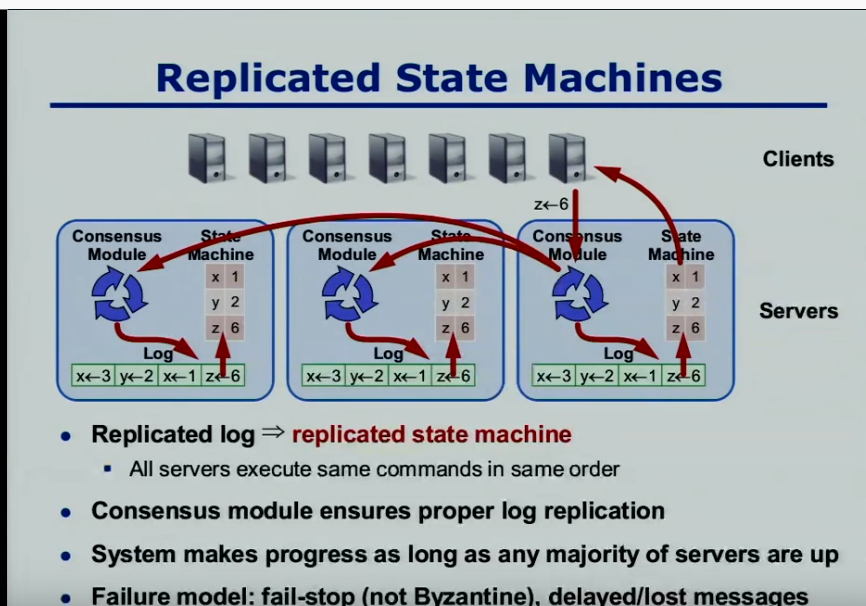
raft

state machine



# #redis

## data

// 集群状态，每个节点都保存着一个这样的状态，记录了它们眼中的集群的样子。

// 另外，虽然这个结构主要用于记录集群的属性，但是为了节约资源，

// 有些与节点有关的属性，比如 slots\_to\_keys 、 failover\_auth\_count

// 也被放到了这个结构里面。

typedef struct clusterState {

    // 指向当前节点的指针

    clusterNode \*myself;  /\* This node \*/

    // 集群当前的配置纪元，用于实现故障转移

    uint64\_t currentEpoch;

    // 集群当前的状态：是在线还是下线

    int state;            /\* REDIS\_CLUSTER\_OK, REDIS\_CLUSTER\_FAIL, ... \*/

    // 集群中至少处理着一个槽的节点的数量。

    int size;             /\* Num of master nodes with at least one slot \*/

    // 集群节点名单（包括 myself 节点）

    // 字典的键为节点的名字，字典的值为 clusterNode 结构

    dict \*nodes;          /\* Hash table of name -> clusterNode structures \*

## 函数

clusterHandleSlaveFailover

从节点发起故障转移

clusterRequestFailoverAuth

This function sends a FAILOVE\_AUTH\_REQUEST message to every node

 向其他所有节点发送 FAILOVE\_AUTH\_REQUEST 信息，

 \* 看它们是否同意由这个从节点来对下线的主节点进行故障转移。

FAILOVE\_AUTH\_REQUEST

// 请求进行故障转移操作，要求消息的接收者通过投票来支持消息的发送者

#define CLUSTERMSG\_TYPE\_FAILOVER\_AUTH\_REQUEST 5 /\* May I failover? \*/

// 消息的接收者同意向消息的发送者投票

#define CLUSTERMSG\_TYPE\_FAILOVER\_AUTH\_ACK 6     /\* Yes, you have my vote \*/