

# Redis架构与介质选择指引

演讲者：民科

技术专家

## 01 Redis集群架构

- 集群架构详解
- proxy与直连
- 如何选型

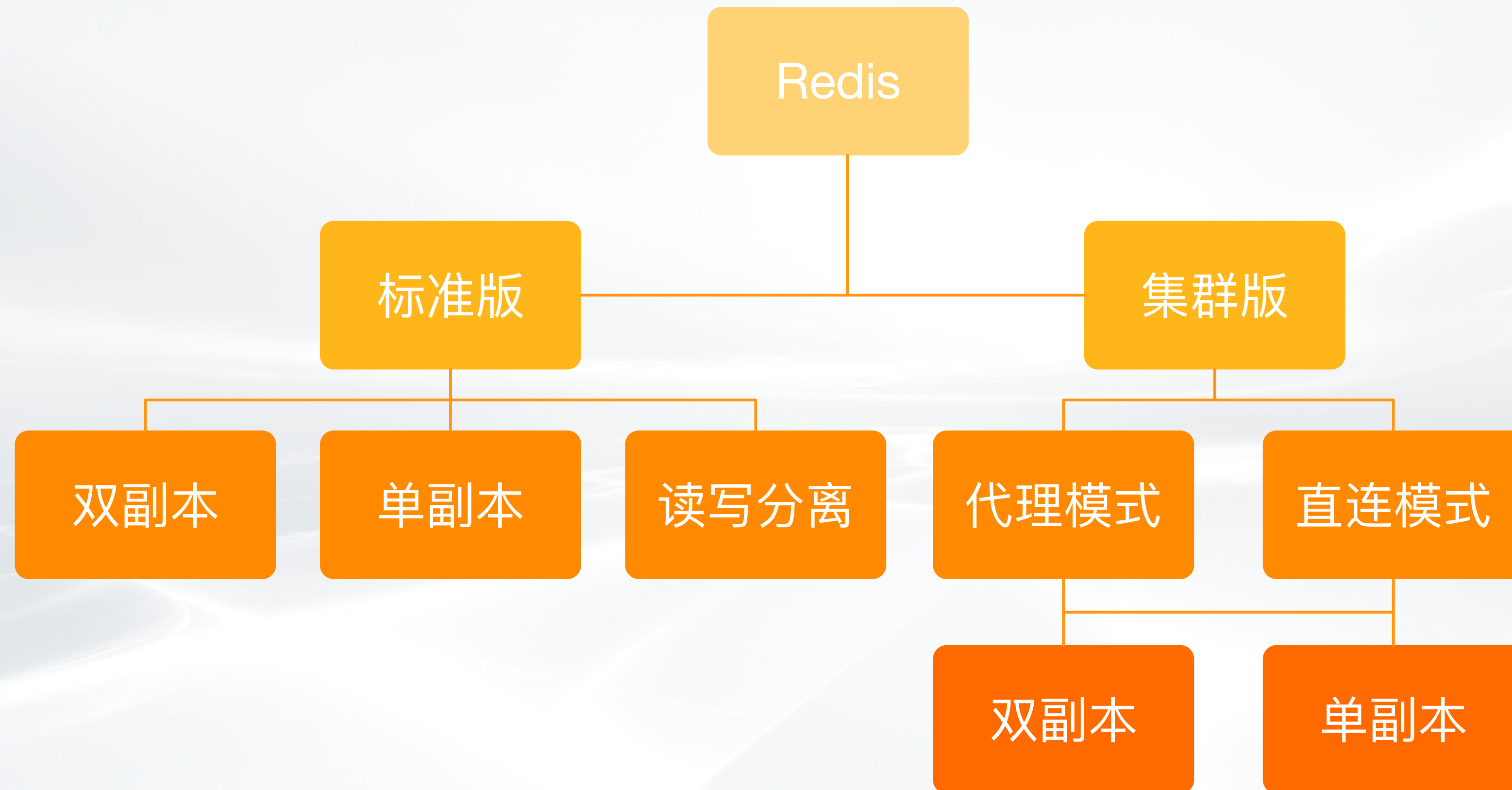
## 02 Redis存储介质

- 存储介质

## 03 从社区到企业版

- 高可用
- 平滑扩缩容

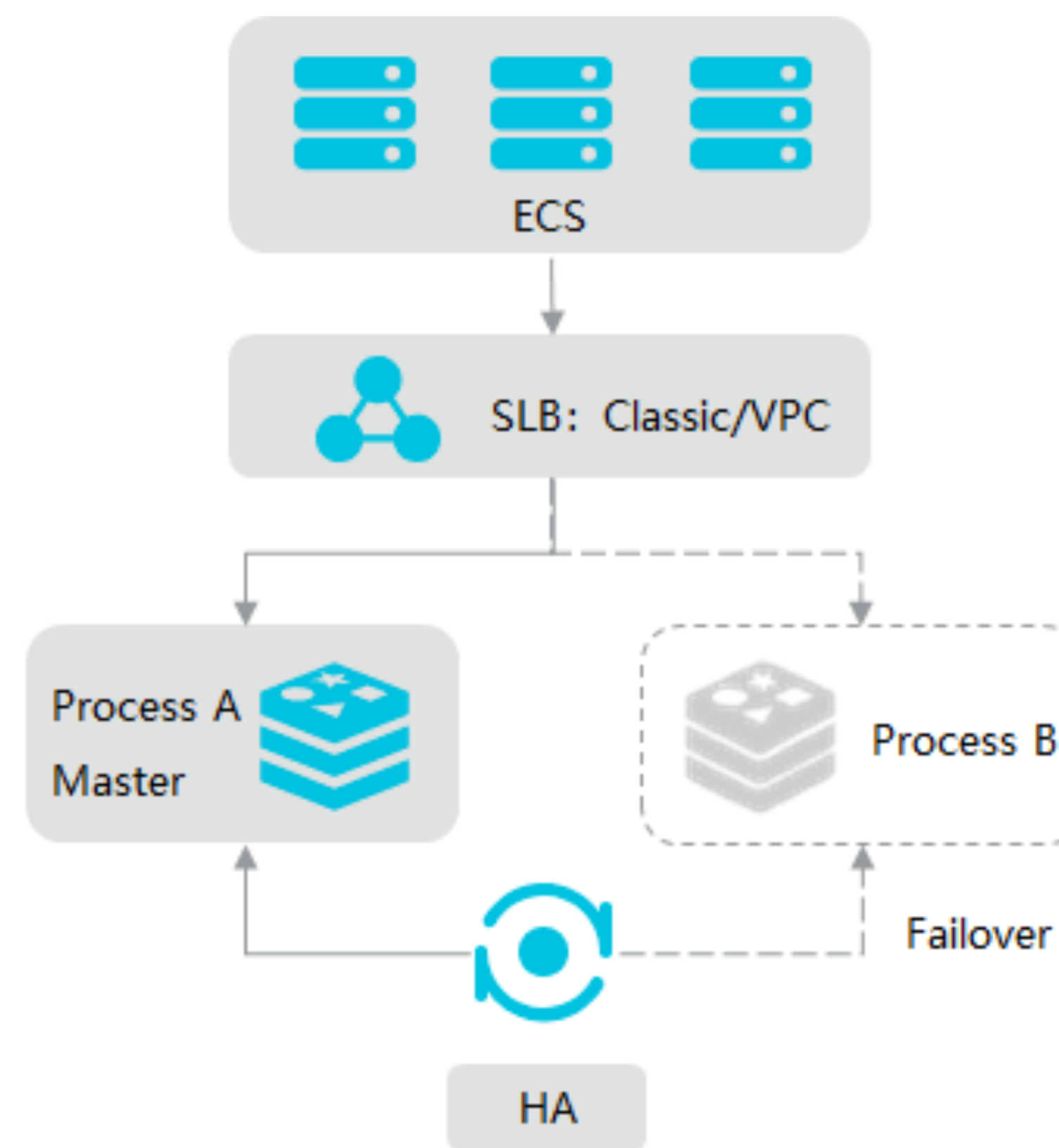
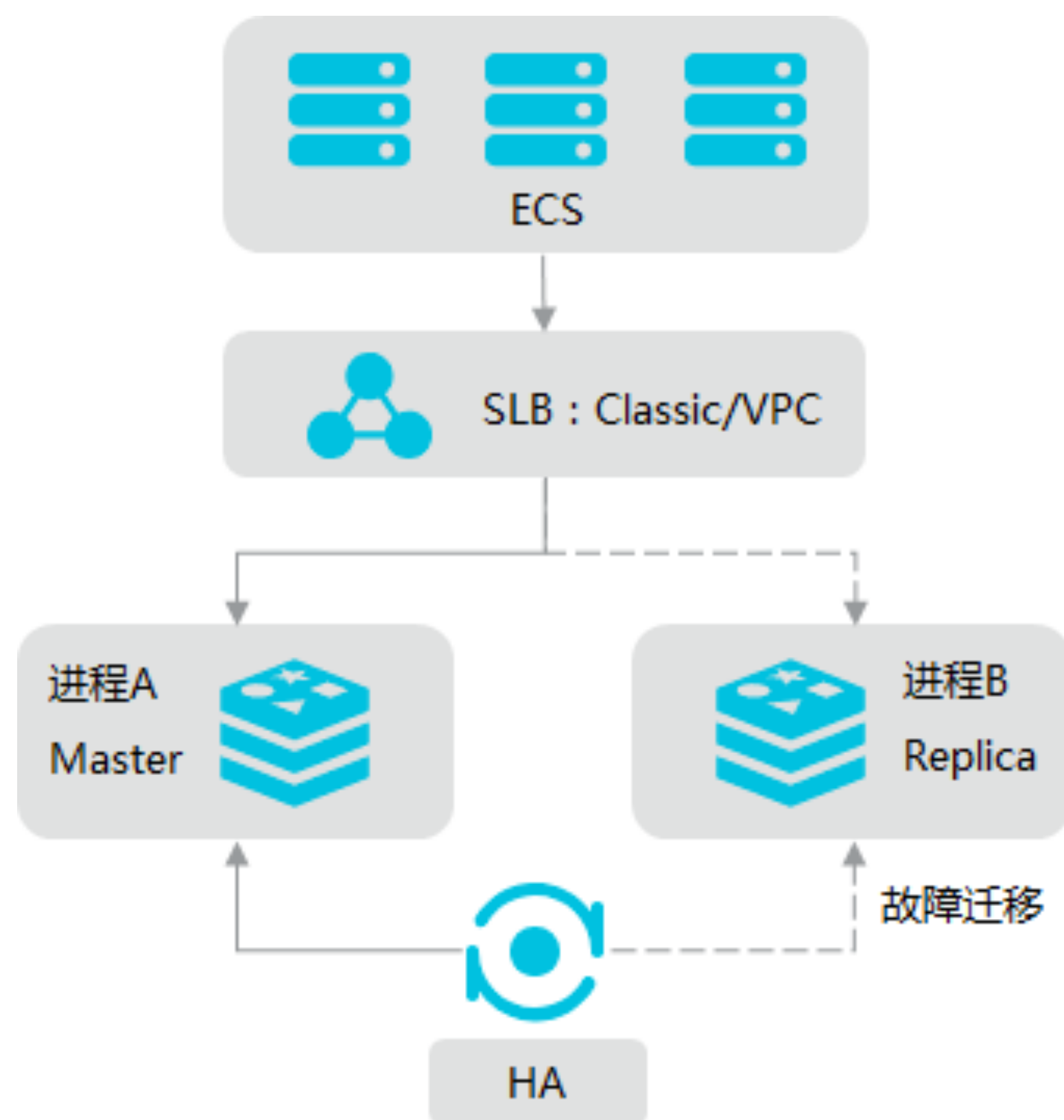
# 01 Redis集群架构



# 标准版

## 使用场景

- 对Redis协议兼容性要求较高的业务。
- 单个Redis性能压力可控的场景。
- Redis命令相对简单，排序和计算之类的命令较少的场景。



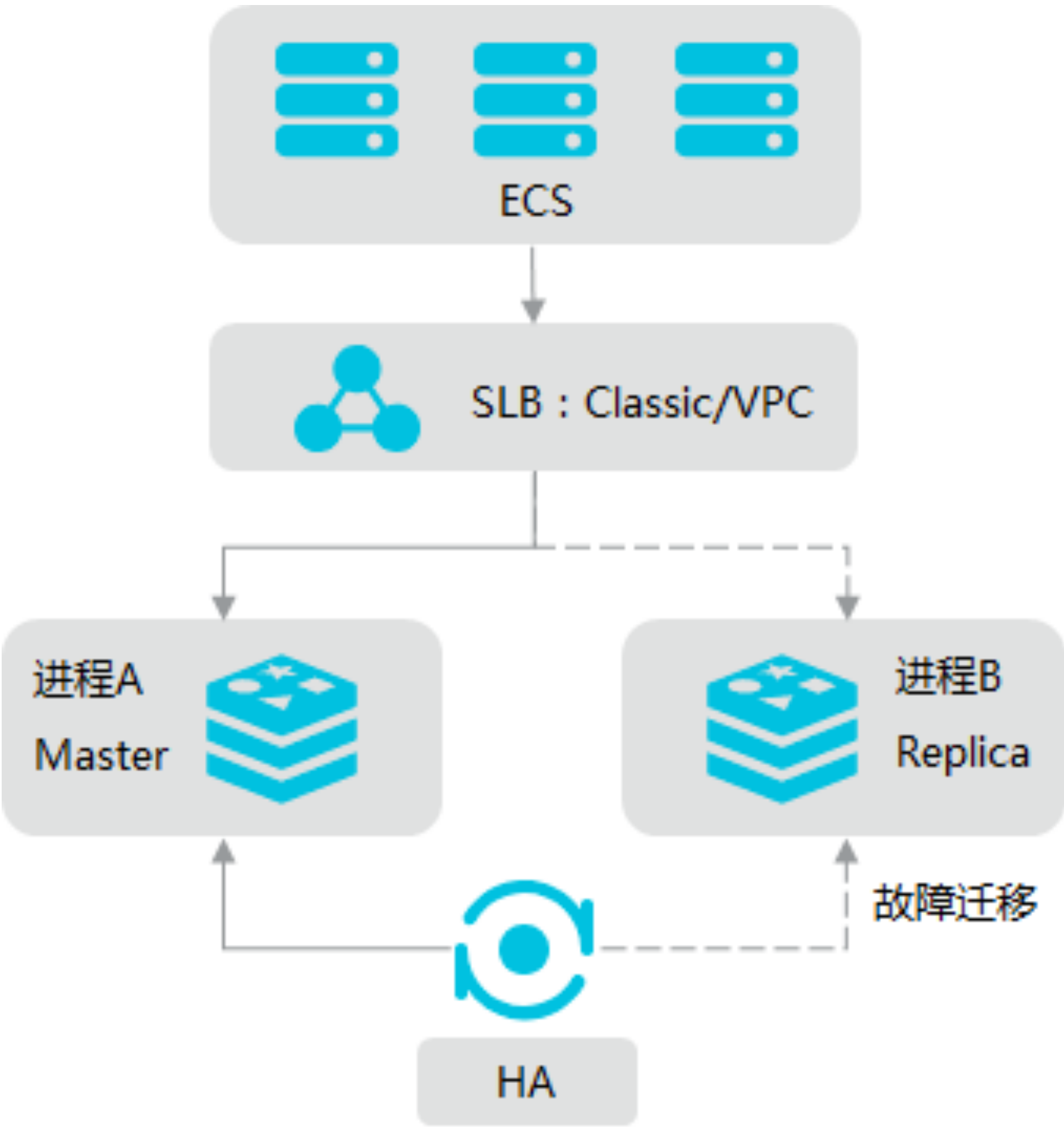


# 标准版-双副本

标准版-双副本模式采用主从（master-replica）模式搭建。主节点提供日常服务访问，备节点提供HA高可用，当主节点发生故障，系统会自动在30秒内切换至备节点，保证业务平稳运行。

## 特点

- 可靠性：采用双机主从（master-replica）架构，主从节点位于不同物理机。主节点对外提供访问，用户可通过Redis命令行和通用客户端进行数据的增删改查操作。当主节点出现故障，自研的HA系统会自动进行主从切换，保证业务平稳运行。
- 数据可靠：默认开启数据持久化功能，数据全部落盘。支持数据备份功能，用户可以针对备份集回滚实例或者克隆实例，有效地解决数据误操作等问题。同时，在支持容灾的可用区（例如杭州可用区H+I）创建的实例，还具备同城容灾的能力。

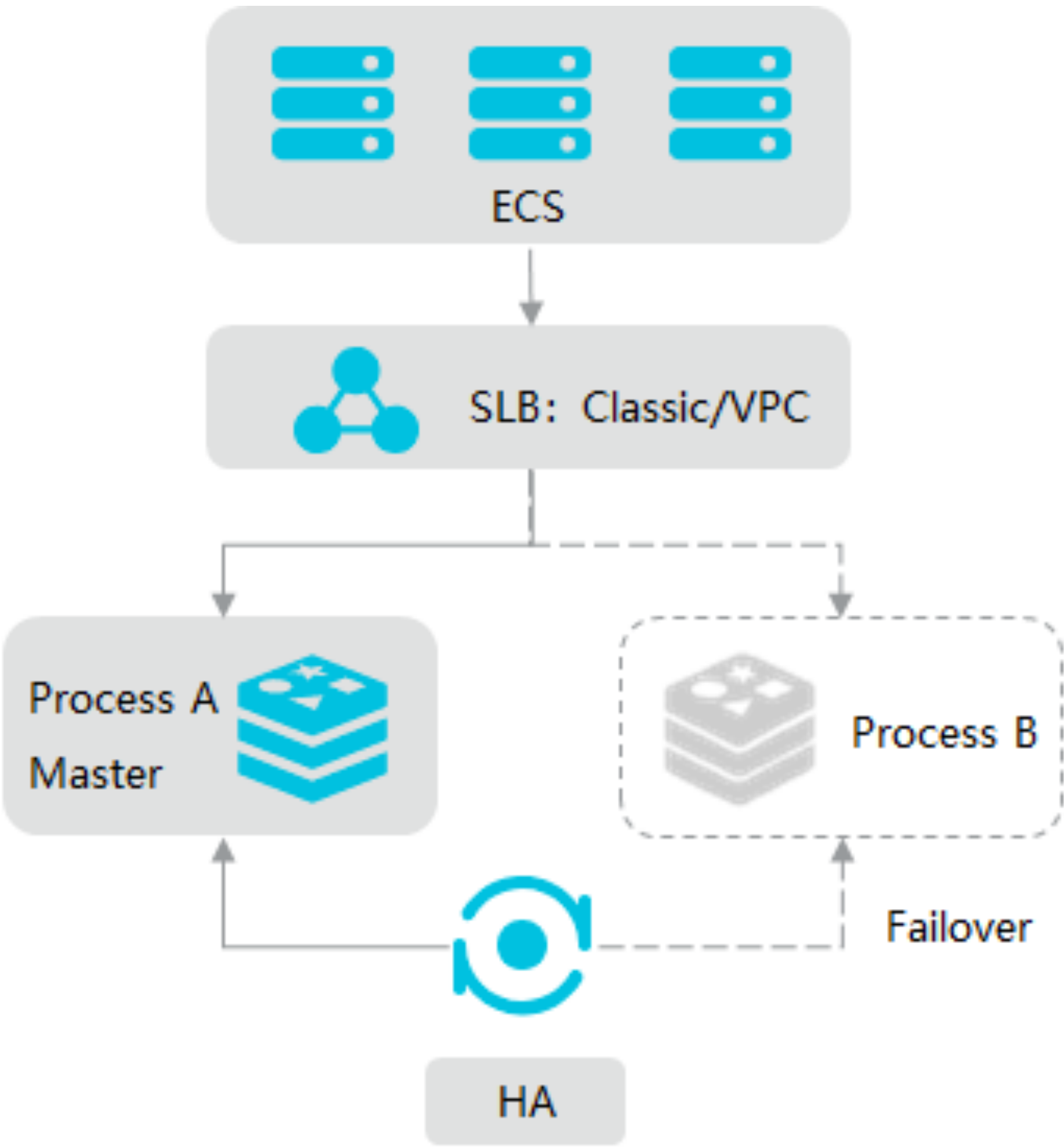


# 标准版-单副本

标准版-单副本采用单个数据库节点部署架构，没有可实时同步数据的备用节点，不提供数据持久化和备份策略，适用于数据可靠性要求不高的纯缓存业务场景使用。

## 特点

- 纯缓存类业务使用，单副本只有一个数据库节点，性价比高
- 阿里云自研HA高可用系统，异常30秒自动切换切换



# 读写分离

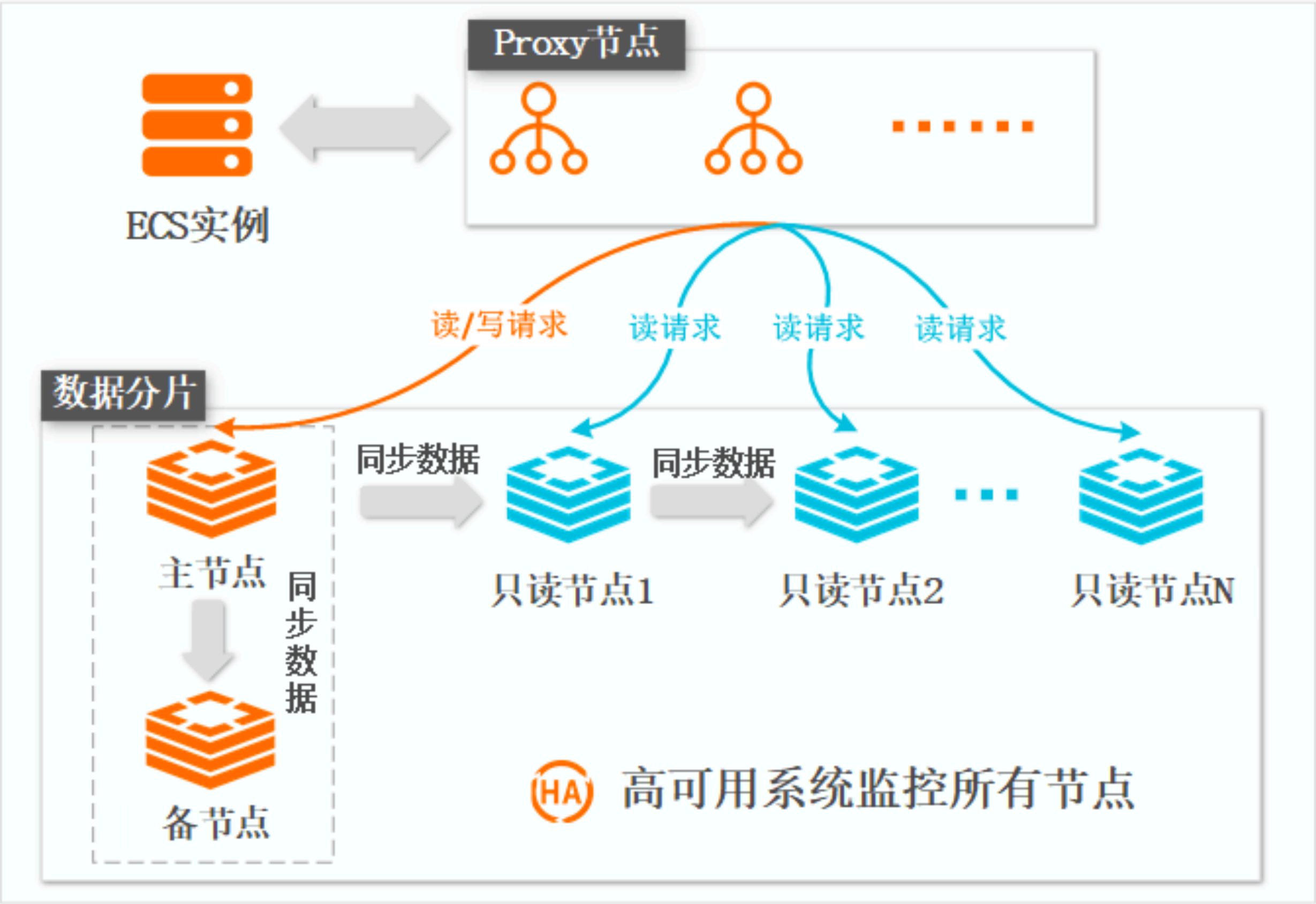
针对读多写少的业务场景，云数据库Redis推出了读写分离版的产品形态，提供高可用、高性能、灵活的读写分离服务，满足热点数据集中及高并发读取的业务需求，最大化地节约运维成本。

## 使用场景

- 读取请求QPS压力较大：适合读多写少型业务
- 对Redis协议兼容要求较高的业务

## 建议与使用须知

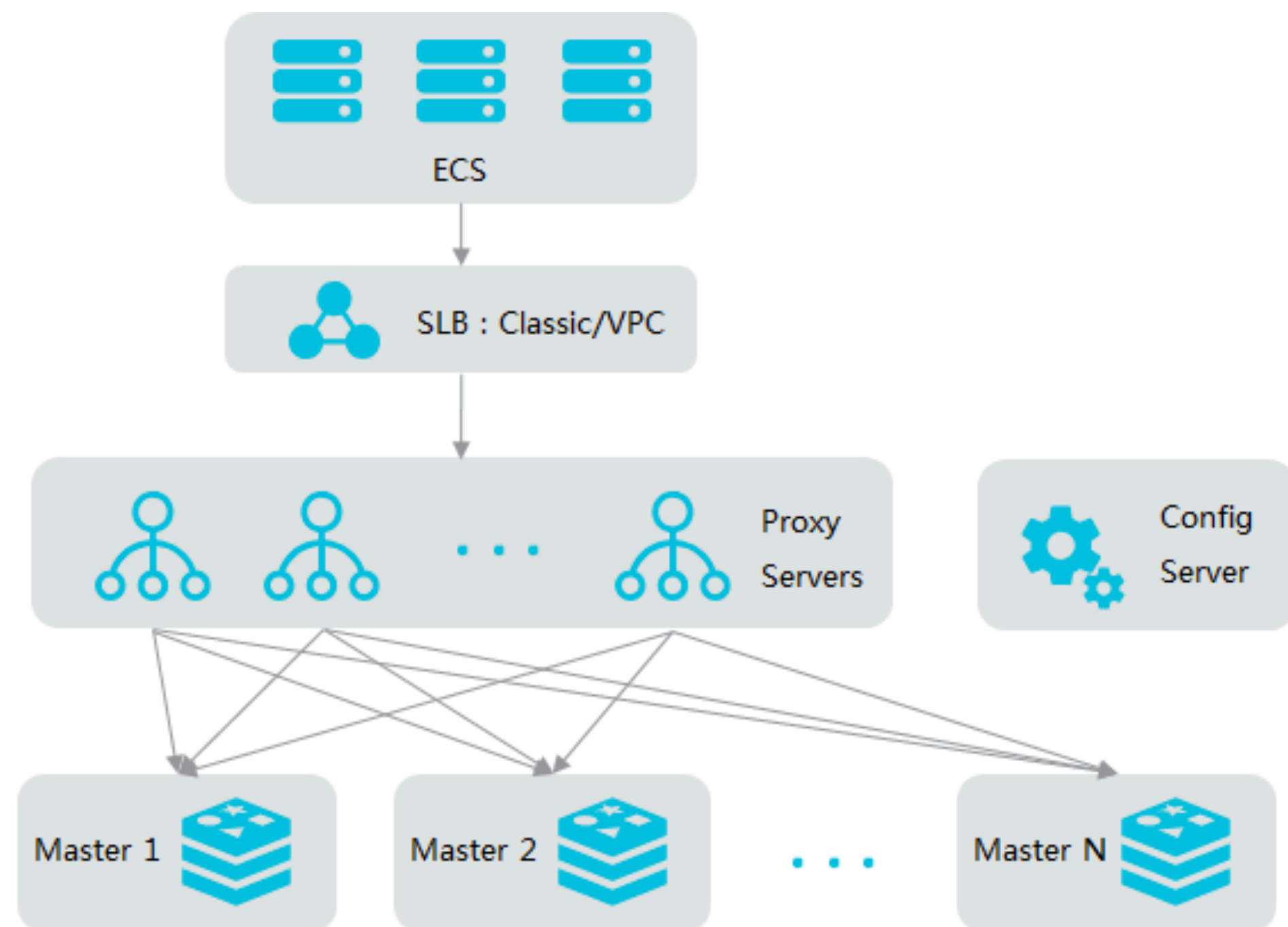
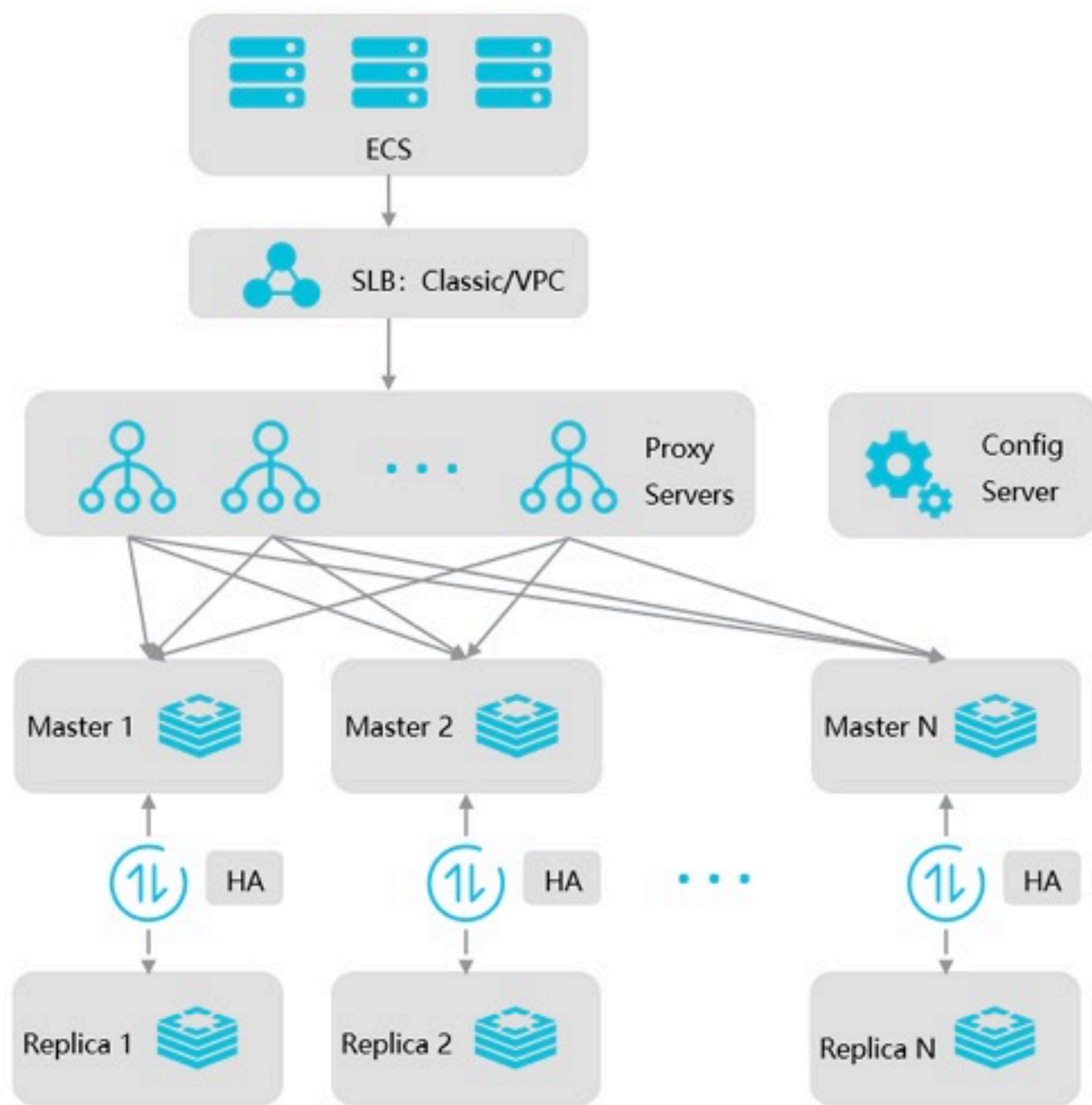
- 非特殊需求不建议使用，QPS压力大的业务建议使用集群版
- 当一个只读节点发生故障时，请求会转发到其他节点；如果所有只读节点均不可用，请求会全部转发到主节点，导致主节点压力过大。
- 只读节点发生异常时，高可用系统会暂停异常节点服务进行重搭恢复。不承诺只读节点的恢复时间指标。
- 只读节点数据旧于主节点且落后时间可能很长。



# 集群版

## 使用场景

- 数据量较大的场景。
- QPS压力较大的场景。
- 吞吐密集型应用场景。



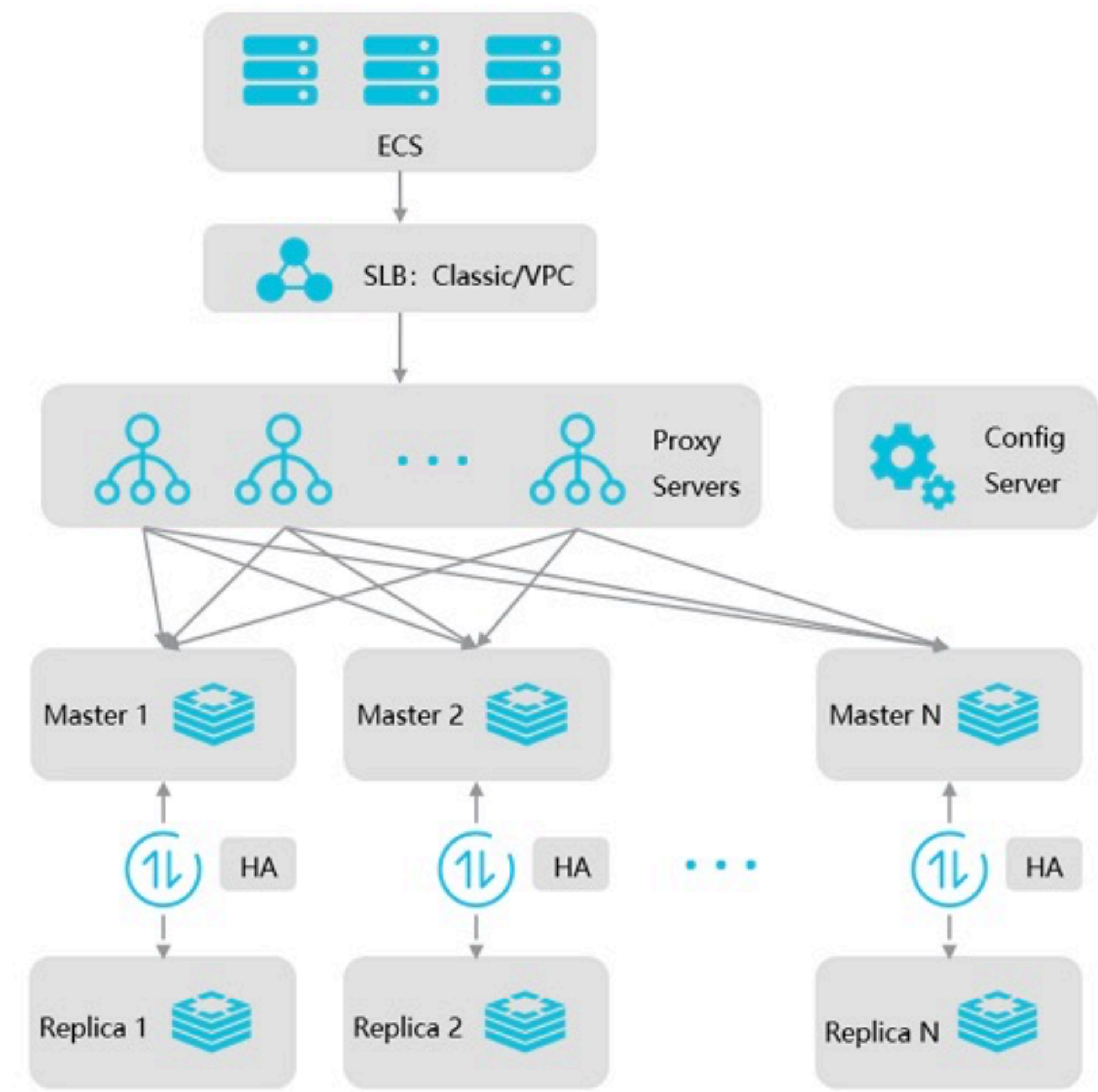


# 集群版-双副本

云数据库Redis版提供双副本集群版实例，可轻松突破Redis自身单线程瓶颈，满足大容量、高性能的业务需求。集群版支持代理和直连两种连接模式。

## 使用场景

- 数据量大：相比Redis标准版，集群版可以有效地扩展存储量，最大可达4098 GB，能有效的满足业务扩展的需求。
- QPS压力较大：标准版Redis无法支撑较大的QPS，需要采用多分片的部署方式来突破Redis单线程的性能瓶颈
- 吞吐密集型应用：相比Redis标准版，集群版的内网吞吐限制相对较低，可以更好地支持热点数据读取、大吞吐类业务。
- 对Redis协议不敏感的应用：集群版对Redis协议支持上相比标准版本有一定的限制。

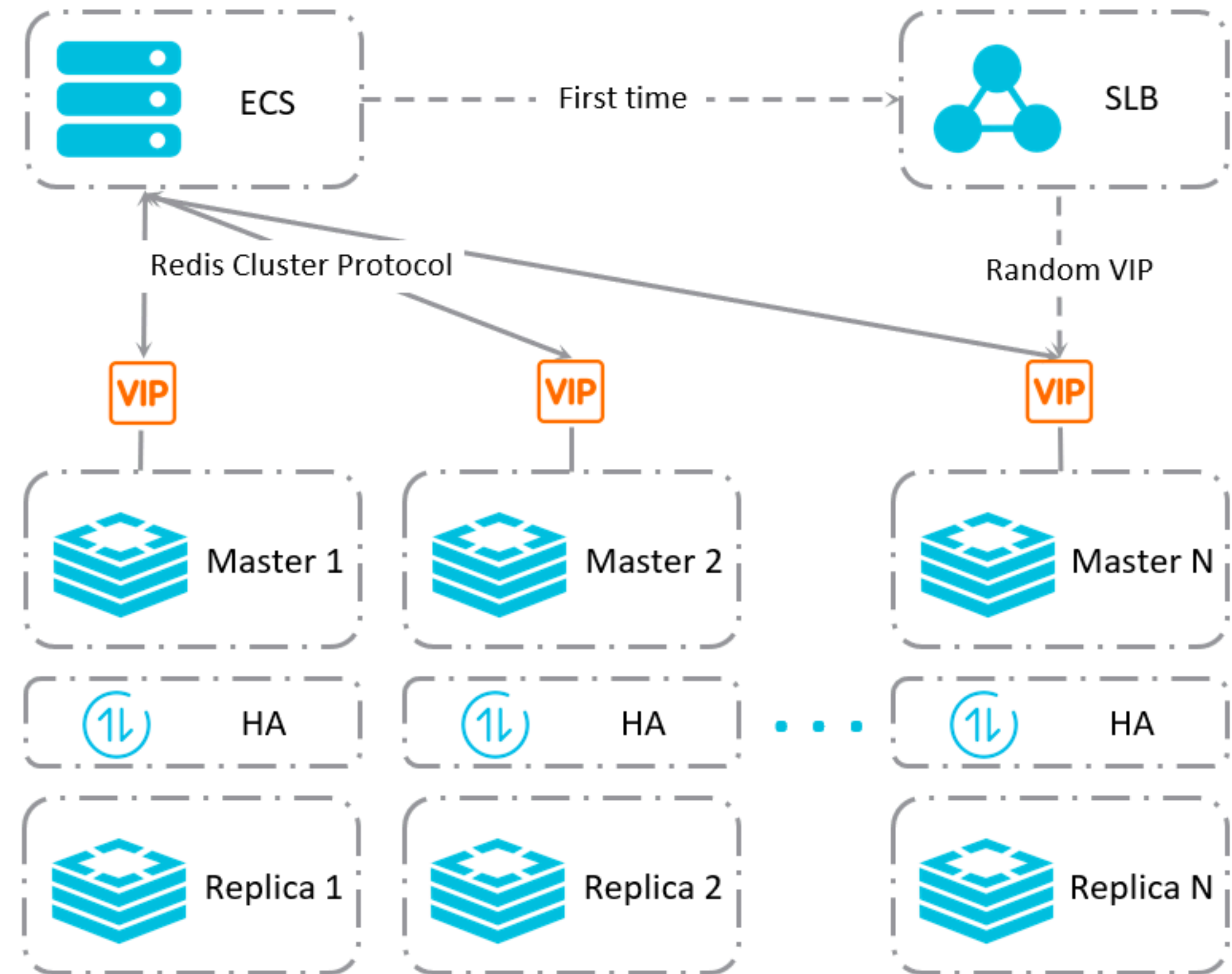


# 集群版-双副本

云数据库Redis版提供双副本集群版实例，可轻松突破Redis自身单线程瓶颈，满足大容量、高性能的业务需求。集群版支持代理和直连两种连接模式。

## 特点

- 代理模式因所有请求都需要通过代理服务器转发，代理模式在降低业务开发难度的同时也会小幅度影响Redis服务的响应速度，如果业务响应速度的要求非常高，可以选择直连模式，绕过代理服务器直连后端数据分片，从而降低网络开销和服务响应时间。



Redis cluster Specification

# 集群版-命令限制

## 不支持的命令

- SWAPDB
- CLIENT ID
- SORT (BY和GET)

## 受限的命令

在集群模式下如果需要执行受限制的命令，需要使用hash tag确保所要操作的key在同个hash slot中，hash tag的详细用法参见[Redis官方文档](#)

## Lua脚本使用限制

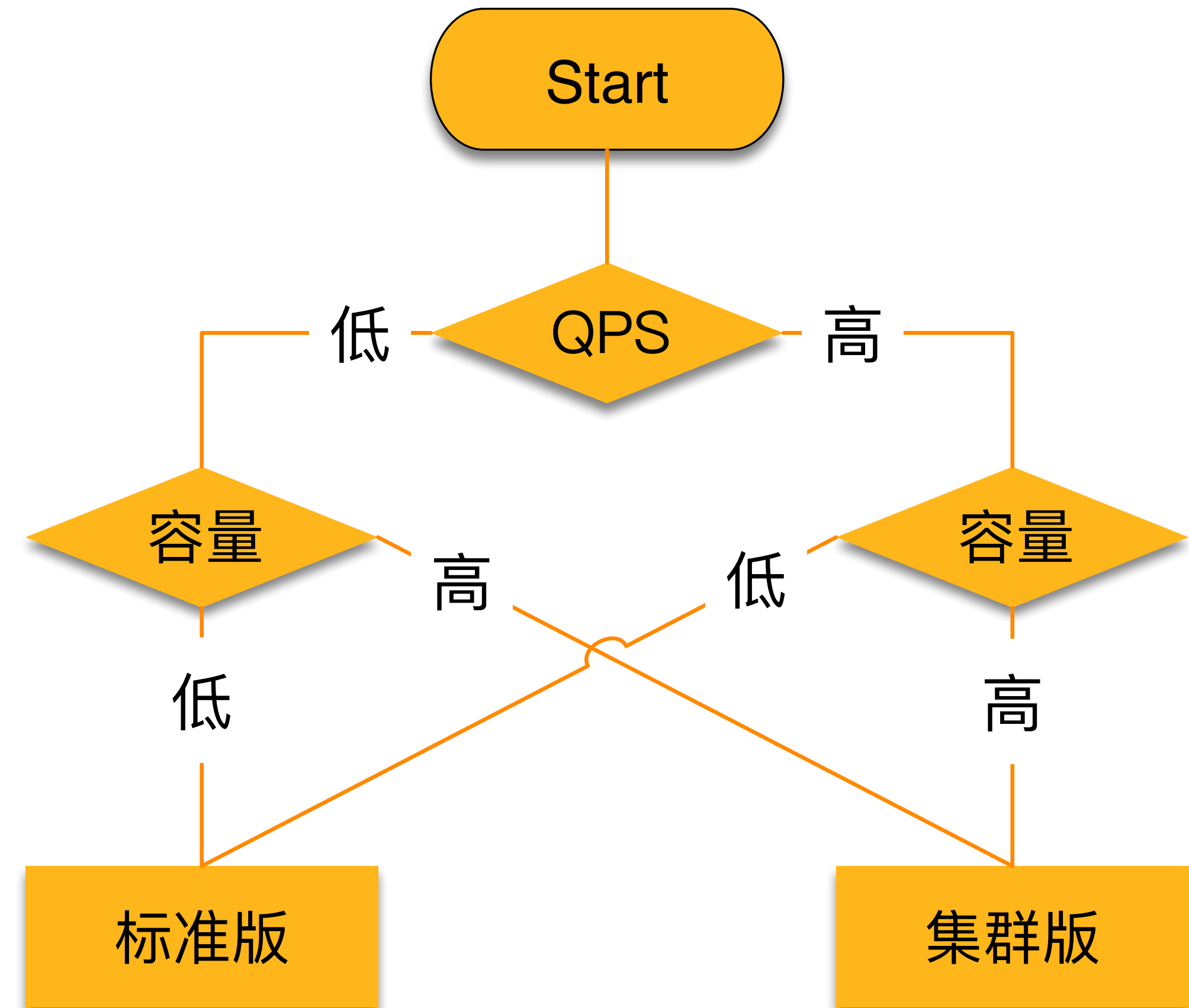
- 所有key都应该由KEYS数组来传递
- 所有key必须在同一个slot上
- 调用必须要带有key
- 不支持发布订阅消息
- 不支持UNPACK函数

## 其他限制

受限命令族	具体命令
HyperLogLog	PFMERGE, PFCOUNT
Keys	RENAME, RENAMENX, SORT
Lists	RPOPLPUSH, BRPOP, BLPOP, BRPOPLPUSH
Scripting	EVAL, EVALSHA, SCRIPT EXISTS, SCRIPT FLUSH, SCRIPT KILL, SCRIPT LOAD
Strings	MSETNX
Transaction	DISCARD, EXEC, MULTI, UNWATCH, WATCH

# 集群模式如何选型

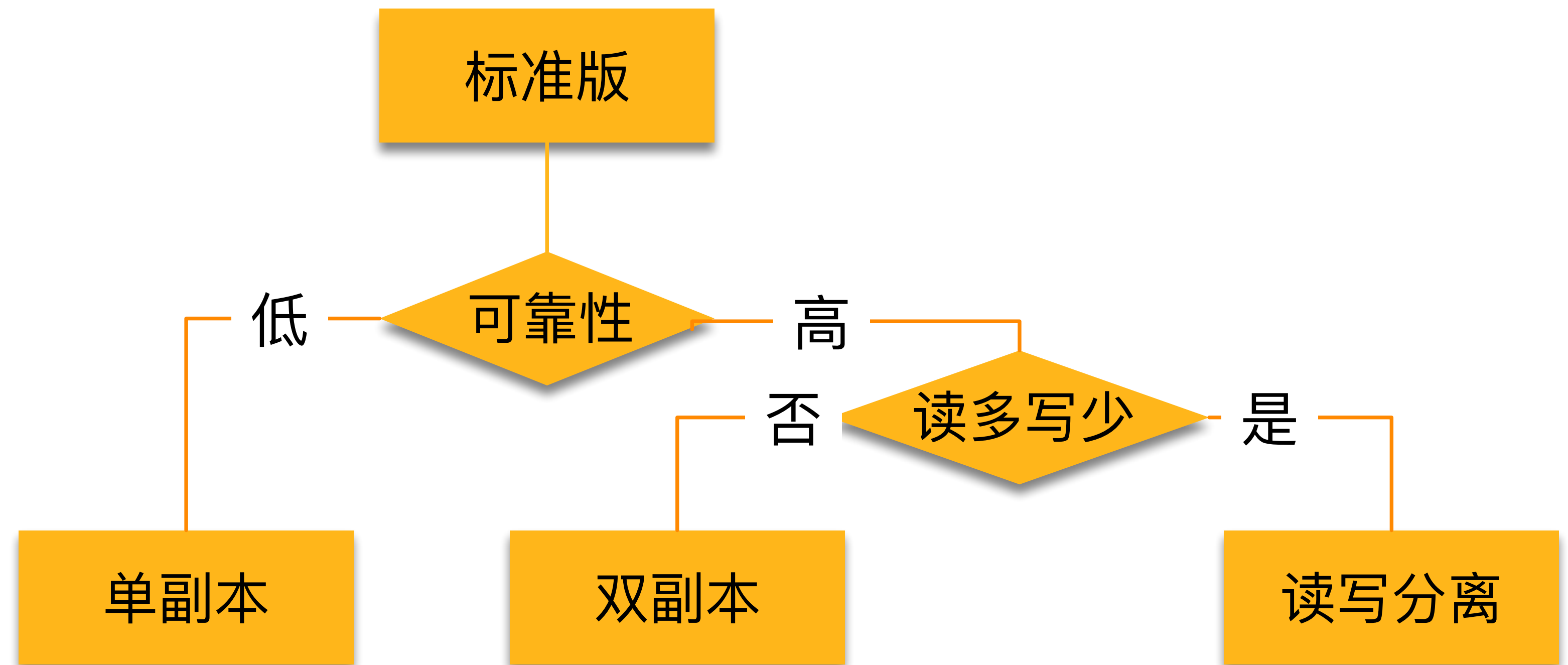
- 评估QPS和容量时一定要为未来留有余量
- 不同架构间存在一定的兼容性问题，业务允许的情况下尽量使用不同架构命令支持集合的交集，以便后续架构切换





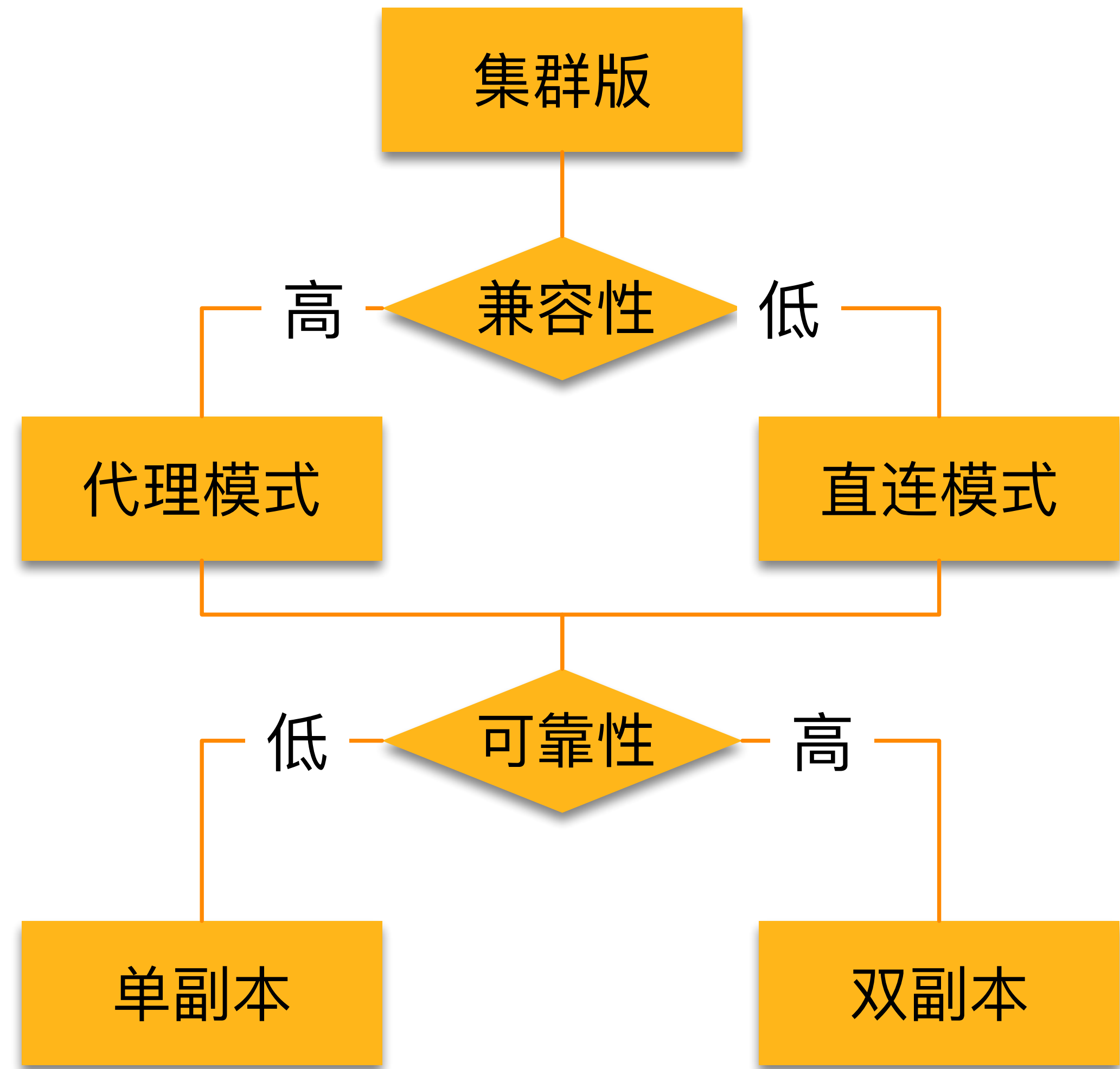
# 集群模式如何选型

- 评估QPS和容量时一定要为未来留有余量
- 不同架构间存在一定的兼容性问题，业务允许的情况下尽量使用不同架构命令支持集合的交集，以便后续架构切换

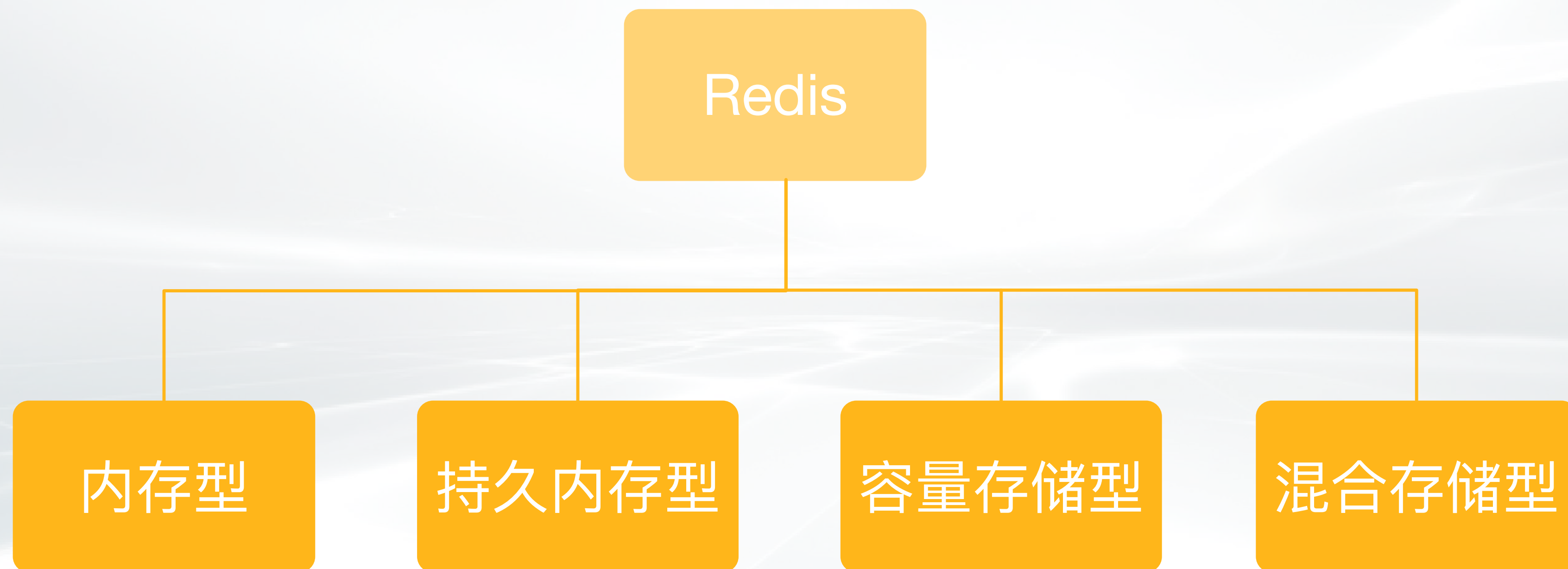


# 集群模式如何选型

- 评估QPS和容量时一定要为未来留有余量
- 不同架构间存在一定的兼容性问题，业务允许的情况下尽量使用不同架构命令支持集合的交集，以便后续架构切换



## 02 Redis存储介质

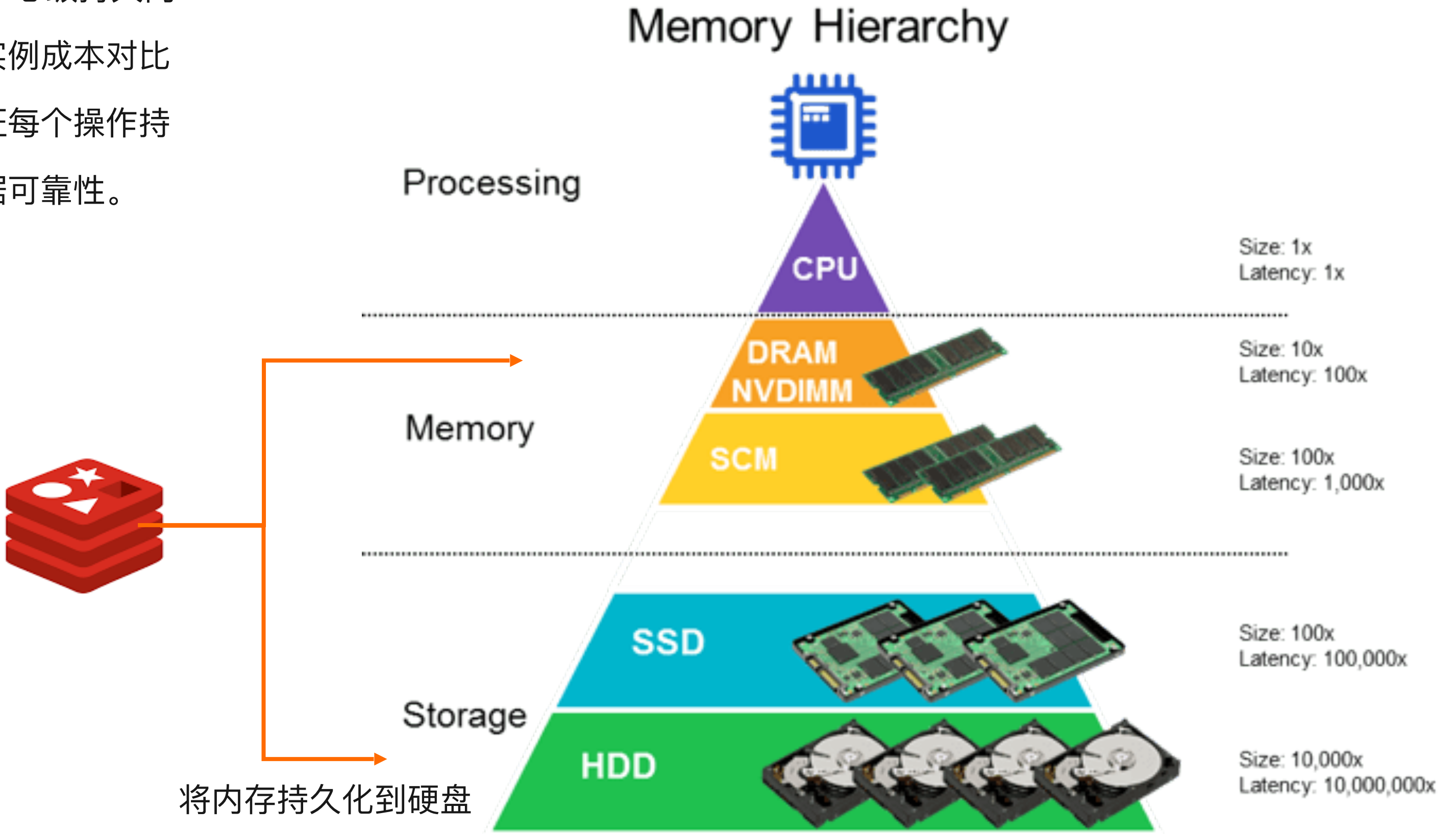


# 持久内存型

Redis企业版持久内存型（简称持久内存型）基于Intel 傲腾™数据中心级持久内存（AEP），为您提供大容量、兼容Redis的内存数据库产品。单实例成本对比Redis社区版最高可降低30%，且数据持久化不依赖传统磁盘，保证每个操作持久化的同时提供近乎Redis社区版的吞吐和延时，极大提升业务数据可靠性。

## 特点

- 超高性价比：相同容量下对比阿里云Redis社区版本，价格降低30%左右
- 大规格优化：解决AOF的造成的性能开销，无需在性能和持久化之间取舍
- 掉电数据不丢失：每个写操作都同步持久化
- 高兼容性：兼容现有阿里云Redis数据库体系



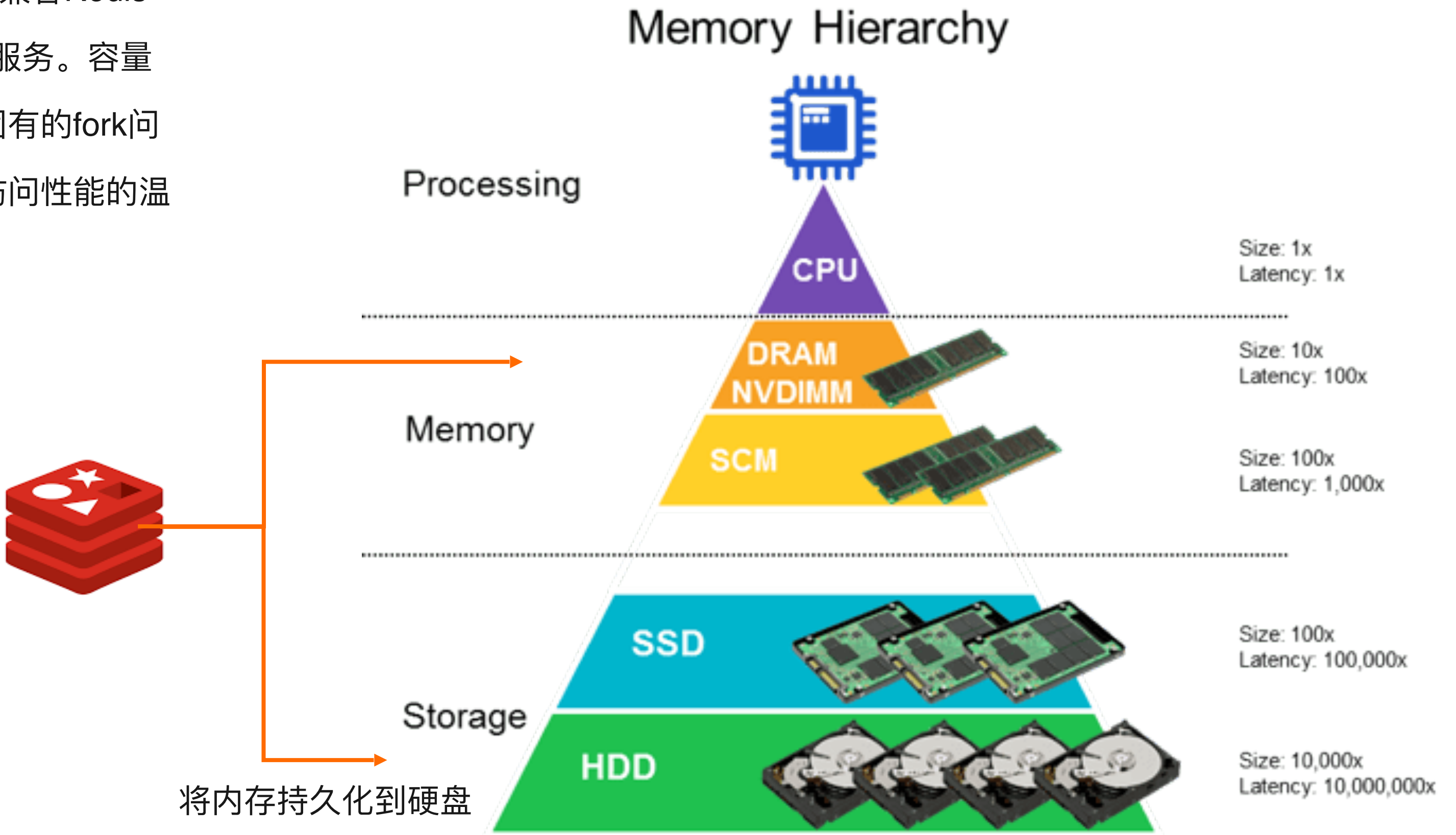


# 容量存储型

Redis企业版容量存储型（简称容量存储型）基于云盘ESSD研发，兼容Redis核心数据结构与接口，可提供大容量、低成本、强持久化的数据库服务。容量存储型在降低成本和提升数据可靠性的同时，也解决了原生Redis固有的fork问题而预留部分内存的问题。适用于兼容Redis、需要大容量且较高访问性能的温冷数据存储场景。

## 特点

- 兼容性：兼容大部分原生Redis命令
- 成本：最低为Redis社区版的15%



# 阿里云集群能力

集群能力对比	开源版Redis	阿里云Redis
高可靠	迁移异常，丢失部分数据，需手动恢复 无中心化控制集群状态收敛慢	数据高可靠，迁移失败自动回滚可重试 中心化控制迅速准确
高可用	高可用心跳探测准确性受慢查询影响，造成故障切换时间过长或者切换不准确	自研探测机制规避慢查询风险导致的误切换 高可用探测更准确 故障切换时间平均8秒，SLA15秒
管控	总机需手动进行重搭 扩缩容需借助额外服务	日常运维由系统自动完成
迁移平滑性	大Key迁移影响服务RT，严重时触发HA Lua脚本丢失 迁移期间多key命令失败	无感迁移 无明显rt上涨 无新增错误
成本	高 集群模式有额外内存开销，大key小value 场景内存容量接近翻倍	低 支持细粒度扩缩容 集群内存使用优化

# 开源Redis集群实现

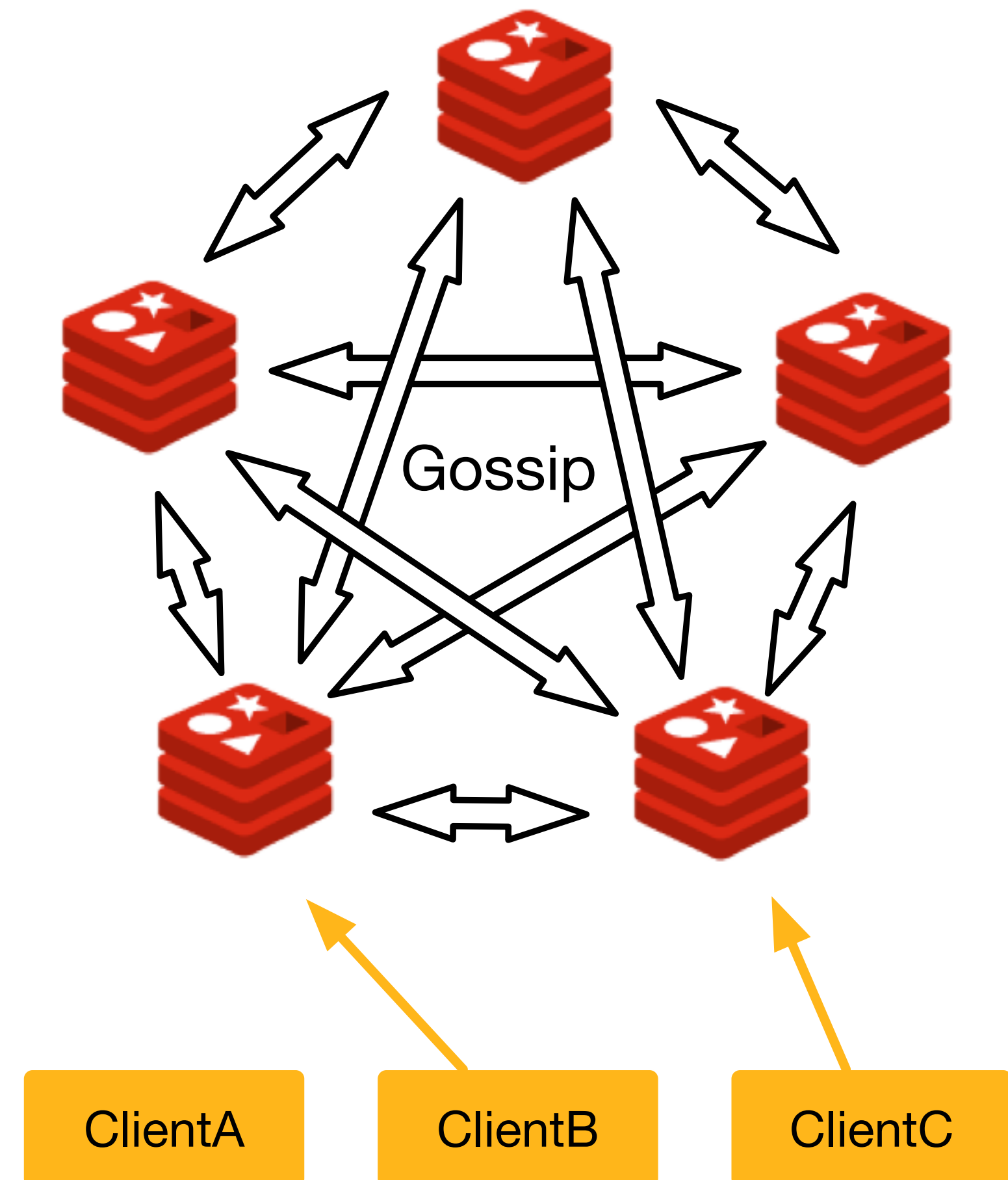
- 所有Redis节点彼此相互心跳探活，使用内部的二进制协议优化传输速度和带宽。
- 节点fail是通过集群中超过半数节点探活协商确定失败
- 节点间数据迁移是按照key的粒度进行的，迁移过程中一个slot的数据会分布在两个节点上

## 优点

- 使用gossip协议无中心控制，无额外控制节点

## 缺点

- 无中心控制，集群状态更新，故障HA慢
- 探活方式单一，受慢查询干扰，容易误切换
- 按key迁移，大key迁移造成服务卡顿
- 迁移异常中断，无法自动恢复
- 迁移期间多key命令失败
- 迁移依赖外部组件



# 阿里云Redis集群实现

- 中心控制节点采用自研的多因子进行准确的探活
- 数据迁移采用slot粒度precopy的方式，迁移快速，异常可回滚

## 优点

- 准确快速的探活，保障服务质量(SLA<15s)
- 同时支持直连模式和代理模式
- 扩缩容业务无感知（大key，多key，Lua），  
不断连接
- 迁移流量在节点间直接传输，不需要外部组件  
中转

