

# **redis** 适用场景与实现

田琪

# 主题

- 数据类型与实现
- 持久化与复制
- 容量规划
- 适用场景
- 相关参数与优化

# redis 概述

a disk backed in-memory database

高性能网络接口 + 数据结构集合



# 网络接口

- 单线程 IO 复用
- aeEvent 封装 IO 与定时器
- 默认支持 10K 连接

AE\_SETSIZE (1024\*10)

- why not multi-thread
  - cache coherency
  - complexity with locks
  - Deadlocks

# redis 数据类型

- string
- hash
- list
- set
- sorted set

# redis 基础数据结构

- redisObject  
数据指针 + type + encoding
- 内存容量占用 ( 不包含数据本身 )  
16 字节 ( 64 位 , 不开启 vm )  
40 字节 ( 64 位 , 开启 vm )  
redis 里面所有存储对象 (key,value) 都  
redisObject 类型
- 案例  
set,get,incr 等

# redis list - 双向链表

- 内存占用

list 每个元素占用内存

listNode( 前后指针, 数据节点指针 )

数据本身指针 (robj)

24 字节 + 16 字节 = 40 字节

40 字节 + 元素本身大小

- 相关参数

list-max-ziplist-entries

list-max-ziplist-value

- 使用场景

关注列表, 粉丝列表 ...

# redis hash

- 数据结构实现
  - zipmap & real hashmap
- 内存容量占用
  - zipmap 更省内存
- 相关参数
  - hash-max-zipmap-entries
  - hash-max-zipmap-value
- 使用场景
  - 用户计数器



# redis set & sorted set

- set 数据结构实现  
hashmap (value 为 NULL)
- sorted set 数据结构实现  
hashmap(element->score)  
skiplist (按 element 插入有序)
- 相关参数  
Set-max-intset-entries
- 使用场景  
用户发件箱 mid

# hash & rehash

- 双 hash 表做 rehash
- 预规划减小 rehash 次数
- rehash 时每次操作迁移一个 bucket
- 相关参数

DICT\_HT\_INITIAL\_SIZE 4

REDIS\_HT\_MINFILL 10

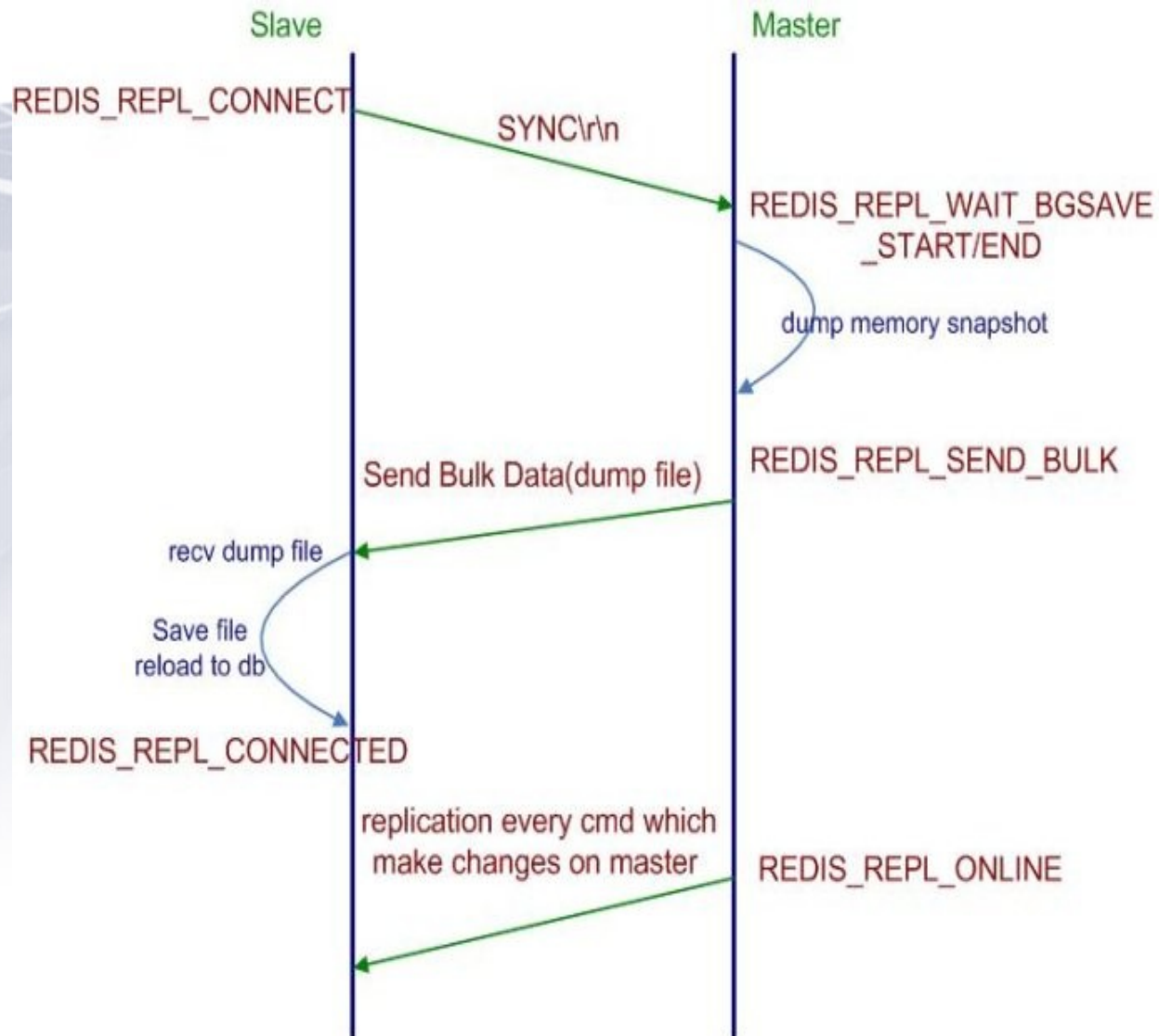
Activerehashing

# redis 持久化机制

- snapshot  
save 参数
- aof  
appendfsync 参数
- vm  
vm is not the way to go for the future  
slow restarts,slow saving,slow replication,complex code
- diskstore  
key,value stored on disk  
memory works as a cache ( read through,write back)

# redis 复制

- 实现机制  
快照同步
- 存在的问题  
无增量复制  
slave 表重建





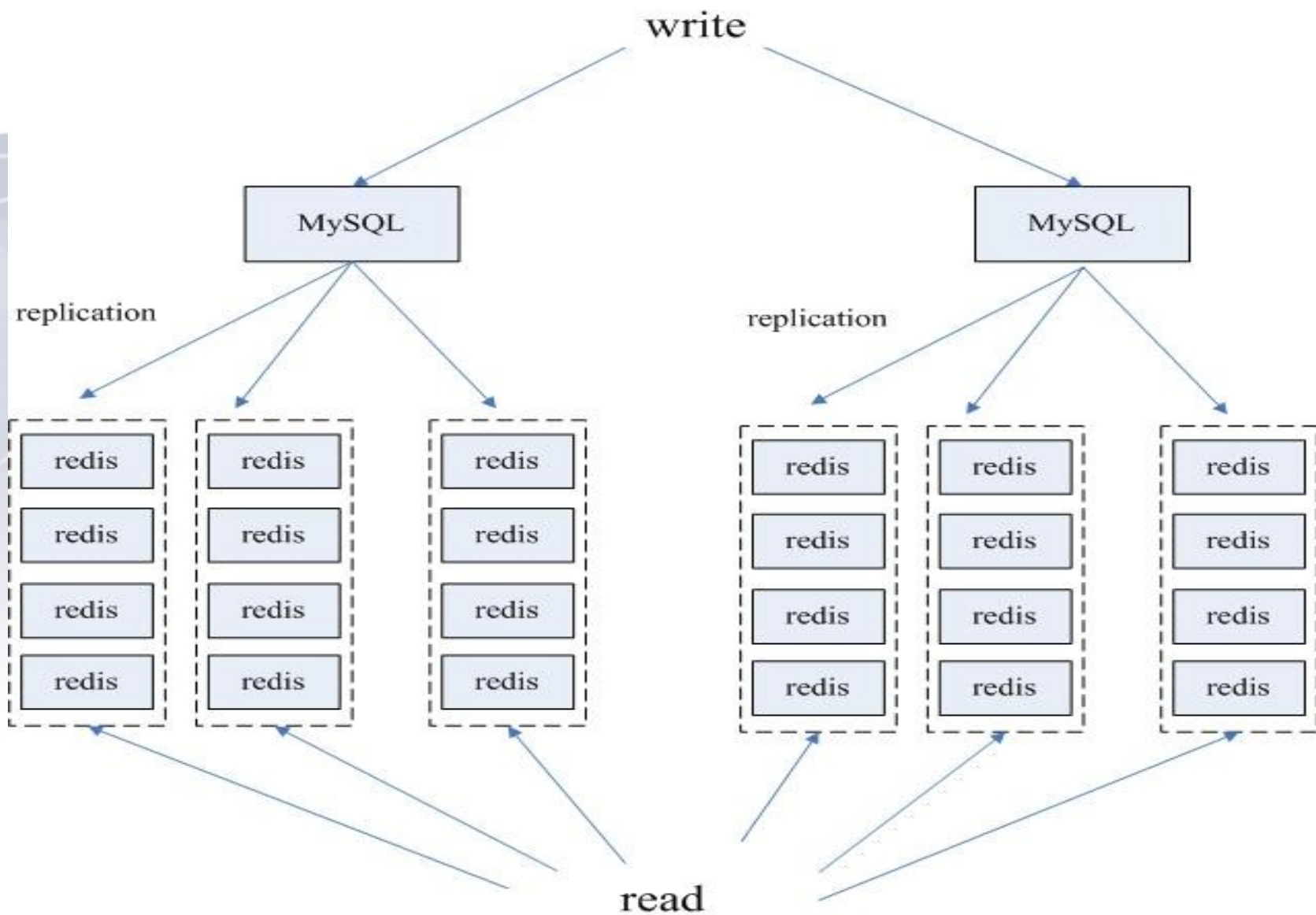
# redis 容量规划与建议

- 业务需要哪几种数据类型
- 预先做好容量规划
- **sharding** 策略与 HA
  - rolling/pre-sharding/ 双写 / 多 slave
- 使用 **pipeline** 减小网络 IO

# redis 适用场景

- MC + MySQL 不能完全解决 web2.0 的需求  
mc 穿透问题, 跨机房问题, ...
- 众多开源 nosql 百花齐放  
Cassandra, Riak, Voldemort  
Hbase  
CouchDB, MongoDB  
...
- what we need
  - 小数据量, 高速读写访问
  - 大数据量, 有明显热点数据
  - 大数据量, 无热点数据

# 异构读写分离



# 参数小结

- vm-enabled
- maxmemory
- save/appendfsync
- hash-max-zipmap-entries
- hash-max-zipmap-value



# 参数小结 (> 2.2)

- list-max-ziplist-entries
- list-max-ziplist-value
- Set-max-intset-entries

# 源码中部分参数

- DICT\_HT\_INITIAL\_SIZE 4
- REDIS\_HT\_MINFILL 10
- AE\_SETSIZE (1024\*10)
- REDIS\_SHARED\_INTEGERS 10000

# 小结

- redis 最佳使用方式是 in-memory
- 根据业务选择合适数据类型
- 做好容量规划以及 pre-sharding
- 不要过度依赖复制和持久化
- 使用 pipeline 减小网络 IO
- 先了解实现后使用，如 pubsub, scripting
- no one-size-fit-all product, understanding the tradeoffs

# Thanks

欢迎随时交流:

我的微博 : @ 摇摆巴赫

<http://t.sina.com.cn/bachmozart>

我的博客:

<http://www.tianqidev.com/>