

## 对外接口

### 添加数据

- 1 . 申请存储族，或设置 key
- 2 . 写数据
- 3 . 提交数据
- 4 . 返回操作结果

### 读数据

- 1 . 设置 key
- 2 . 设置选择读取存储族还是单条数据
- 3 . 提交读取数据
- 4 . 返回读取数据

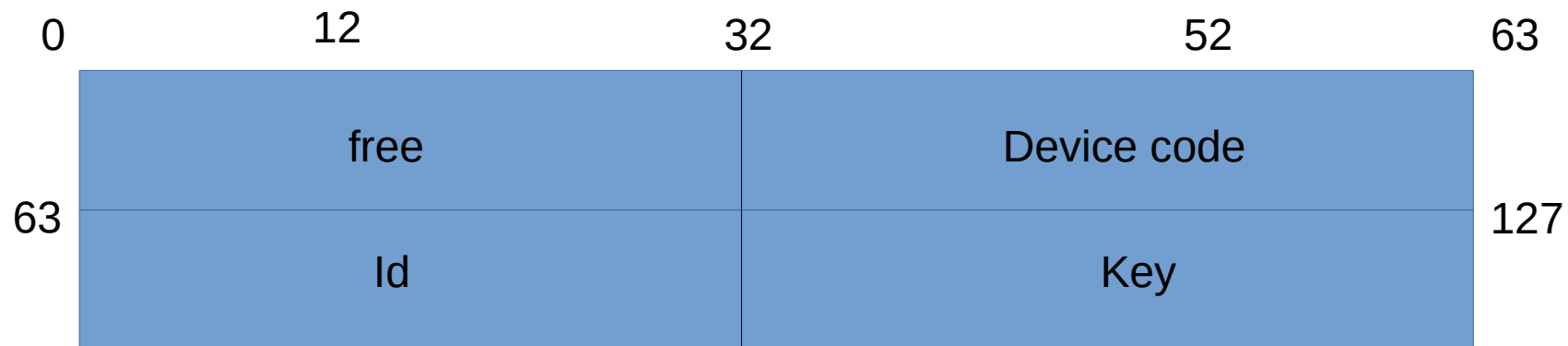
### 删除数据

- 1 . 设置 key
- 2 . 设置删除方式
- 3 . 提交数据
- 4 . 返回操作结果

### 更改数据

- 1 . 设置 key
- 2 . 设置更改方式
- 3 . 提交数据
- 4 . 返回操作结果

## key 的组成



Device Code	IP	Save speed	isFull
1	192.168.1.1	5	0
2	192.168.2.1	5	0
3	192.168.2.2	6	0

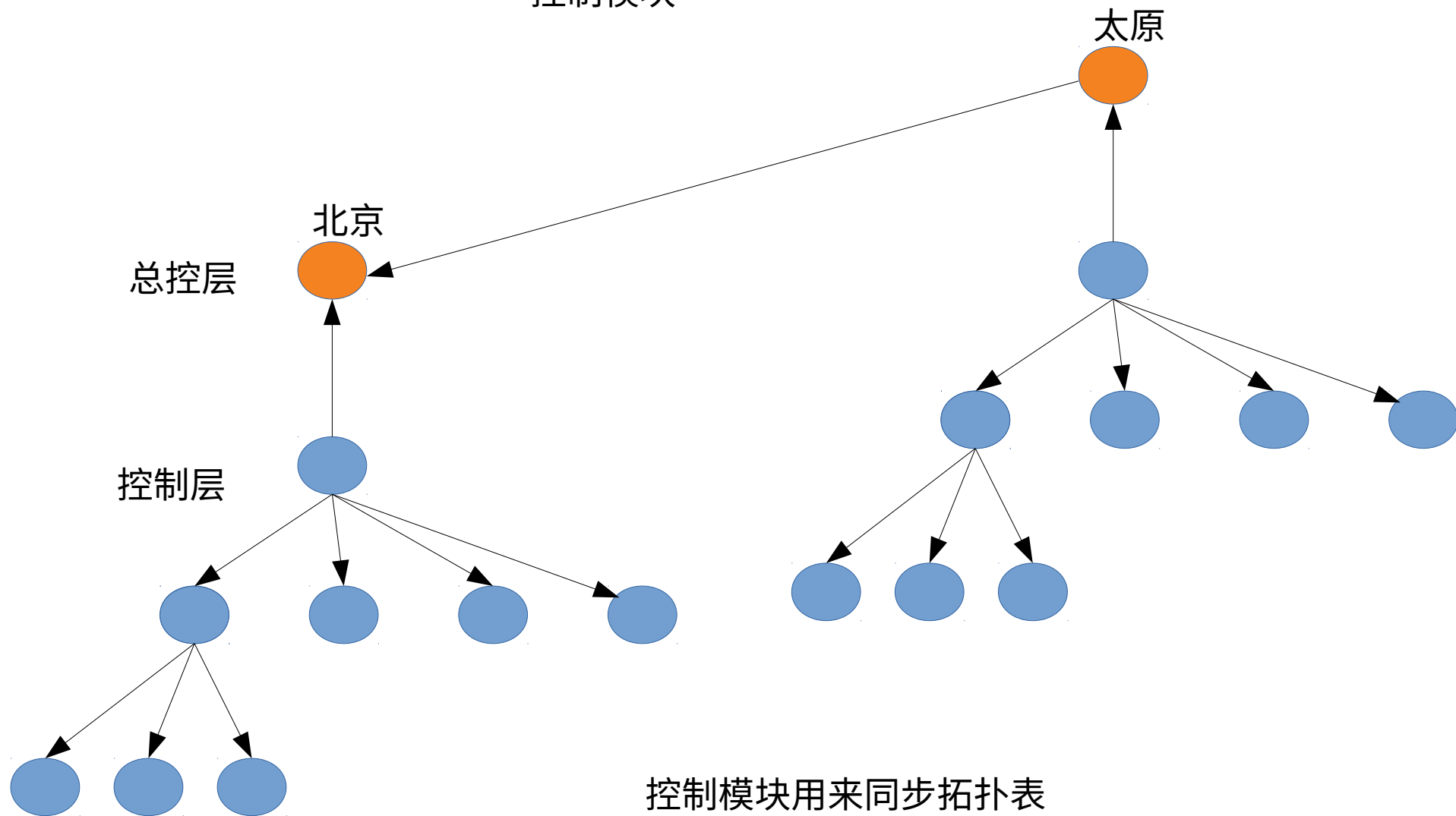
拓扑表

head_id	组成员	存储族数量	存储片大小	sub_id
1	192.168.1.1 x 192.168.2.1 x 192.168.2.2 x	1024	4M	1
1	192.168.1.1 x 192.168.2.1 x 192.168.2.2 x	1024	4M	2

存储族表

$$\text{Id} = (\text{head\_id} \ll 16) + \text{sub\_id}$$

控制模块



存储原则：

1. 就近存储

控制原则：

1. 树形组织控制层
2. 新结点容纳旧树
3. 上层主动连接下层

总控原则

1. 不主动连接控制层

总控层

控制层

被控 / 控制层

被控 / 控制层

...

被控层

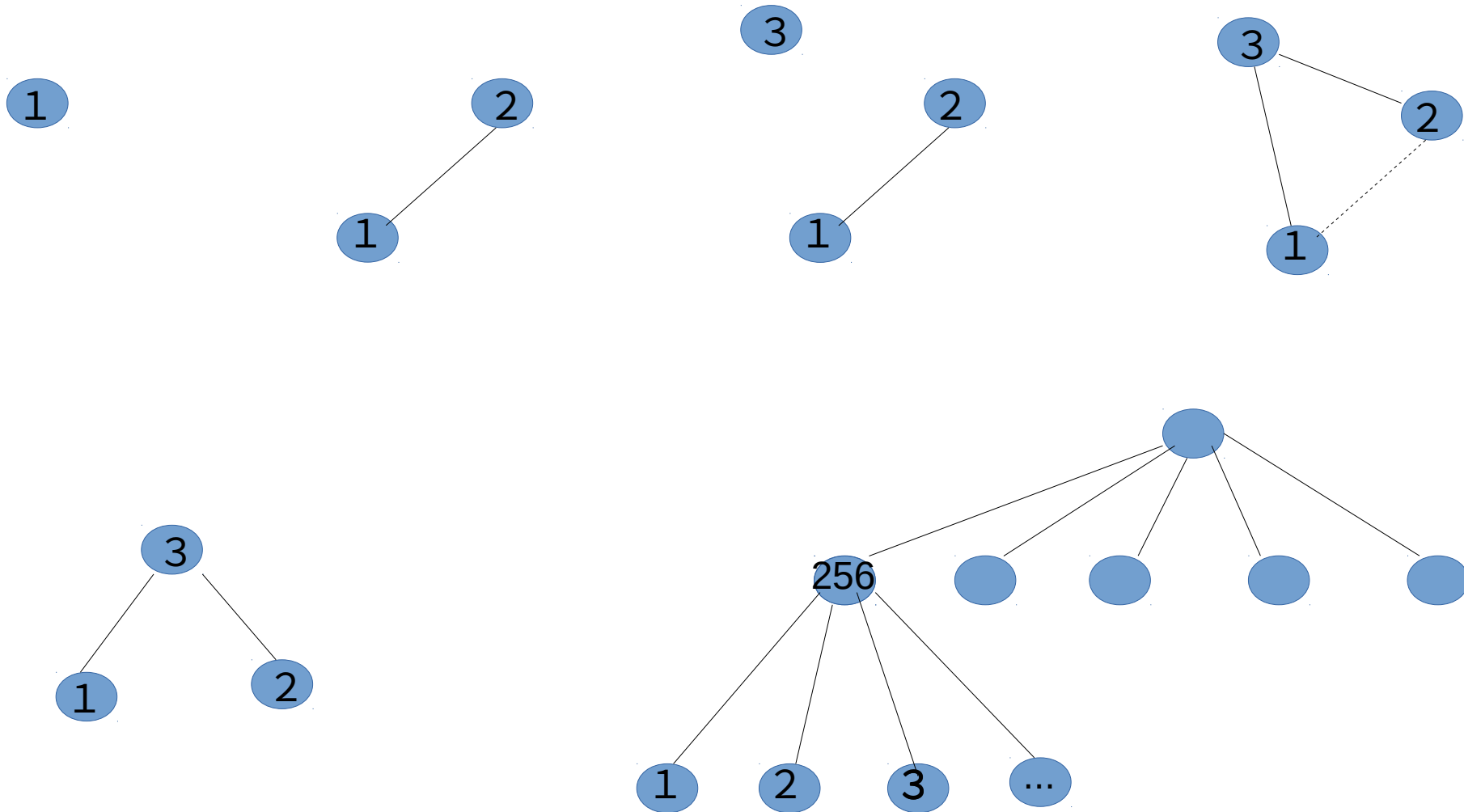
( 也是下一层的控制层 )

## 控制模块特点

上层拥有下层数据

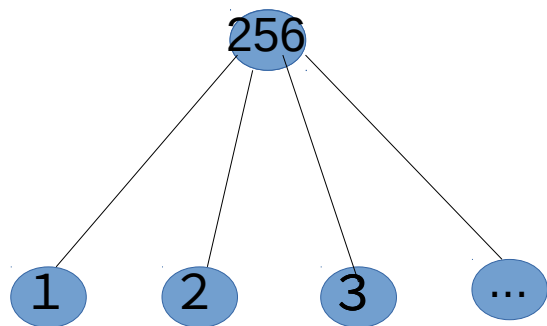
下层懒加载上层或同层拓扑表

## 控制层形成的过程 - 单个结点的进入

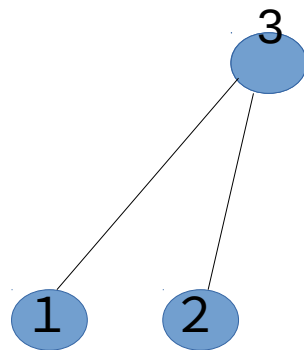




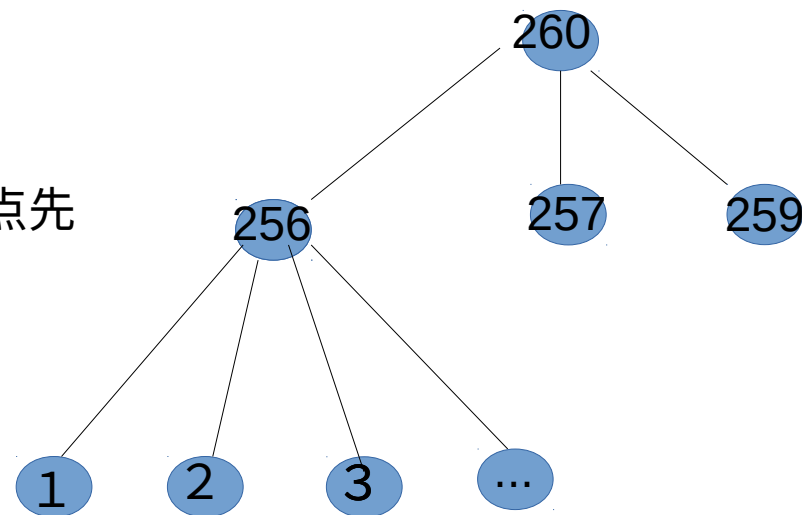
## 控制层形成的过程 - 多个结点的进入



旧结点



新加入的结点先  
组织成树



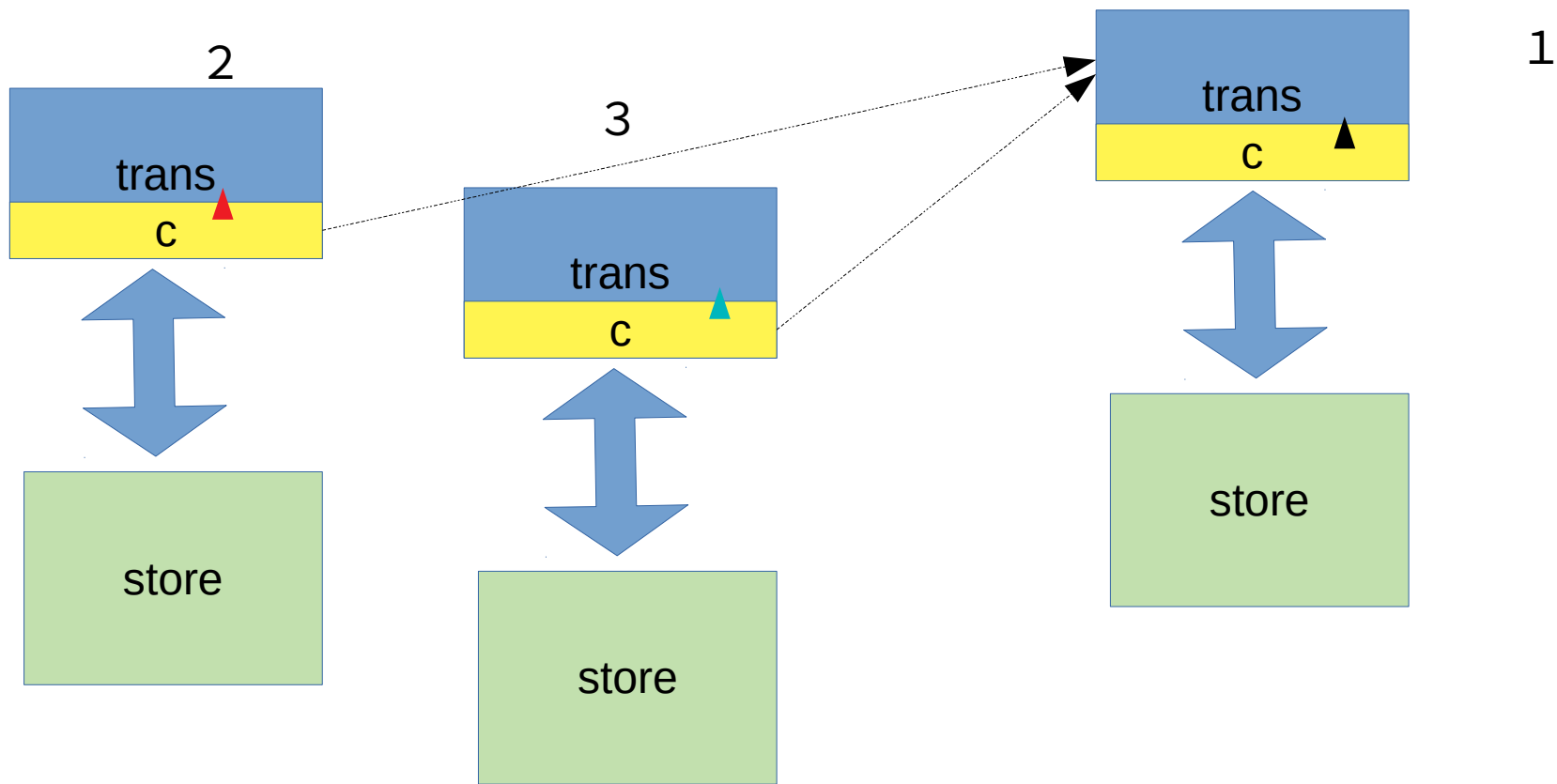
容纳旧结点后，更新设备 id

## 存储族的形成

按照最近原则，在数据的入口处寻找最近的设备，确定设备码，由此设备分配存储族。

此设备查找拓扑表，寻找速度最快的两个设备，形成存储族。

存储族由三个存储单元构成，区分等级。由最高等级者维护三者关系。



一个存储族由三个存储单元构成