消息队列设计与实现总结

Sunzhidong

msn: everestsun@126.com

提纲

- 系统概述
- 应用场景
- 后续发展
- 项目总结
  - 设计
  - 实现
  - 优化
  - 思考

Baidu 百度

消息队列概述

- 消息队列技术是分布式应用间交换信息的一种技术。它简化了应用之间数据的传输，屏蔽底层异构操作系统和网络平台，提供一致的通讯标准和应用开发，确保分布式计算网络环境下可靠的、跨平台的信息传输和数据交换。
- “百度知道”

BMQ 介绍

- BMQ 是百度 Ecom 和 IBase 两部门共同合作研发，基于 pub/sub 模式的消息队列集群服务
- 产品特性
  - 管理性：多产品线统一运维，支持多 topic
  - 安全性：用户管理，读写权限验证

BMQ 介绍

- 产品特性
  - 可靠性：保证消息在异常情况下不丢失、不重复
  - 有序性：保证消息有序送达所有消费者
  - 可用性：异常情况下的 Failover 容错机制
  - 易用性：提供 C++/PHP 的客户端 API，屏蔽通讯细节
  - 高性能：多客户端聚合性能异步 5K/s，单客户端同步 1K/s 无 slave 模式 7K/s

Baidu 百度

MQ 应用场景

- 异步通讯应用
  - —Mailbox\IM
- 增量可靠传输
  - —Yahoo Pnuts\cm-transfer
- 社区消息广播
  - —Twitter\Feed
- 复杂系统解耦

后续发展

- 集群可扩展能力
- 提升单点的可靠性
- 单点切换提高可用性
- 增强系统状态监控
- 制定运维管理系统

后续发展

- 定制化的下游消费需求
- 丰富的多语言 API
- Restful 接口支持
- 特殊需求的扩展
  - 生产消费：单一的中间消费者
  - 点对点：单一的终极消费者

项目总结

- 设计
  - 明确系统目标特性，逐一突破
  - 慎重引入开源、第三方产品库
  - 不要凭空想象出一些特殊需求
  - 多考虑便于测试、运维的需求
  - 缩小迭代的 Gap ，连续非跳跃

Baidu 百度

项目总结

- 设计
  - 架构设计
    - 集群与分布：中心节点（单点）
    - 分布可管理性：问题 <- 中间层
    - 如何可扩展：无状态设计
    - 轻量级设计，与其避免异常，不如可恢复

项目总结

- 设计
  - 数据模型
    - 需求决定了模型
    - 数据校验、压缩、索引
  - 内存管理
    - Cache 设计
    - 同步一致性需求

项目总结

- 设计
  - 负载均衡
    - 极端情况下的负载问题
    - 热点的发现和避免
  - 静态 Vs 动态
  - 主动 Vs 被动
    - Pull /push
    - 懒惰更新，延迟策略
    - 外部驱动

项目总结

- 开发
  - 什么更重要？
  - KISS + DRY + SOLID
  - Holley Wood ： don’t call me, we will call u
  - 防御性编程的不信任法则：
    - input/output check
    - inner stat check
    - double check

Baidu 百度

项目总结

- 开发
  - 提高可单测性
  - 降低对外依赖
    - 低耦合 <- 高内聚
  - 代码设计分层
  - 关于代码重构
    - refactor early and often
      - DRY 事不过三

项目总结

- 开发
  - 关于异常
    - 严格发现，延迟处理
    - 不稳定，不要过多容错
  - 多线程问题
    - 各种锁之外
      - Lock free
      - 引用计数
      - 数据 + 状态

系统优化

- 优化（软件 + 硬件）
  - 架构、流程、算法、代码
  - 明确性能目标和限制
  - 抓住瓶颈重点优化，权衡 ROI
  - 不要过早、过度优化
  - 通用影响性能，特有流程优化

系统优化

- 优化
  - —网络模型选择， socket 40ms pro
  - —降低锁粒度，防止多重锁

    Write 24/s->5000/s
  - —尽量应用异步接口和模型
  - —迭代开发，尽早建立测试基准
    - 便于分析和定位变化的根源

后续思考

- 架构
  - 多 slave 是否必要
  - 网络线程模型优化
- 优化
  - 消息同步转发方式
    - Slave Group Commit Vs 滑动窗口
    - Common Batch Commit Vs 滑动窗口
  - Slave 对性能的影响
  - 滑动窗口对性能的影响
  - 滑动窗口的动态调整

后续工作

- 日志完整性
  - 线上问题的可跟踪性
- 深藏的 bug 某一天会灵魂附体
- 试运行和监控
  - 监控，监控，还是监控

结束语

谢 谢