SOFAJRaft高可用最佳实践：BC-MQ移动云消息队列高可用设计之谈

胡宗棠

胡宗棠

中国移动云能力中心中间件团队负责人，

SOFAJRaft Committer，

Apache RocketMQ Committer，

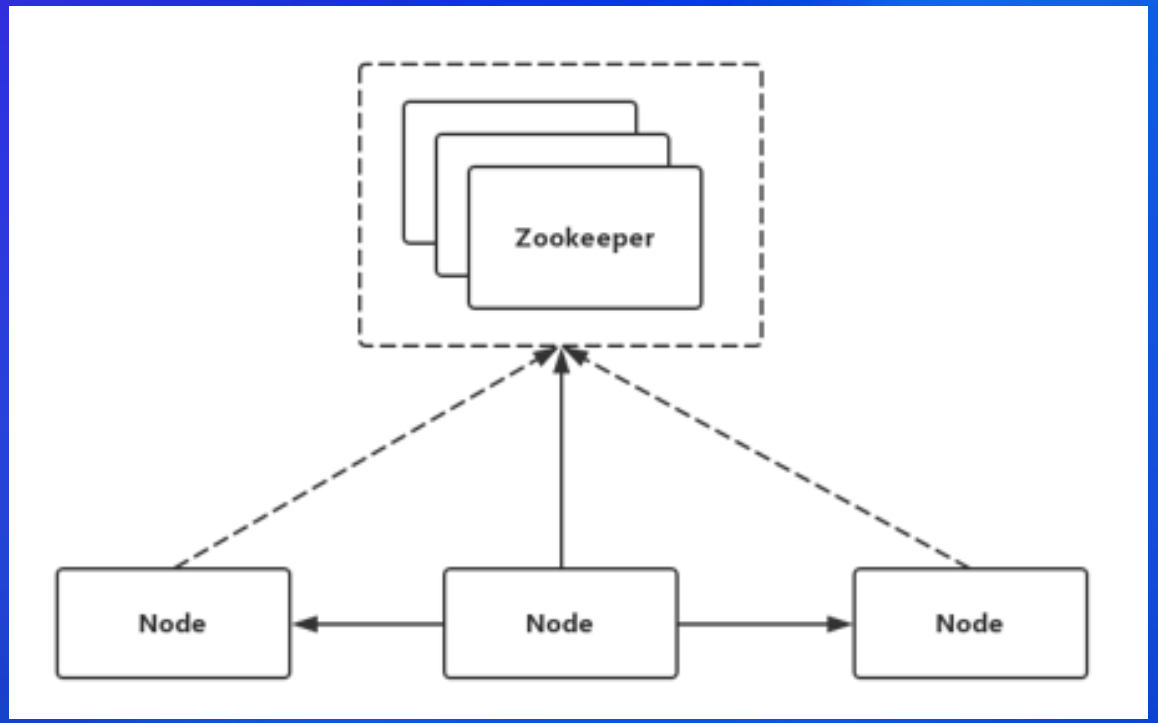
Linux OpenMessaging Advisory Borad Member，

熟悉分布式消息队列、API网关和分布式事务等中间件设计原理、架构以及各种应用场景，具有丰富高性能、高可用和高并发经验。

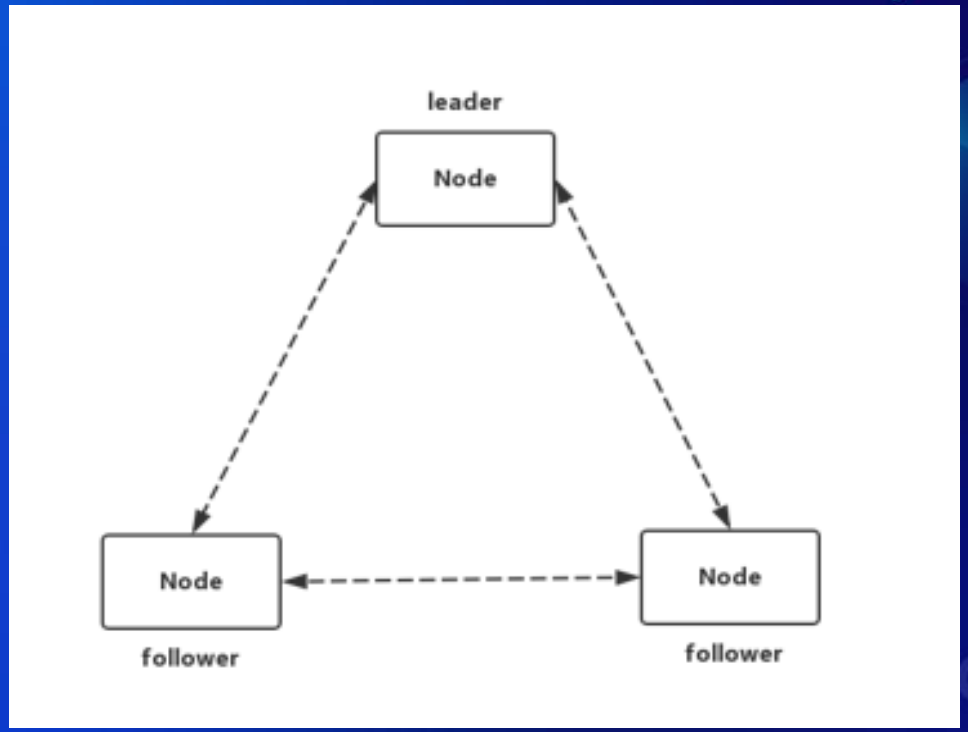
目录

Raft为何出现？FailOver

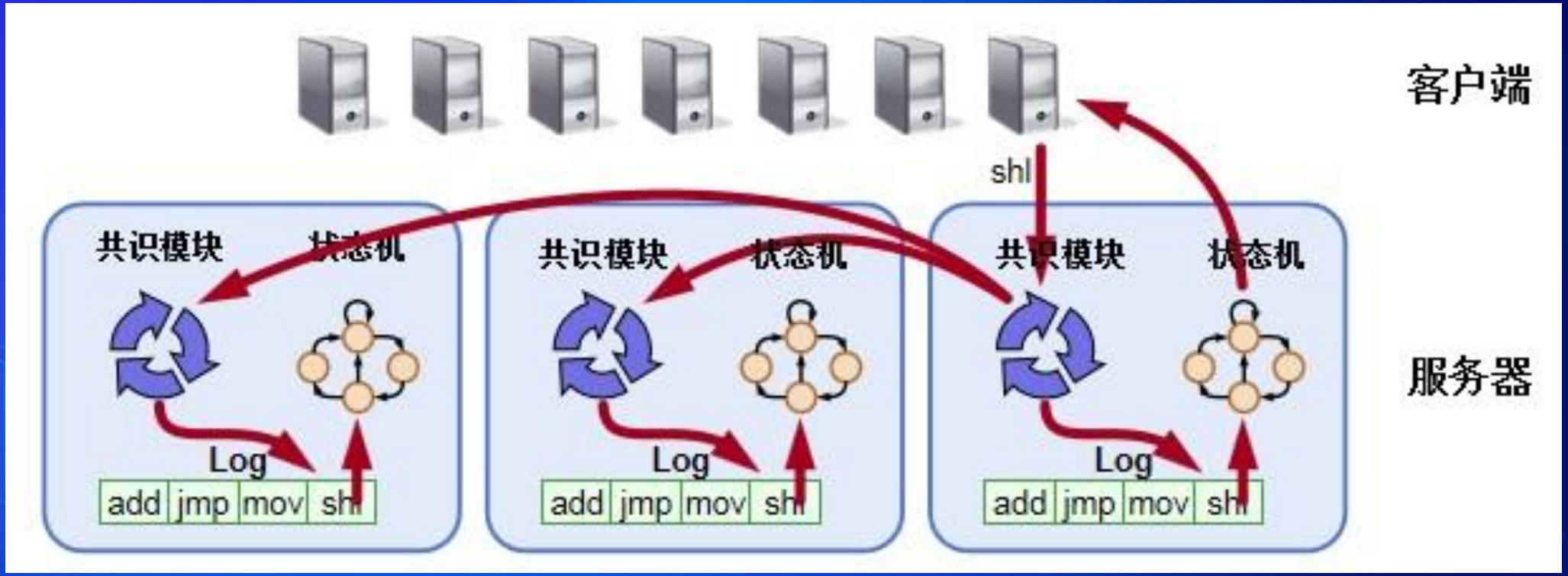
Zookeeper



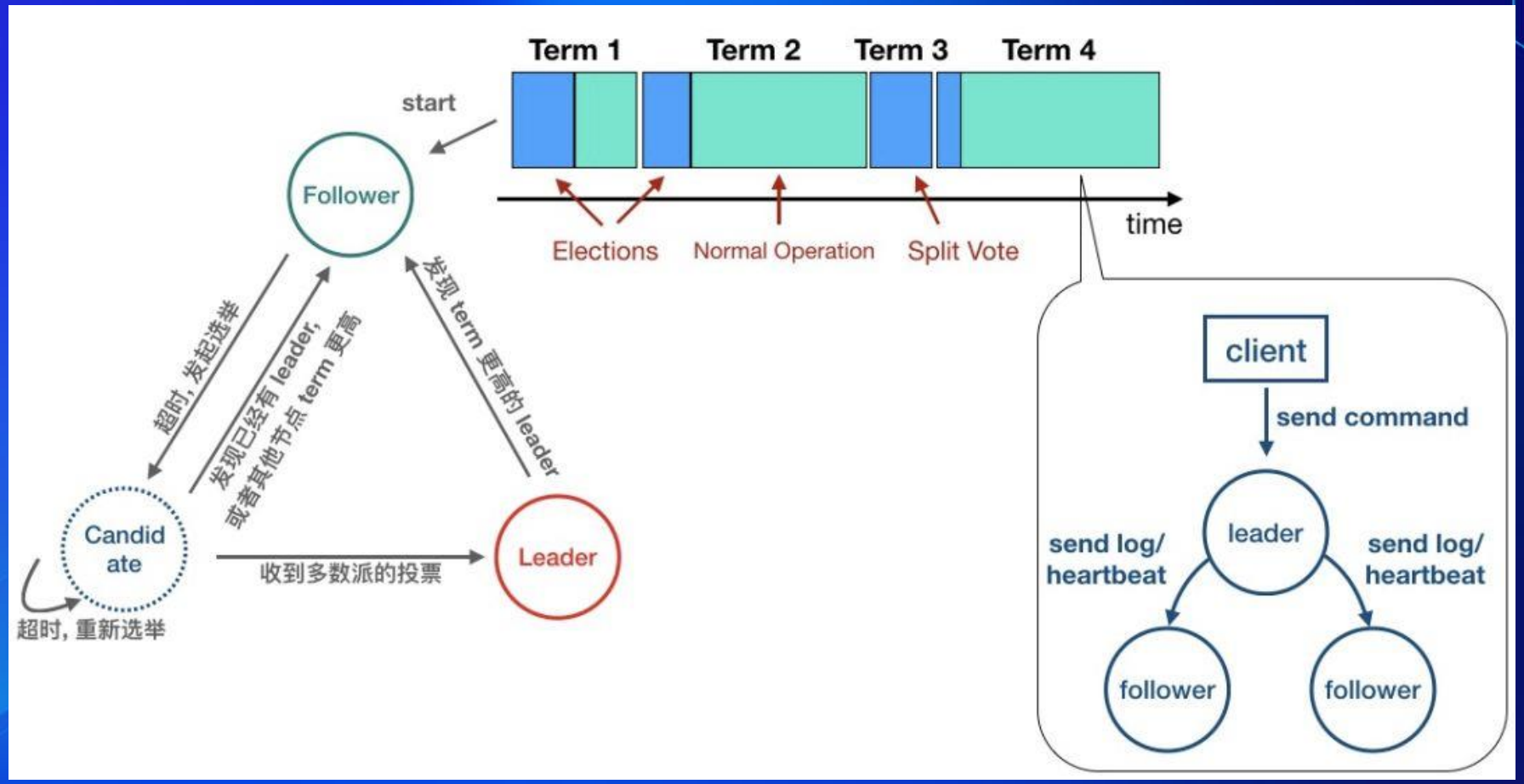
Raft



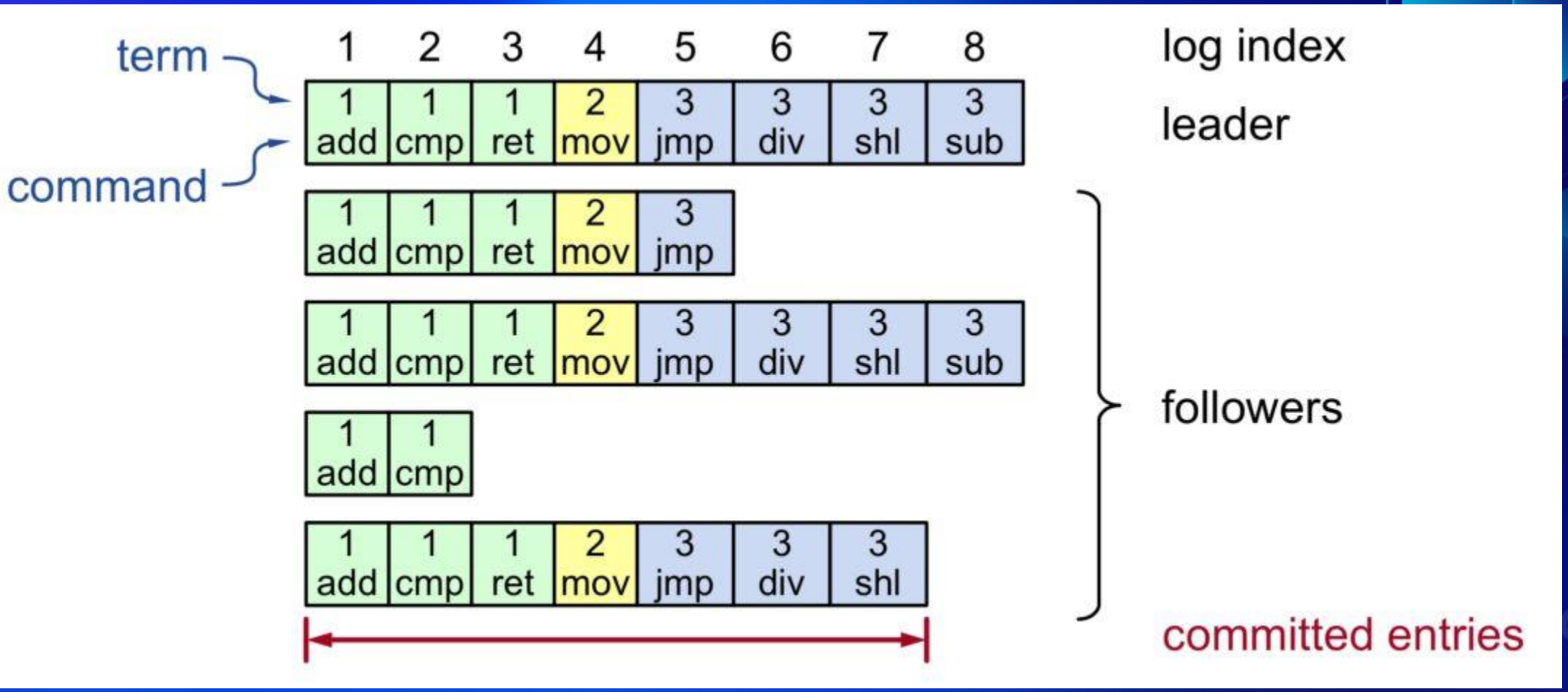
Raft算法整体概览


客户端
shl
共识模块
状态机
共识模块
状态机
共识模块
状态机
服务器
Log
add
jmp
mov
shl
Log
add
jmp
mov
shl
Log
add
jmp
mov
shl

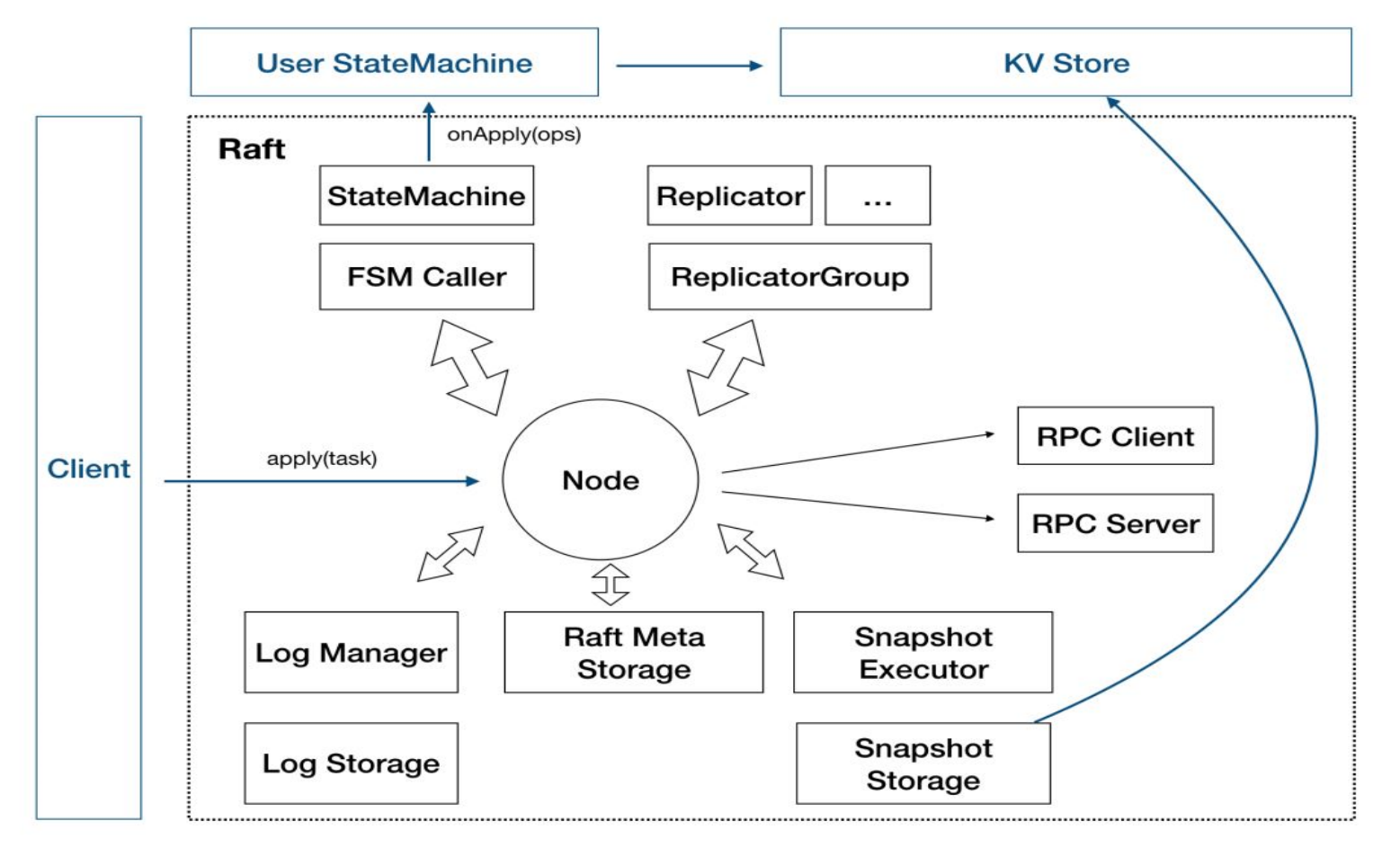
Raft算法中的Leader选举



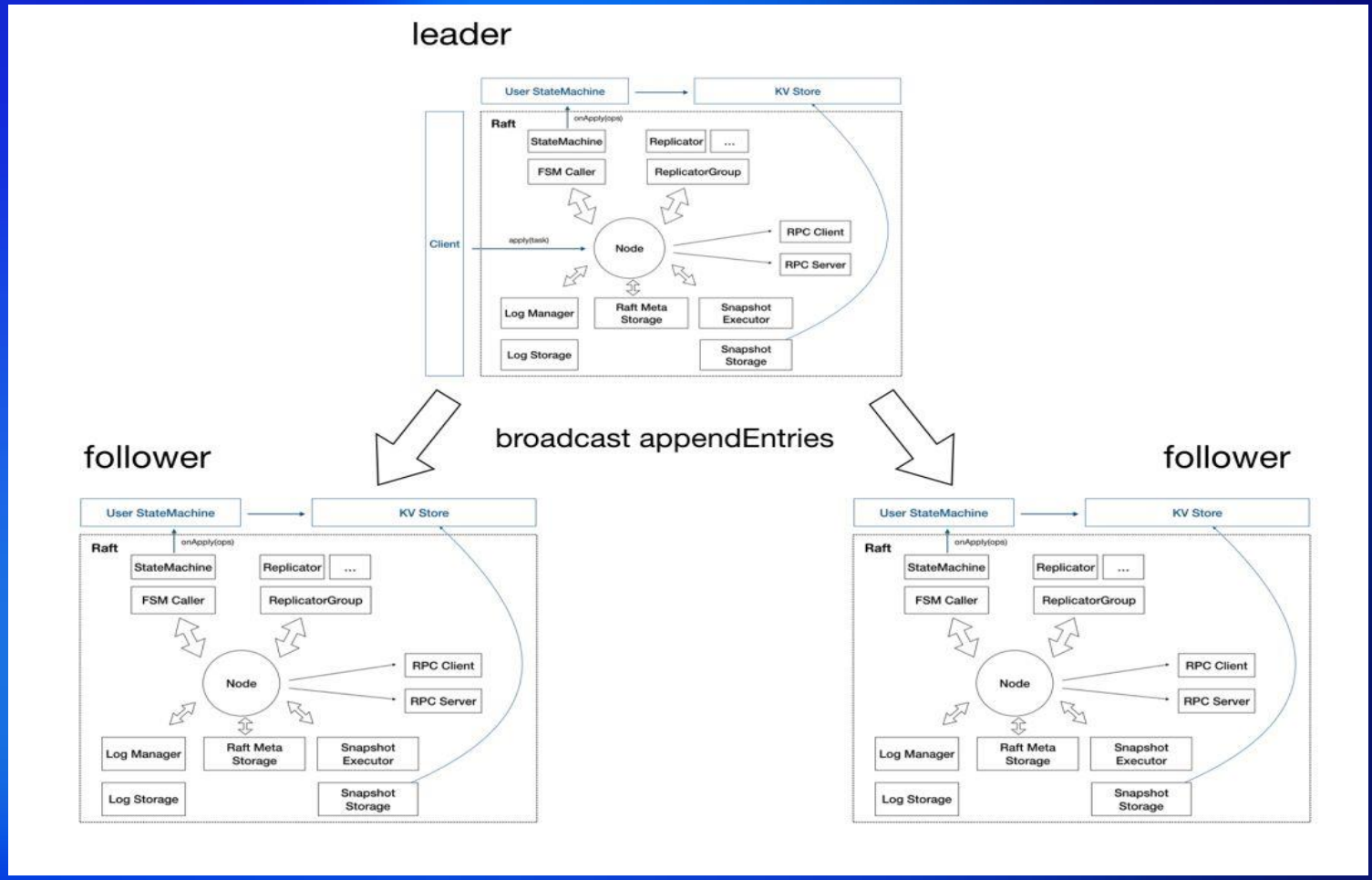
Raft算法中的日志复制



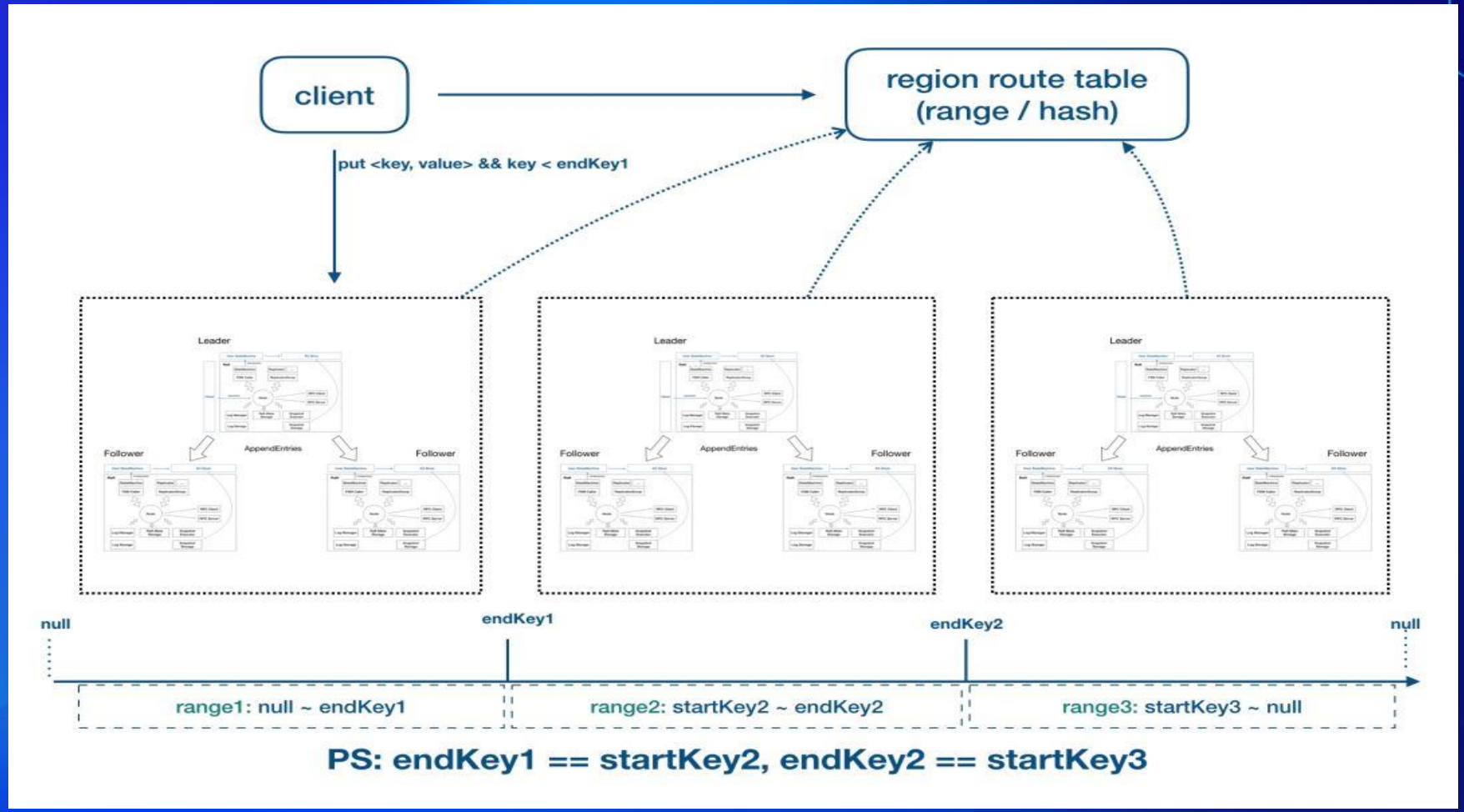
SOFAJRaft整体框架介绍



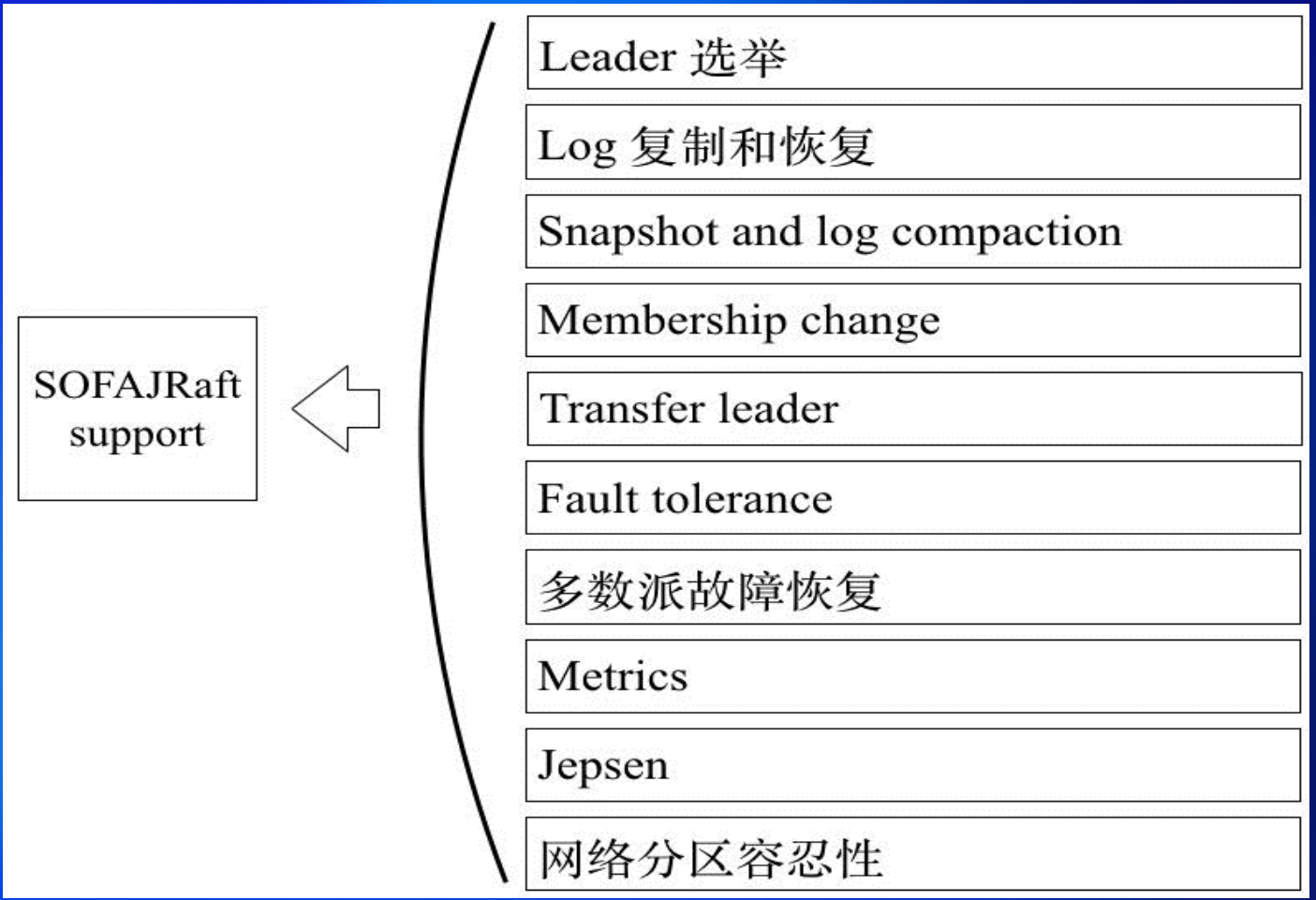
SOFAJRaft SingleGroup集群框架图



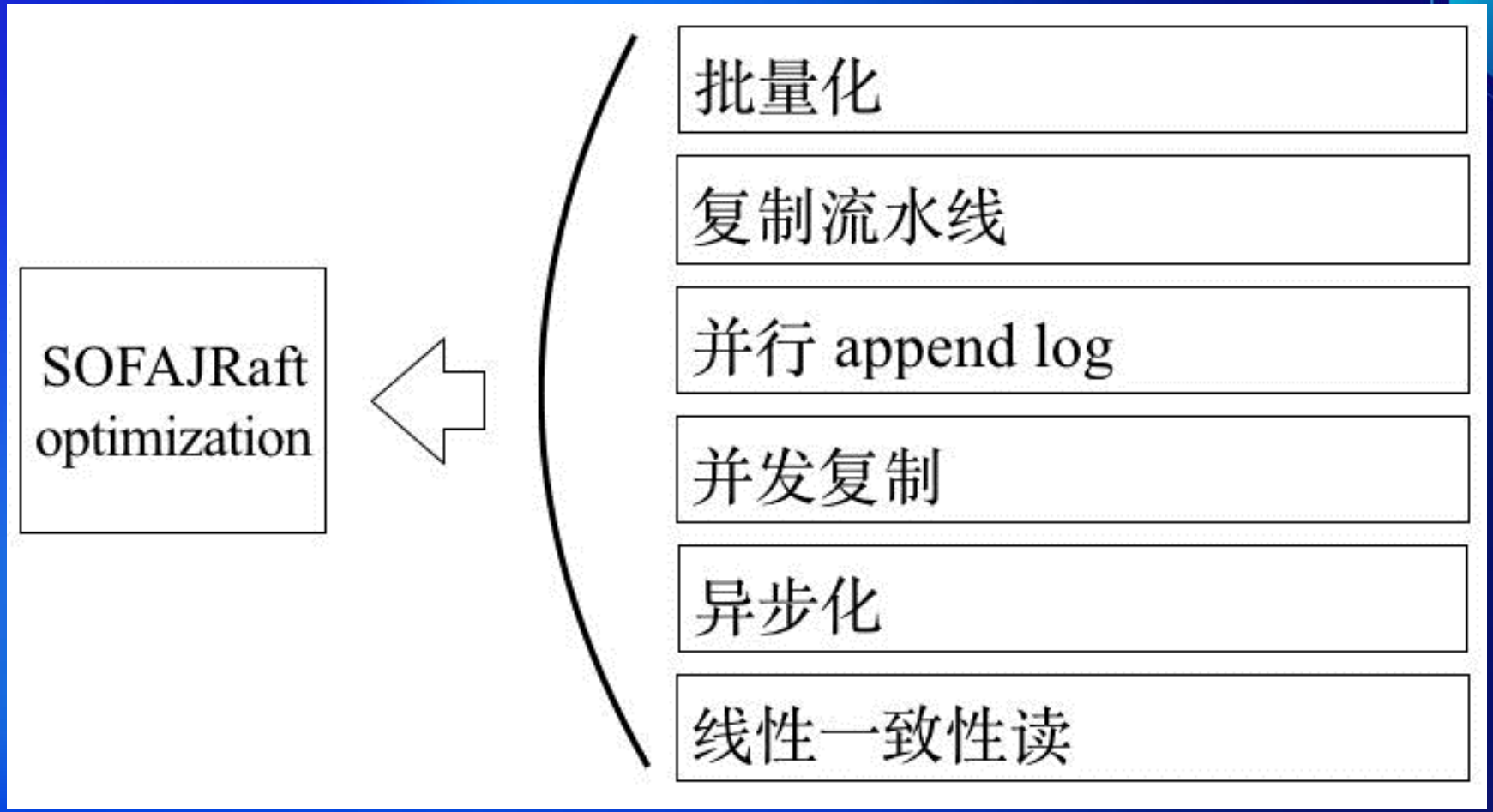
SOFAJRaft Multi-Group集群框架图



SOFAJRaft的主要功能特性



SOFAJRaft的优化特性



SOFAJRaft的性能

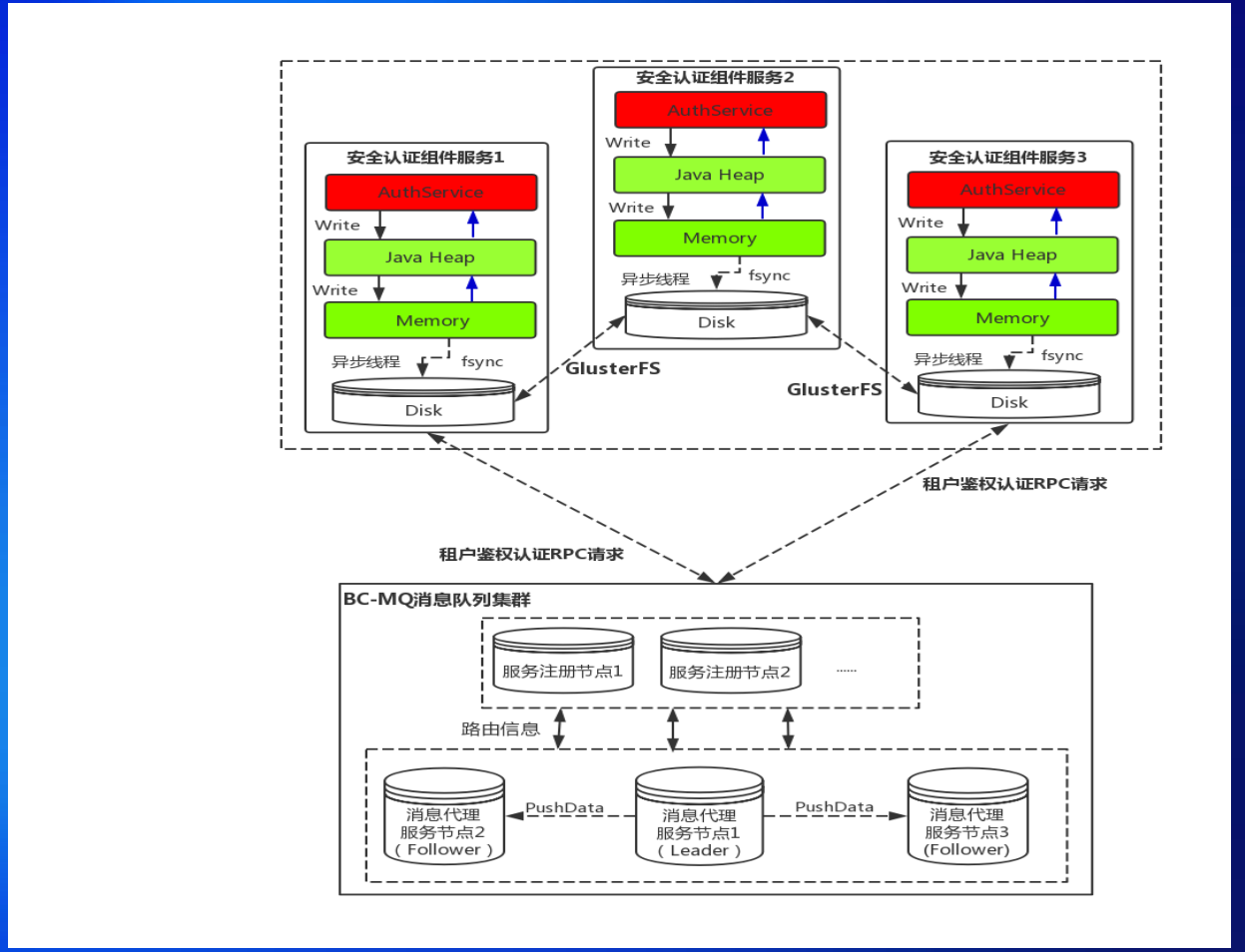
Client 数量	Client-Batching	Server Load, CPU	Storage Type	写读比例	key, value 大小	Replicator-Pipeline	Result
2	关闭	接近 20，over 50 %	MemoryDB	9:1	均为 16 字节	关闭	共 7.5w ops
2	关闭	接近 20，over 50 %	MemoryDB	9:1	均为 16 字节	开启	共 10w+ ops
8	开启	接近 15, 40 %	MemoryDB	9:1	均为 16 字节	开启	共 40w+ ops
8	开启	接近 10, 30 %	RocksDB	9:1	均为 16 字节	开启	共 25w+ ops

BC-MQ的简要介绍

BC-MQ基于Apache开源社区RocketMQ内核，同时参考云端PAAS产品架构，结合消息中间件云服务和组件化管理运维平台，是一款针对云端场景的高性能、高可靠、低延迟、高灵活性的平台级产品，具有消息轨迹、资源统计报表、MQTT物联网消息队列、事务消息等丰富功能特性。目前，已经在中国移动公有云和私有云上发布，并且在公司内部得到广泛落地应用。

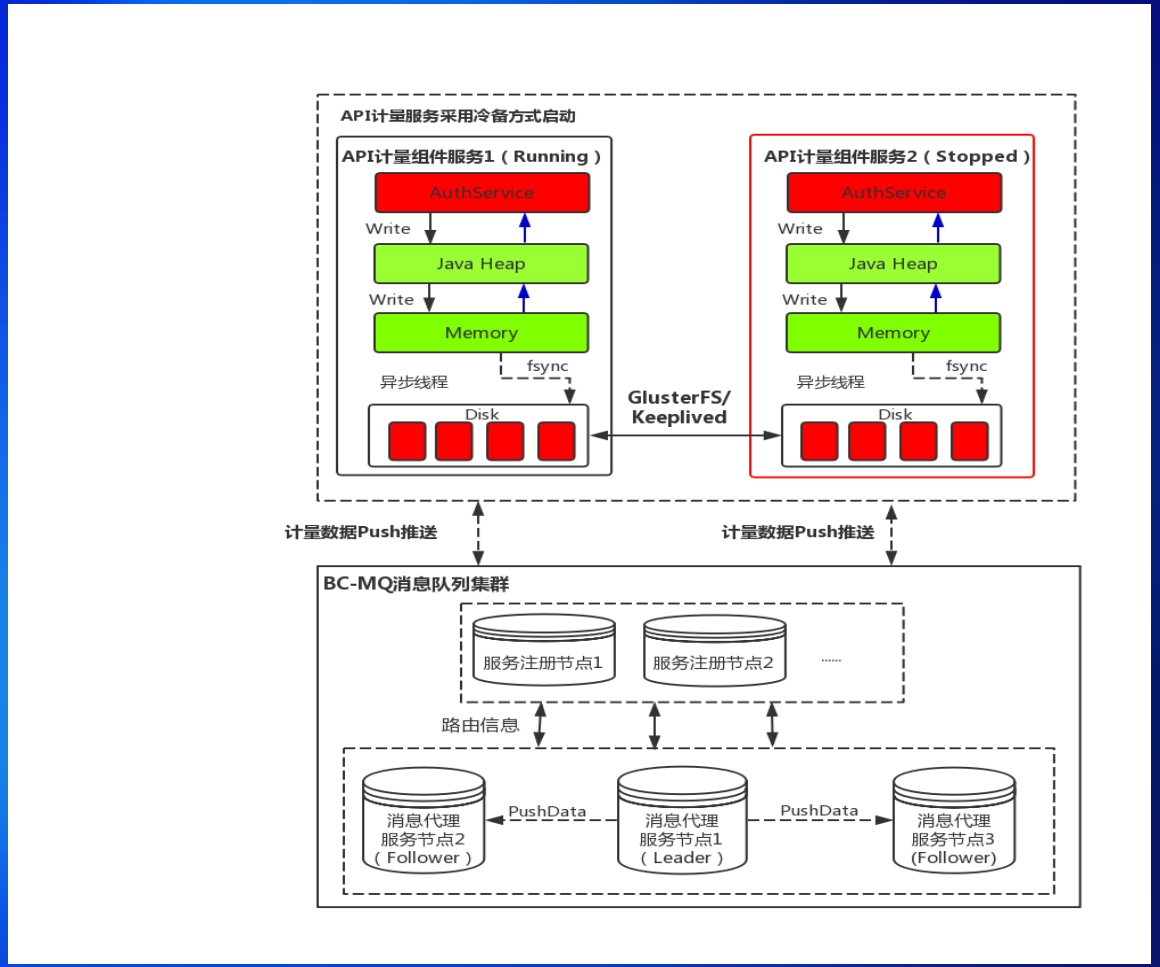
BC-MQ选型SOFAJRaft的原因

GlusterFS+KeepaLived 高可用设计方案



BC-MQ选型SOFAJRaft的原因

GlusterFS+KeepaLived 高可用设计方案



BC-MQ选型SOFAJRaft的原因

原有高可用方案的缺点：

（1）GlusterFS分布式文件存储系统和Keepalived组件这类组件，增加了系统整体的运维复杂度，给运维人员带来很多人工介入排查和部署的工作负担。这两种组件也增加了系统整体设计的复杂性；

（2）Keepalived组件采用冷备方式作为高可用方案需要考虑主机故障宕机后切换到备机的时间成本消耗，切换时候会造成业务感知；

目标：

生产环境出现部分节点故障后，服务组件依旧能正常提供服务，业务无感知？？？

BC-MQ基于SOFAJRaft的高可用设计

集成SOFAJRaft组件后的改变：

（1）部署方式：不再依赖GlusterFS和Keepalived两个外部组件，极大简化BC-MQ云服务的部署方式；

（2）数据持久化：通过SOFAJRaft的日志复制和状态机，实现集群中Leader节点和Follower节点的数据同步保证主备节点的数据一致性；

（3）高可用模式：在发生故障后，通过SOFAJRaft自动Leader选举，业务组件仍然能够对外正常提供服务，故障转移的过程无需运维人员介入；

BC-MQ基于SOFAJRaft的高可用设计

组件服务端的状态机接口实现：

（1）void onApply(...)：SOFAJRaft中需要用户自己实现的最为核心的接口；

（2）void onLeaderStart(...)：在节点通过选举成为Leader时调用；

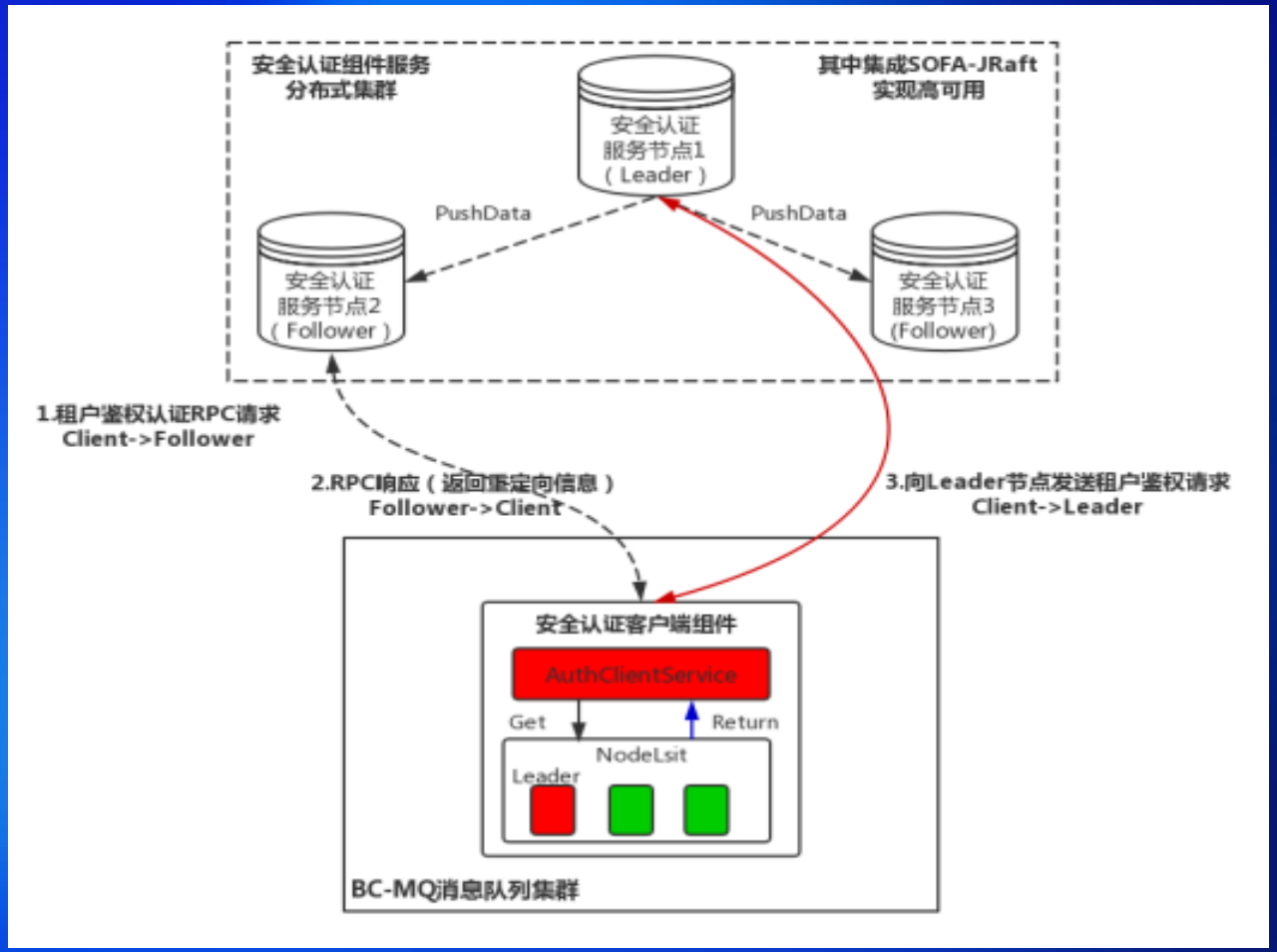
（3）void onLeaderStop(...)：在节点失去Leader角色时调用；

（4）void onSnapshotSave(...）:节点启动或者安装Snapshot后加载快照；

（5）boolean onSnapshotLoad(...)：节点定期保存快照；

BC-MQ基于SOFAJRaft的高可用设计

客户端请求
重定向机制优化：

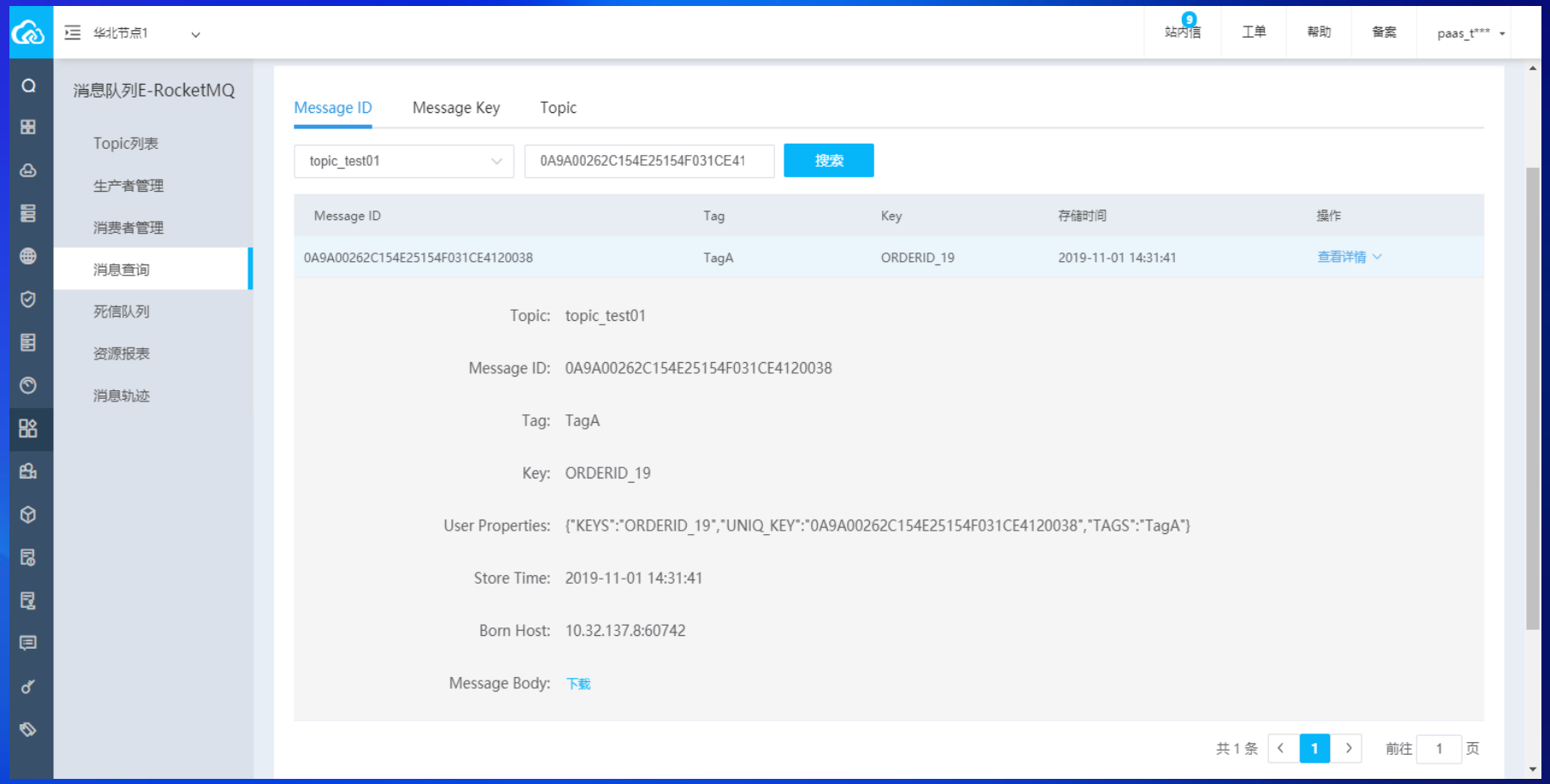


BC-MQ的高可用性的验证

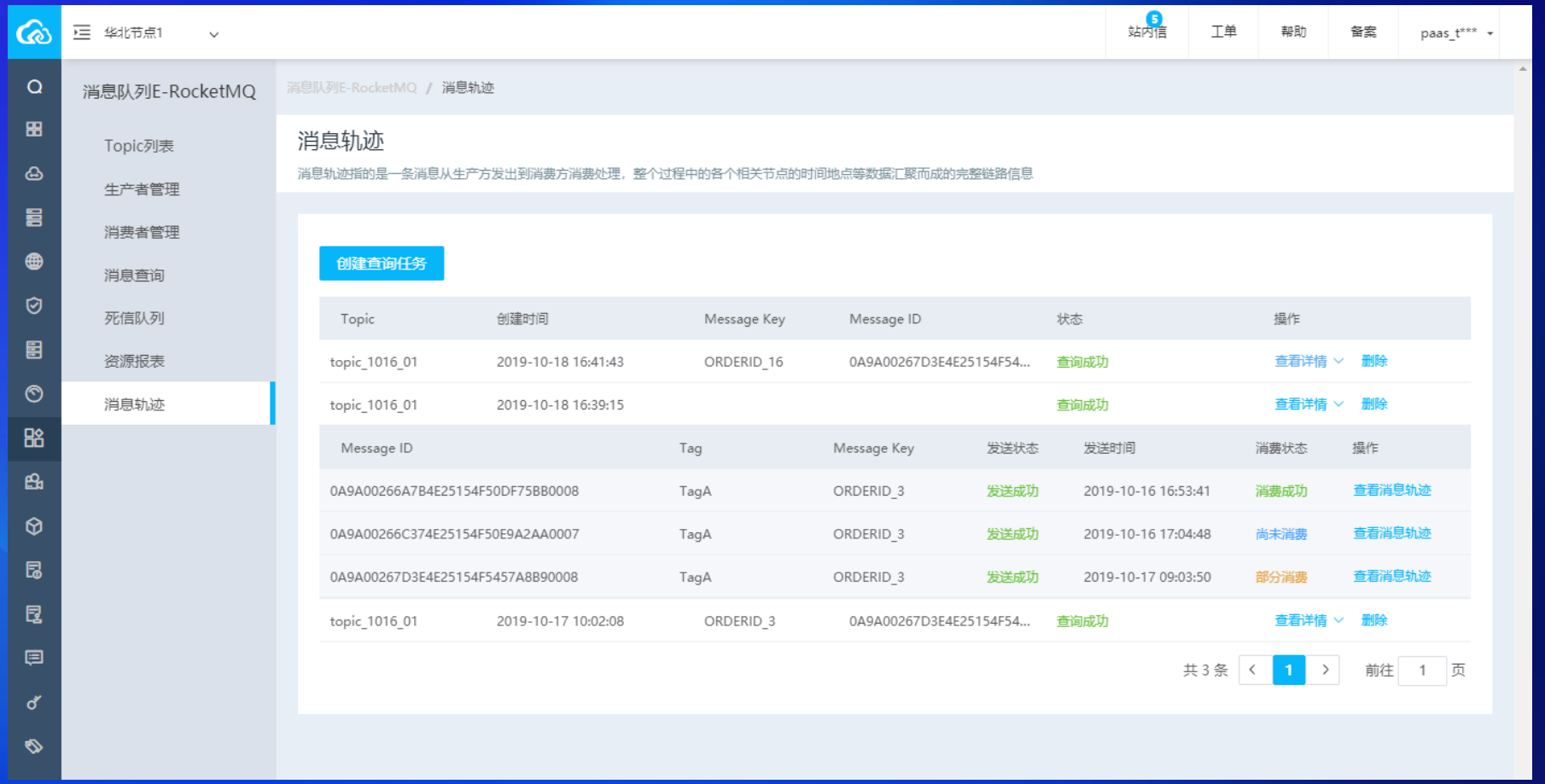
高可用测试用例验证：

（1）随机分区，一大一小两个网络分区；
（2）随机增加和移除节点；
（3）随机停止和启动节点；
（4）随机 kill -9 和启动节点；
（5）随机划分为两组，互通一个中间节点，模拟分区情况；
（6）随机划分为不同的 majority 分组；

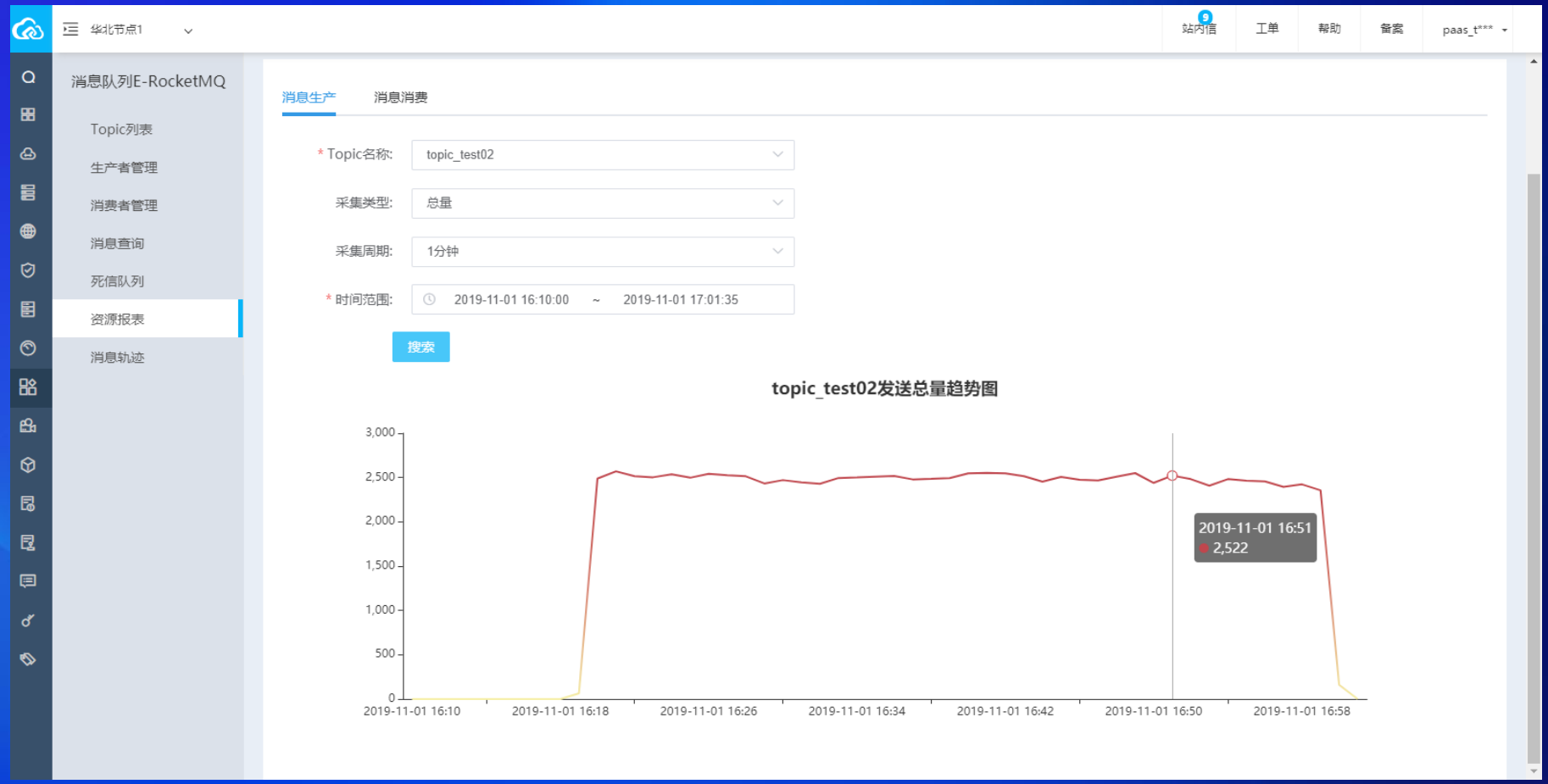
BC-MQ的推广应用案例—移动云消息队列服务



BC-MQ的推广应用案例—移动云消息队列服务



BC-MQ的推广应用案例—移动云消息队列服务



SOFAJRaft社区相关介绍

作为共识算法的工程实现，通过一系列优化逐步解决了落地过程中会遇到的诸多问题，保证了生产环境使用的可靠性。

【剖析 | SOFAJRaft 实现原理】系列

出品人：

力鲲，蚂蚁金服 SOFA 团队，SOFAJRaftLab 负责人

作者：

米麒麟、袖扣、徐家锋、胡宗棠

源码剖析issue

https://github.com/sofastack/sofa-jraft/issues/327

在线文档

https://www.sofastack.tech/projects/sofa-jraft/overview/

欢迎加入社区成为 Contributor，**SOFAJRaft**。

SOFAJRaft社区相关介绍

欢迎关注SOFAStack公众号
获取分布式架构文章

使用钉钉扫码入群
第一时间获取活动信息

Thanks For Watching

本PPT来自2019携程技术峰会

更多信息请关注“携程技术中心”微信公众号~