蚂蚁金服 ANT FINANCIAL | 金融科技 FINANCIAL TECHNOLOGY

SOFAJRaft
蚂蚁金服基于 RAFT 一致性算法的生产级高性能 Java 实现

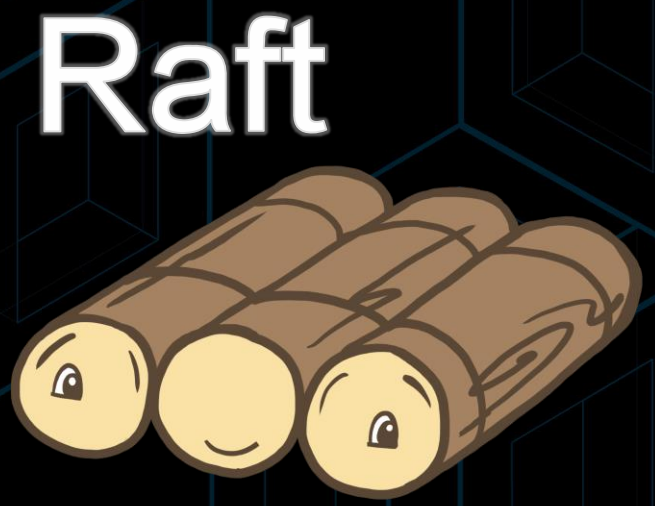
力 鲲

蚂蚁金服 SOFAJRaft 核心成员

目 录
contents

• Part 1 – Raft 算法

Consensus algorithms



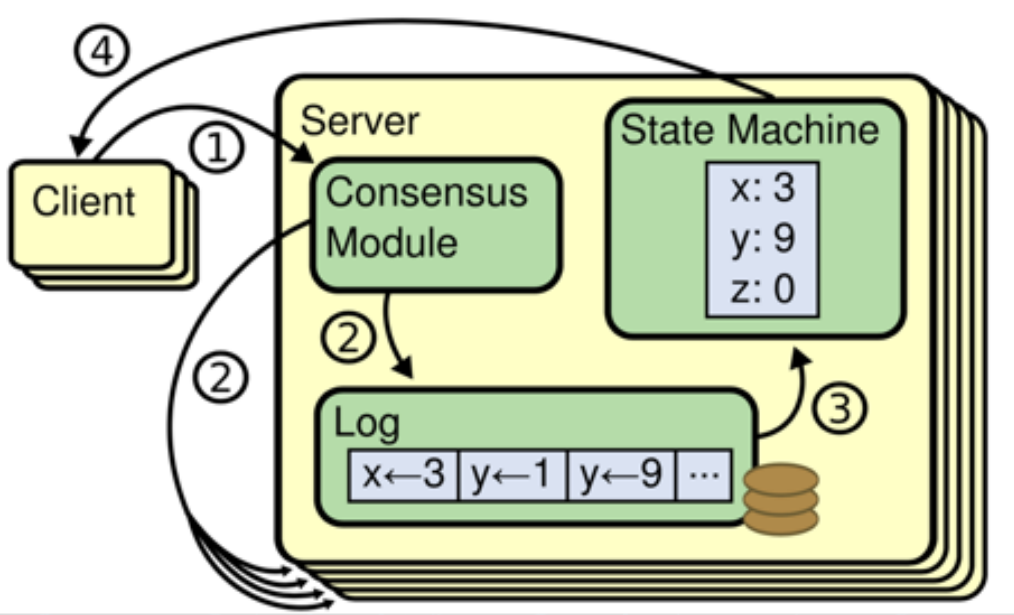
共识算法：

- 多个参与者针对某一件事达成完全一致：一件事，一个结论。
- 已达成一致的结论，不可推翻。

Raft 特性：

- Strong leader
- Leader election
- Membership changes

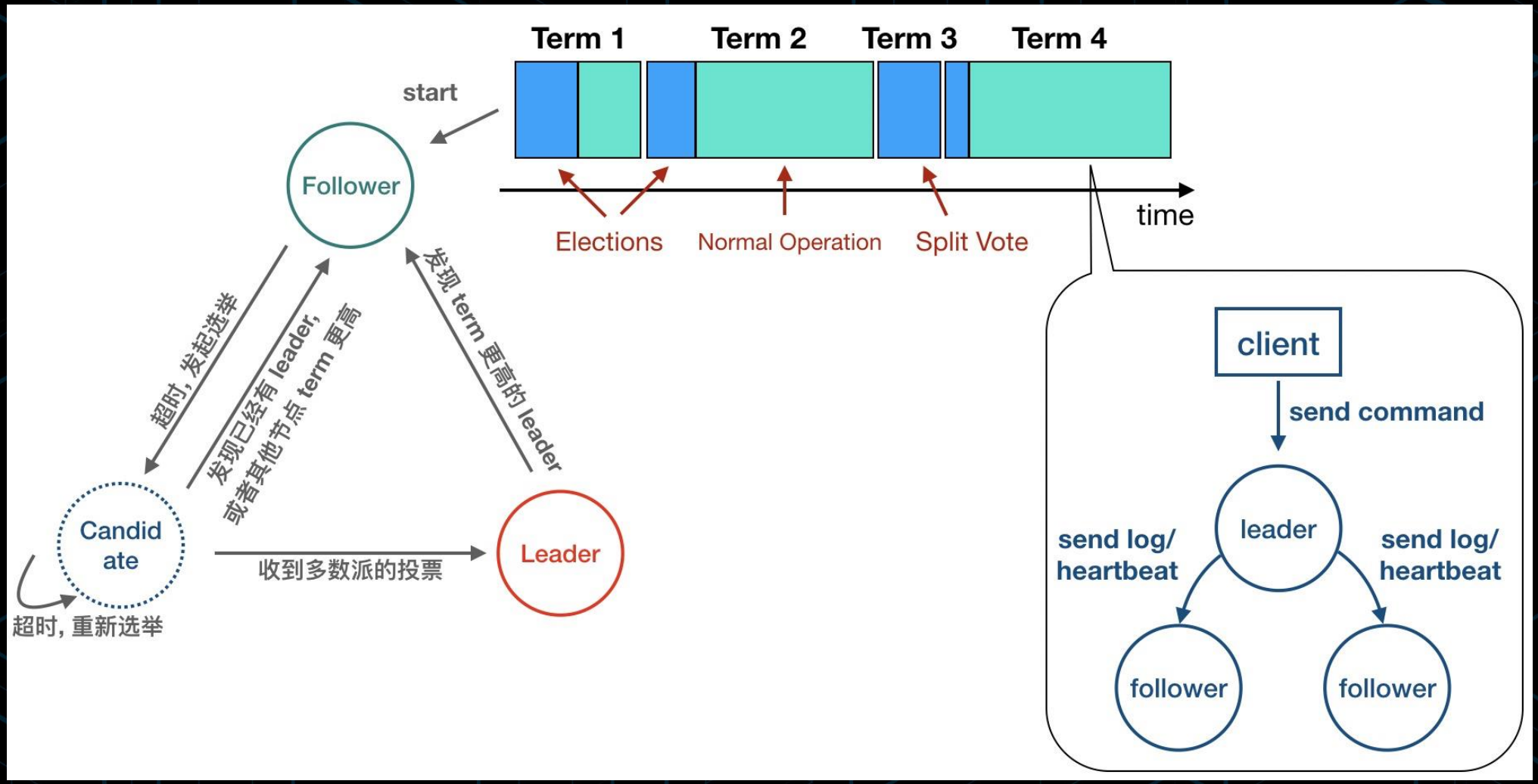
复制状态机



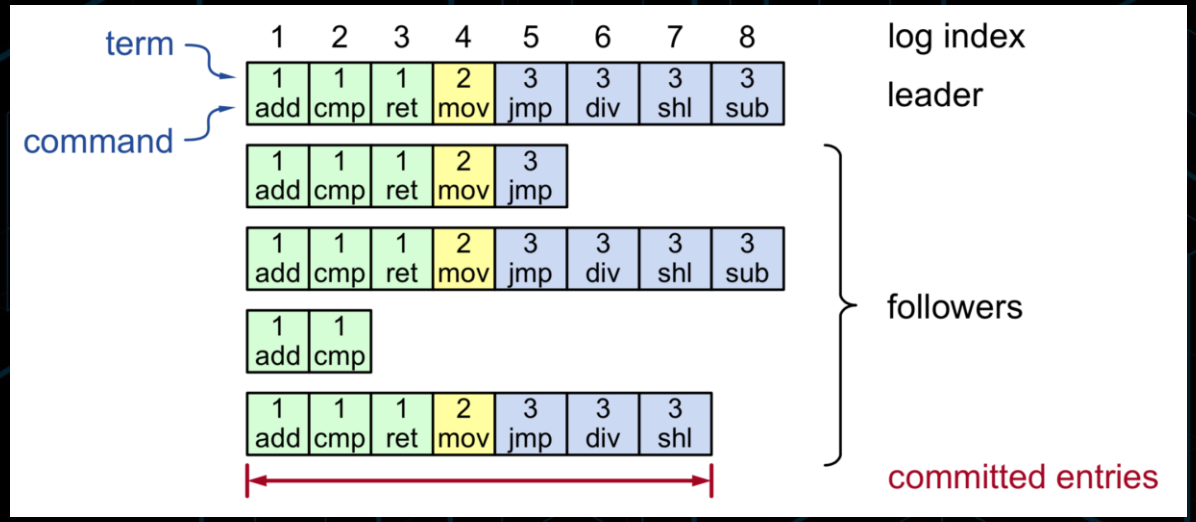
典型应用场景：复制状态机

- 保证被复制日志的**内容**一致；
- 保证被复制日志的**顺序**一致。

Leader election



Log replication



Log replication日志格式

- TermId
- LogIndex
- LogValue

业内 Raft 实现

- braft
  - 介绍：An industrial-grade C++ implementation of RAFT consensus algorithm based on brpc, widely used inside Baidu to build highly-available distributed systems.
  - GitHub：https://github.com/brpc/braft
  - Language：C++
- etcd
  - 介绍： Distributed reliable key-value store for the most critical data of a distributed system.
  - GitHub： https://github.com/etcd-io/etcd
  - Language：Go
- TiKV
  - 介绍： Distributed transactional key-value database, originally created to complement TiDB.
  - GitHub： https://github.com/tikv/tikv
  - Language：Rust

Raft 的 Java 实现

部分现有 Raft Java 实现

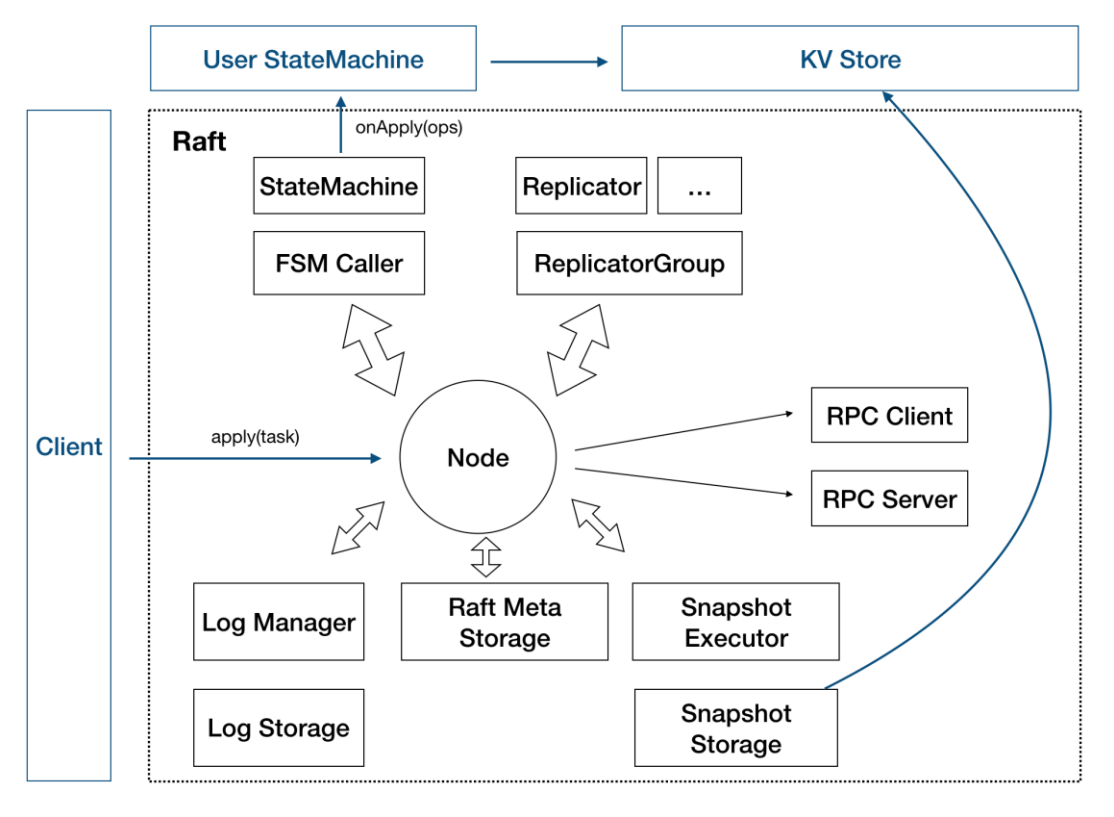
Name	License	Leader Election + Log Replication?	Membership Changes?	Log Compaction?
copycat	Apache2	Yes	Yes	Yes
OpenDaylight	Eclipse	Yes	No	Yes
Ratis	Apache2			
Permazen/RaftKVDatabase	Apache2	Yes	Yes	Yes
xraft	MIT	Yes	Yes	Yes
barge	Apache2	Yes	No	No
tetrapods/raft	Apache2	Yes		Yes
Raft-JVM		No	No	No
Raft4WS	Apache2	Yes	No	No

• Part 2 – SOFAJRaft 介绍

SOFAJRaft 概况

- SOFAJRaft：基于 Raft 算法的生产级高性能 Java 实现，支持 MULTI-RAFT-GROUP
- 开发时间： 2018 年 3 月 - 2019 年 2 月
- GitHub：https://github.com/alipay/sofa-jraft/
- 应用场景：
  - Leader 选举
  - 分布式锁服务
  - 高可靠的元信息管理
  - 分布式存储系统
- 使用案例
  - RheaKV
  - SOFA 服务注册中心元信息管理模块

SOFAJRaft 设计



Log 存储

- Log Storage
- Log Manager

Raft Metadata 存储

- 元信息存储

Snapshot 存储

- Snapshot Storage
- Snapshot Executor

状态机 StateMachine

- 用户核心逻辑的实现

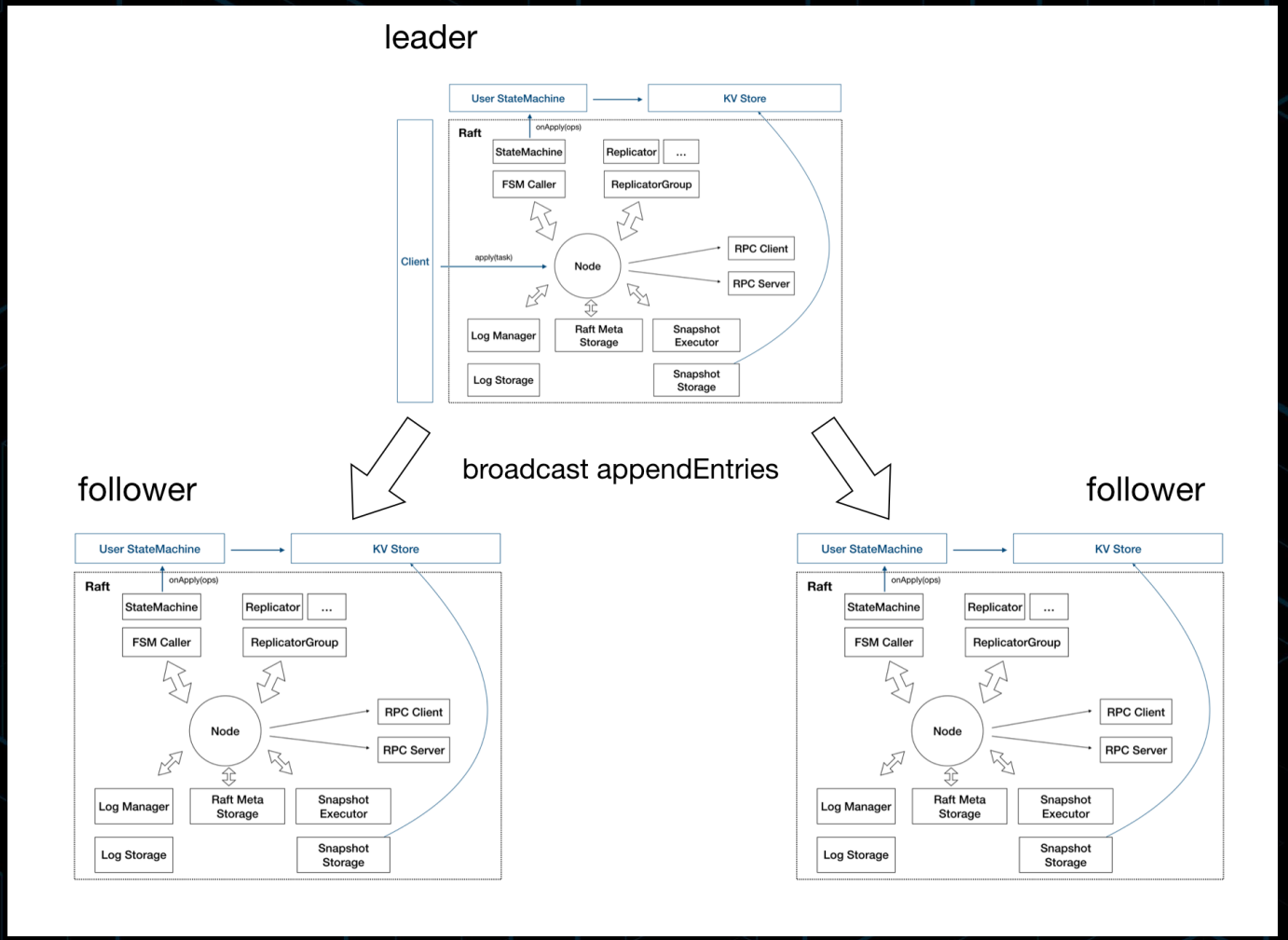
FSMCaller

- 封装对 StateMachine 的调用

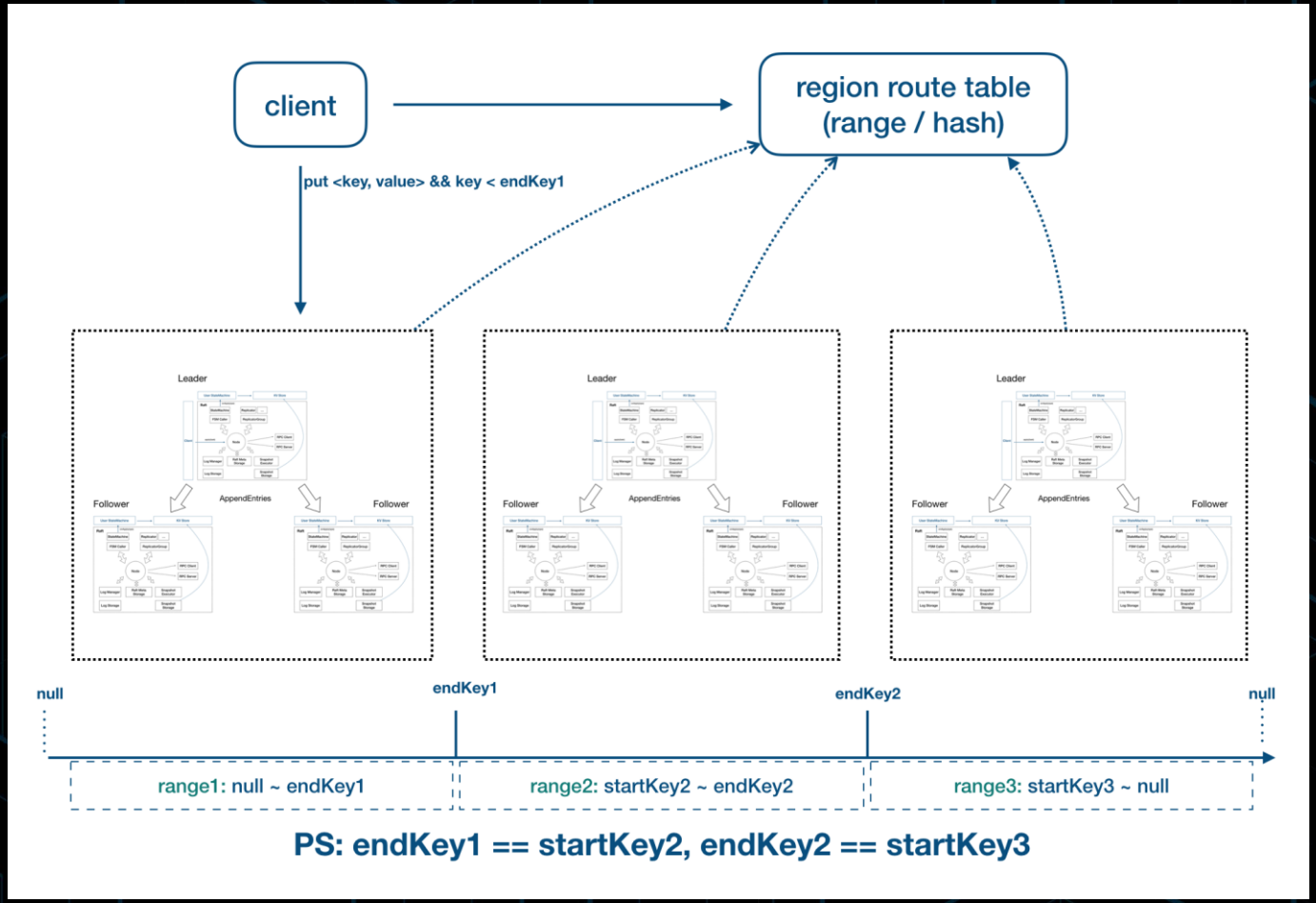
复制

- Replicator：用于 Leader 向 Followers 复制 Log
- ReplicatorGroup

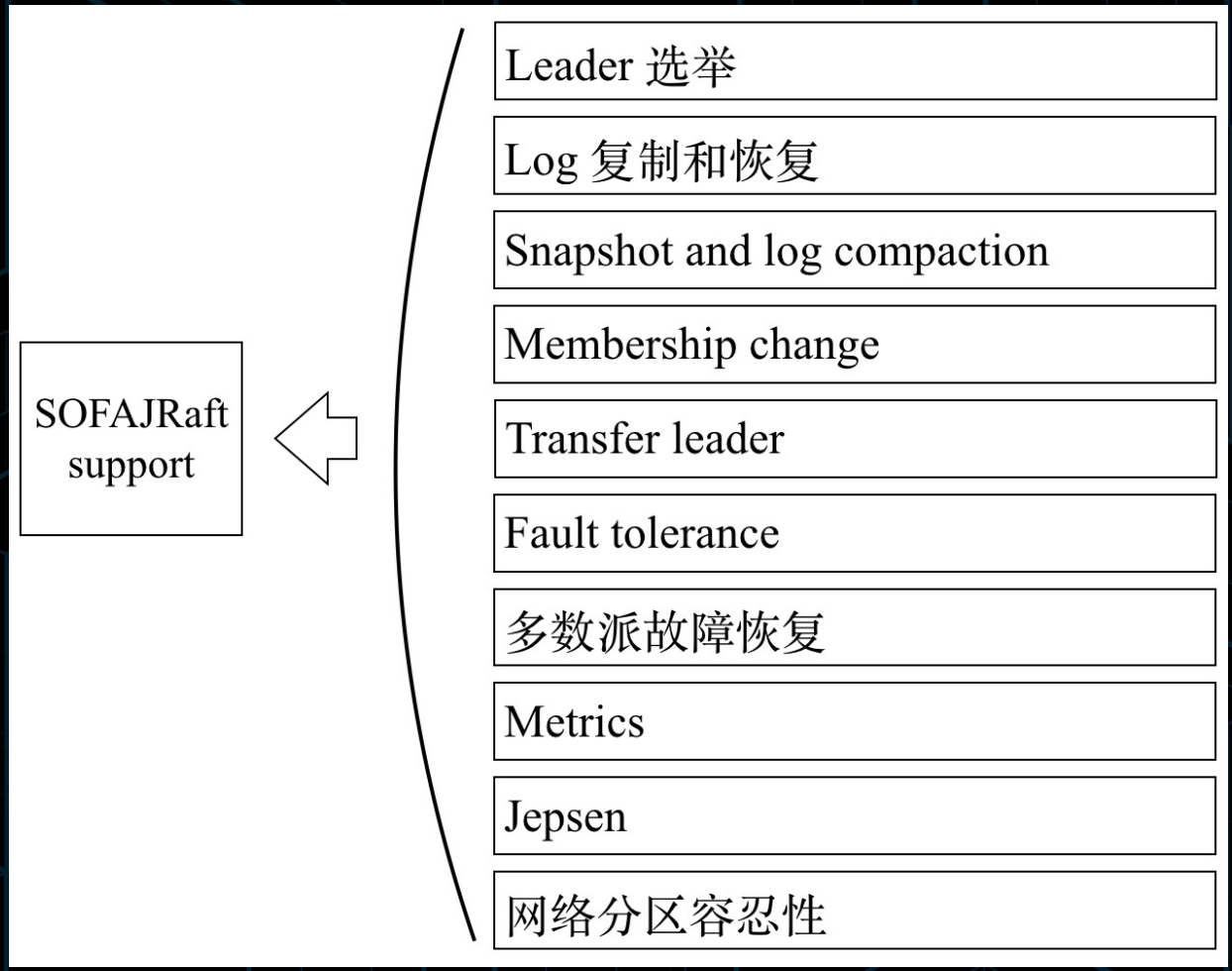
三副本 SOFAJRaft 集群部署



Multi-Raft-Group



SOFAJRaft 支持特性

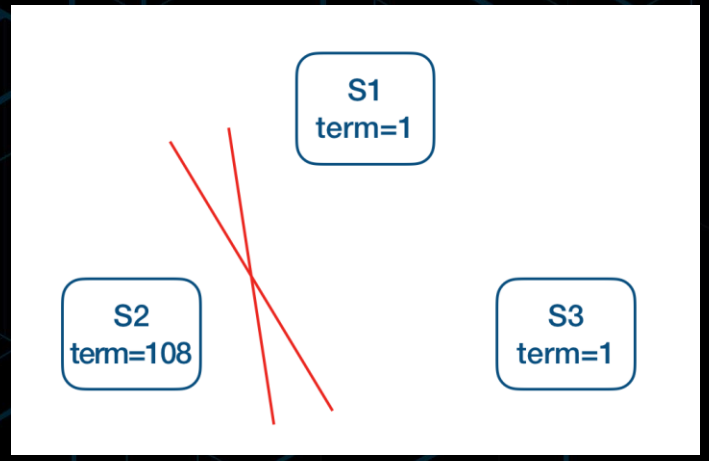


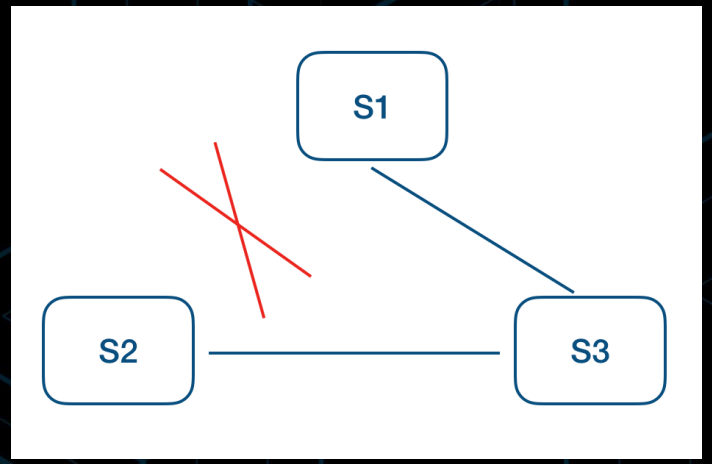
Jepsen

Jepsen：分布式验证和故障注入测试框架，模拟验证多种情况：

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

网络分区容忍性





Symmetric network partition tolerance:

对称网络分区容忍性

- pre-vote(currentTerm + 1, lastLogIndex, lastLogTerm)

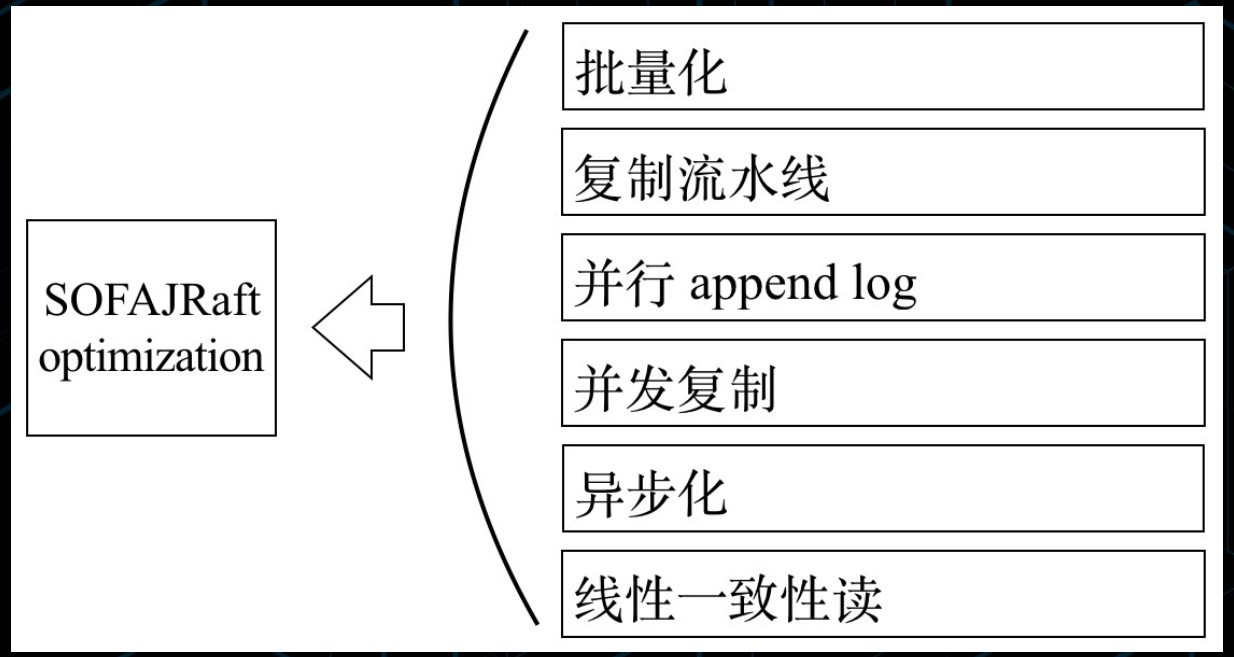
Asymmetric network partition tolerance:

非对称网络分区容忍性

- Follower 维护一个时间戳记录收到 leader 上数据更新的时间

• Part 3 – SOFAJRaft 优化

SOFAJRaft 优化



- Batch
- Replication pipeline
- 线性一致性读

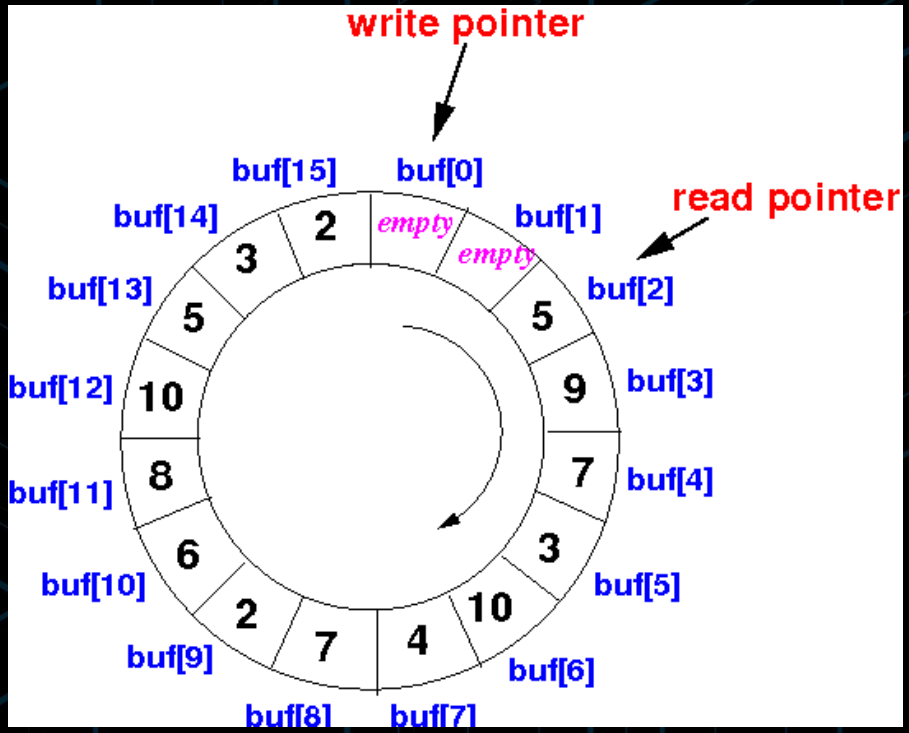
Batch

Batch：

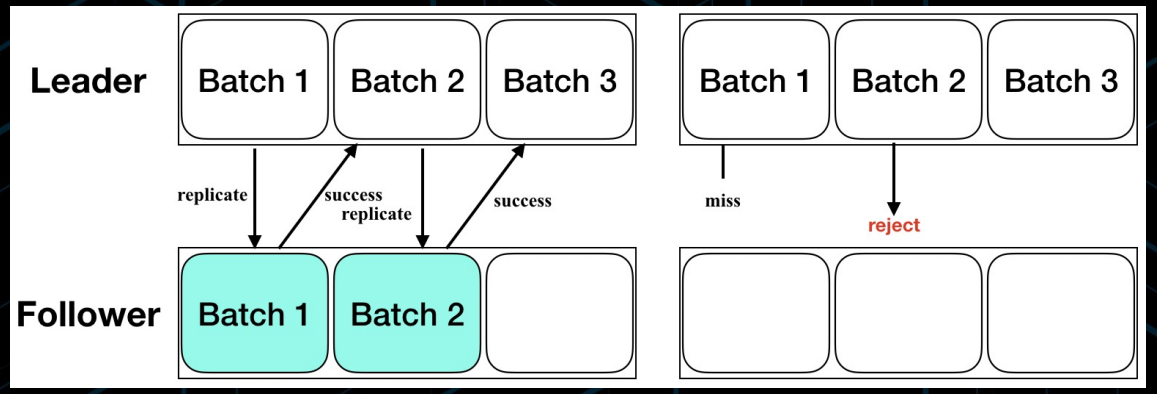
- 批量提交 task
- 批量网络发送
- 本地 IO batch 写入
- 状态机批量应用

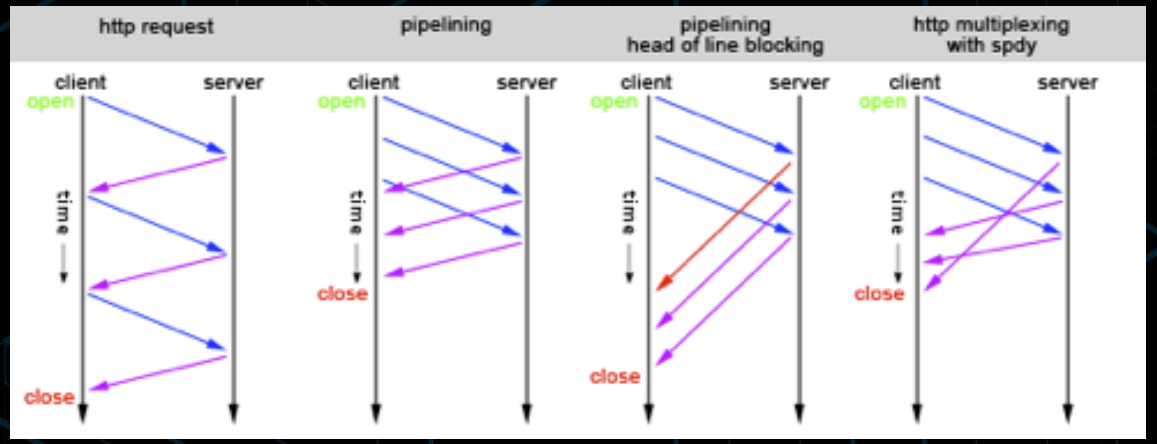
Disruptor：

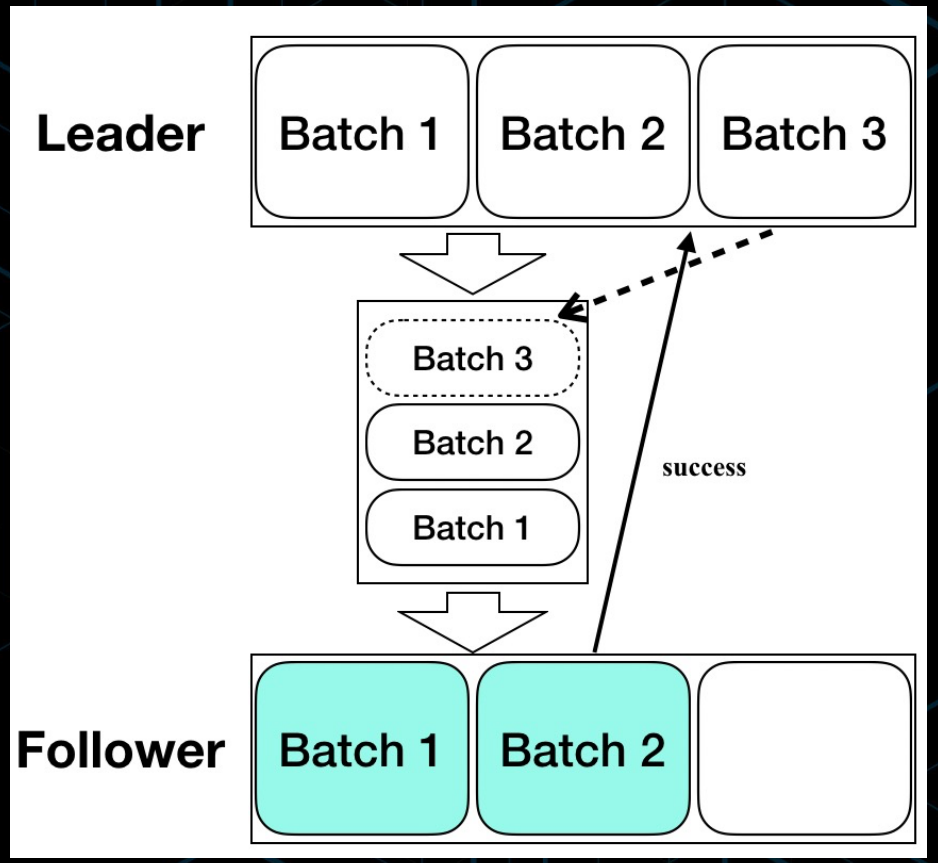
- 并发 ring buffer
- GitHub：https://github.com/LMAX-Exchange/disruptor



Replication pipeline: 流水线复制







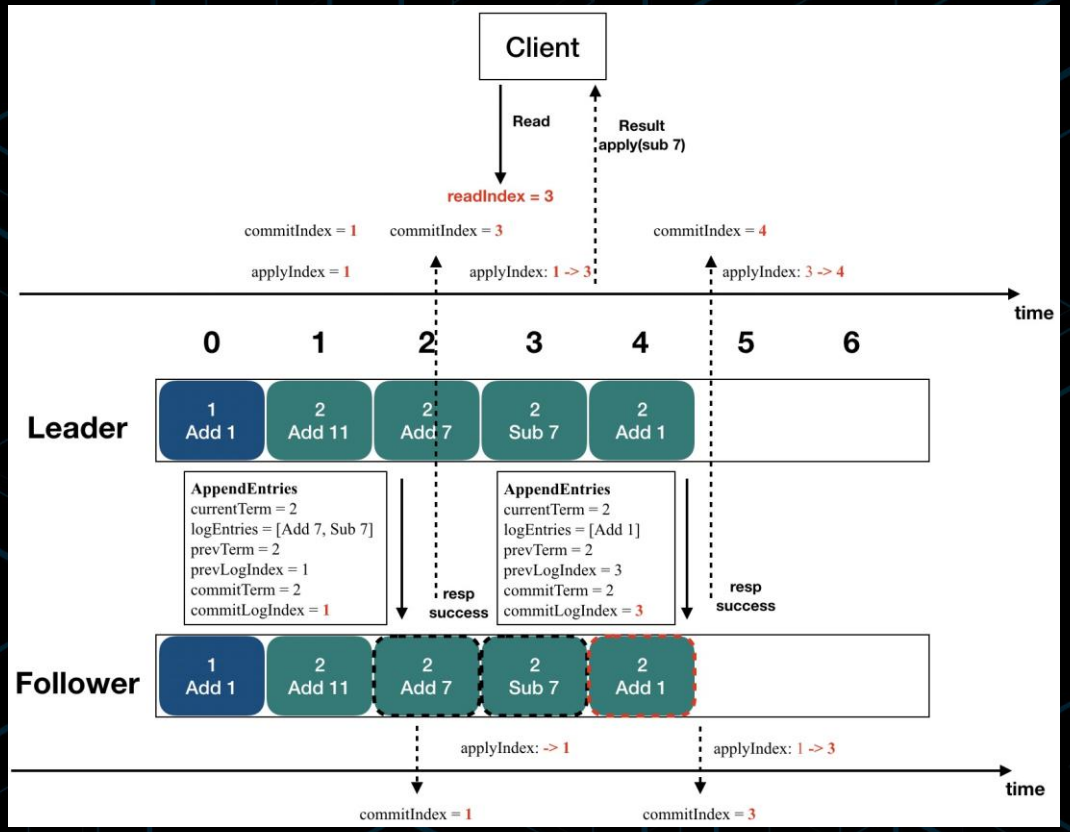
Linearizable read

线性一致读：分布式的 volatile 语义

- Client 发起读请求；
- Leader 确认最新复制到多数派的 LogIndex；
- Leader 确认身份；
- 在 LogIndex apply 后执行读操作。

租约读：省去 Leader 和 Followers 额外的网络交互

- 给 Leader 一段租期，期间身份不会被剥夺；
- 依赖机器时钟的准确性。



Benchmark

性能测试

- 环境 & 条件：
  - Server：3 台 16C 20G 内存；
  - Client：2-8 台 8C；
  - Multi-Raft-Group：24 个 Raft 复制组。
- 更多详情：https://github.com/alipay/sofa-jraft/wiki/Benchmark-数据

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

现状

- Latest version：v1.2.4 (2019.03.24)
- Wiki：https://github.com/alipay/sofa-jraft/wiki
- Example 详解：https://github.com/alipay/sofa-jraft/wiki/Counter-例子详解
- 公众号文章：《蚂蚁金服开源 SOFAJRaft：生产级 Java Raft 算法库》

欢迎关注 SOFAStack 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