redis运维之道

@jackbillow
jackbillow@gmail.com
2011-06-25

议题

1. redis应用历程
2. redis部署场景
3. 运维道与术
4. 挑战

redis规模

国际上最大的redis用户？

数据结构需求

Data Structure需求越来越多：

list

hashes

sets

sort sets

Memcached,MemcacheDB某些场合成为历史

性能需求

Scale-up→Scale-out→Scale-up MySQL又遇到瓶颈

<1%的Cache miss >> MySQL集群能支持的最大并发

多次Query带来的性能消耗越来越凸现

可靠性需求

Cache的“雪崩”问题让人纠结

Cache面临着快速恢复的挑战

开发成本需求

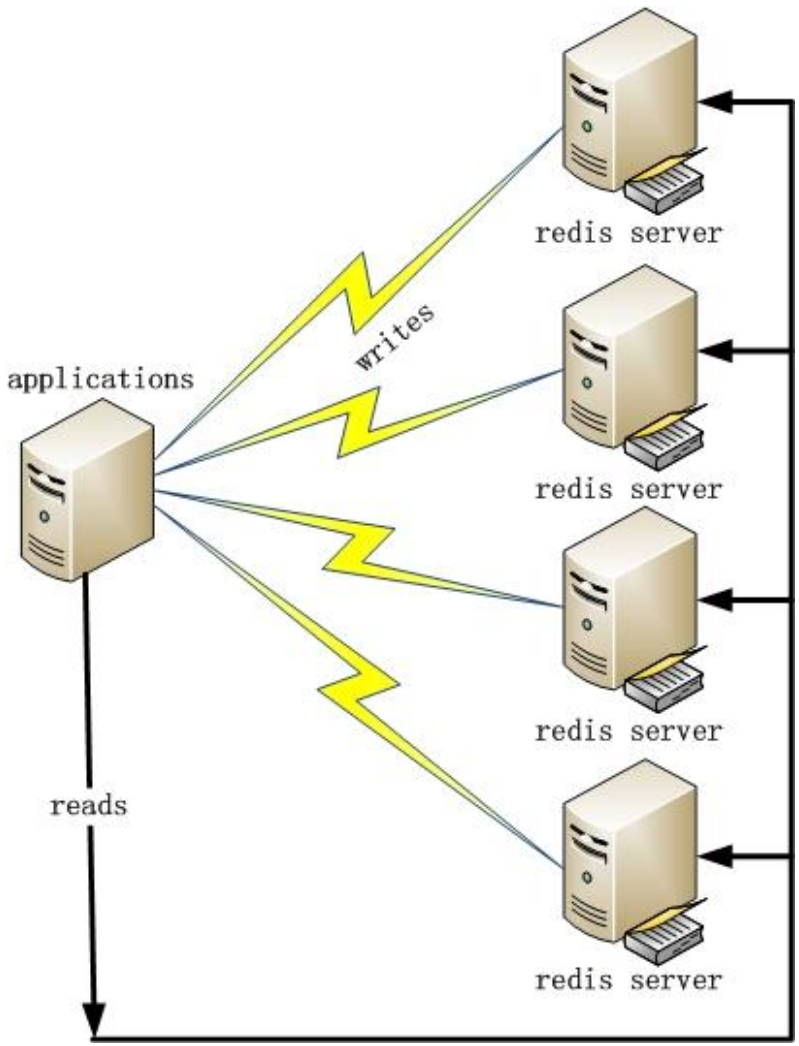
Cache和DB的一致性维护成本越来越高

开发需要跟上不断涌入的产品需求

redis部署场景

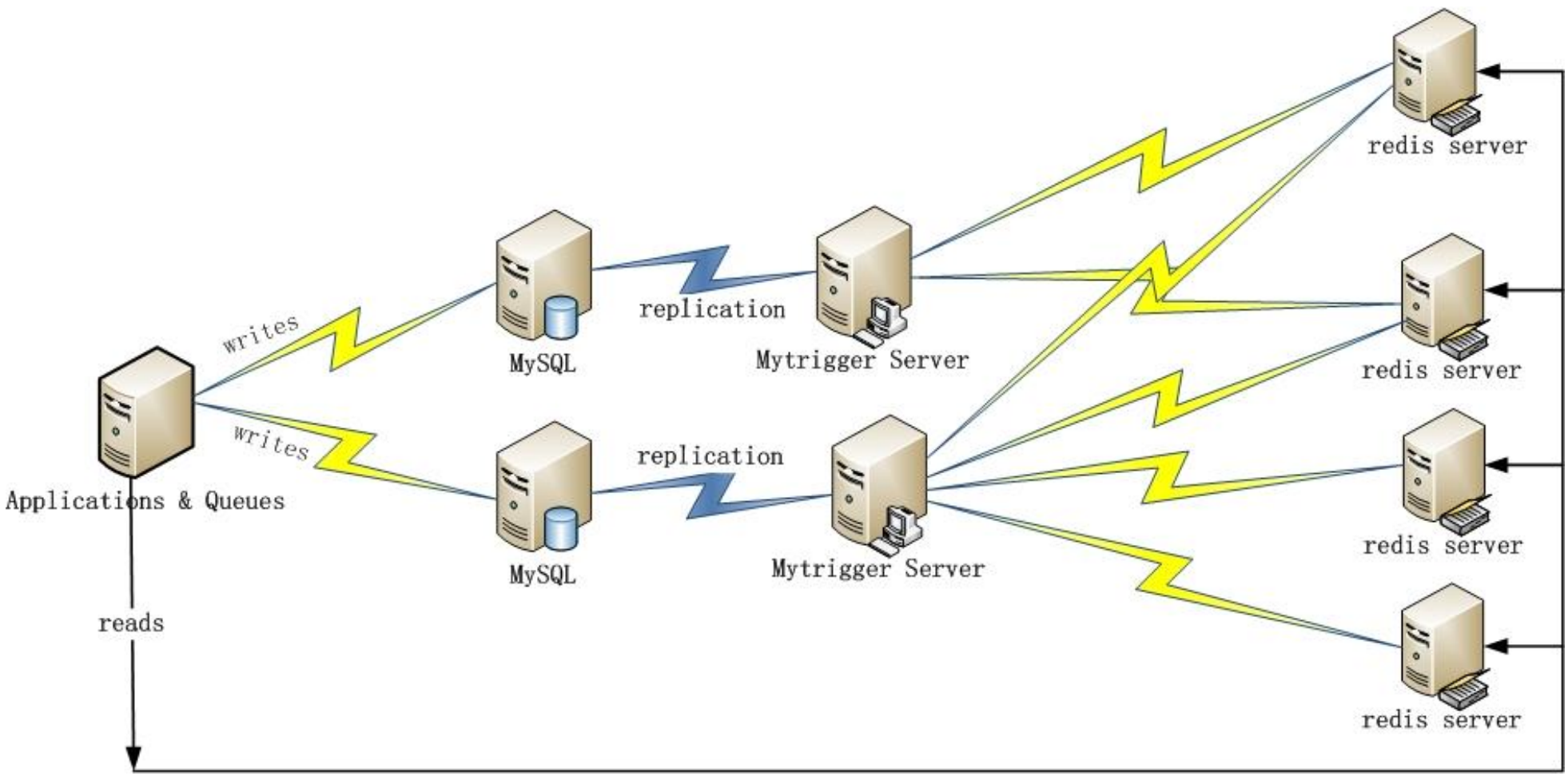
redis部署场景

Application → Redis



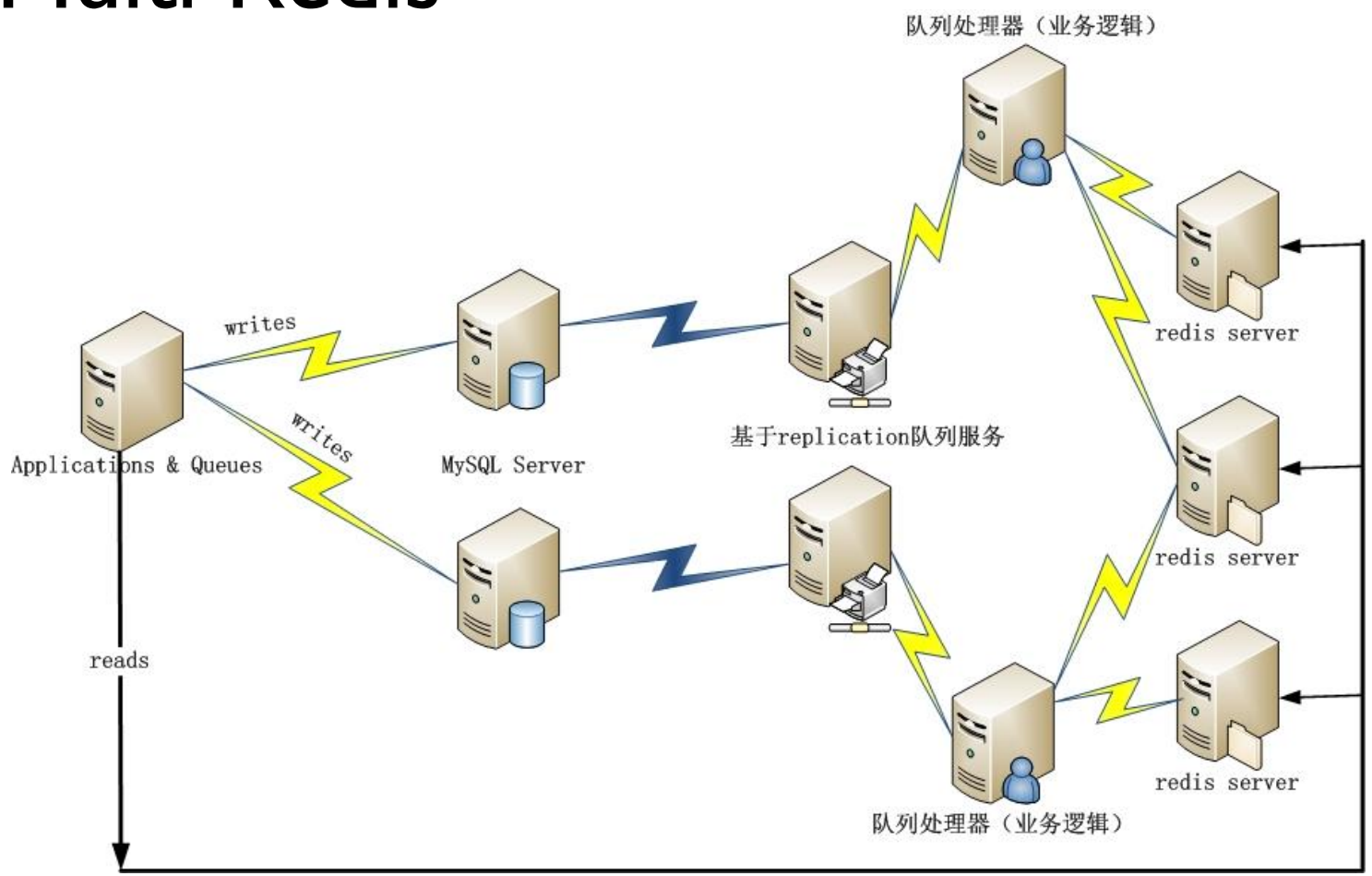
redis部署场景

MySQL → Mytrigger → Multi Redis



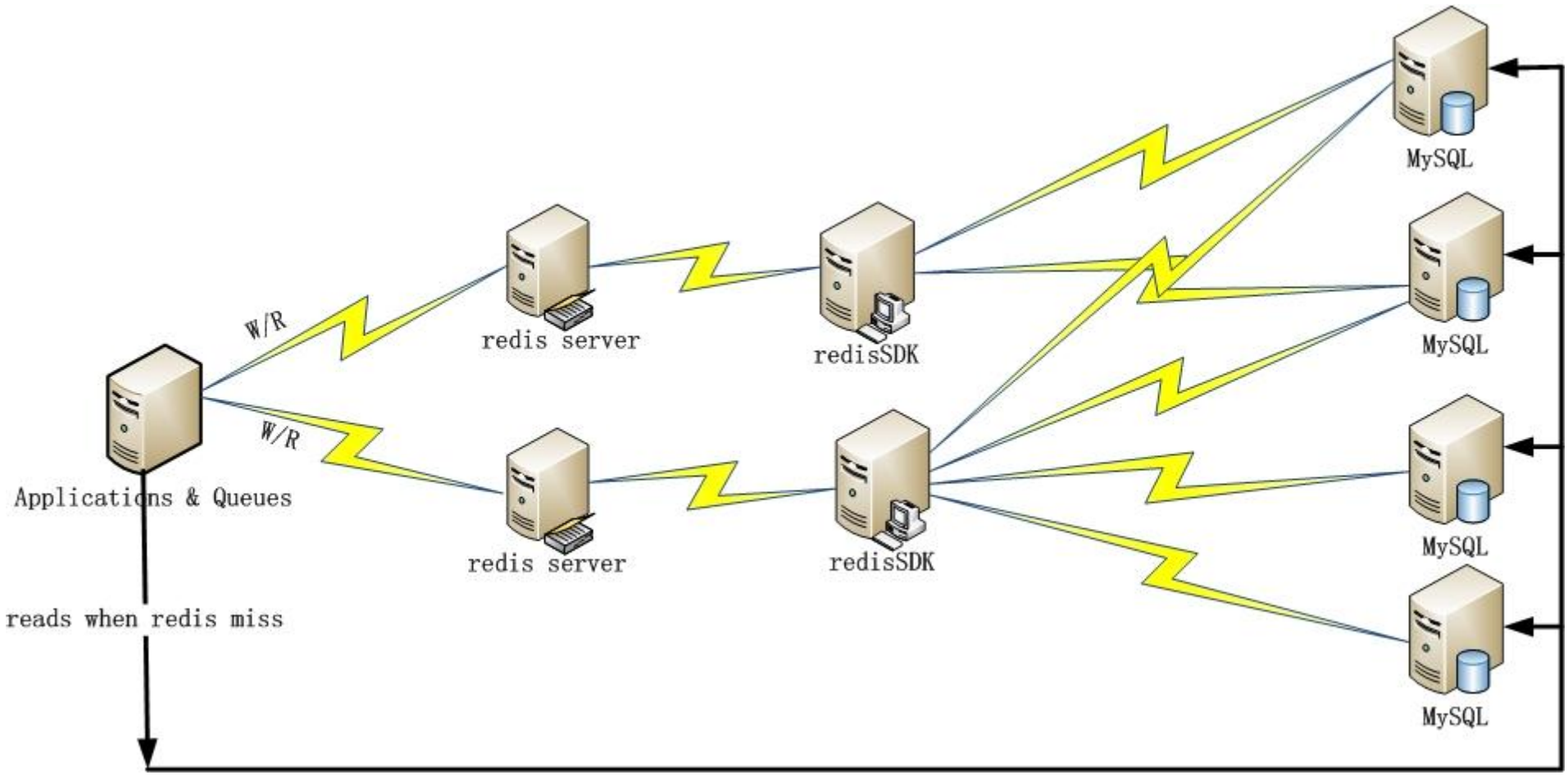
redis部署场景

MySQL→MytriggerQ→Queue Procesor→ Multi Redis



redis部署场景

Redis → MySQL



运维的道与术

sharding问题

多端口规划避免sharding

增加replication的filter功能

replication问题

基于rdb的replication →基于position的rdb + aof方式
aof文件大小可以定制
aof文件可以自动过期（避免bgrewriteaof）

performance和load问题

sorted sets

限制数据类型: filed int, score int
内存结构直接存入rdb

hashes

内存结构直接存入rdb

rediscounter

key: hash处理 value: int (16bit)

php客户端

改善phpredis, rediska的长连接问题
redisproxy:
php(socket) → redisproxy server（长连接）→redis

failover问题

master/slaves

多套部署

开发监控系统

实现cron bgsave

其他改进

bgsave带来aof写入很慢

fdatasync在做bgsave时不做sync aof

多核利用问题

taskset

统计信息

total_read_requests

total_write_requests

挑战

挑战

Cache还是Store?

挑战

完全代替Memcached?

挑战

不可避免的sharding问题如何优雅处理？

挑战

双写的容灾机制导致基础设施成本增加？

挑战

规模越来越大时如何减少运维成本？

谢谢！

Q & A

欢迎加入我们团队！

@jackbillow

jackbillow@gmail.com