

SOFARegistry 服务注册中心开源

尚 彧

蚂蚁金服 服务注册中心开源负责人

目录

contents

- 概念

- 作用
- 能力

- 演变

- 数据结构
- 存储
- 分层
- 高可用
- 数据同步
- 元数据管理

- 开源

- 优势
- 现状

- 将来

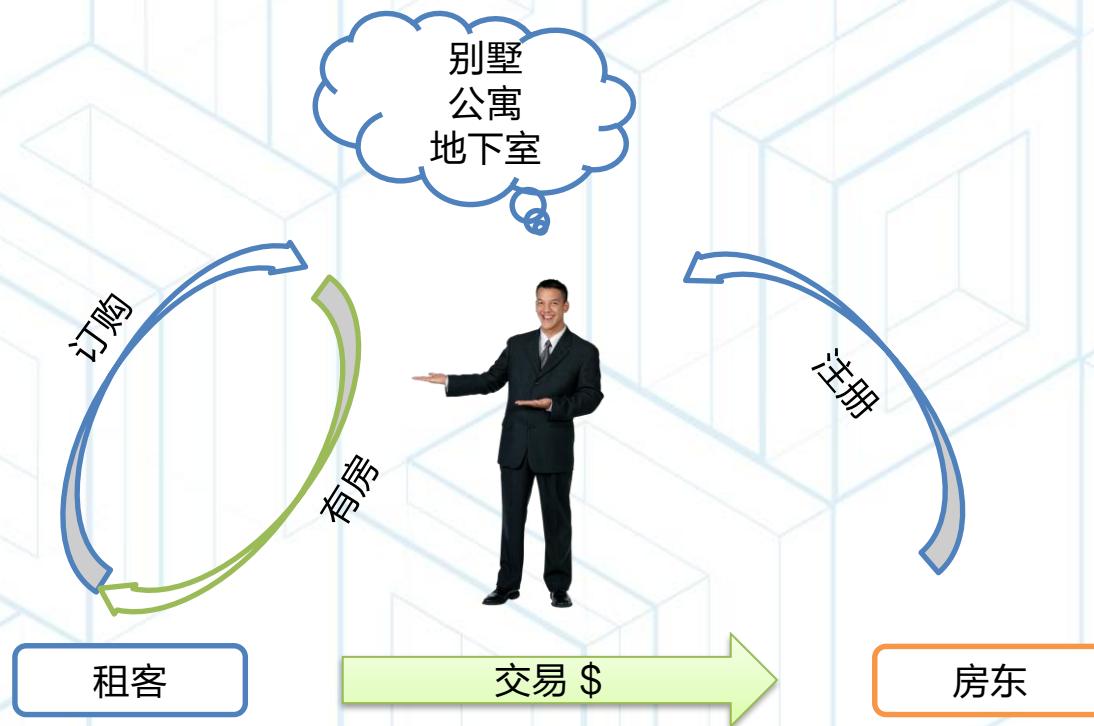
- 规划

SOFARegistry 概念

服务注册中心是什么？

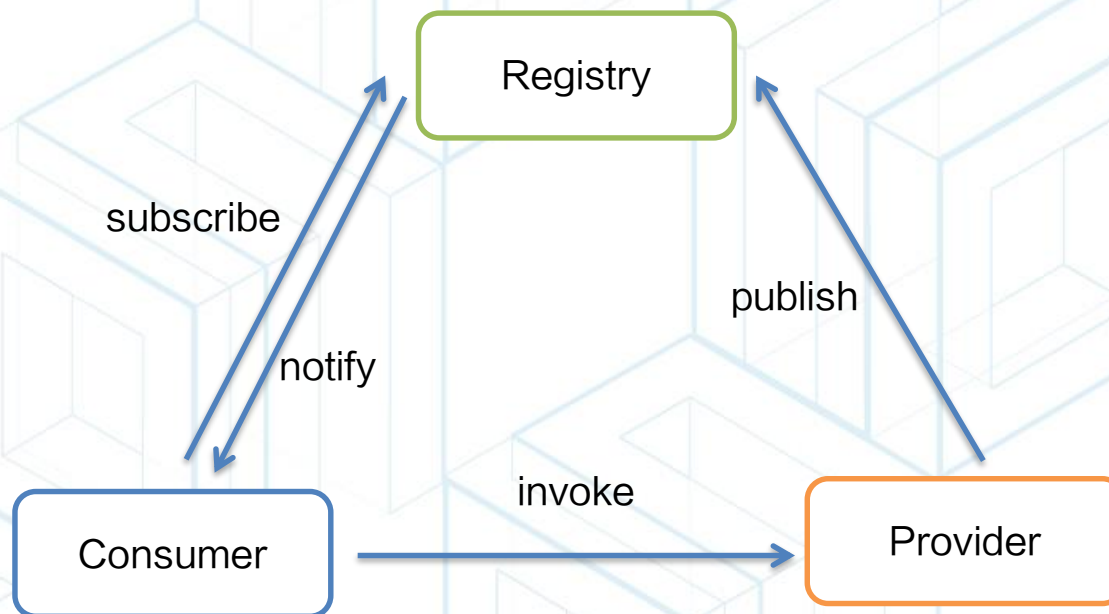
作为中间件的基础设施提供什么能力？

类似房屋租赁中介，持有房屋供求关系关联需求双方达成租赁交易。



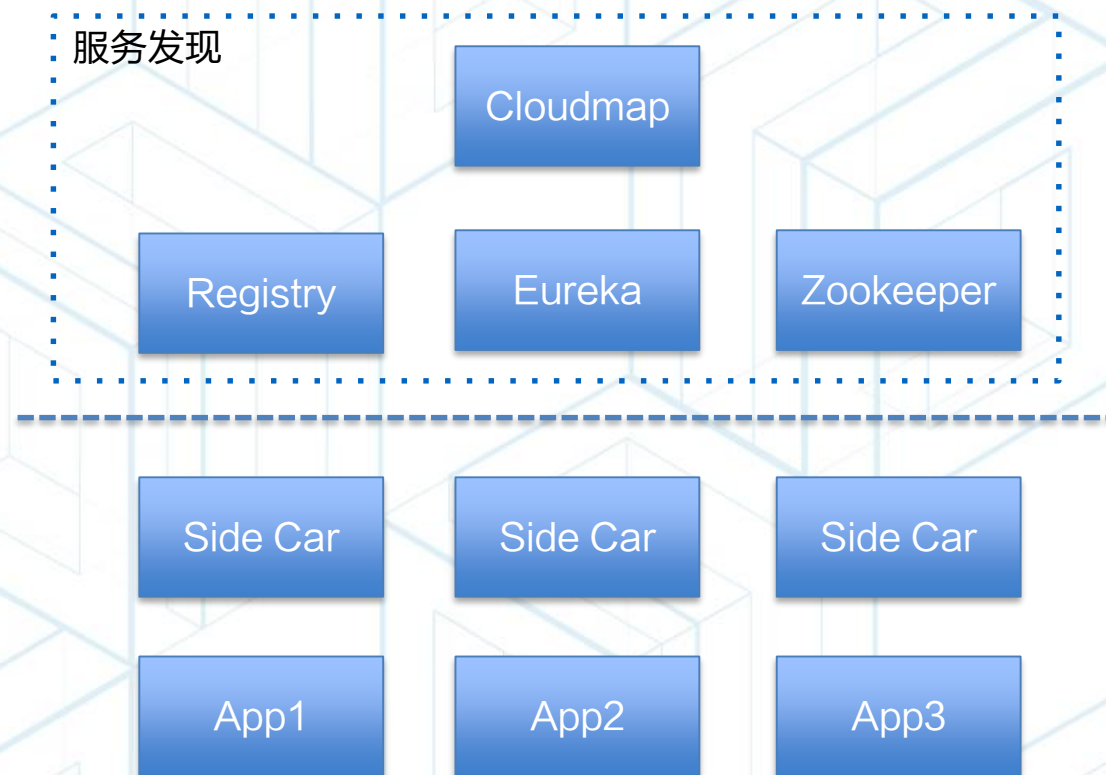
SOFARegistry 概念

服务供求关系的持有者，提供把服务提供方位置信息推送给所有消费方，使之成功寻址完成调用。



SOFARegistry 服务架构中位置

服务架构持续演进，从早期 SOA 架构到新的云原生微服务架构，无论是服务粒度的划分，还是服务基础通信能力下沉到 Mesh，但服务注册发现能力 **不可或缺**。

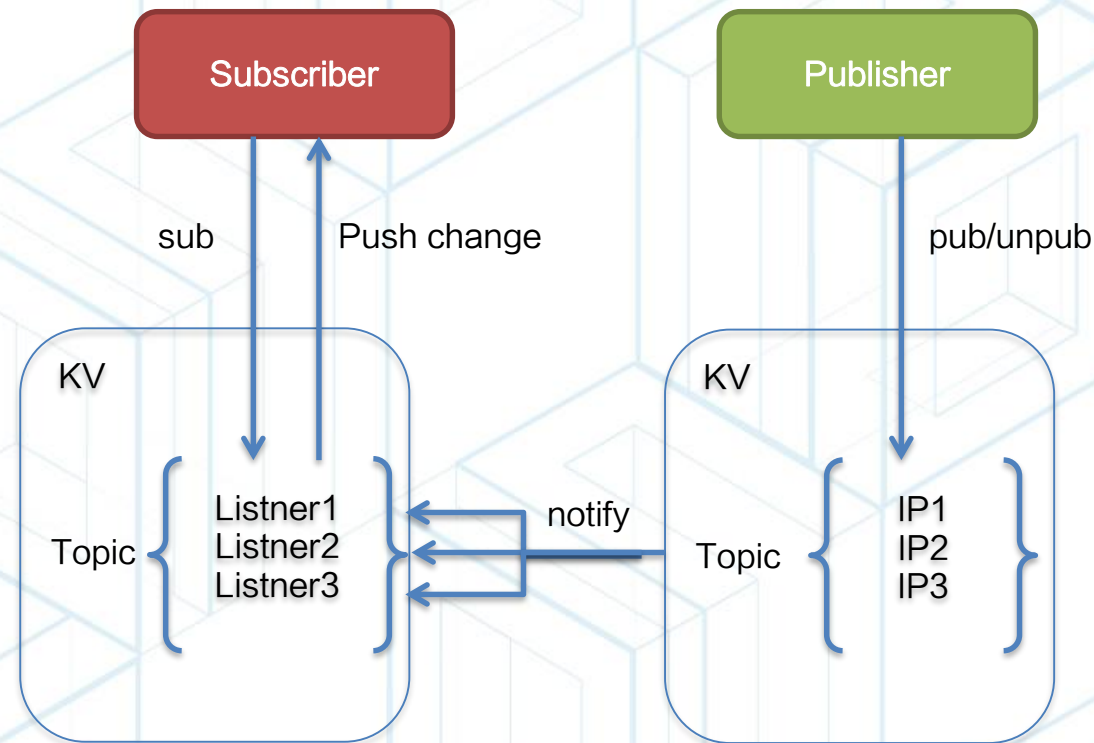


SOFARegistry 能力

- ★ 服务注册、服务订阅、存储
- ★ 及时准确通知服务需求方注册的服务信息变化、推送

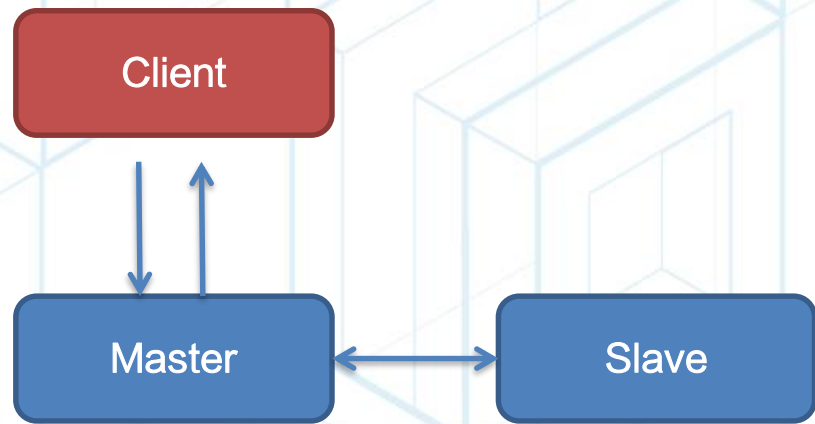
SOFARegistry 数据模型

- 作为开发者一定好奇这些能力怎么实现的，实际上注册中心是一个**标准的监听者模型加 KV 存储**。
- 这个其实从数据模型上理解并不复杂，最基本本地开发模式就是基于这个实现的 Simply Registry。



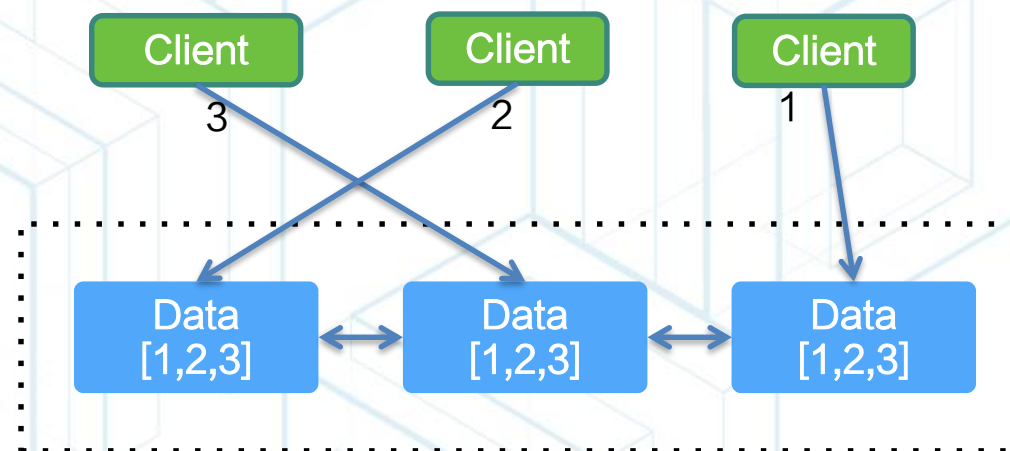
SOFARegistry 存储

- 既然需要**存储**，就要确定怎么存？存哪儿？存了会不会丢？
- 怎么存是由存什么决定的，存储数据结构简单 KV，首先就淘汰关系数据库，由于变化频繁需要高的时效性，需要内存数据无需持久化，存储功能相对简单单一没必要再依赖三方的内存库产品。
- 最初的部署形式简单-主备方式。



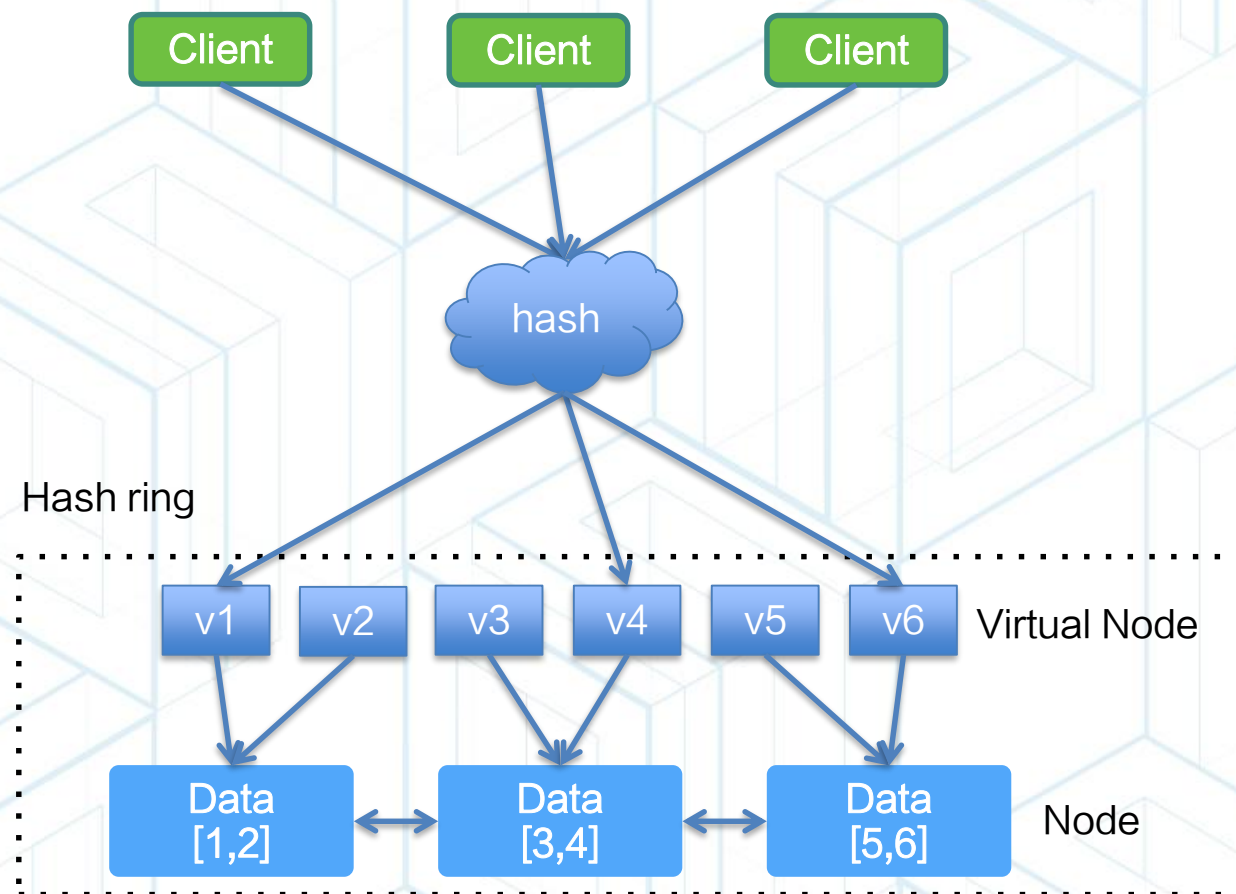
SOFARegistry 存储

- 随着业务增长主备模式很快就跟不上需求，必须不断扩容，变为**集群强一致模式**。
- 数据规模小的情况下，每个节点数据一致，节点相对状态相同，目前很多 CP 系统比如 ZooKeeper 还是这样方式，如果容量不是问题可以。



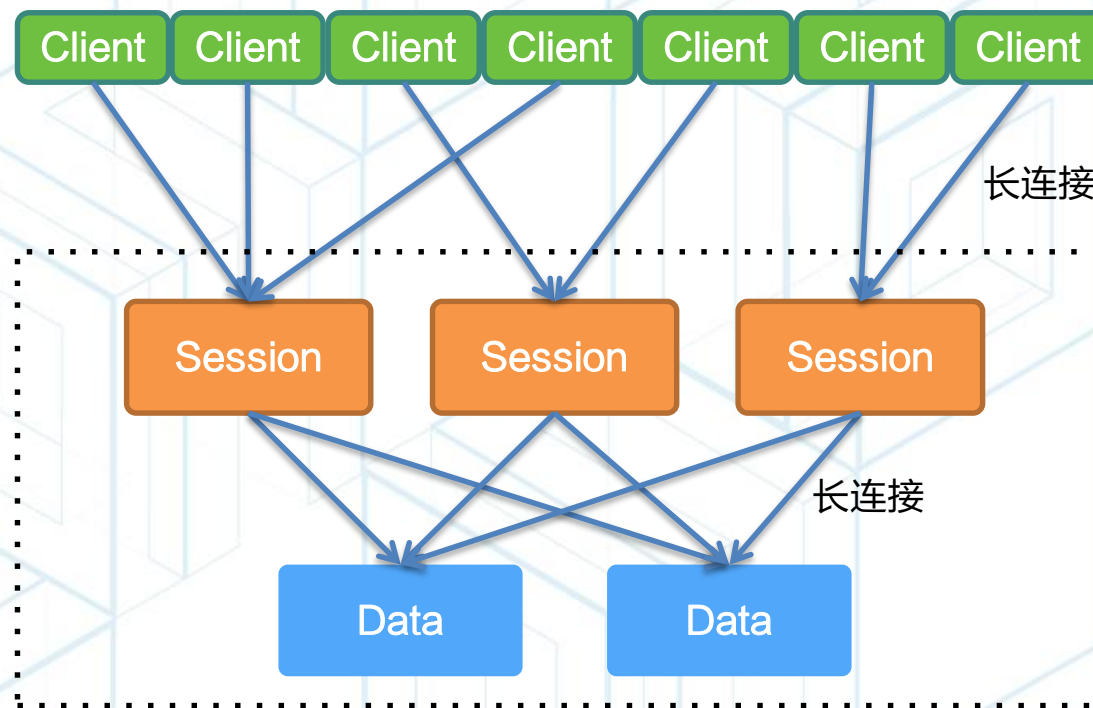
SOFARegistry 存储

- 业务持续增长，数据量扩展继续，强一致模型效率越来越低。
- 想要数据可以理论上无限扩展，必须进行数据分片存储，所以每个节点按照**一致性 hash** 算法形成 hash 环存储。



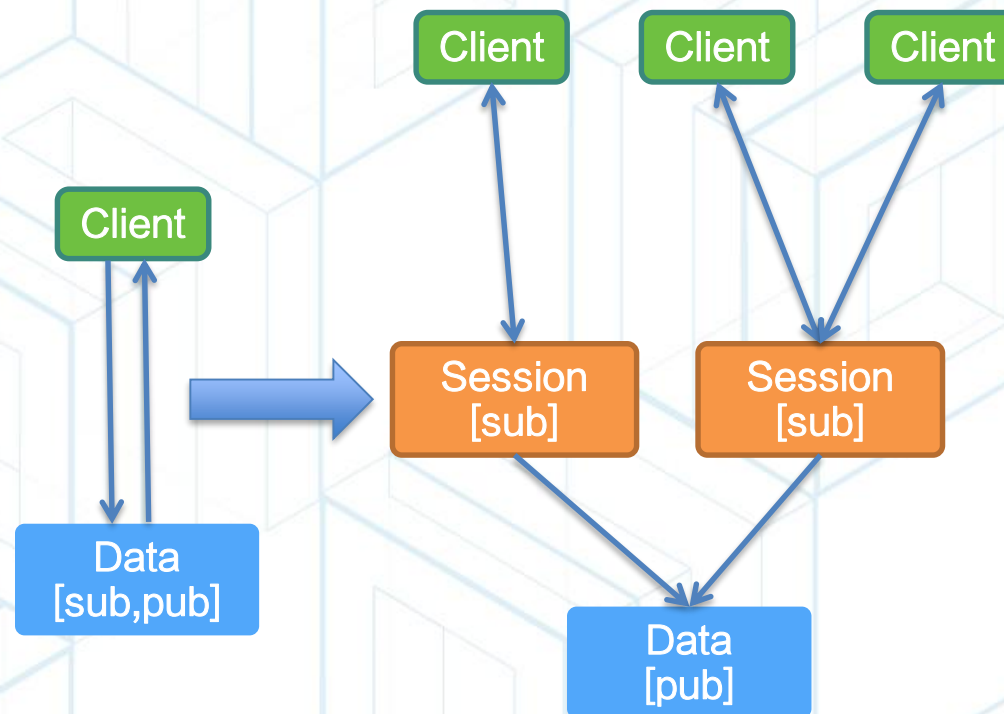
SOFARegistry 分层

- 运作实际过程通常数据没有达到节点容量上限，客户端连接已经不够用，客户端不断增长的情况下继续水平扩展，需要分离出一层承载大量链接。
- 此外承载链接的层因为不是服务数据存储节点，只是链接数据通道，需要无状态随时断链替换，客户端可以保持立即选择新节点进行注册订阅。



SOFARegistry 分层

- 分出 Session 层承载链接，增加了整体框架复杂性。
- 订阅关系和直接连接目的客户端在一起推送效率最高，所以分离订阅数据存储于 Session 上。



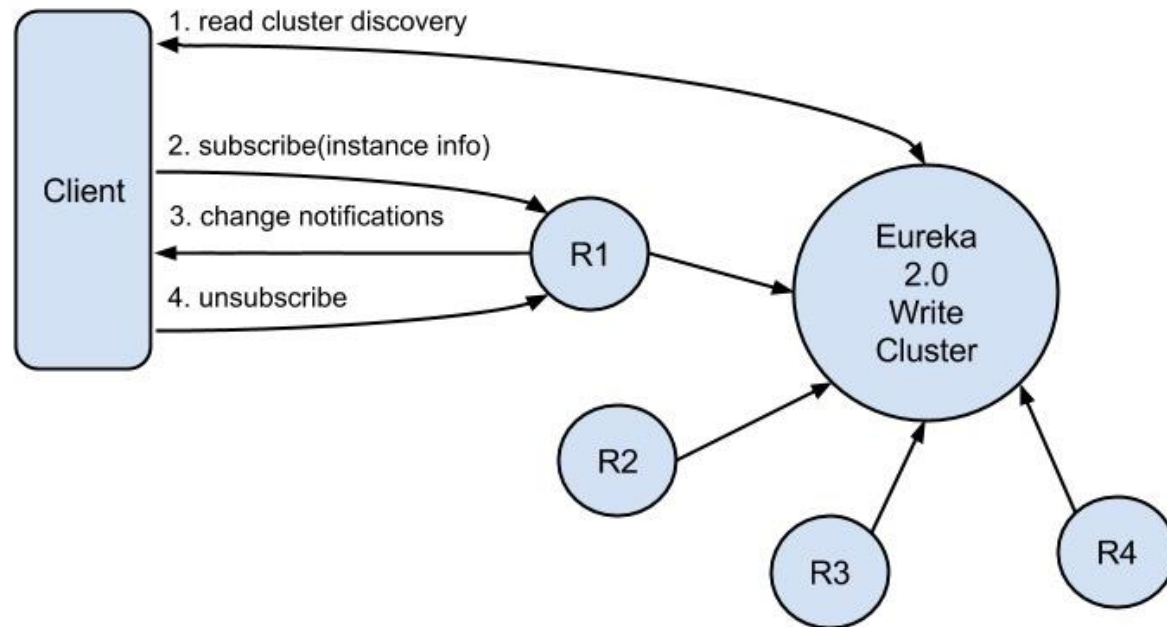
SOFARegistry 分层

分离出两层角色：

- Data Server 存储服务发布数据 写入
- Session Server 推送服务数据信息变化给订阅客户端，承载链接 读取

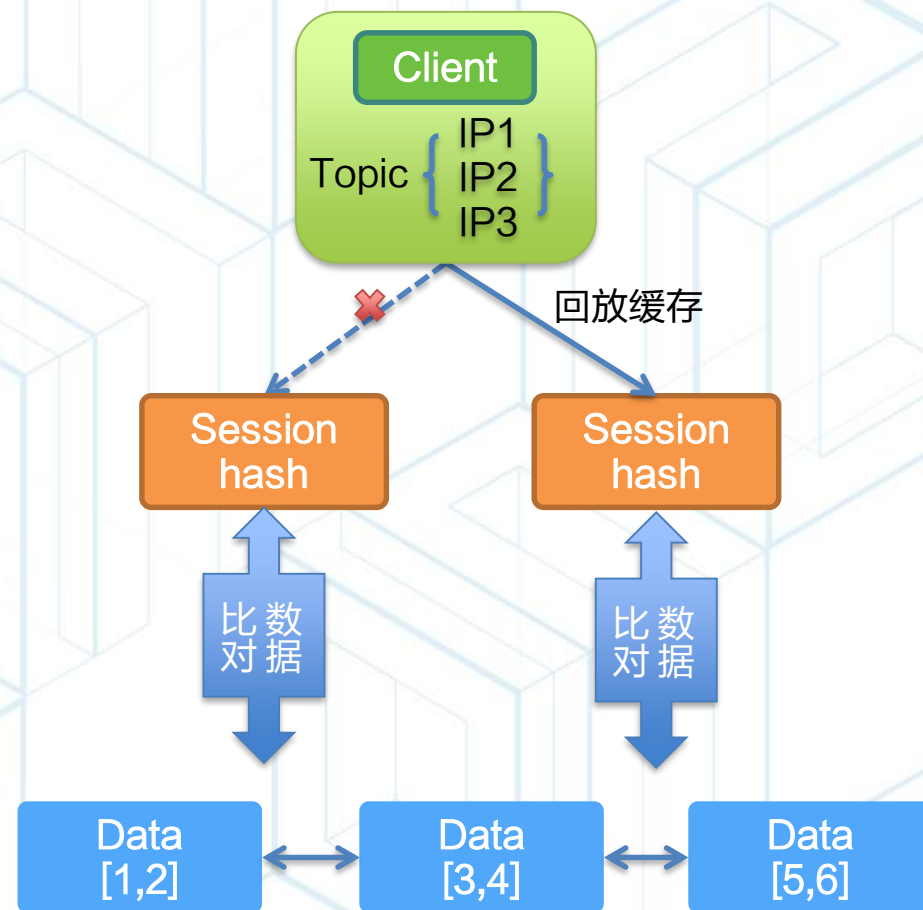
Registry discovery

Eureka clients can subscribe to interest sets on Eureka servers. After a successful subscription, all changes to subscribed interests are pushed by the server to the client. Eureka servers responsible for handling these actions form the read cluster.



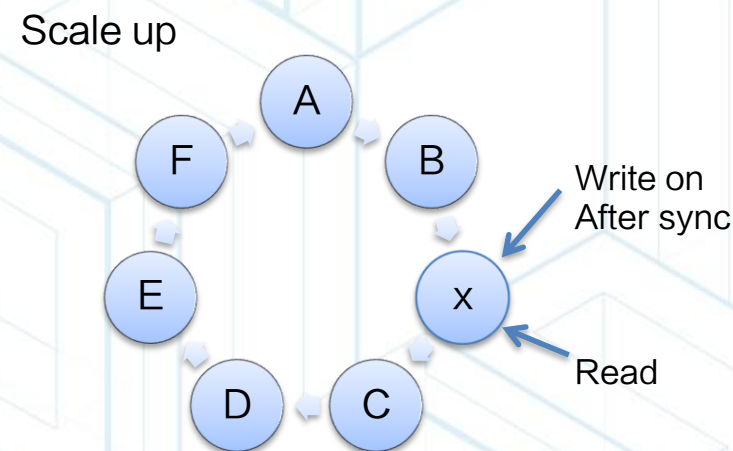
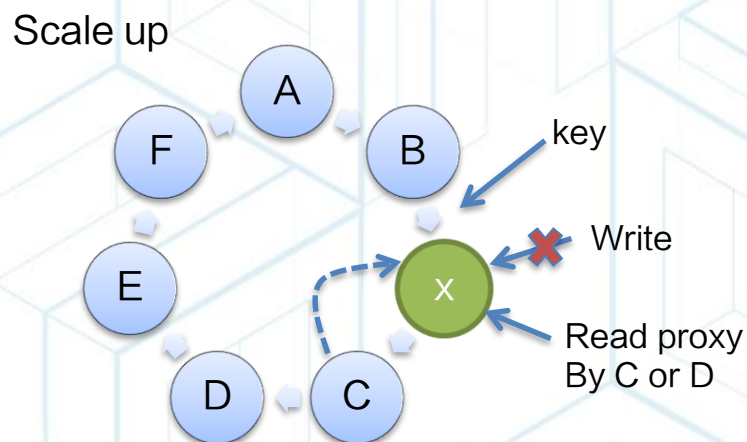
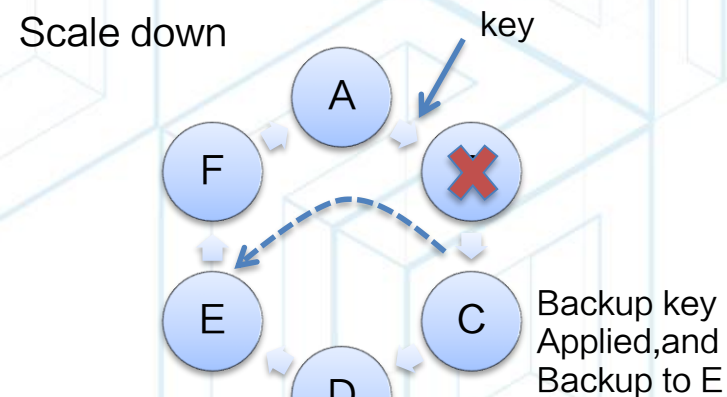
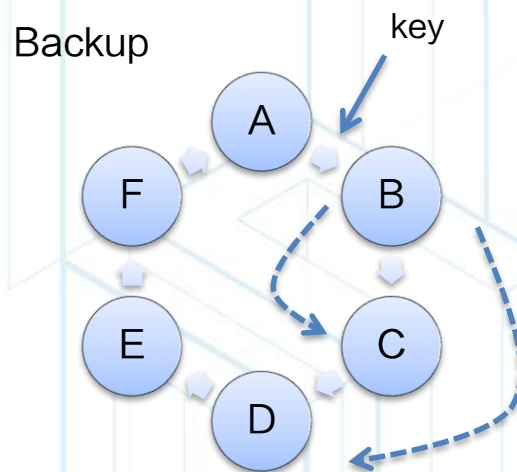
SOFARegistry 高可用

- 数据备份，采用逐级缓存数据回放模式，Client 对于订阅和发布服务数据进行本地缓存，新连接 Session 后回放数据到新节点，再发布到 Data。
- Session 存储客户端 Pub 数据，定期通过数据比对保持和 Data 一致。



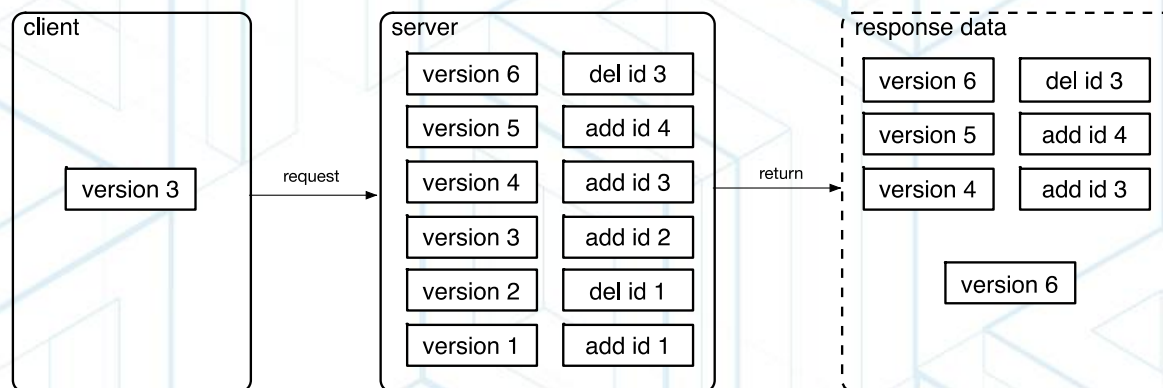
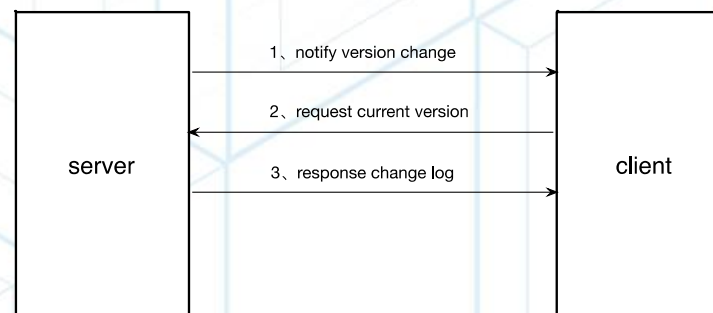
SOFARegistry 高可用

- 除了逐级保证，Data 自身的数据保持**多副本**，副本数可配置。
- Data **缩容**，或者有节点宕机，对于一个 key 的备份数据，在备份节点直接生效为主节点，同时进行新列表备份到后续节点。
- Data **扩容**，新增节点初始化状态禁止新数据写入，读取请求进行转移到后续节点获取数据，直到其他节点的备份数据按照新节点信息同步完成，再提供服务 Working。



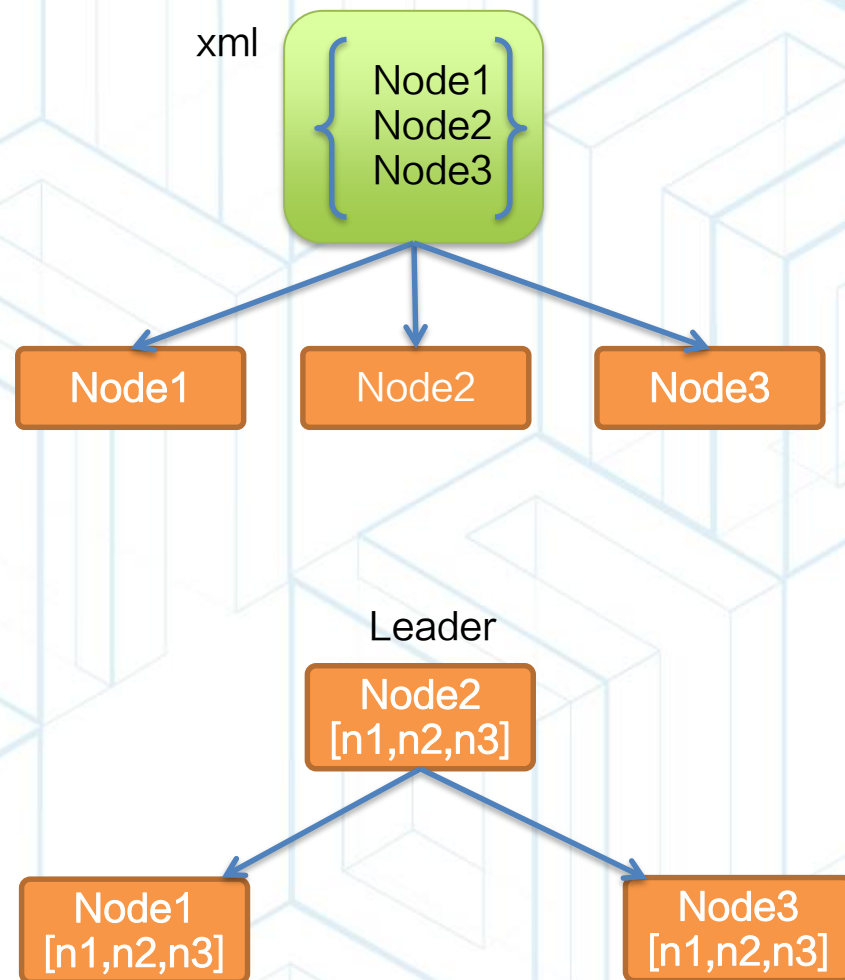
SOFARegistry 高可用

- 数据备份，以及内部数据的传递主要通过操作日志同步方式。
- 持有数据一方 server 发起变更通知，需要同步的 client 对比版本变更进行拉取数据操作日志。
- 操作日志存储采用堆栈方式，获取日志是通过当前版本号在堆栈内所处位置，把所有版本之后的操作日志同步过来执行。



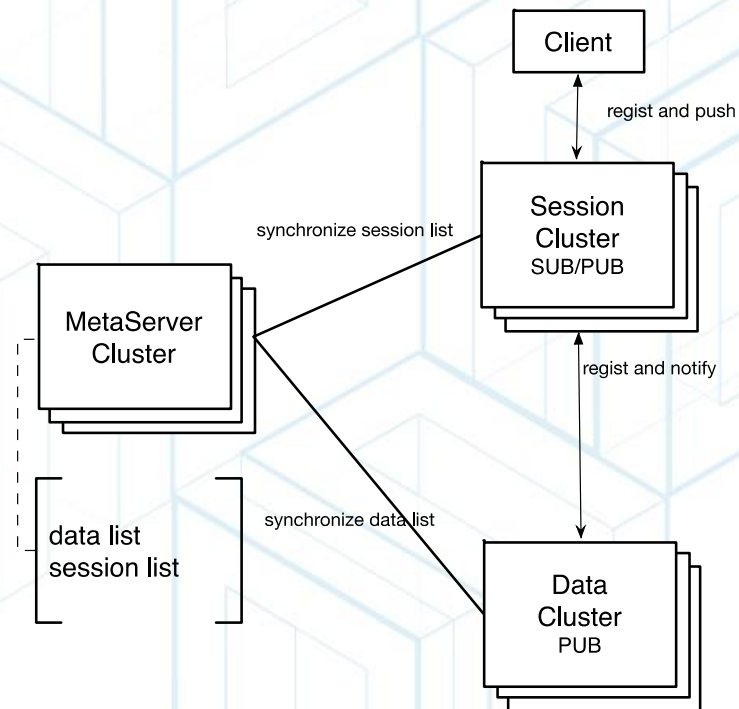
SOFARegistry 元数据

- Data 和 Session 如何知道集群内部其他节点地址列表？节点间如何动态应发现其他节点的存在？
- **MetaServer** 司职集群内所有节点信息发现。



SOFARegistry 元数据

- MetaServer 主要提供其他节点注册自身位置信息，并且通知给所有其他节点，最主要是 Data 列表信息。
- MetaServer 采用集群部署，集群整体节点列表数据通过 **Raft协议** 保持强一致，即所有其他 Session 和 Data 节点读写节点列表信息强一致。



SOFARegistry 开源

- SOFARegistry 同类产品比较

比较	SOFARegistry	Eureka 1.0	ZooKeeper
一致性	最终一致	最终一致	强一致
可用性	高可用，集群节点可动态扩容，数据保持多副本	高可用	节点选举过程整个集群不可用，无法提供服务
可扩展	一致性hash数据分片，理论上无限制扩展	数据节点相互同步方式保持一致，有上限瓶颈	数据强一致，同样存在上限
时效性	秒级服务发现，通过连接状态进行服务数据变更通知	采用长轮询健康检查方式获取节点状态，时效不敏感	强一致要求，多写效率低

SOFARegistry 开源

- SOFARegistry 支撑数据量
 - 2000+ 应用 2.3w 服务注册发现
 - 单机房 Data 集群支持百万级 Pub 数据，千万级 Sub 数据
 - 高可用，集群宕机 $\frac{1}{2}$ 以内节点服务自恢复
 - 支持数百应用同时启动发布订阅

SOFARegistry 将来

- SOFARegistry 面临挑战
 - 面向现有主流微服务框架
 - 面向云原生微服务运维
 - 面对单元化部署

SOFARegistry 将来



SOFARegistry

- 服务注册中心的功能
 - 服务注册 存储
 - 服务发现 推送
- 存储容量如何无限扩展
 - 一致性hash进行数据分片存储
- 数据连接如何无限扩展
 - 独立层承担连接，订阅关系（读）存储在 Session
- 如何保持存储高可用
 - 逐级回放，多副本
- 如何解决内部节点信息动态发现
 - 独立集群专门管理节点列表一致性数据



欢迎关注 SOFASlack 公众号
获取分布式架构干货



使用钉钉扫码入群
第一时间获取活动信息



蚂蚁金服
ANT FINANCIAL

金融科技
FINANCIAL TECHNOLOGY