主讲人：付磊

主页：http://carlosfu.iteye.com/

5-3.AOF

AOF

RDB有什么问题?

什么是AOF?

AOF三种策略

AOF重写

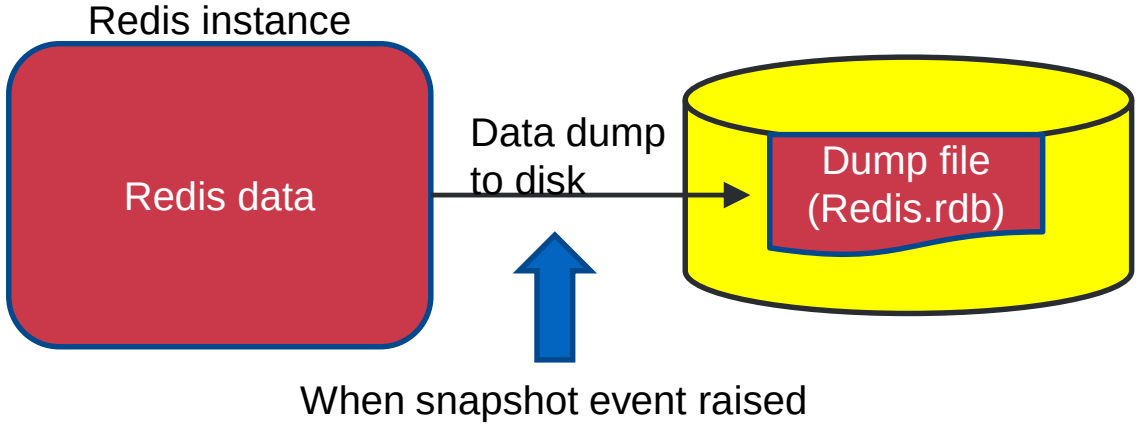
1. RDB有什么问题？

RDB有什么问题

耗时、耗性能

不可控、丢失数据

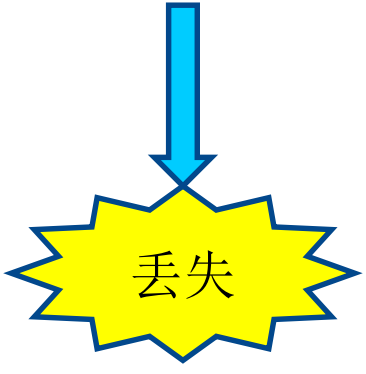
耗时、耗性能



- O(n)数据: 耗时
- Fork() may be swap
- Disk I/O：IO性能

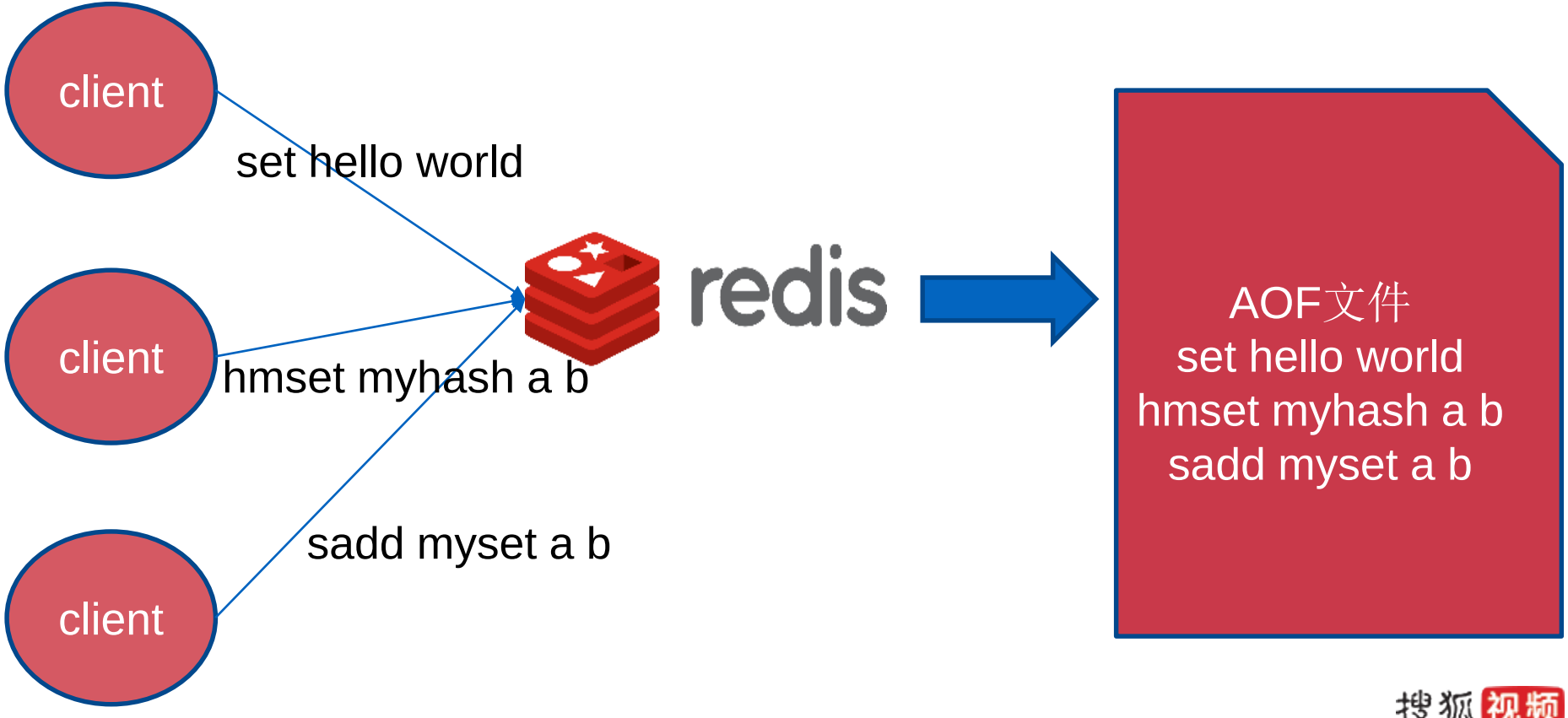
不可控、丢失数据

时间戳	Save
T1	执行多个写命令
T2	满足RDB自动创建的条件
T3	再次执行多个写命令
T4	宕机



2. 什么是AOF？

AOF运行原理-创建



AOF运行原理-恢复

载入

AOF文件
set hello world
hmset myhash a b
sadd myset a b

key	value
hello	String:world
Myhash	Hash:{“a”:”b”}
myset	Set:[“a”, “b”]

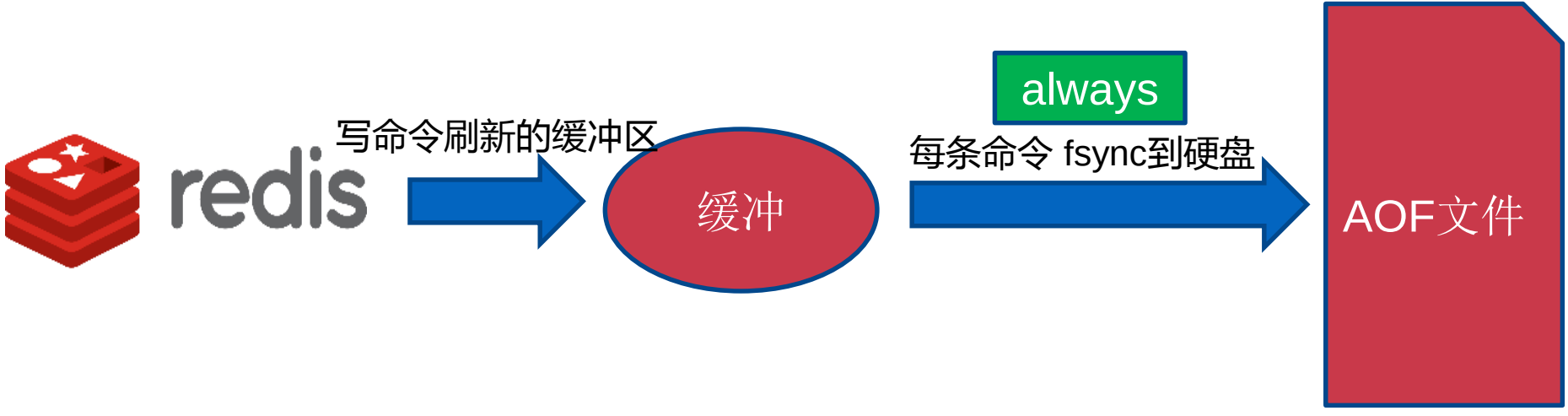
3.AOF的三种策略

always

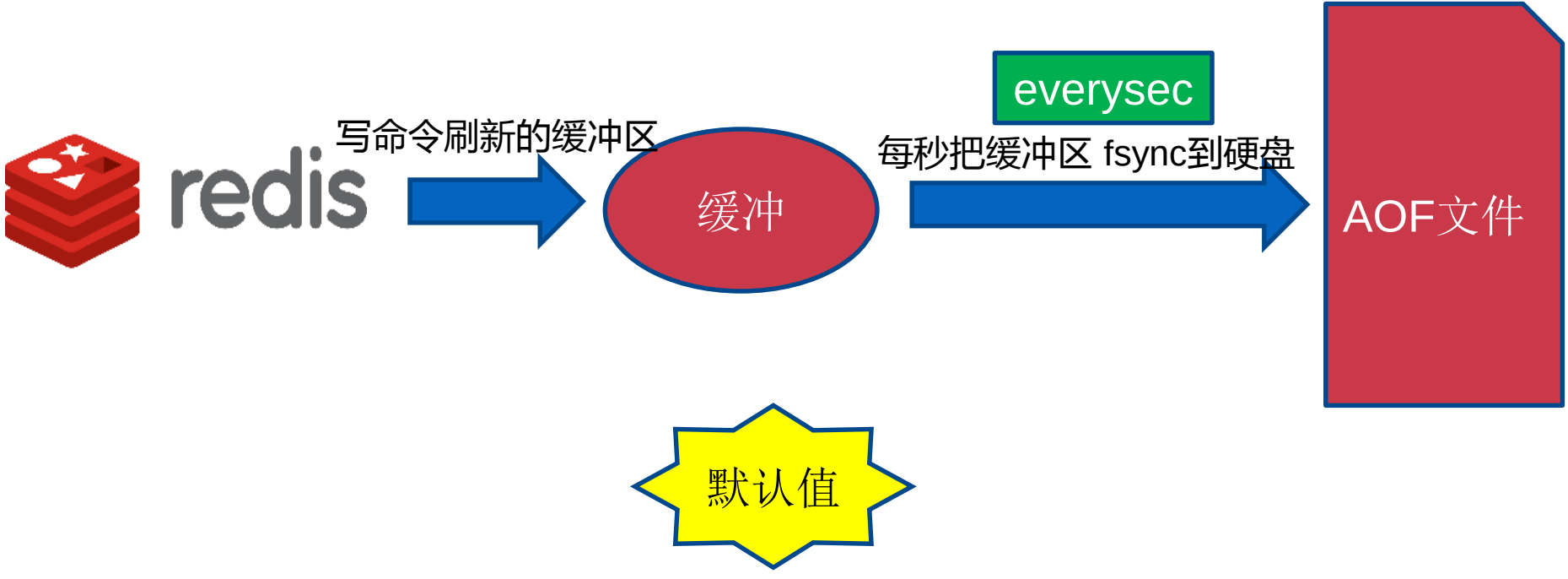
everysec

no

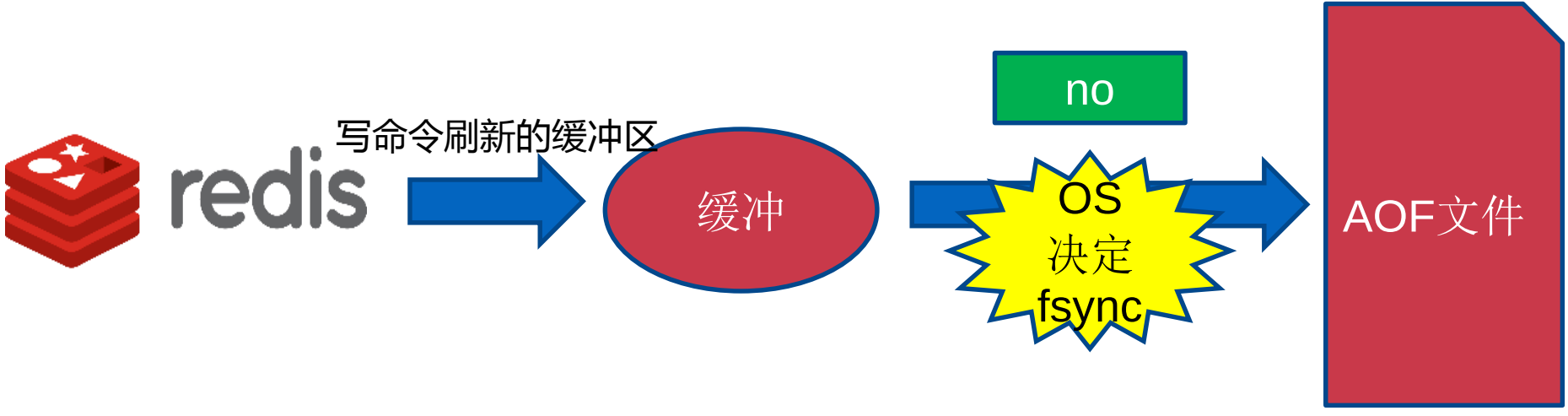
always



everysec



no



三种策略选择

always、everysec、no

命令	always	everysec	no
优点	不丢失数据	每秒一次fsync 丢1秒数据	不用管
缺点	IO开销较大	丢1秒数据	AOF什么时候重写 未知

4.AOF重写

原生AOF	AOF重写
Set hello world Set hello java Set hello hehe Incr counter Incr counter Sadd myset a Sadd myset b Sadd myset c Rpush mylist a Rpush mylist b Rpush mylist c	Set hello hehe Set counter 2 Sadd myset a b c Rpush mylist a b c

AOF重写作用

减少硬盘占用量

加速恢复速度

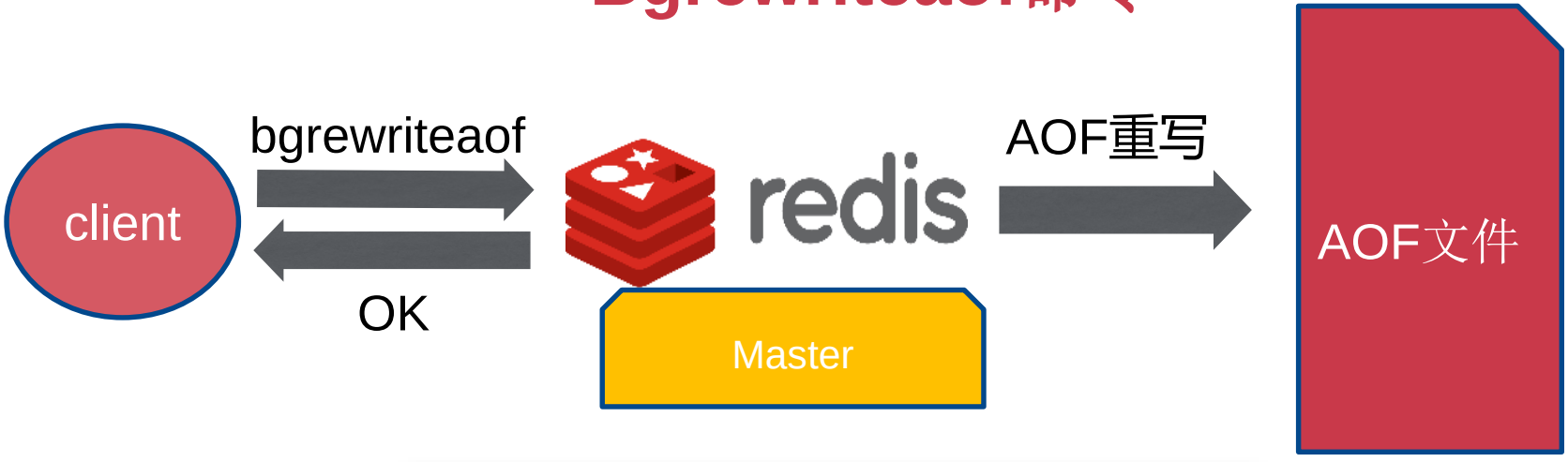
“不阻塞”读写

AOF重写实现两种方式

bgrewriteaof

AOF重写配置

Bgrewriteaof命令




```
redis> bgrewriteaof
OK
```

AOF重写配置

配置名	含义
Auto-aof-rewrite-min-size	AOF文件重写需要的尺寸
Auto-aof-rewrite-percentage	AOF文件增长率

配置


```
appendonly no
appendfilename "appendonly-${port}.aof"
appendfsync everysec
no-appendfsync-on-rewrite no
auto-aof-rewrite-percentage 100
auto-aof-rewrite-min-size 64mb
```

试验

AOF生成

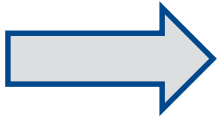
AOF配置

AOF长啥样？

AOF重写配置

AOF文件格式

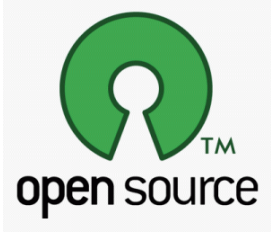
```
127.0.0.1:6379> dbsize
(integer) 0
127.0.0.1:6379> set hello world
OK
127.0.0.1:6379> bgrewriteaof
Background append only file
rewriting started
127.0.0.1:6379> get hello
"world"
```



```
*2
$6
SELECT
$1
0
*3
$3
SET
$5
hello
$5
world
```

AOF总结

1. RDB：丢数据。
2. AOF生成原理。
3. AOF三种策略比较。
4. 为什么要做AOF重写，如何实现？



搜狐视频Redis私有云平台开源了！！

Github主页：https://github.com/sohutv/cachecloud

QQ群：534429768