FaRM

优缺点

- 优点
  - 软硬一体化
  - 同时优化CAP
- 缺点
  - backup 为了保证一致性，不提供读，降低吞吐量
  - 集群只是在数据中心中，没有跨区域
  - 应用程序需要处理部分事务一场

可用性

- Mechine Down
  - 错误检测
  - 使用极短租约（5ms）
  - 重新配置
  - 使用 zookeeper 序列号保证一致

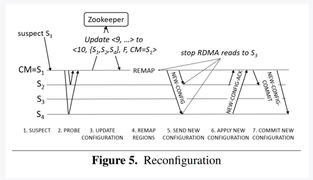

Figure 5. Reconfiguration

- 事务状态恢复
  - 当接收到NEW-CONFIGCOMMIT消息时，所有机器处理其日志中的所有记录
  - RDMA 不被处理
  - 选择要恢复的事务

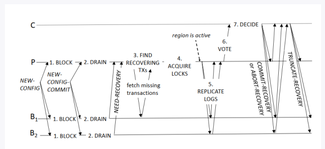

Figure 6. Transaction state recovery showing a coordinator C, primary P, and two backups B_1 and B_2

- 恢复数据 & 恢复分配器状态

目标

- 在 CAP 三个方面都不妥协
- 高可用的支持分布式事务的高性能内存存储系统

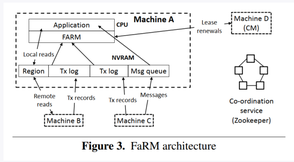

Figure 3. FaRM architecture

架构

- Cluster 集群
- 架构组成
  - Machine
    - 集群中的每一台机器
      - 对象 包含 64 bit 版本和数据内容，版本用于并发控制 相当于数据
      - 区域（region）
        - 多个对象聚合在一起存储，成为 Region 每个 Regions 2G
        - 多个数据放在一个区域
      - 主从复制 1 个 Primary 多个 backup，对象写入读取都在 primary，backup 只做热备份
      - Tx log 事务 log，用于事务恢复
      - Msg Queue 消息队列，用于 RPC
    - Machine 之间可以互相通信（分布式事务）
  - CM（ConfigManager）
    - 集群中只有一台
    - 管理 leases，regions 在 machine 间的分布，检测失败，故障恢复等管理工作
  - Zookeeper 协调服务器

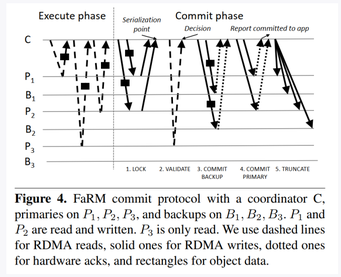

Figure 4. FaRM commit protocol with a coordinator C, primaries on P_1, P_2, P_3, and backups on B_1, B_2, B_3. P_1 and P_2 are read and written. P_3 is only read. We use dashed lines for RDMA reads, solid ones for RDMA writes, dotted ones for hardware acks, and rectangles for object data.

一致性事务

- 保证一致性
  - Lock（加锁阶段）
    - 因为使用版本控制，所以只针对写请求加锁
    - 协调器将 lock 记录写入对应写入对象的机器的日志中
    - 其中包括了 primary 所有写入对象的版本和新值
    - 检查锁 锁失败的情况
      - 旧版本：事物更旧
      - Lock：已经被上锁
    - 发送消息报告上锁是否成功
  - Validate（验证阶段）
    - 只针对读请求，只发送给 primary
    - coordinator 从对象的 primaries 读取读验证
    - 如果任何对象被改变，则验证失败同时事务回滚
  - Commit backups
    - 只针对写请求，只发送给 backup，用来保证一致性
    - coordinator 写一个 COMMITBACKUP 记录给非易失性日志，写给每一个 备份服务器，并且等待备份服务器的 ack 回复
  - Commit primaries
    - 只针对写请求，只发送给 primary
    - 在收到所有 backup 的 ack 应答后，协调器给每一个 primary 写 COMMITPERIMARY 记录到日志中
    - Primary 通过在相应位置更新对象来处理这些记录
      - 增加版本
      - 解锁
  - Truncate
    - 内存优化
    - 直到主从服务器被 truncate 之前，都在他们的日志里保存这个记录。协调器会在接受所有的 primary acks 之后惰性的对 primaryes 和 backup 进行 truncate logs
  - 原子读写 + 事务乐观并发控制 + 两阶段提交

软件支持

- 读写
  - 本地读写 or 远程单边 RDMA
  - RDMA 的原子操作（远端无CPU操作数据）
    - RDMA Fetch-and-Add（增加操作）
    - RDMA Compare-and-Swap（比较并交换操作）
  - 原子读
    - 一致性协议
    - 大 Value 支持 RPC
- 事务
  - 乐观并发控制
    - 无锁读
    - 验证有效性
    - 版本
    - 版本验证，数据是否被改变
  - 本地写缓存（针对远端写）
    - FaRM（Fast Remote Memory）在处理远程写入操作时使用本地写缓存（local write cache）来提高性能和效率
    - 本地写缓存是指在FaRM系统的计算节点上维护的一个缓存区域，用于临时存储远程写入操作的数据。当计算节点收到远程写入请求时，它将数据先写入本地写缓存，然后异步地将数据传输到远程节点上的内存。
- 架构设计
  - 使用主从备份取代复制状态机
  - 降低 CPU 瓶颈
  - 访问远端数据不需要远端 CPU 参与

通信支持

- RDMA
  - RDMA（Remote Direct Memory Access）是一种网络通信技术，允许计算机系统直接在远程主机的内存中读取和写入数据，而无需涉及中央处理器（CPU）的干预
  - 处理不同 mechine 之间的数据不使用 RPC
  - 数据包较小时：RDMA 吞吐大于 RPC
  - 使用单边 RDMA
    - 单边RDMA操作是指在进行数据传输时，只需要发送方的CPU参与，而不需要接收方的CPU参与。通过使用单边RDMA操作，FaRM可以避免在远程计算节点上涉及额外的CPU开销和处理
  - 每台机器都配备了两个 InfiniBand FDR（56 Gbps）的网络适配器（NIC）
    - InfiniBand是一种高性能网络技术，常用于构建高性能计算和数据中心环境。FDR表示InfiniBand的一代规范，提供了每秒传输速率为56 Gbps的数据传输能力

硬件支持

- NVRAM：non-volative RAM
- DRAM + SSD + UPS
- 分布式 UPS（Uninterruptible Power Supply）是一种分布式的不间断电源系统

消除存储和网络瓶颈