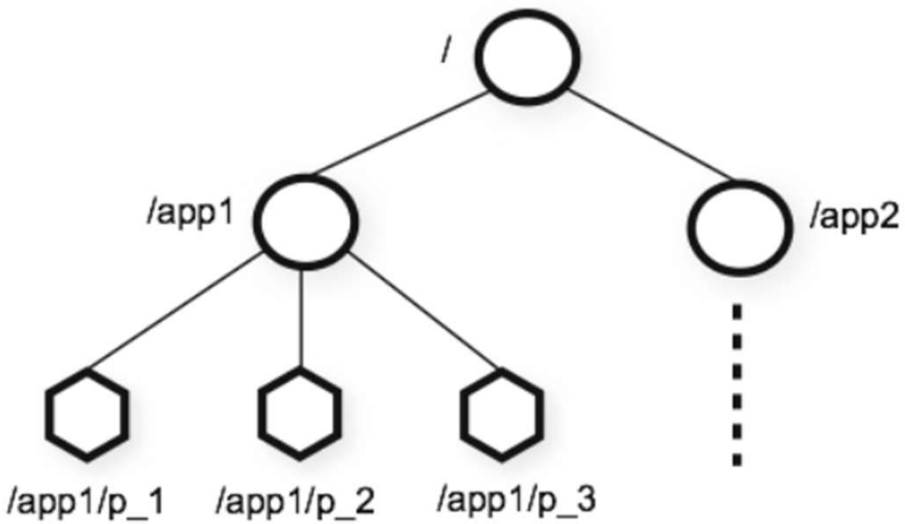
ZAB

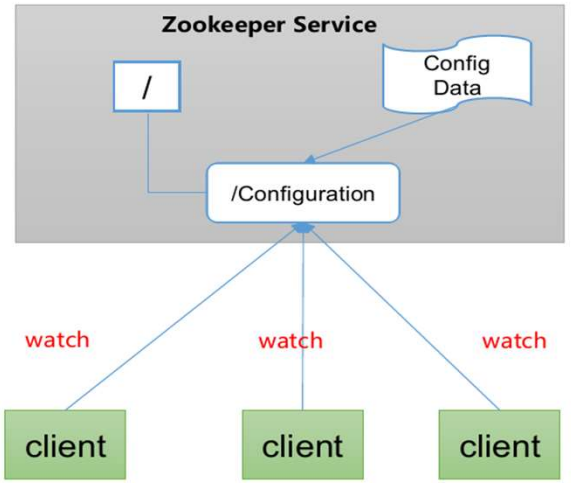
配置内容:

- 心跳时间
- 连接端口
- 通信时限
- ……

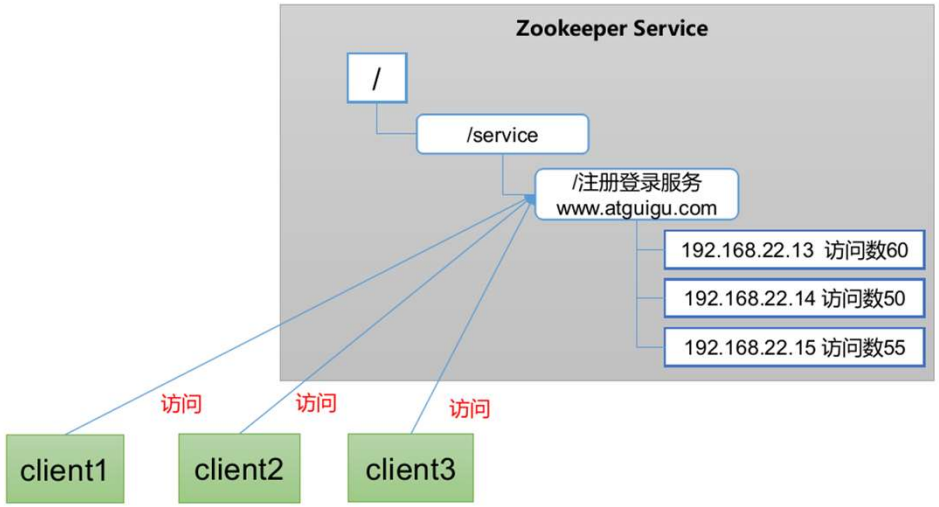
系统功能:

- 系统初始化（选举 leader ）
- Leader 挂掉之后重新选举
- ……

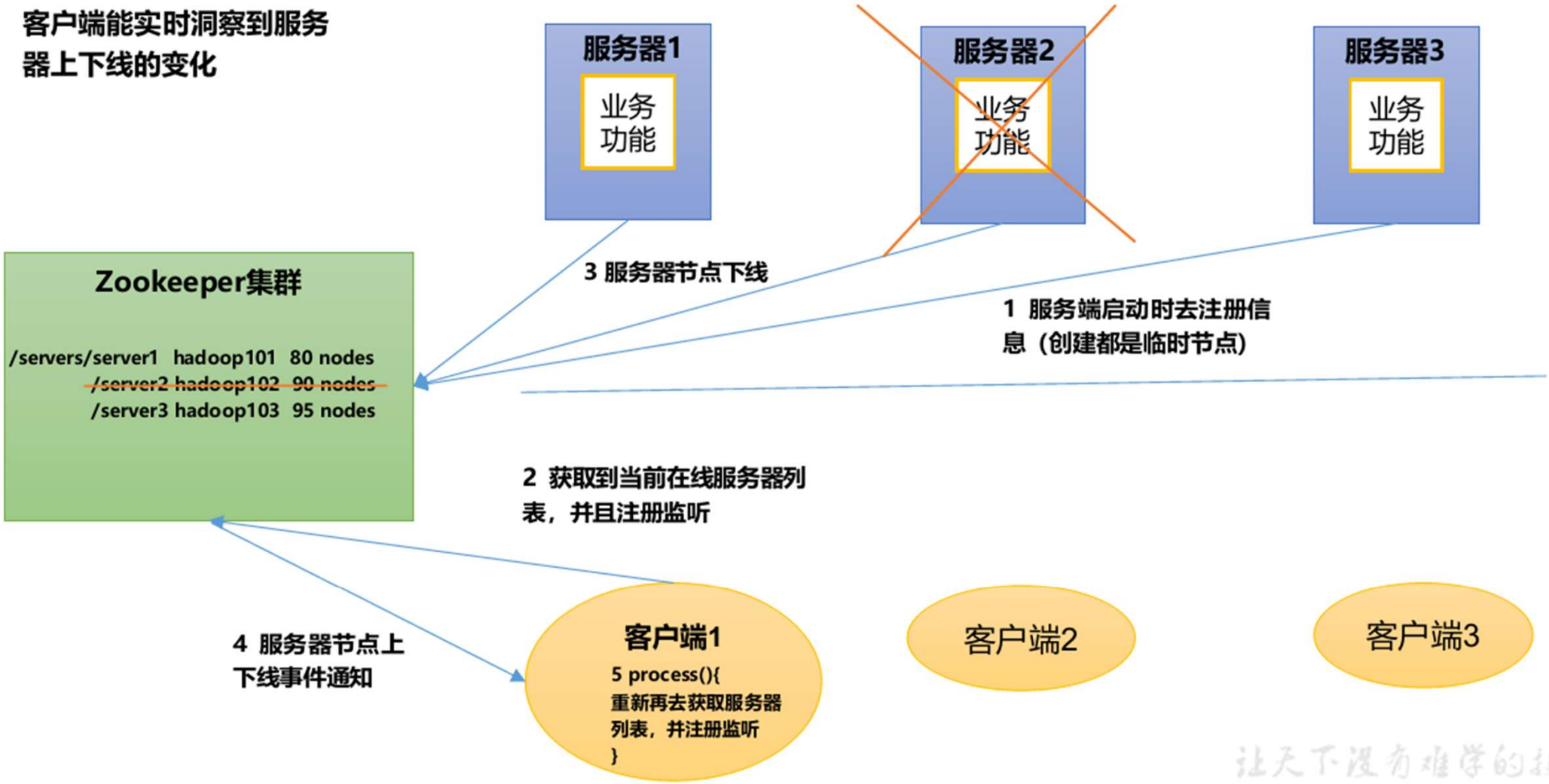




- 统一命名服务
- 统一配置管理
- 统一集群管理
- 服务器动态上下线
- 软负载均衡

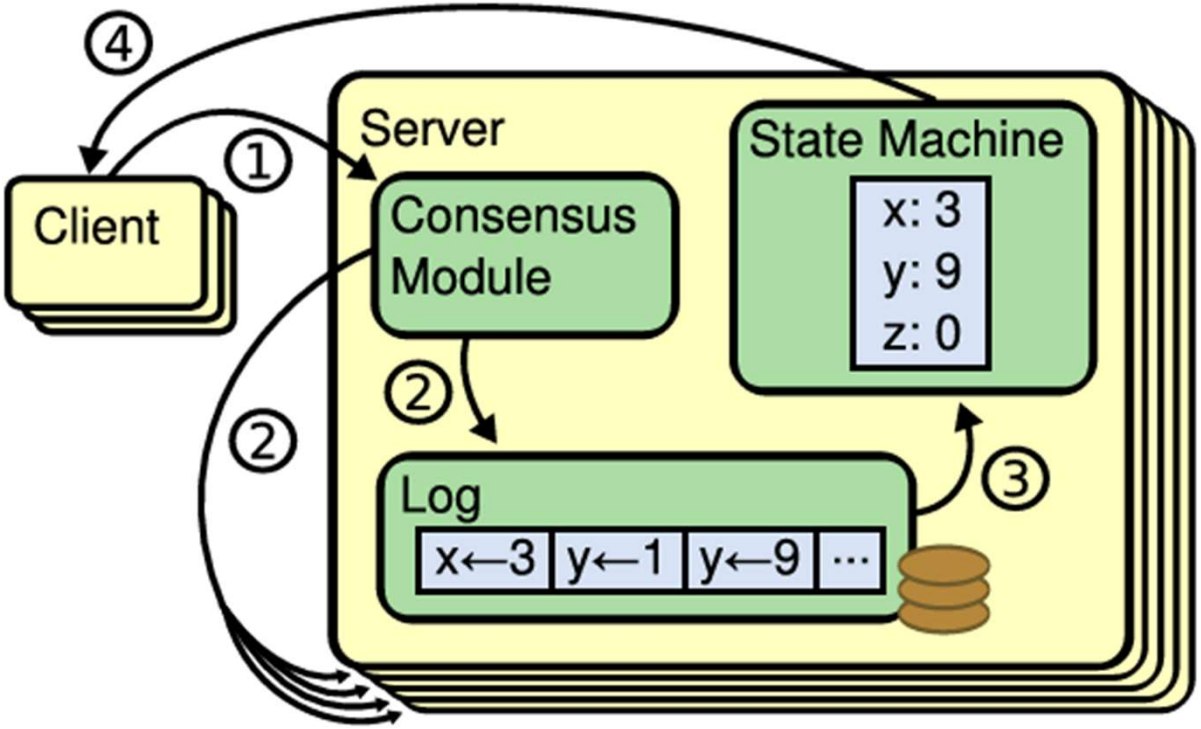


客户端能实时洞察到服务器上下线的变化



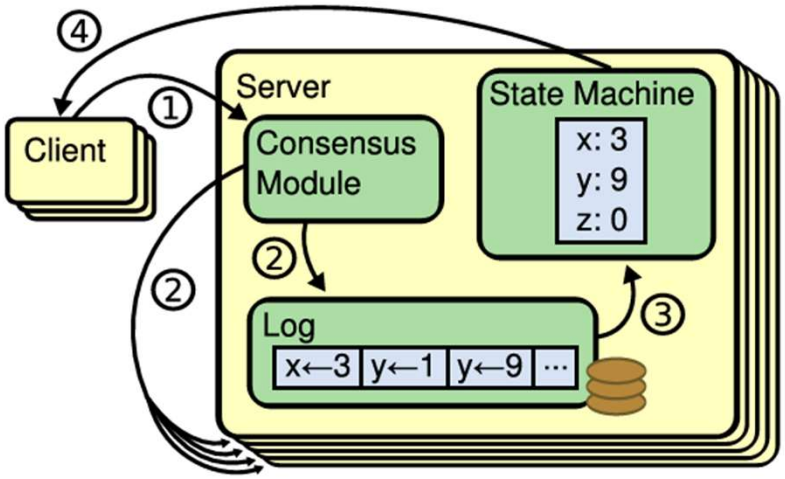
Client 写操作

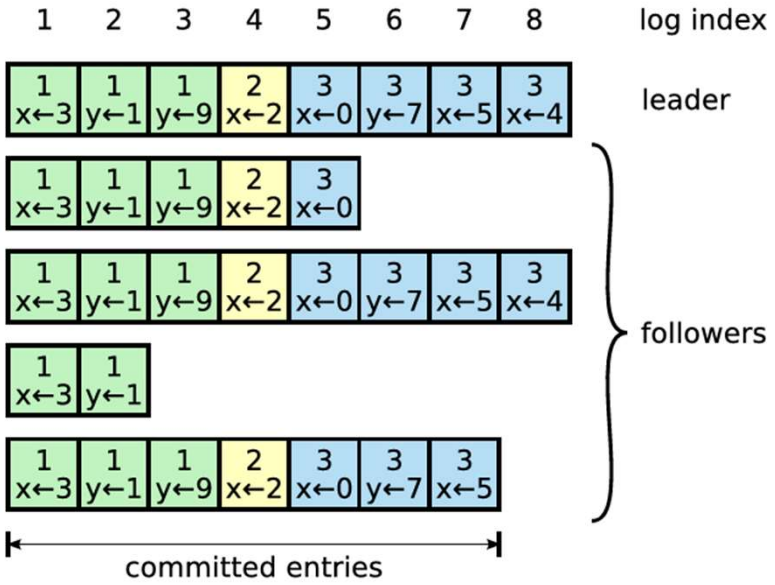
Client 不需要考虑信息发给了 leader 还是 follower,如果是 follower， follower 会先去向 leader 发写的请求，然后 leader 会返回确认，同时把请求发给别的 follower，超过半数的确认后 leader 会提交这次请求。



Client 读操作

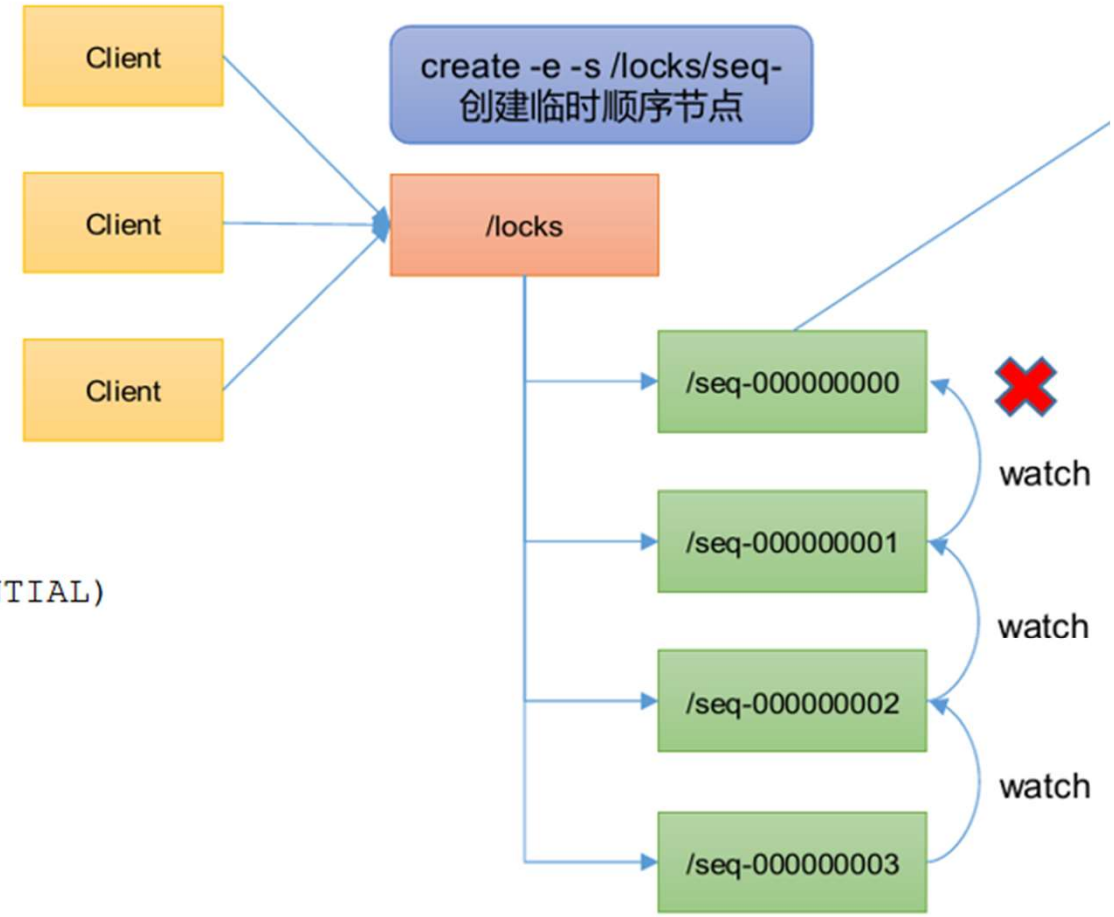
1. **强一致性**：禁止从 follower 读数据
2. **非强一致性**：读数据的服务器可能并不属于 leader 所认为的大多数在线的服务器，读取到的信息并不一定是最新的消息





分布式锁:

ZooKeeper is not a lock service. It can be used by clients to implement locks, but there are no lock operations in its API.



Lock

```
1 n = create(l + "/lock-", EPHEMERAL|SEQUENTIAL)
2 C = getChildren(l, false)
3 if n is lowest znode in C, exit
4 p = znode in C ordered just before n
5 if exists(p, true) wait for watch event
6 goto 2
```

Unlock

```
1 delete(n)
```

Client API

- **create(path, data, flags):** Creates a znode with path name path, stores data[] in it, and returns the name of the new znode. flags enables a client to select the type of znode: regular, ephemeral, and set the sequential flag;
- **delete(path, version):** Deletes the znode path if that znode is at the expected version;
- **exists(path, watch):** Returns true if the znode with path name path exists, and returns false otherwise. The watch flag enables a client to set a watch on the znode;
- **getData(path, watch):** Returns the data and meta-data, such as version information, associated with the znode. The watch flag works in the same way as it does for exists(), except that ZooKeeper does not set the watch if the znode does not exist;
- **setData(path, data, version):** Writes data[] to znode path if the version number is the current version of the znode;
- **getChildren(path, watch):** Returns the set of names of the children of a znode;
- **sync(path):** Waits for all updates pending at the start of the operation to propagate to the server that the client is connected to. The path is currently ignored.