

蘑菇街分布式消息中间件 在多场景下的架构演进

江龙

北京

伦敦

纽约

旧金山

圣保罗

上海

东京



全球软件开发大会

[上海站]

主办方

Geekbang > InfoQ
极客邦科技

2017年10月17-19日
上海·宝华万豪酒店

信息安全

人工智能

机器学习

黑产

互联网金融 (FinTech)

云计算

基础设施

团队管理

微服务

软件性能

硅谷

互联网架构

→ 扫描二维码
开启软件开发新思路





Geekbang | EGO EXTRA GEEKS' ORGANIZATION
极客邦科技

EGO会员招募季

EGO旨在组建全球最具影响力的技术领导者社交网络，联结杰出的技术领导者学习和成长。

2017年6月30-7月10



扫 码 报 名

SPEAKER
INTRODUCE

江龙 石头

- 2011年加入美丽联合
- 消息，数据分发，SLB



SPEAKER
ArchSummit 2017 Shenzhen

TABLE OF
CONTENTS 大纲

- 开源消息产品的使用及Corgi的背景
- 架构迭代
- Paxos协议在消息中间件中的应用
- 降低延迟的一些手段
- 延迟与消息语义的关联
- 消息轨迹与海量队列的应用

开源消息产品的使用及Corgi的背景

从Redis到RabbitMQ到Corgi

- Redis时期
 - 随用随走，运维噩梦
- RabbitMQ时期
 - 运气驱动可用性
- Corgi时期
 - 努力降低开发人员心智负担，提升生活质量

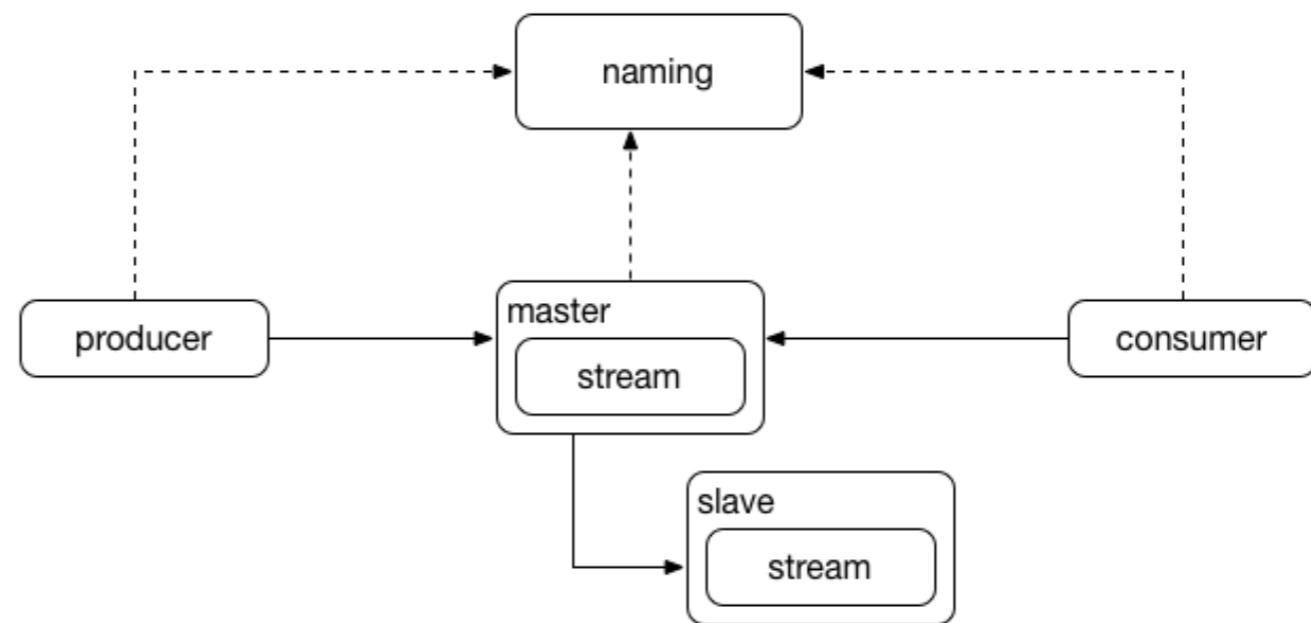
Messaging System

- 高可用(Always Writable)
- 高吞吐、可扩展、低延迟
- 多语义、多语言支持、多消费方式
- 功能丰富、多次消费、消息轨迹
- 灵活适配可用性
- ...

架构迭代

Corgi 1.x

- 中心化
- master/slave结构

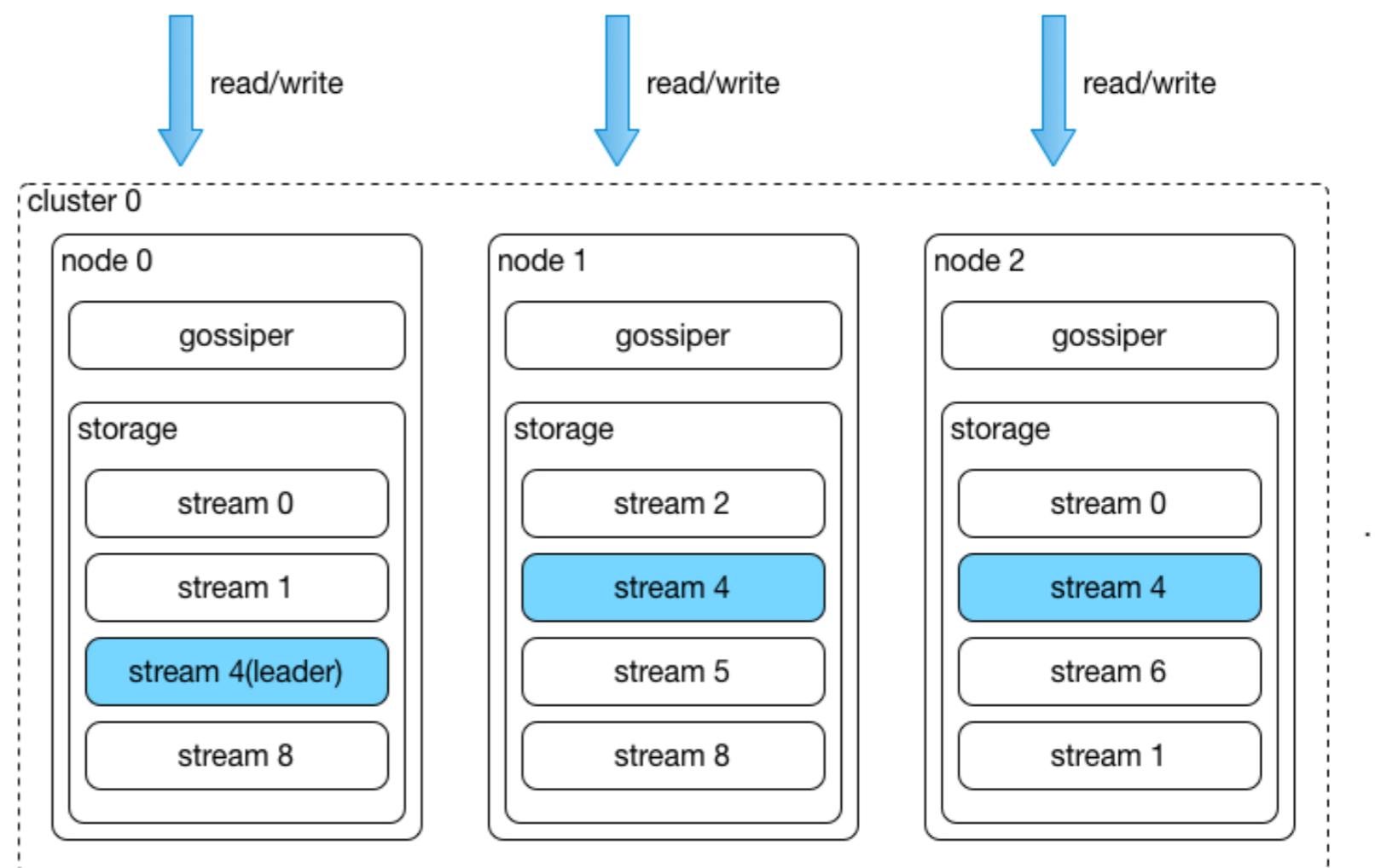


Corgi 1.x

- 顺序与可用性
- 资源利用

Corgi 2.x

- 一切皆消息
- 强一致性
- Gossip
- 灵活适配可用性

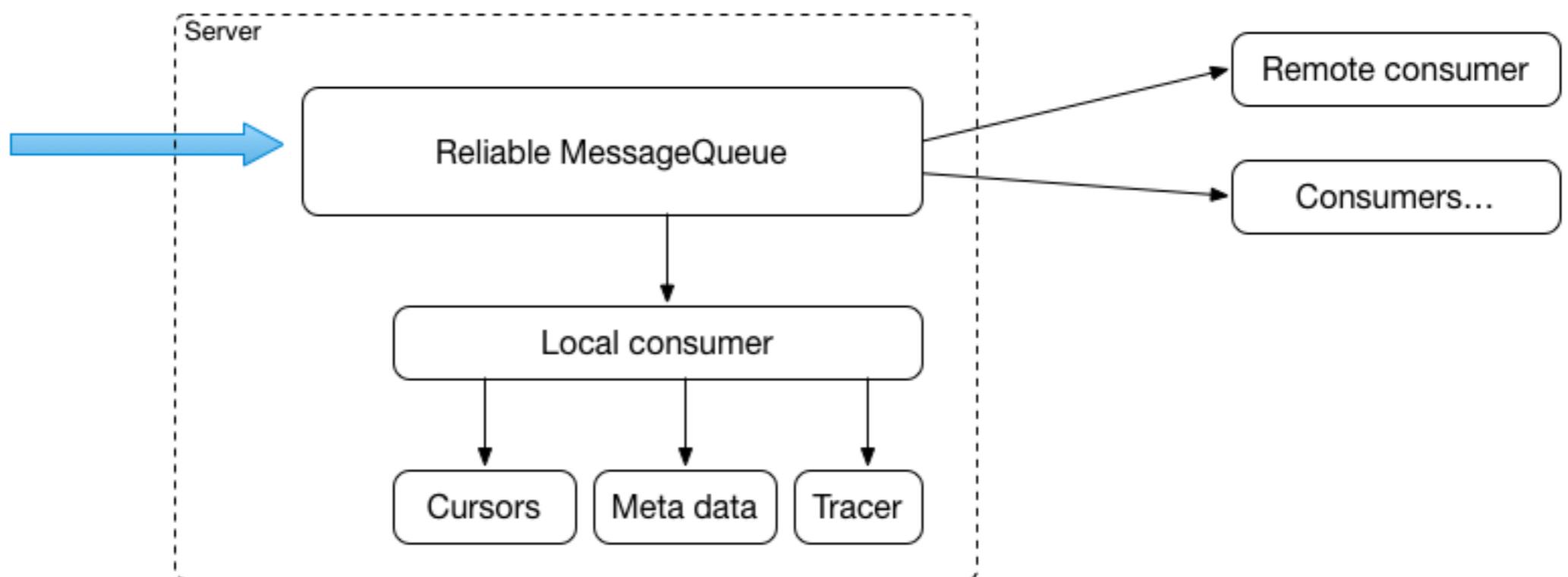


一切皆消息

- Meta data(topic, group...)
- Checkpoints
- Transactions
- Messages
- Events(produce/consume)

Everything is a message

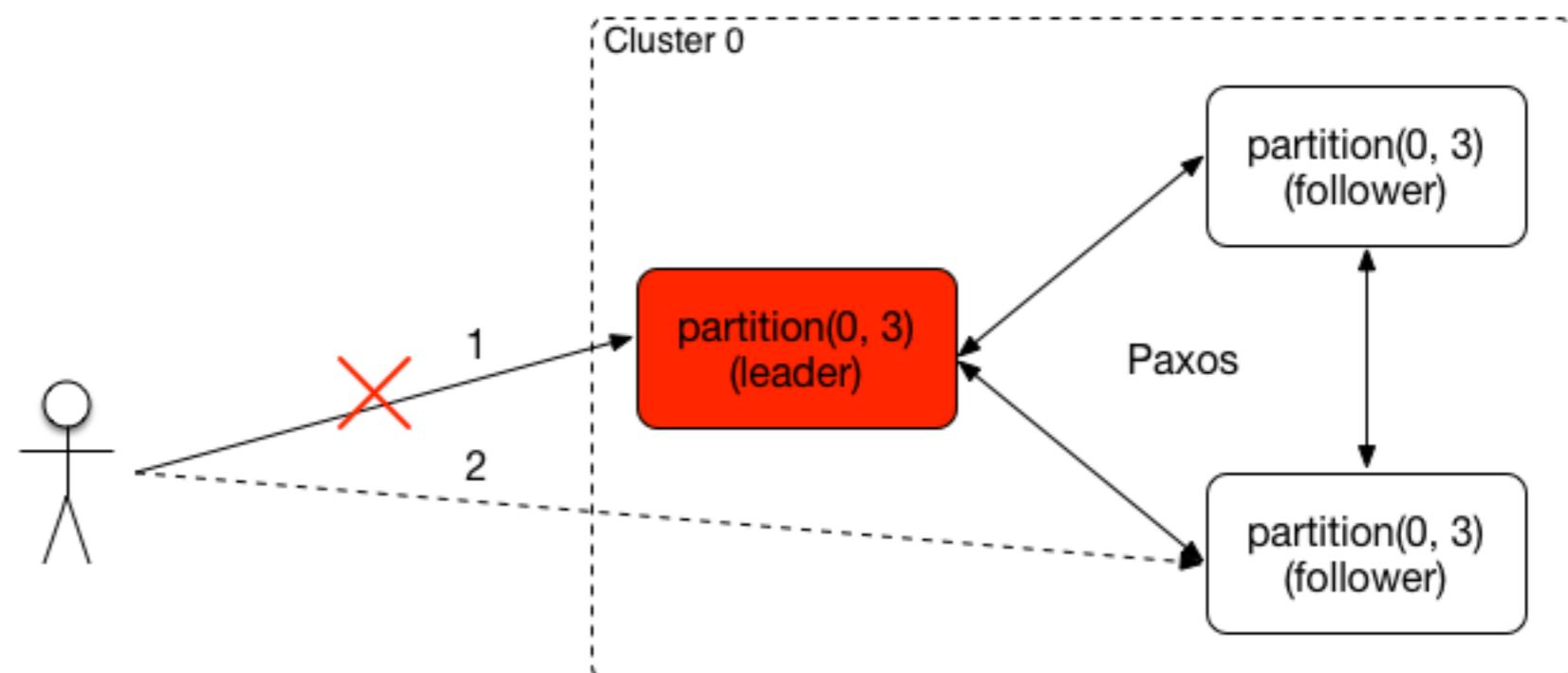
- 消息可靠则一切(消费结果、元数据等)可靠



Paxos协议在消息中间件中的应用

Why Paxos?

- 可用性、一致性、顺序兼得
- 多点可写，优先leader



正确性？

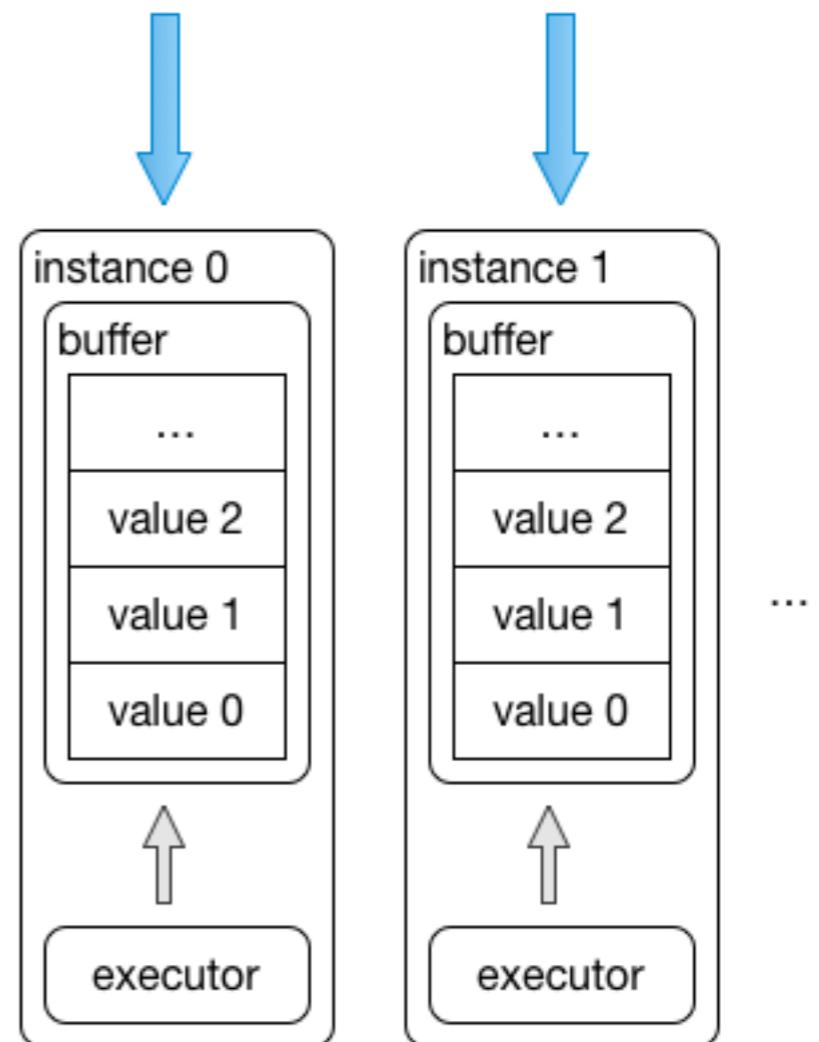
正确性

- 基于场景(正常/异常)的测试
- 异常模拟
 - 通讯故障(异常)模拟
 - 存储/机器故障模拟
- 开源参考(XCom)

性能？

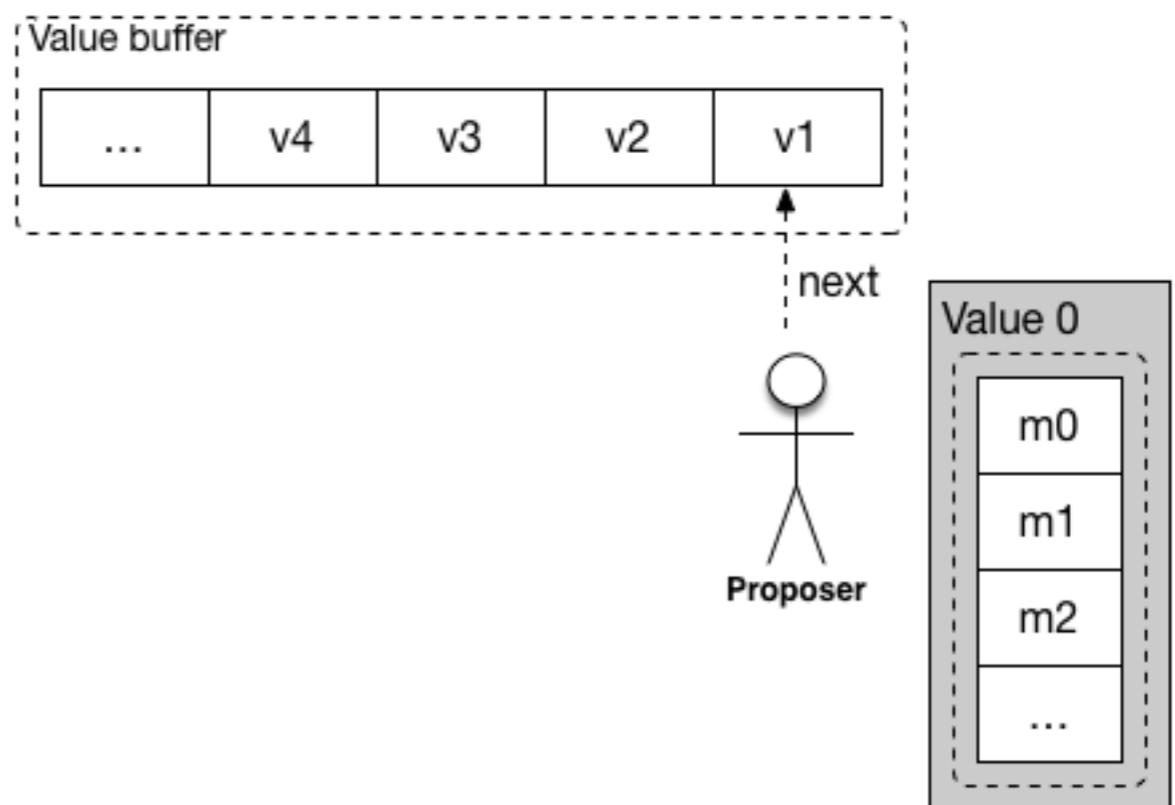
Paxos性能优化 - 并行执行

- 运行实例: Partition mod Instances
- 资源共享, 状态独立
- 实例之间无序



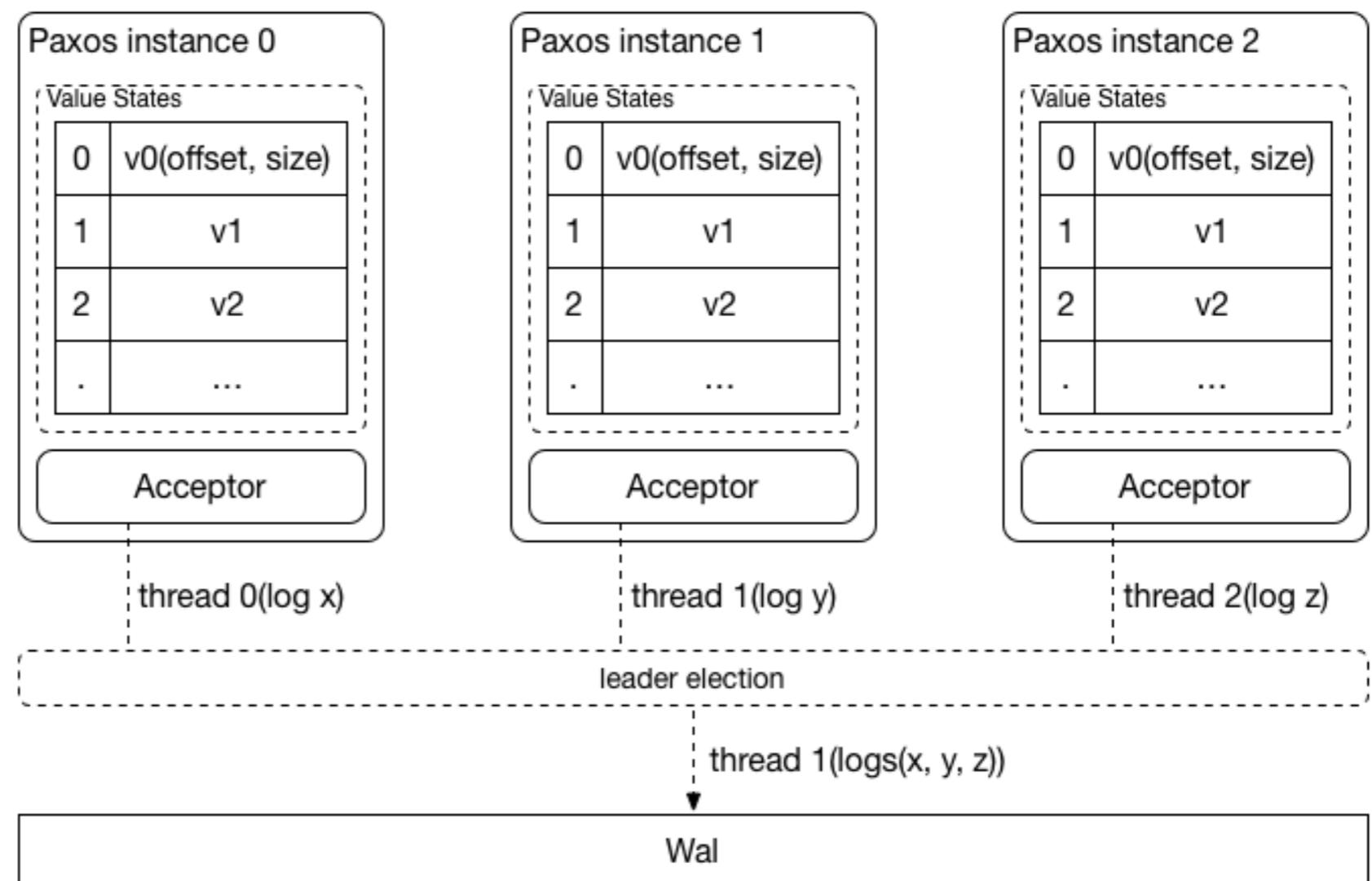
Paxos性能优化 - 消息合并

- Value包含多个message
- 并发越高，效果越好



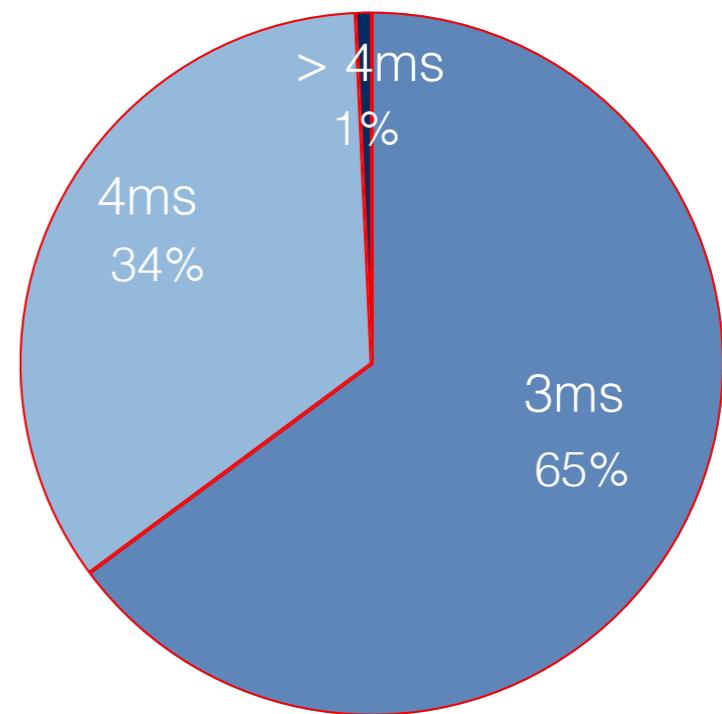
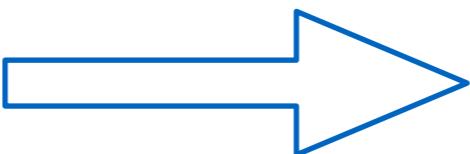
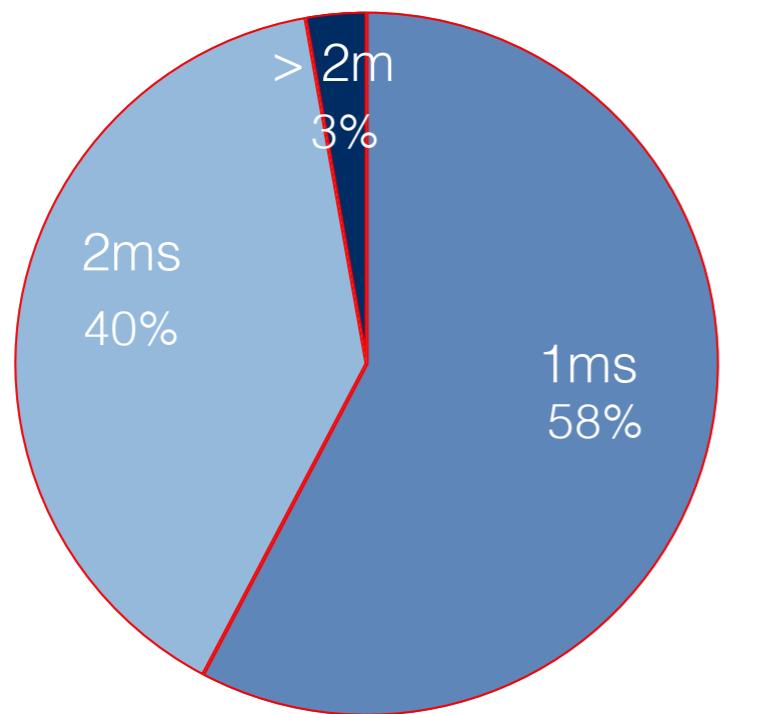
Paxos性能优化 - I/O合并

- 共用Wal



Paxos性能优化 - 仅Leader可写

- Prepare阶段优化
- 写入延迟对比(无GC情况下)
- RT?



性能总结

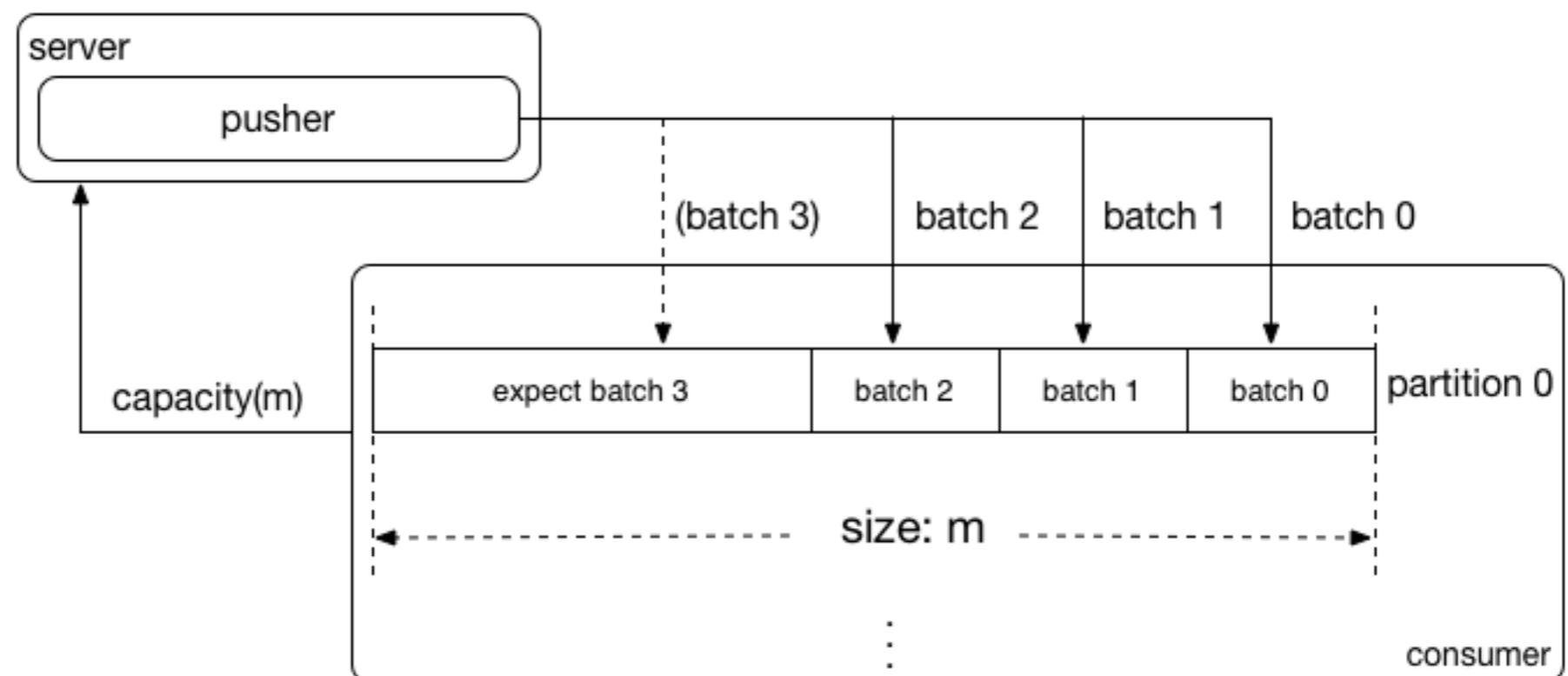
- 可接受的Latency
- 小量下跌的吞吐 (30%~)

降低延迟的一些手段

Pull or Push?

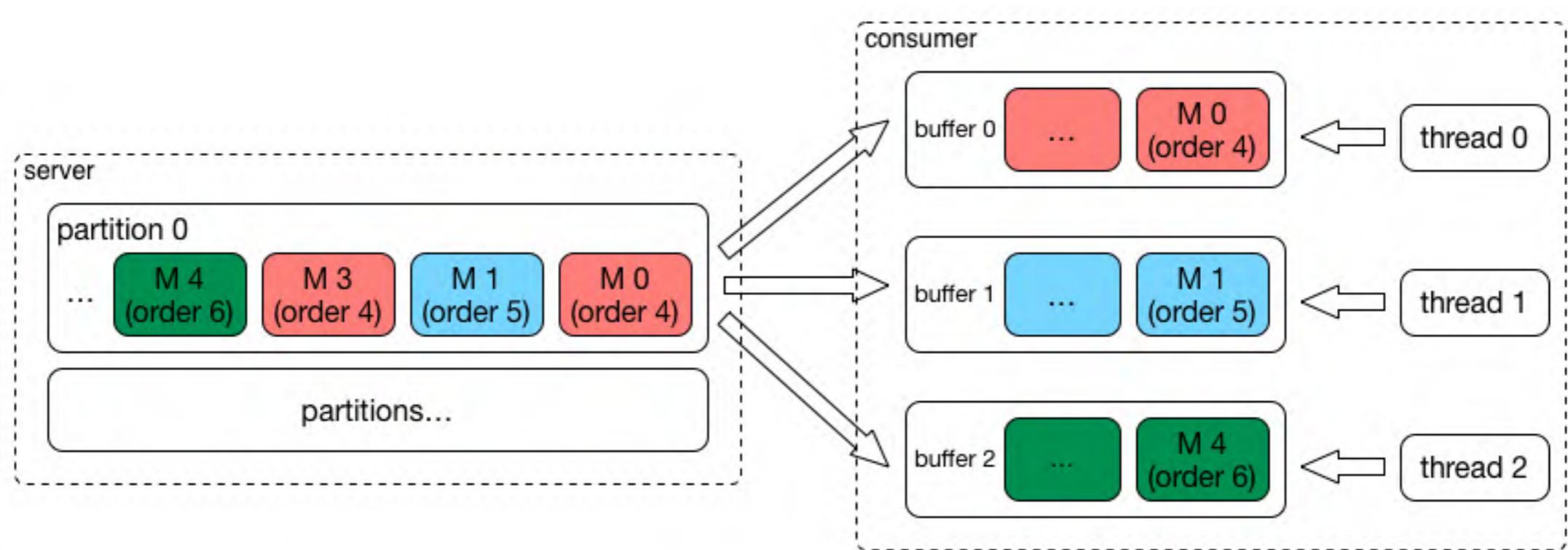
Dynamic Pull/Push

- Pull vs Push
- 区别对待fast/slow consumer



顺序与并行

- 最大化处理能力
- Message Key决定顺序



延迟 - Fast Consumer

- Echo Consumer
 - 10 ~ 15ms(NO GC)
 - 随着消息量的上涨, latency基本不变

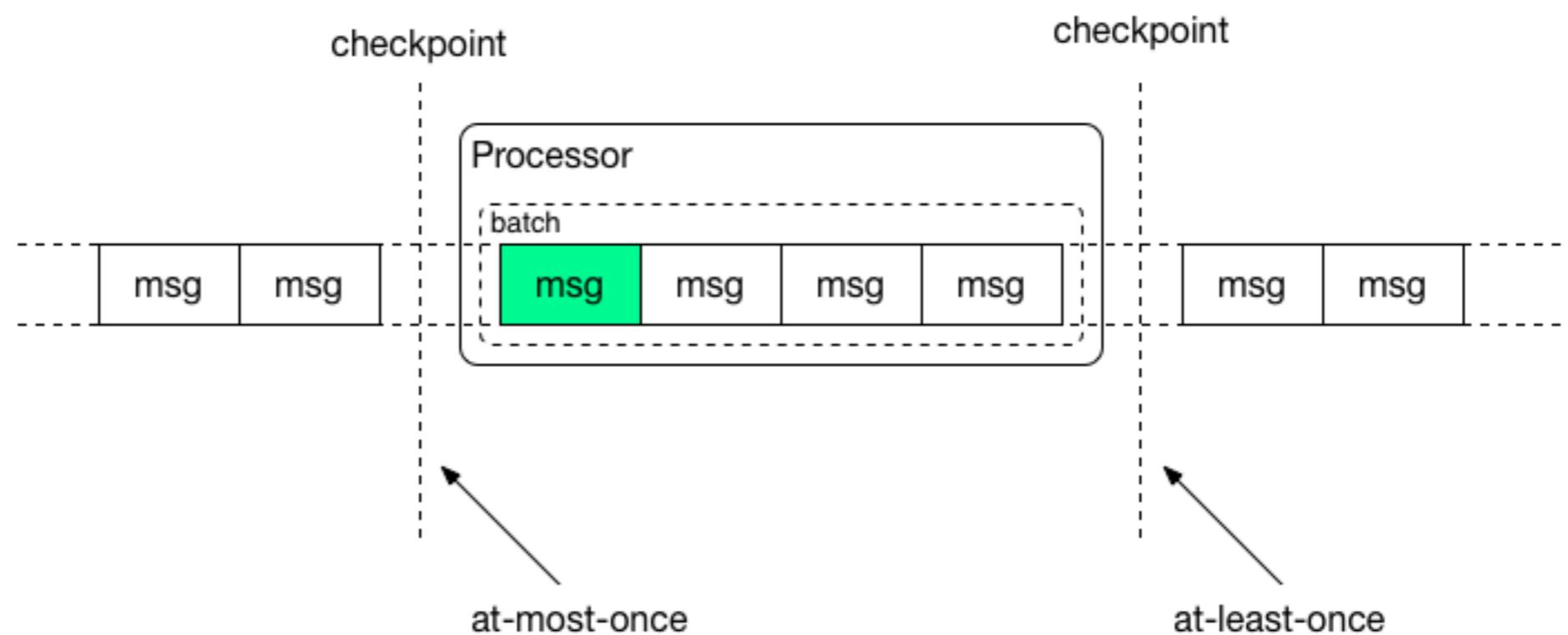
低延迟与消费语言

消费语义

- At-most/least-once
- Exactly-once

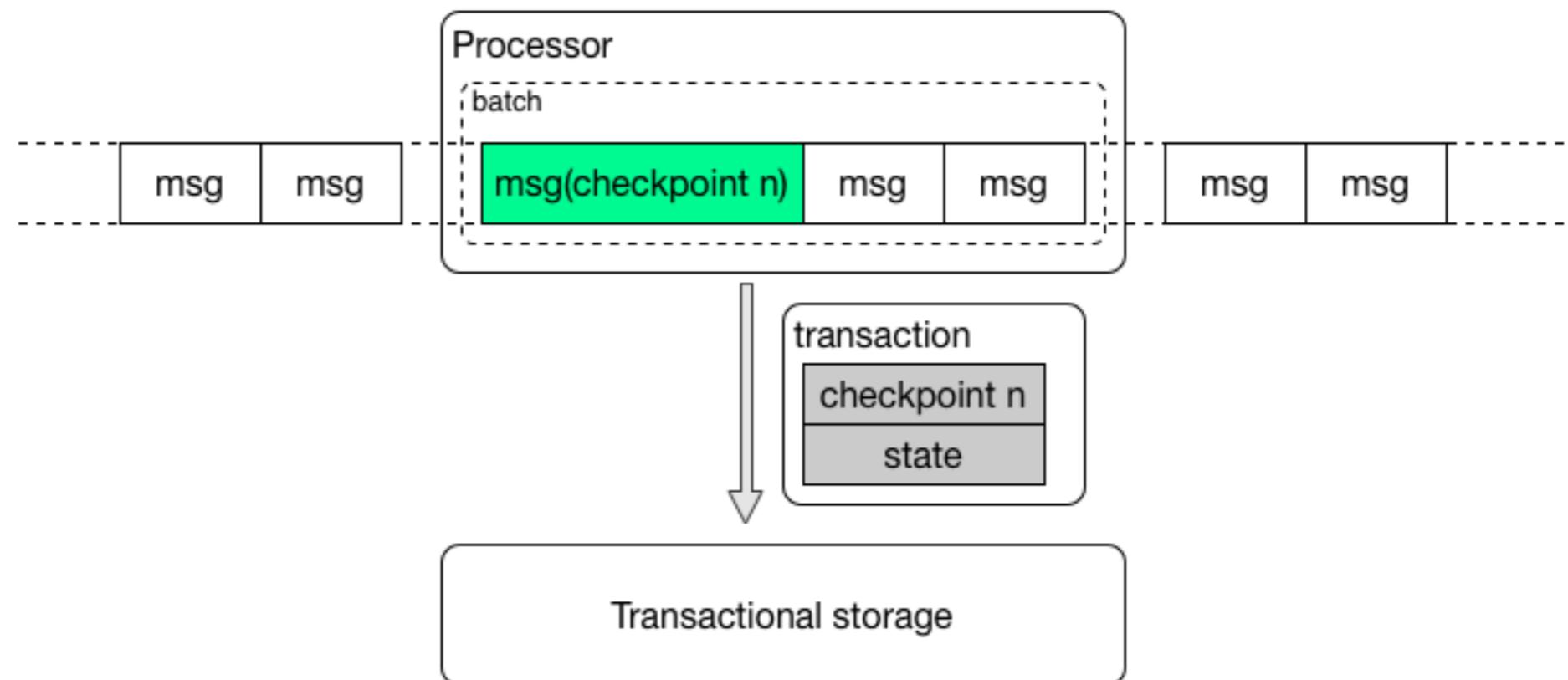
At-most/latest-once

- Replication策略对at-most-once的延迟影响

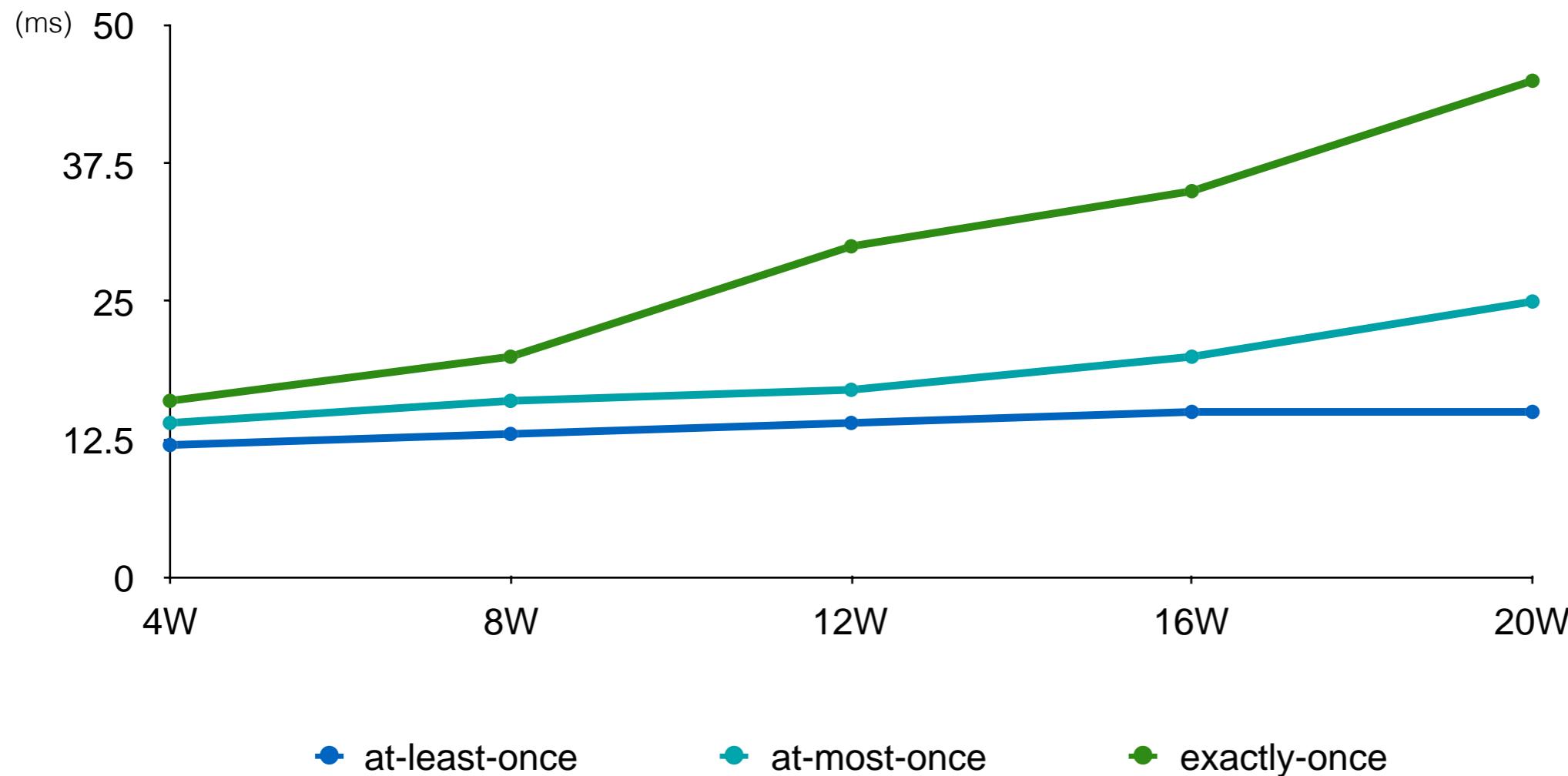


Exactly-once

- atomic(checkpoint & consume result)
 - latency



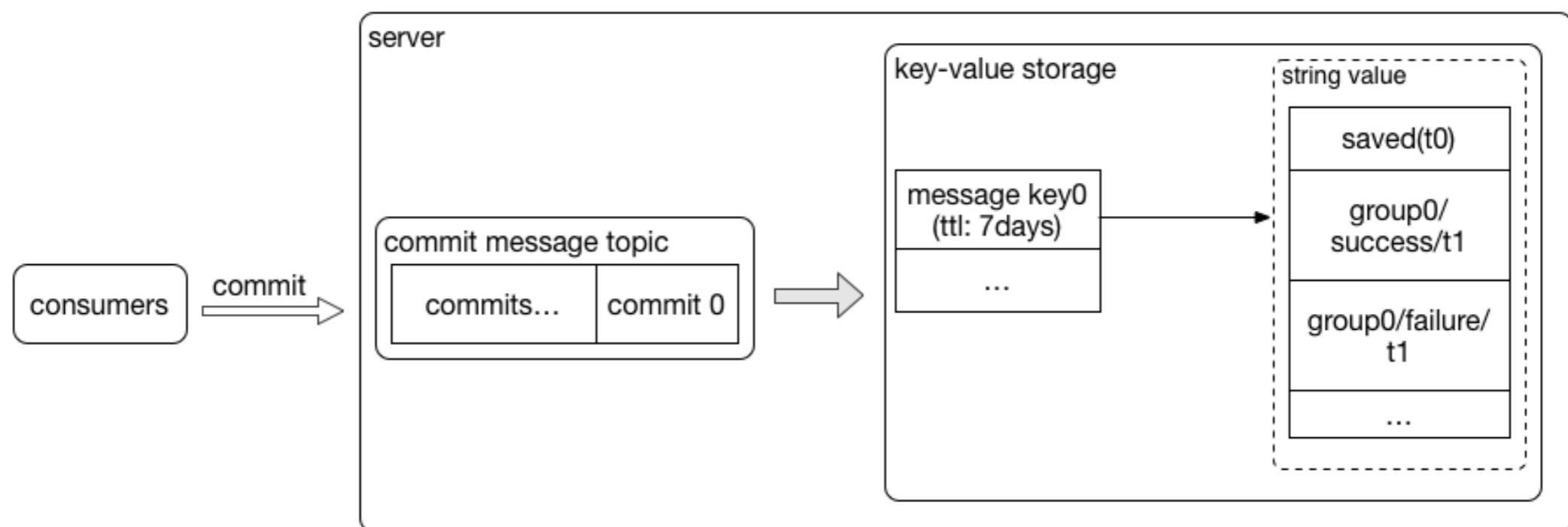
延迟对比 - Echo Consumer



消息轨迹与海量队列

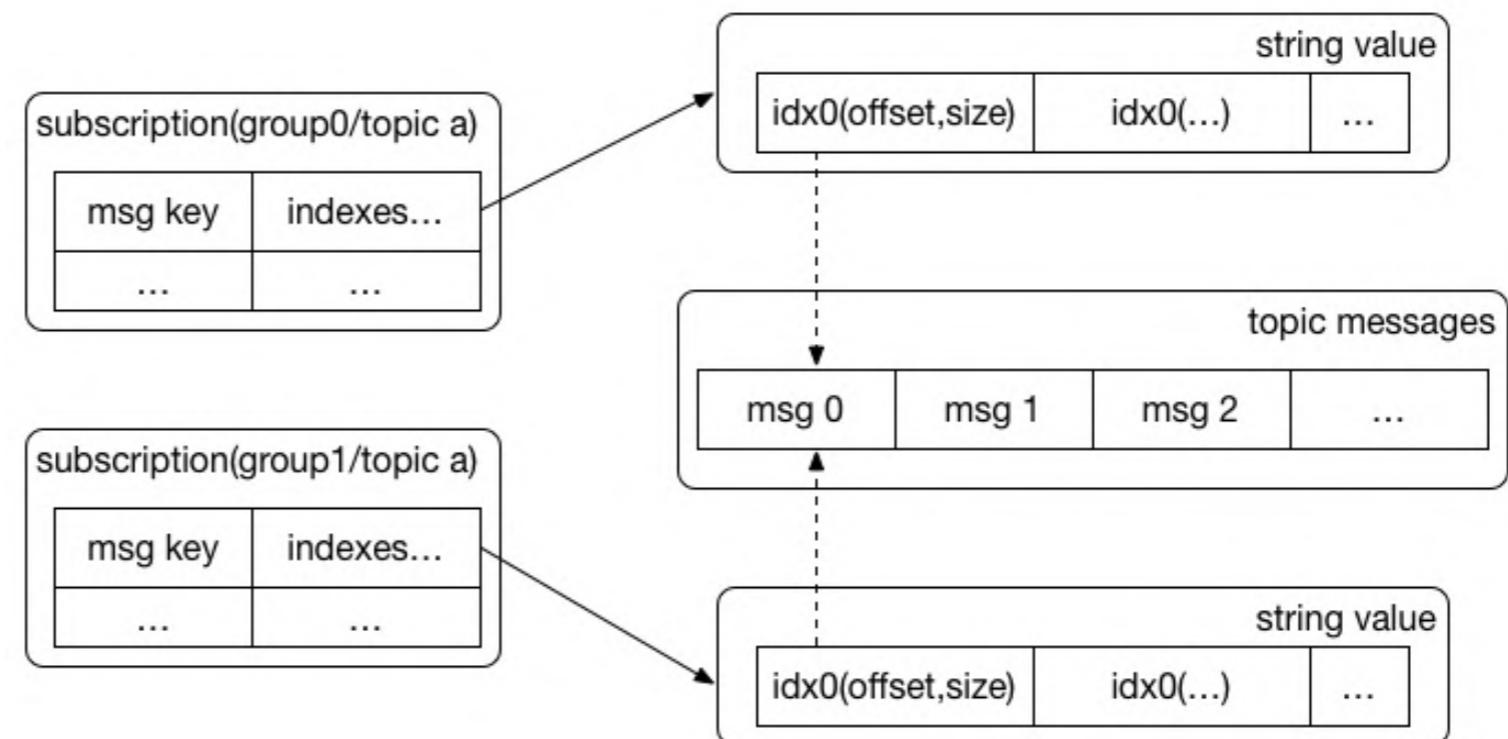
消息轨迹

- 结果(生产、消费)可追踪
- 架构内聚，不依赖三方存储



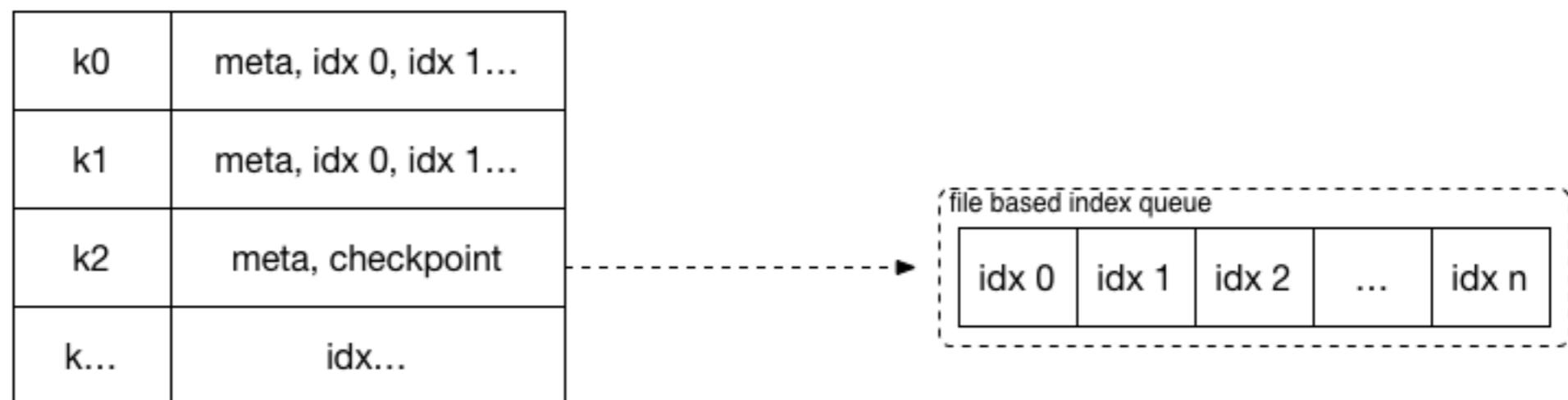
海量队列

- Per User/Object Per Queue
- 延后消费
- 低延迟



海量队列 - 热点处理

- “网红”队列
- 内存与文件队列转换



THANKS!

让创新技术推动社会进步

HELP TO BUILD A BETTER SOCIETY WITH
INNOVATIVE TECHNOLOGIES

Geekbang®

极客邦科技



专注中高端技术人员的技术媒体



高端技术人员学习型社交平台



实践驱动的 IT 教育平台

