

Redis 高可用架构最佳实践

温国兵

三七互娱

37 Interactive Entertainment

玩心创造世界!

FUN CREATES THE WORLD



❖ 关于我

❖ Redis 高可用架构

❖ 架构优劣

❖ 最佳实践



温国兵，CSDN □博客专家，InfoQ □社区编辑。曾任职于酷狗音乐，现为三七互娱 **DBA**。目前主要关注领域：数据库自动化运维、高可用架构设计、数据库安全、海量数据解决方案、以及开源技术在互联网中的应用。



- Redis □ Sentinel □ 集群 + □ 内网 DNS □ + □ 自定义脚本
- Redis □ Sentinel □ 集群 + □ VIP □ + □ 自定义脚本
- 封装客户端直连 Redis □ Sentinel □ 端口
 - JedisSentinelPool, 适合 Java
 - PHP □ 基于 phredis □ 自行封装
- Redis □ Sentinel □ 集群 + □ Keepalived/Haproxy
- Redis □ M/S □ + □ Keepalived
- Redis □ Cluster
- Twemproxy
- Codis



Redis 高可用架构



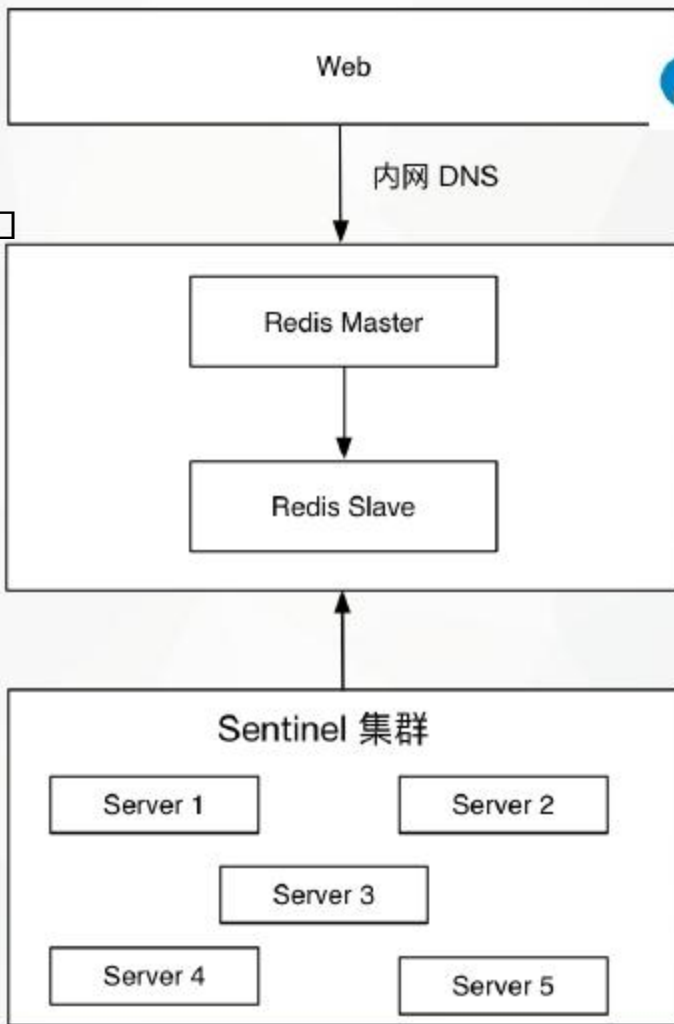
Redis Sentinel 集群 + 内网 DNS + 自定义脚本

优点:

- 秒级切换
- 脚本自定义, 架构可控
- 对应用透明

缺点:

- 维护成本略高
- 依赖 DNS, 存在解析延时
- Sentinel 模式存在短时间的服务不可用





Redis 高可用架构



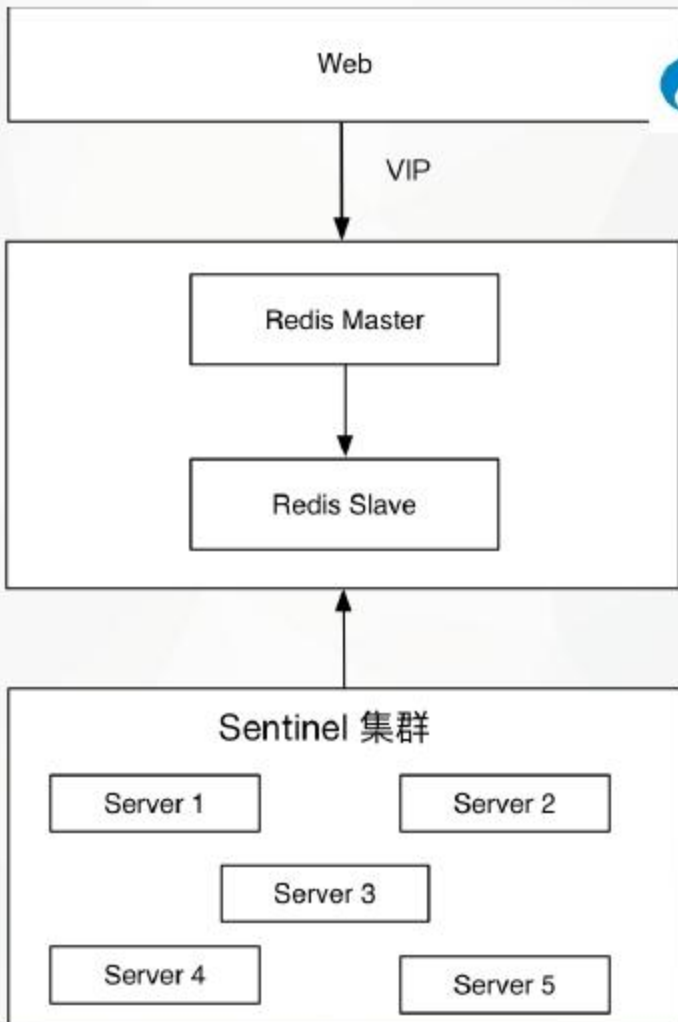
Redis Sentinel 集群 + VIP + 自定义脚本

优点:

- 秒级切换
- 脚本自定义, 架构可控
- 对应用透明

缺点:

- 维护成本略高
- Sentinel 模式存在短时间的服务不可用





Redis 高可用架构

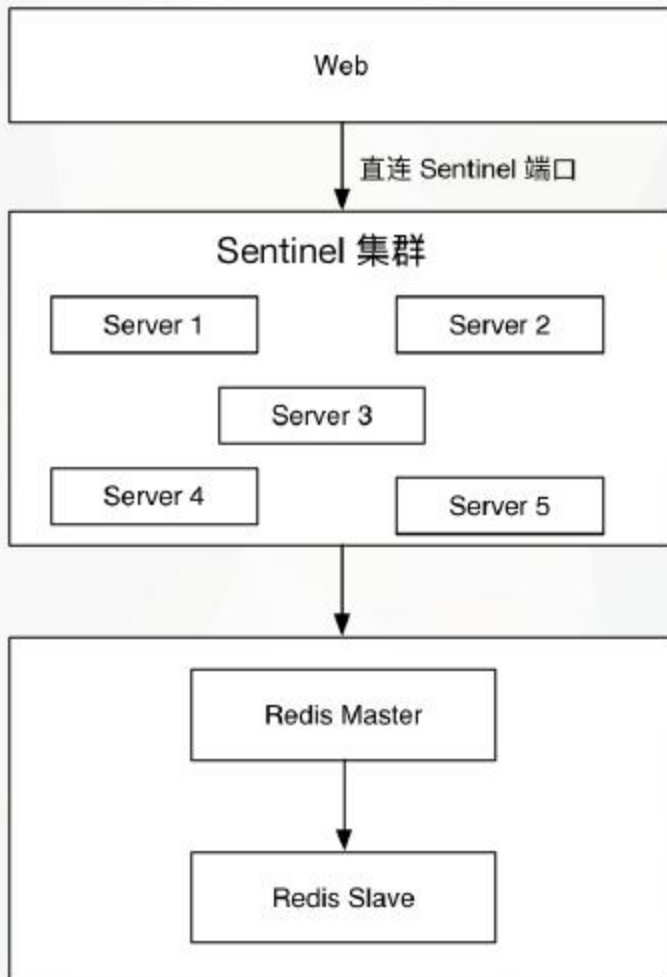
封装客户端直连 Redis Sentinel 端口

优点:

- 服务探测故障及时
- DBA 维护成本低

缺点:

- 依赖客户端支持 Sentinel
- Sentinel 服务器需要开放访问权限
- 对应用有侵入性





Redis 高可用架构

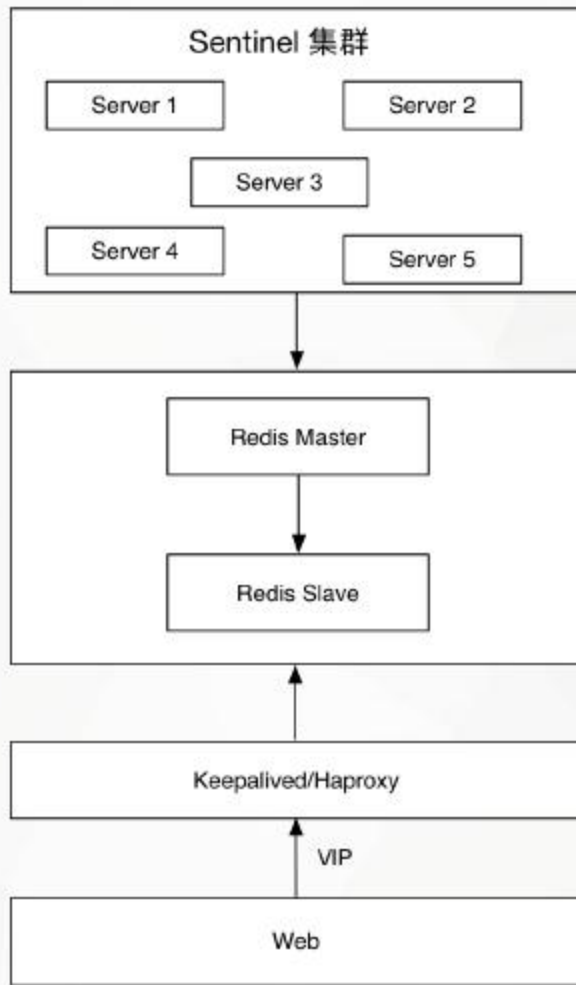
Redis Sentinel 集群 + Keepalived/Haproxy

优点:

- 秒级切换
- 对应用透明

缺点:

- 维护成本高
- 存在脑裂
- Sentinel 模式存在短时间的服务不可用





Redis 高可用架构

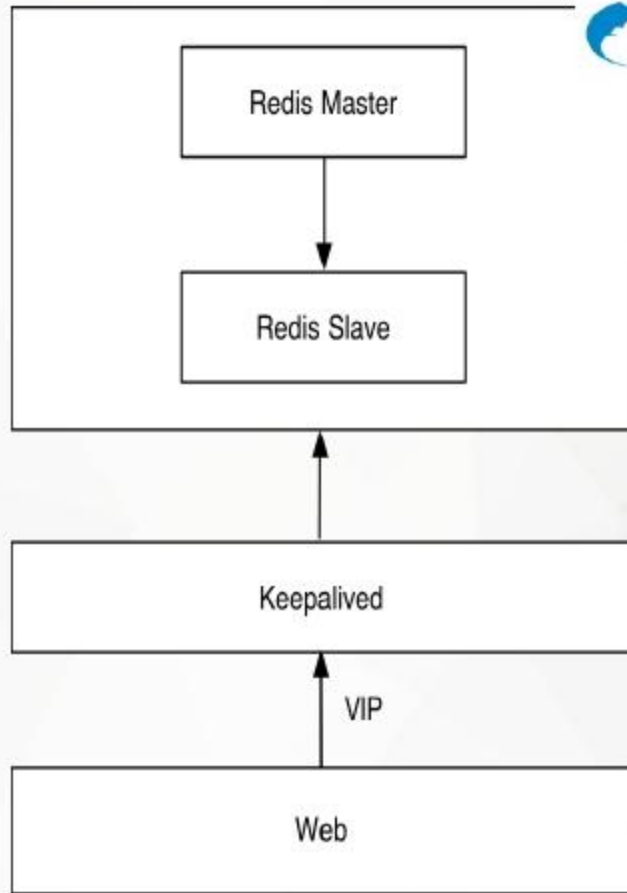
Redis M/S + Keepalived

优点:

- 秒级切换
- 对应用透明
- 部署简单, 维护成本低

缺点:

- 需要脚本实现切换功能
- 存在脑裂





Running Redis Cluster

Redis Cluster

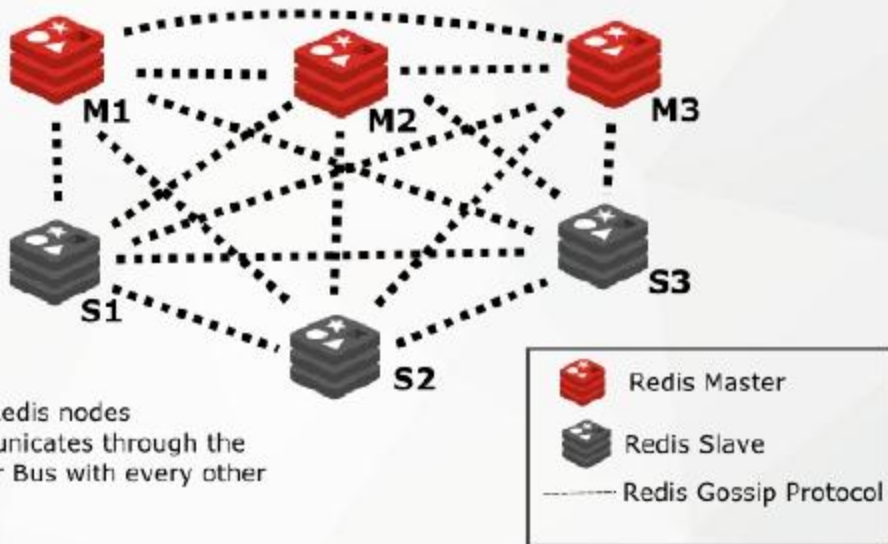
优点:

- 组件 **all-in-box**, 部署简单, 节约机器资源
- 性能比 **proxy** 模式好
- 自动故障转移、**Slot** 迁移中数据可用
- 官方原生集群方案, 更新与支持有保障

缺点:

- 架构比较新, 最佳实践较少
- 多键操作支持有限 (驱动可以曲线救国)
- 为了性能提升, 客户端需要缓存路由表信息
- 节点发现、**reshard** 操作不够自动化

In a minimal Redis Cluster made up 3 masters nodes each with a single slave node, each master node is assigned a hash slot range between 0 and 16,384. Both Master and Slaves Run two TCP services, the first is for normal RESP messages and the second is Cluster Bus that communicates with the Redis Cluster Gossip protocol.





Redis 高可用架构

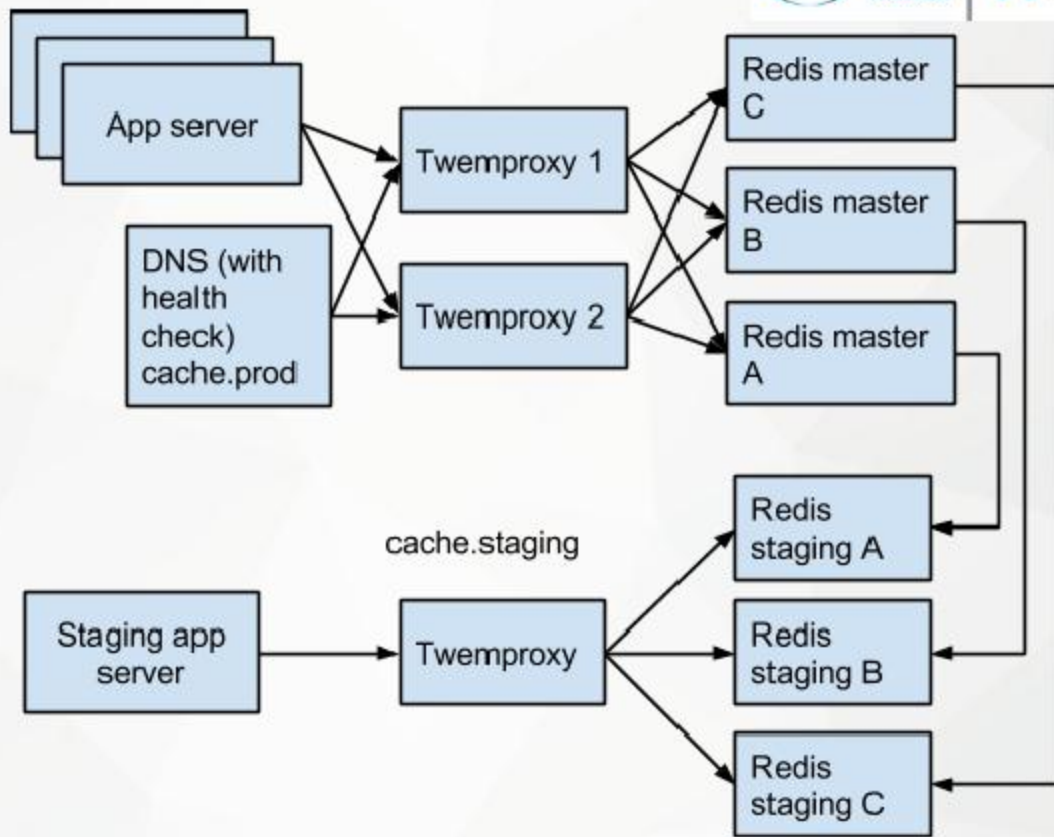
Twemproxy

优点:

- 开发简单, 对应用几乎透明
- 历史悠久, 方案成熟

缺点:

- 代理影响性能
- 会有节点性能瓶颈
- **Redis** 扩容非常麻烦
- **Twitter** 内部已放弃使用该方案, 新使用的架构未开源





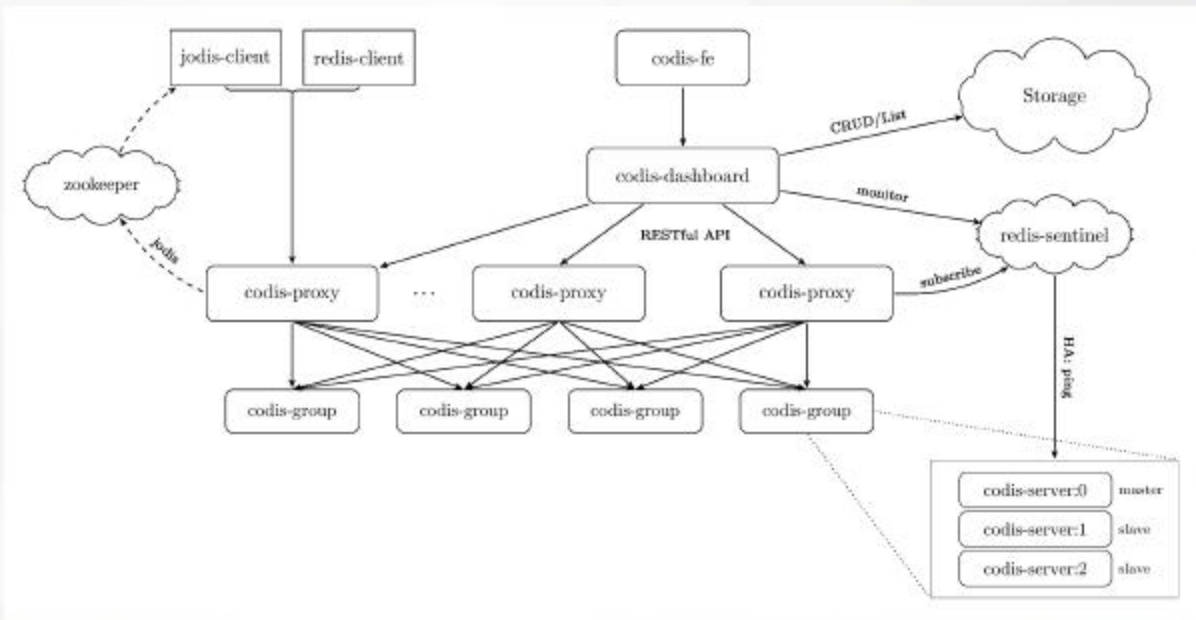
Codis

优点:

- 开发简单，对应用几乎透明
- 性能比 Twemproxy 好
- 有图形化界面，扩容容易，运维方便

缺点:

- 代理依旧影响性能
- 组件过多，需要很多机器资源
- 修改了 Redis 代码，导致和官方无法同步，新特性跟进缓慢
- 开发团队准备主推基于 redis 改造的 reborndb



From: <https://github.com/CodisLabs/codis>



最佳实践



所谓的最佳实践，都是最适合具体场景的实践。

主推以下方案：

- Redis Sentinel 集群 + 内网 DNS + 自定义脚本
- Redis Sentinel 集群 + VIP + 自定义脚本



- **Redis Sentinel** 集群建议使用 ≥ 5 台机器
- 不同的大业务可以使用一套 **Redis Sentinel** 集群，代理该业务下的所有端口
- 根据不同的业务划分好 **Redis** 端口范围
- 自定义脚本建议采用 **Python** 实现，扩展便利
- 自定义脚本需要注意判断当前的 **Sentinel** 角色
- 自定义脚本传入参数：`<service_name> <role> <comment> <from_ip> <from_port> <to_ip> <to_port>`
- 自定义脚本需要远程 **ssh** 操作机器，建议使用 **paramiko** 库，避免重复建立 **SSH** 连接，消耗时间
- 加速 **SSH** 连接，建议关闭以下两个参数
 - **UseDNS no**
 - **GSSAPIAuthentication no**
- 微信或者邮件告警，建议 **fork** 一个进程，避免主进程阻塞
- 自动切换和故障切换，所有操作建议在 **15s** 以内完成



THANKS!

—— 谢谢观看！

三七互娱
37 Interactive Entertainment

玩心创造世界
FUN CREATES THE WORLD