

云智未来⁹th

第九届中国系统架构师大会
SYSTEM ARCHITECT CONFERENCE CHINA 2017

万亿级数据洪峰下的消息引擎

牟羽@阿里巴巴

自我介绍

- ◆ 花名：牟羽
- ◆ 真名：金吉祥
- ◆ @阿里巴巴-消中间件
- ◆ 开源软件爱好者，Apache RocketMQ committer
- ◆ Aliware MQ核心开发
- ◆ 邮箱：lollipop@apache.org



金吉祥 
浙江 杭州

备注: desc

扫一扫上面的二维码图案，加我微信

The image shows a WeChat QR code for a user named 金吉祥 (Jin Jixiang) from Hangzhou, Zhejiang. The QR code has a WeChat logo watermark in the center. Below the QR code, there is a note that says '备注: desc'. At the bottom of the QR code area, there is a prompt: '扫一扫上面的二维码图案，加我微信' (Scan the QR code above to add me on WeChat).

分享内容

阿里消息中间件发展历史

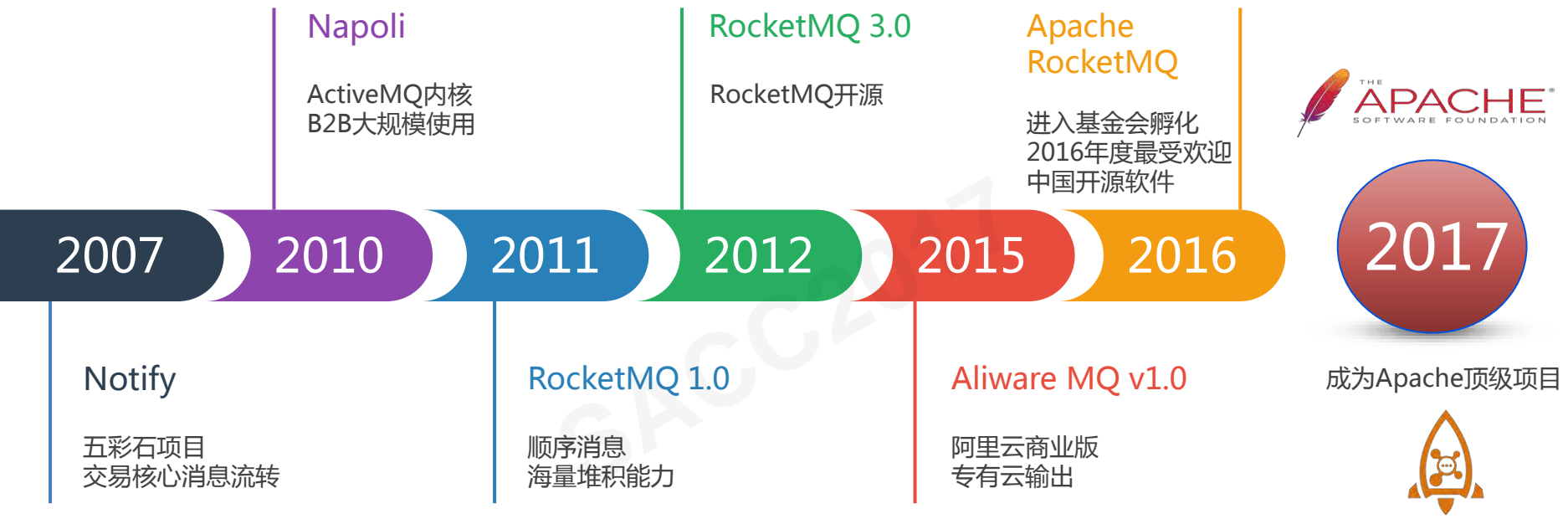


消息中间件核心功能设计

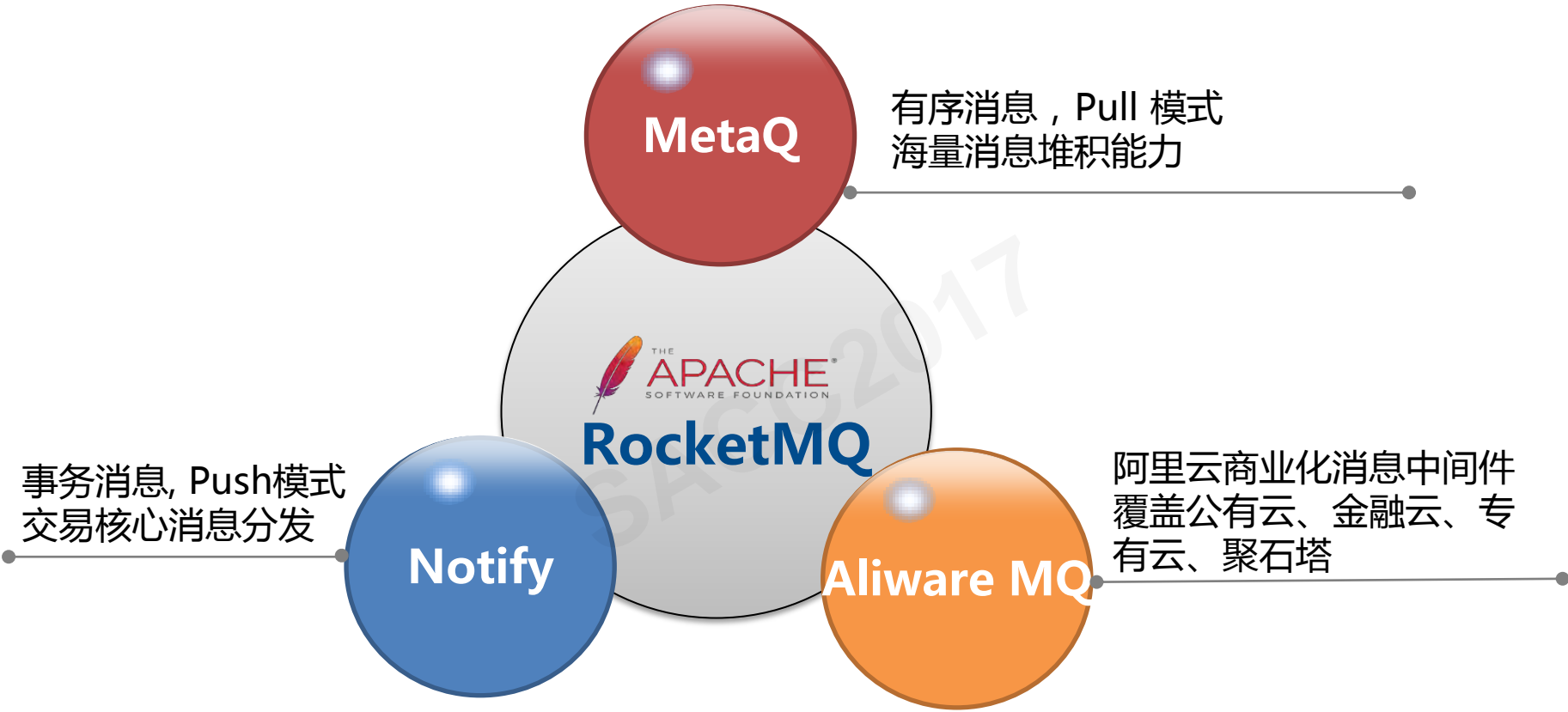
双11万亿数据洪峰的挑战

RocketMQ 5.0 展望

阿里消息中间件发展历史



阿里消息中间件现状



分享内容

阿里消息中间件发展历史

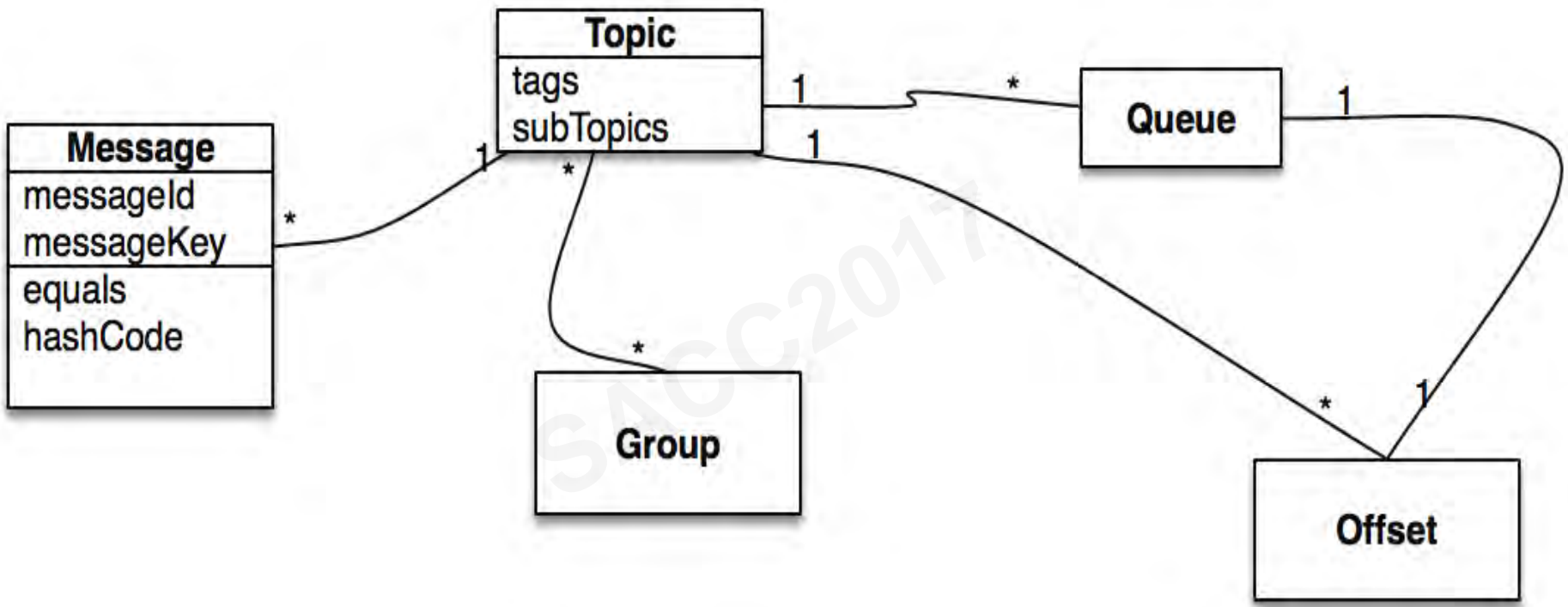
消息中间件核心功能设计



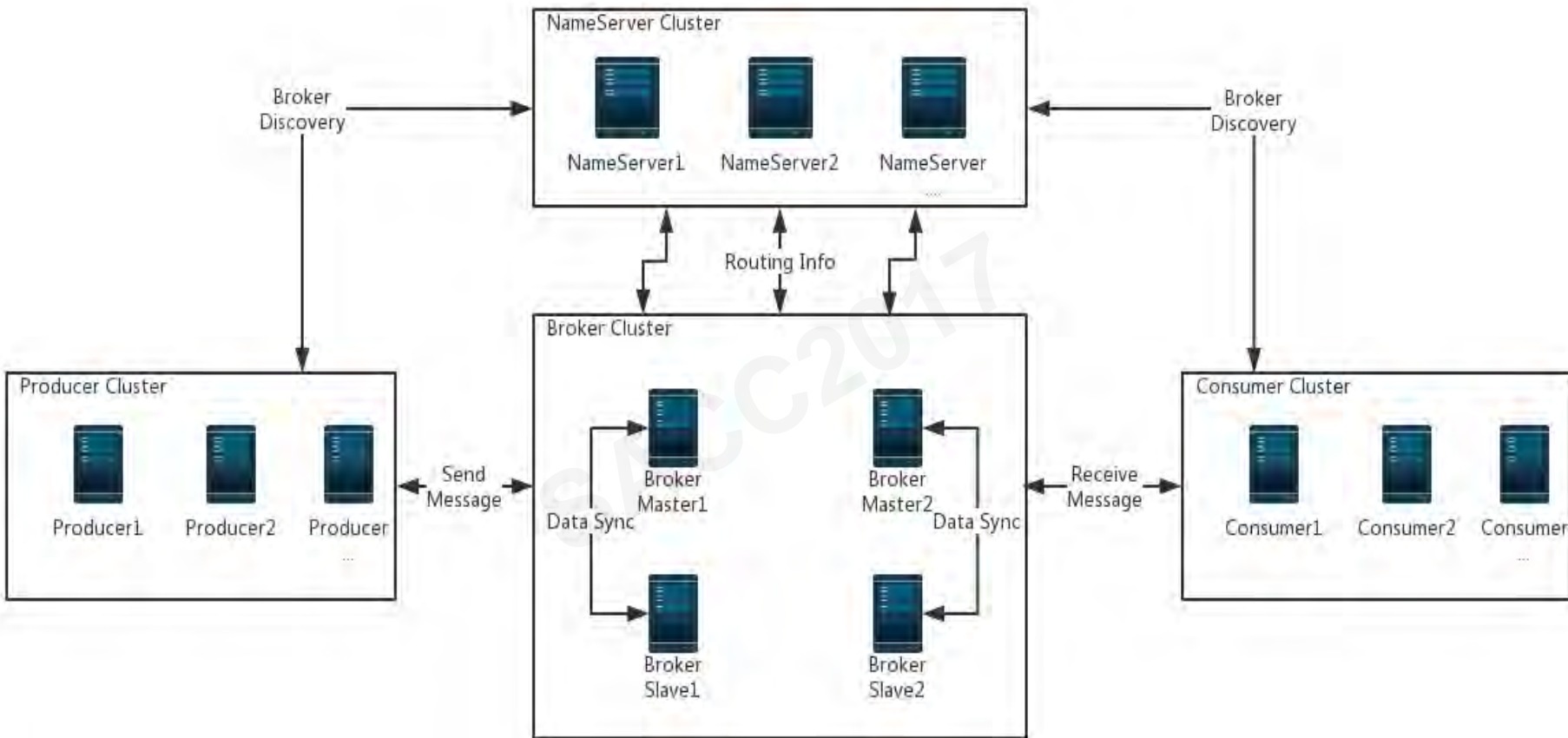
双11万亿数据洪峰的挑战

RocketMQ 5.0展望

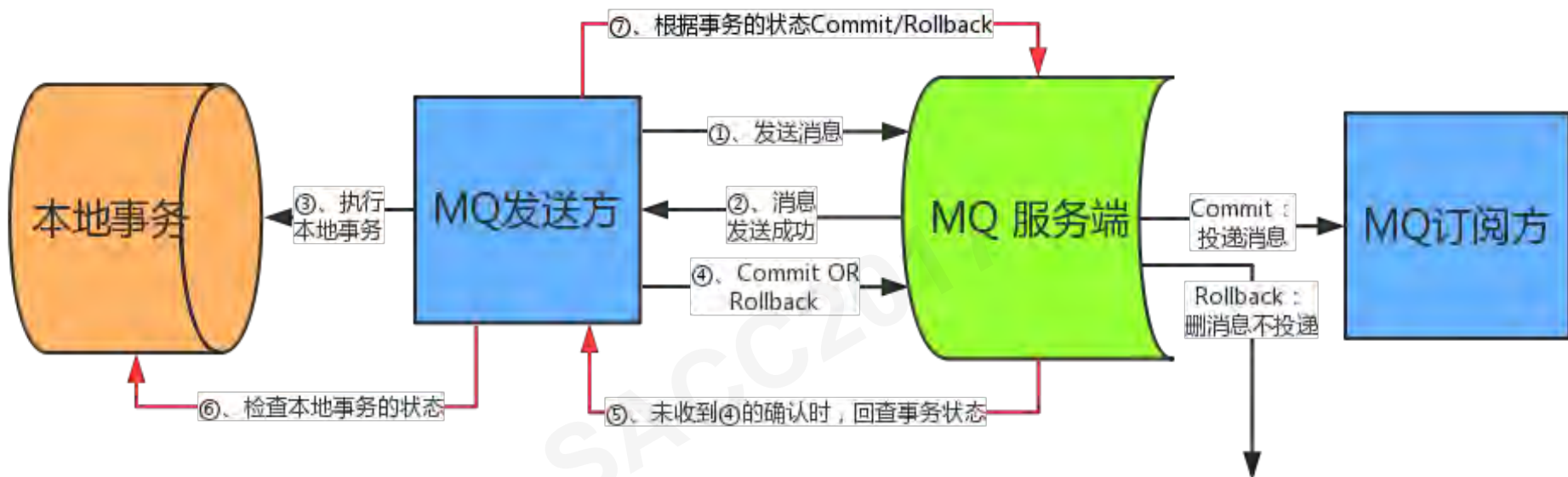
消息领域模型



消息组件交互流程



事务消息



Tips:

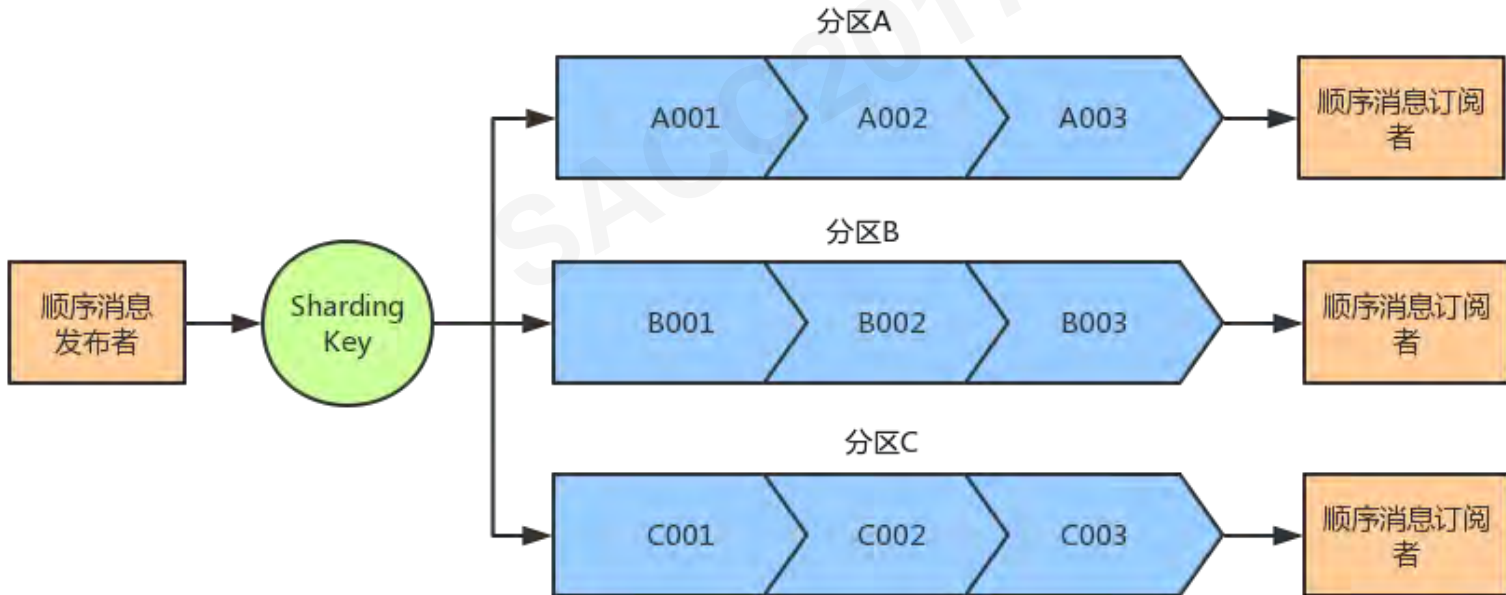
1. 保证本地事务与消息处理的最终一致性，并非强一致性。
2. 消息中间件保证消息至少投递一次。

顺序消息

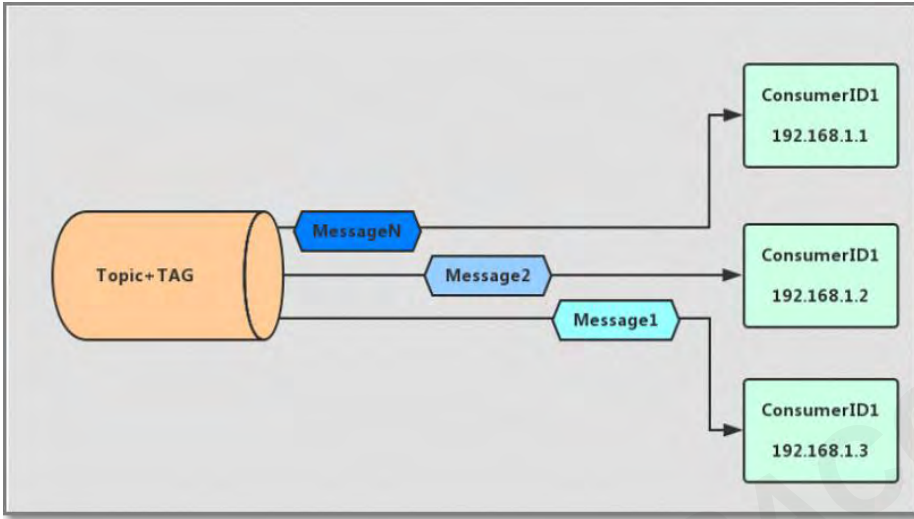
全局有序消息



局部顺序消息

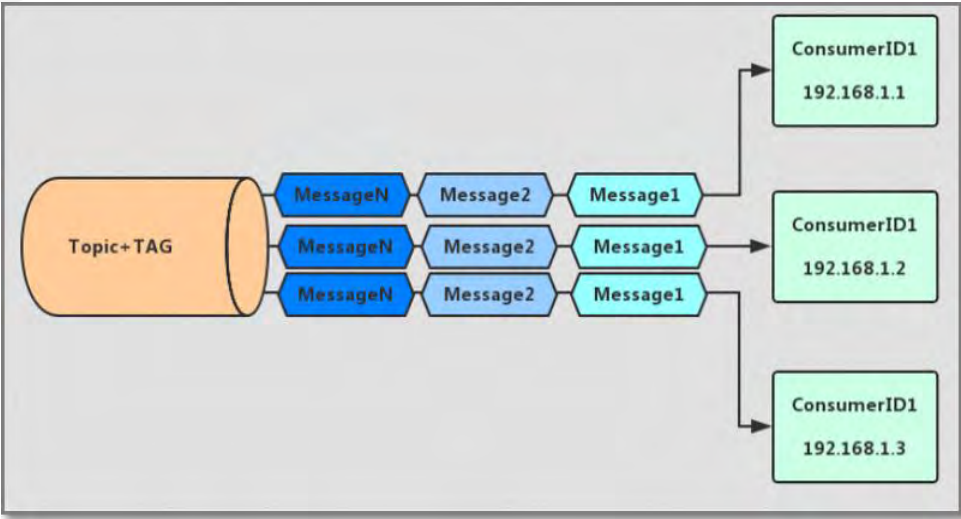


消费模式



集群消费模式

广播消费模式



消息过滤

基于消息tag、属性进行过滤

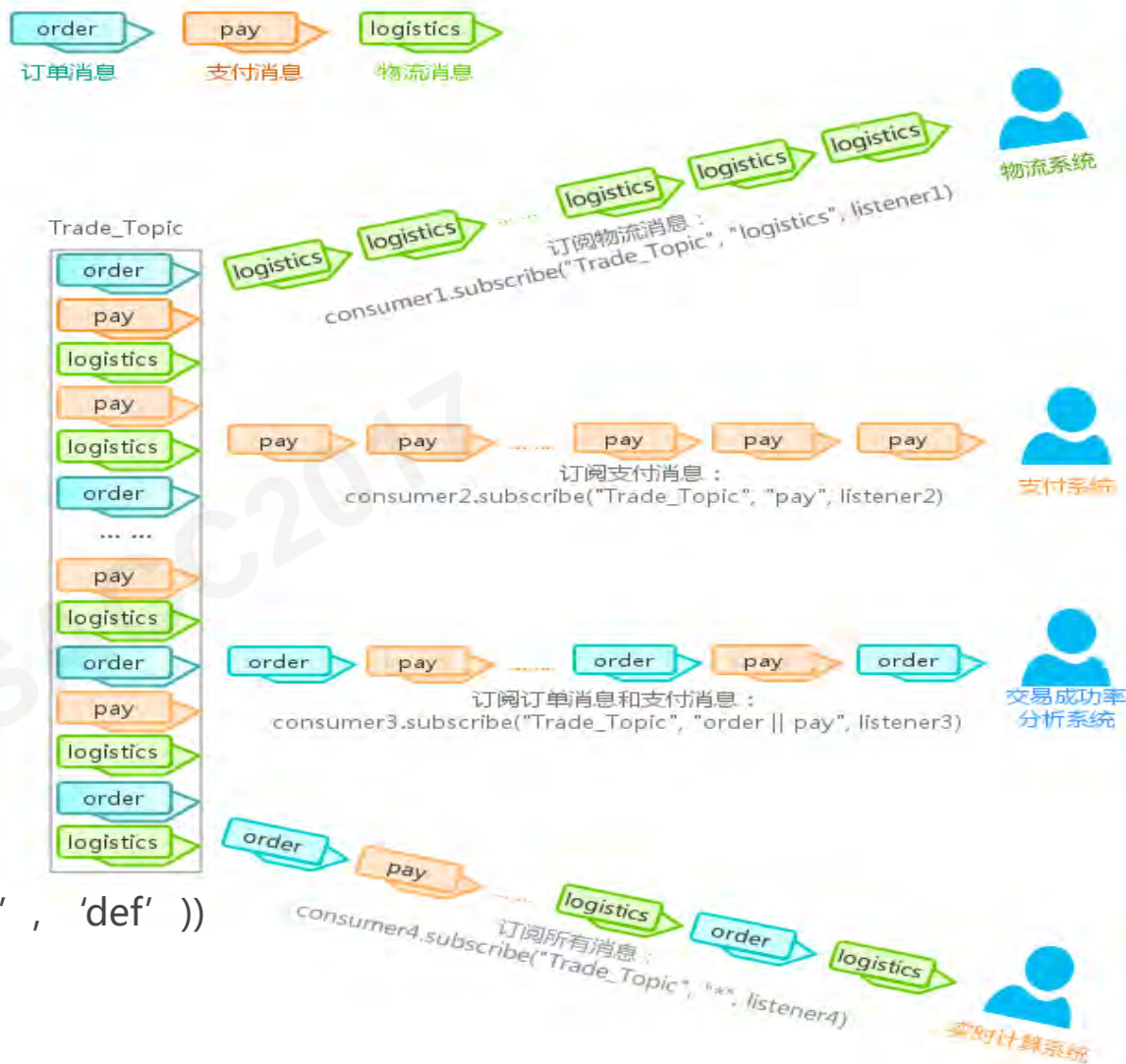
•语法: SQL92子集

•关键字:

- NOT
- AND
- OR
- BETWEEN
- IN
- TRUE
- FALSE
- NULL
- IS

•举例: `a IS NOT NULL AND (a IN ('abc' , 'def'))`

•Server端过滤, 节省带宽



消息轨迹

消息轨迹

0A418E4511F06DA213894F6BC98A000E 的消息轨迹



分享内容

阿里消息中间件发展历史

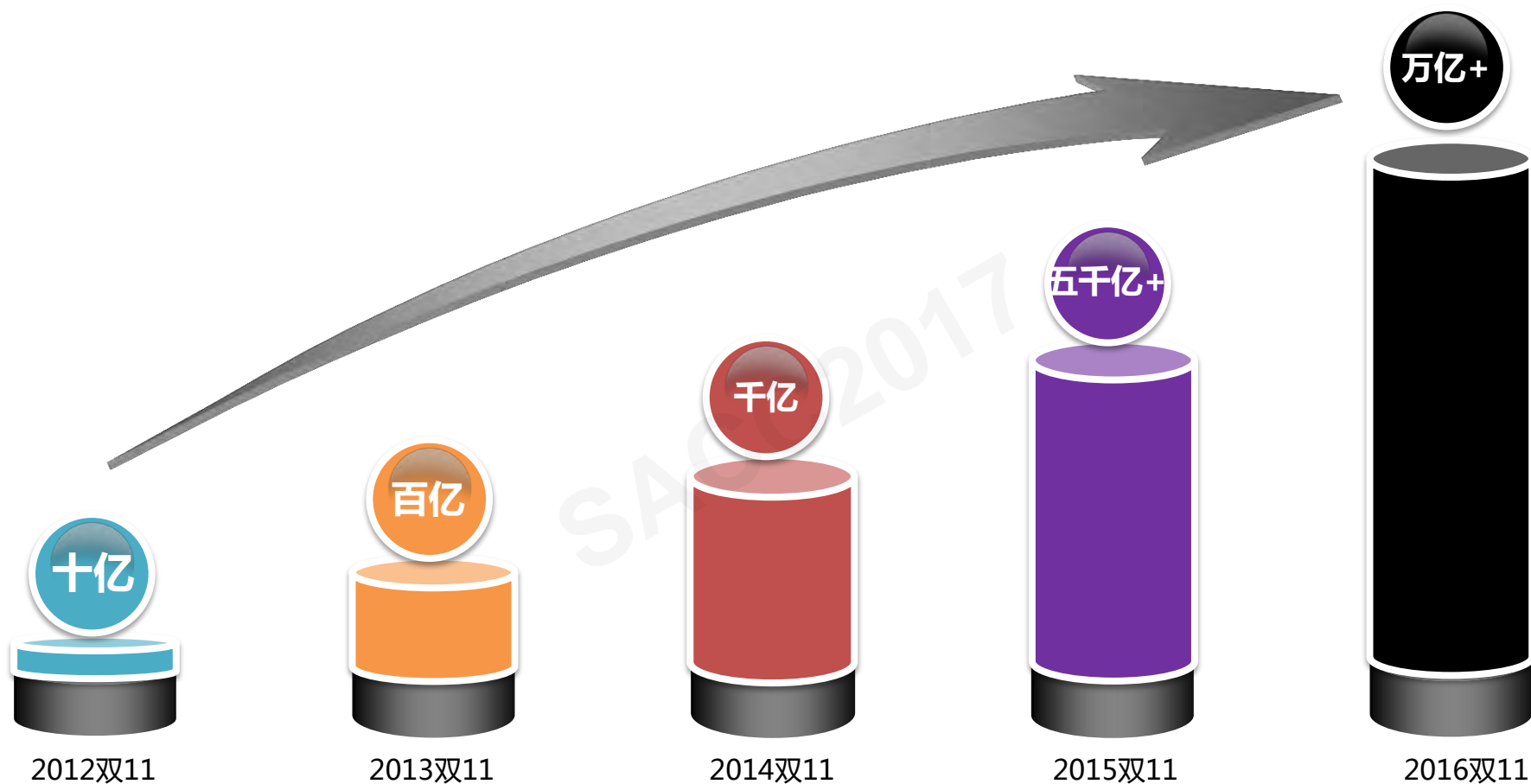
消息中间件核心功能设计

双11万亿数据洪峰的挑战

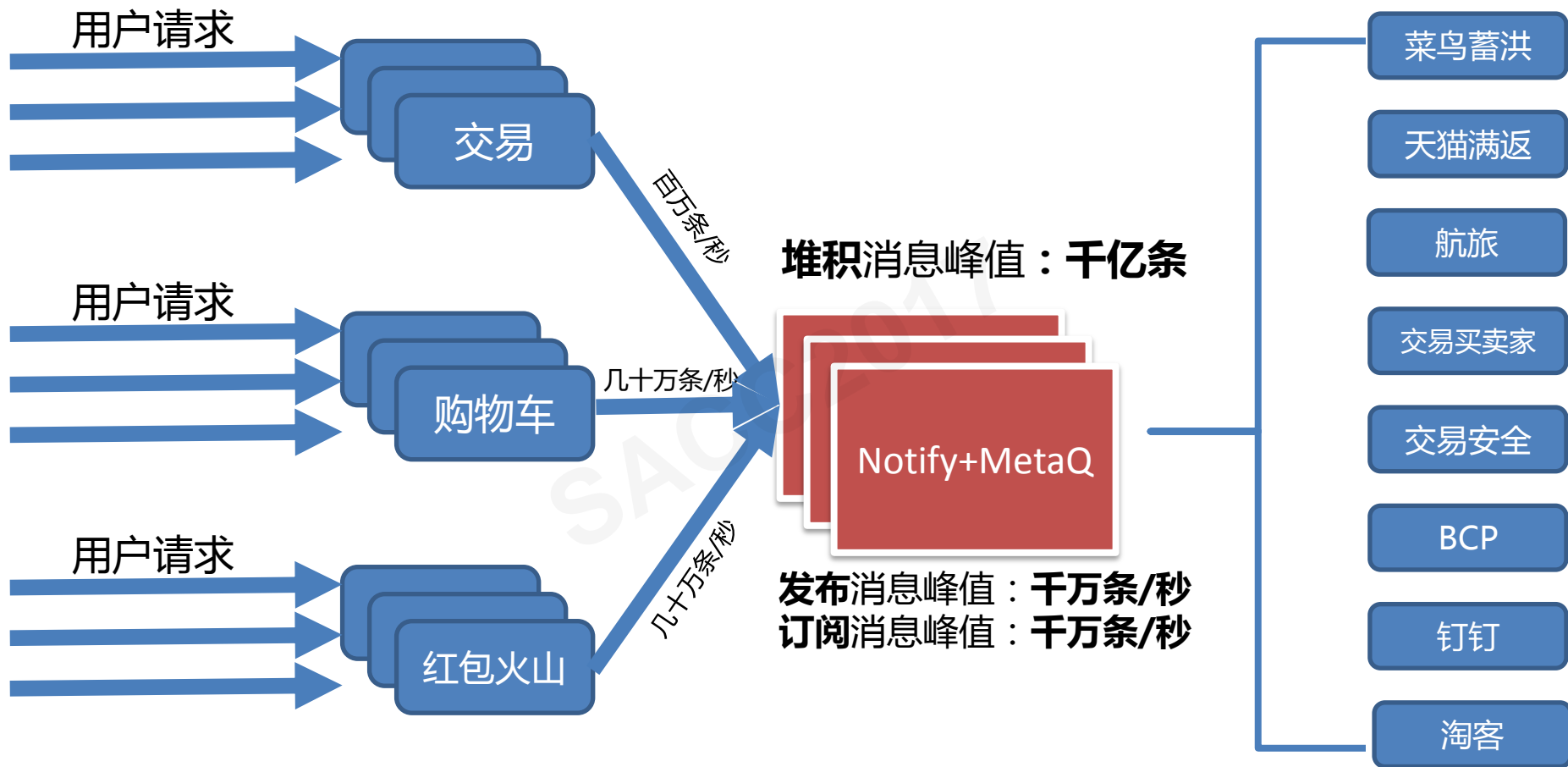


RocketMQ 5.0展望

历年双十一消息量变化



消息中间件核心链路



万亿洪峰下有哪些问题

根本的要求：

可用性无限接近100%

可靠性无限接近100%

可用性 > 可靠性

双十一当天系统可用性要求 ~ 100%

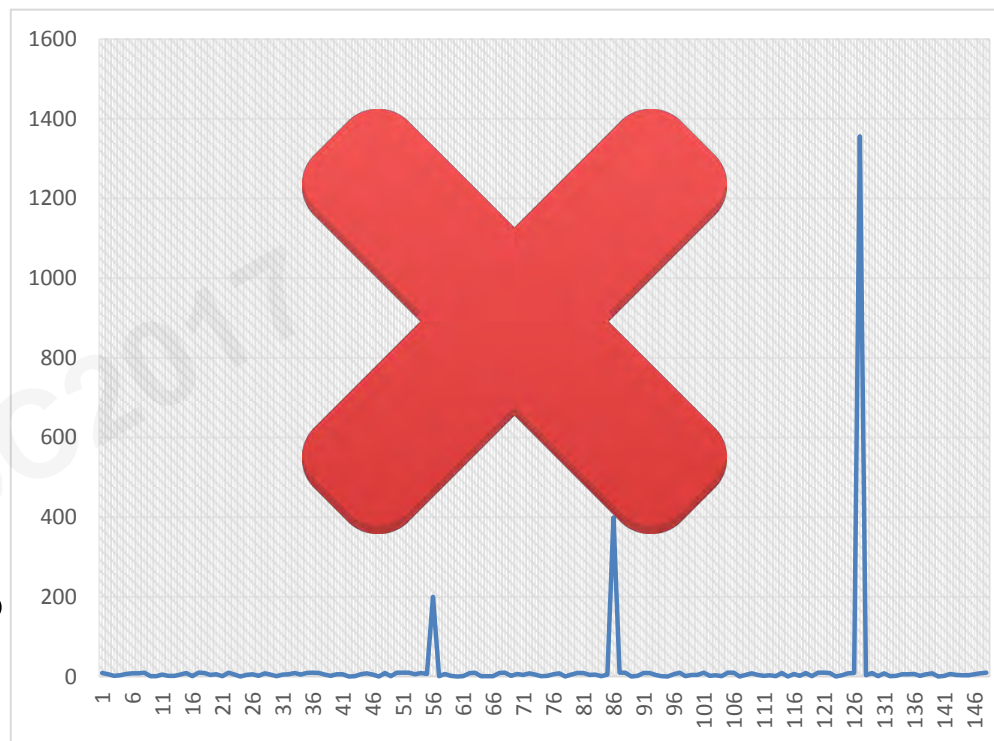
$$Availability = \frac{MTBF}{(MTBF + MTTR)}$$

MTBF: Mean time between failure

MTTR: Mean time to recover.

MTTR = 1 seconds

$$Availability = \frac{60*60*24 - 1}{(60*60*24)} = 99.999\%$$



消息中间件可用性提升方案



容量规划，限流



低延迟分布式存储系统

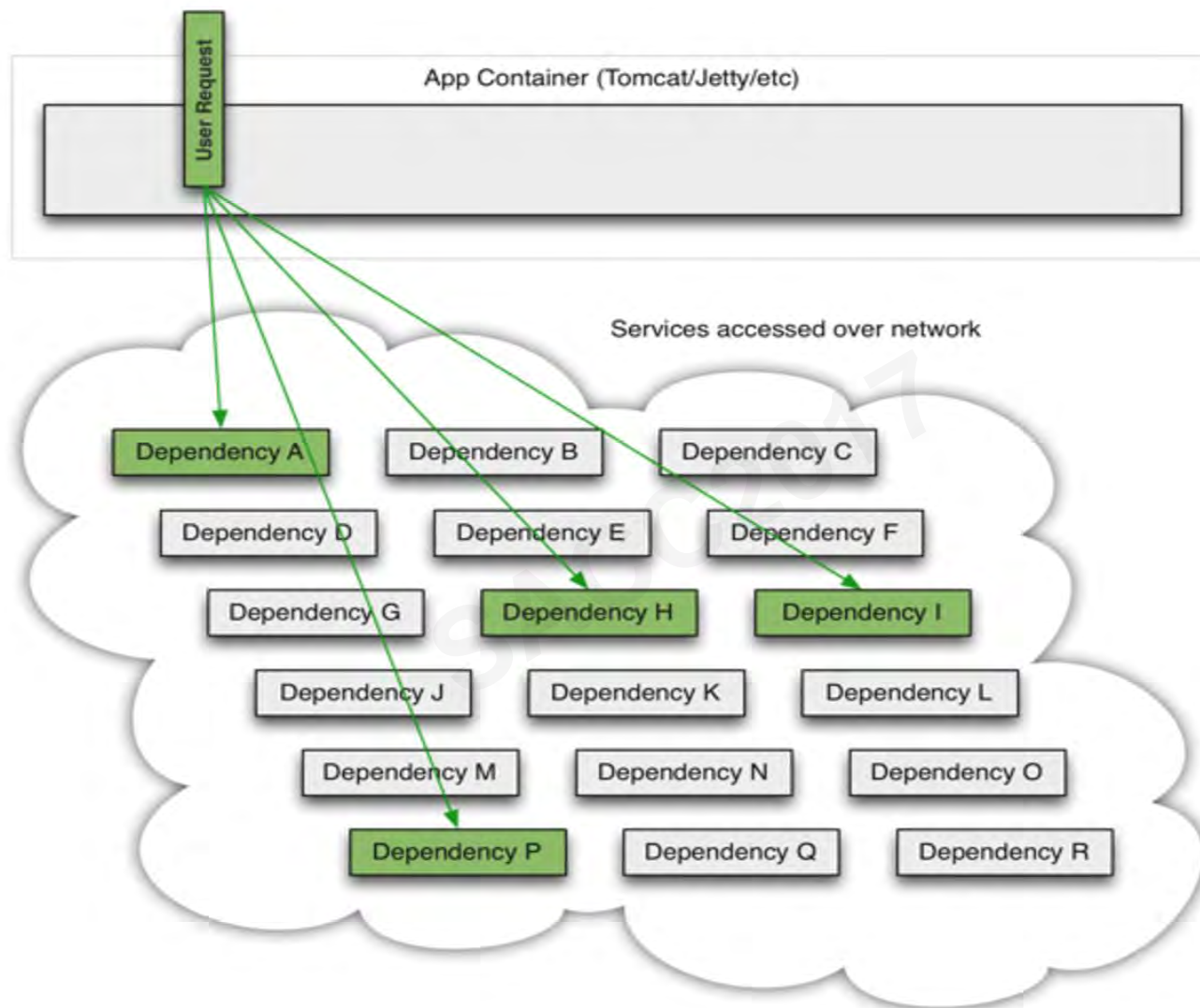


在线熔断机制，秒级隔离



单机故障自动恢复，秒级主备切换

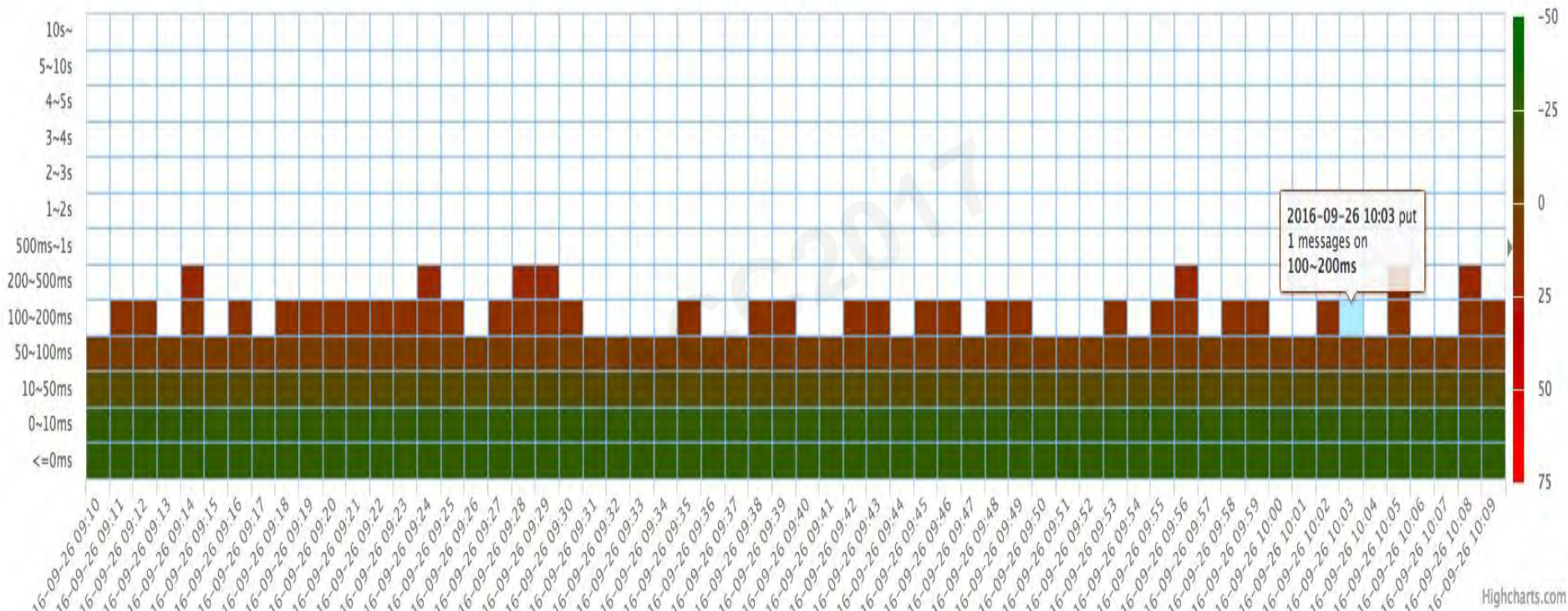
慢请求导致雪崩



高并发场景下写消息毛刺（优化前）

热力图

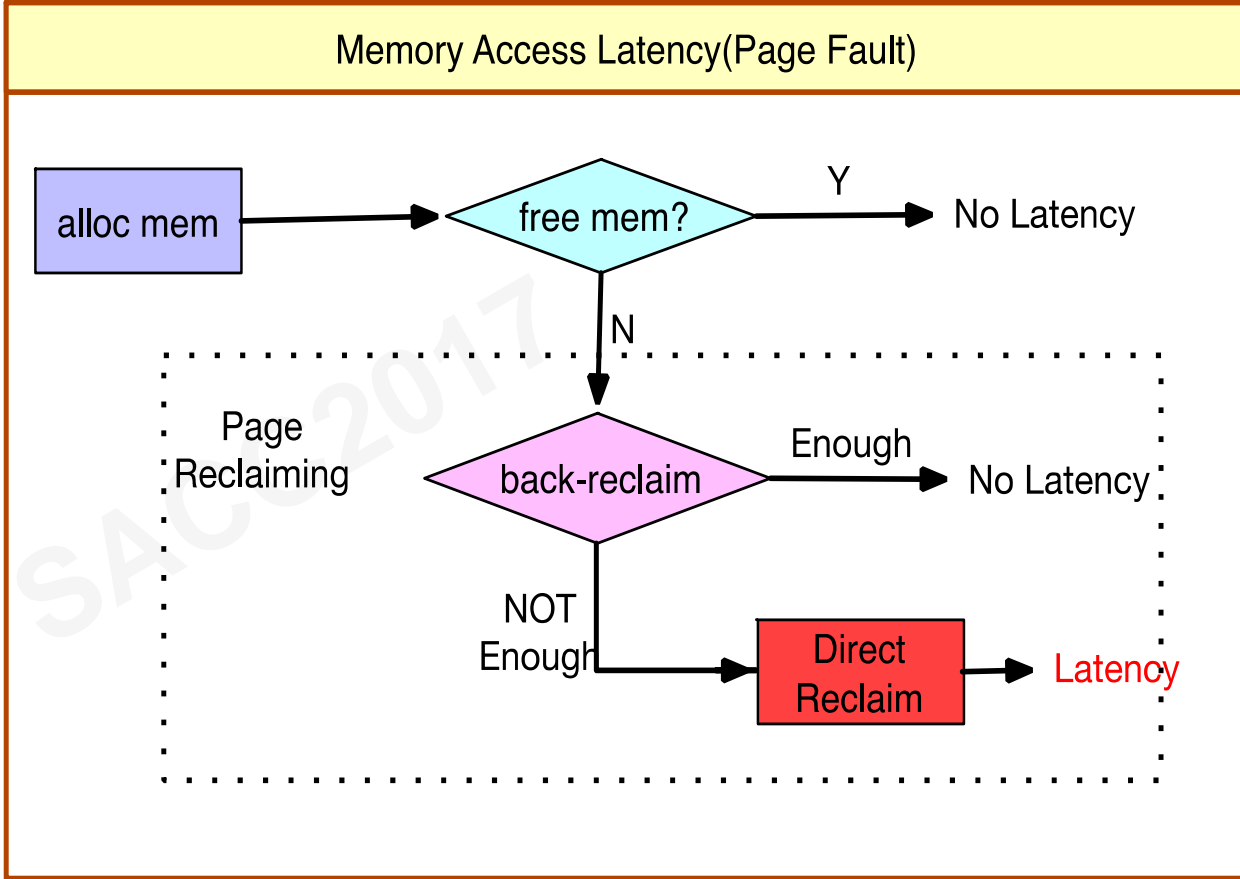
写消息耗时热力图



低延迟分布式存储系统—访存毛刺分析

- Memory access latency issues:

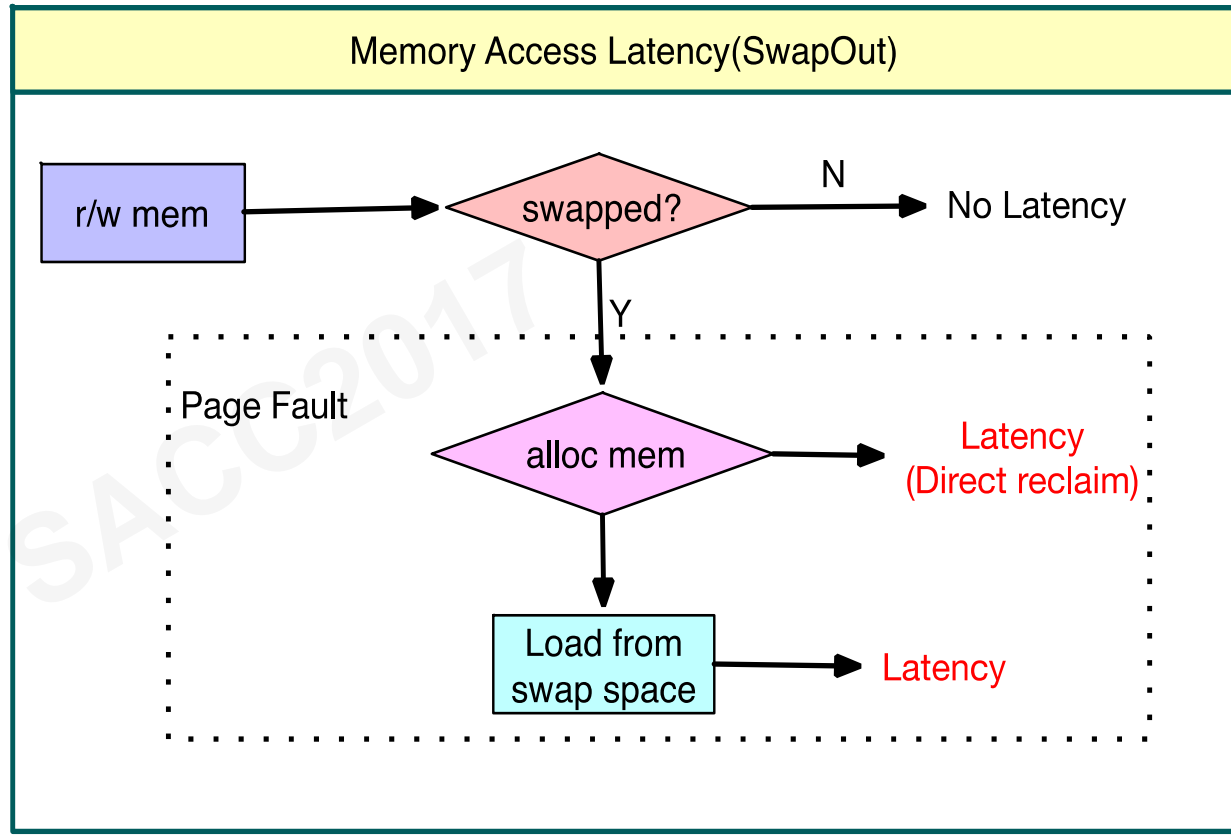
- Direct reclaim
 - Background reclaim (kswapd)
 - Foreground reclaim (direct reclaim)



低延迟分布式存储系统—访存毛刺分析

● Memory access latency issues:

- swapout/swapin
 - Put anonymous pages to disk
 - Need I/O to read data from disk at next access



低延迟分布式存储系统—访存毛刺分析

- Memory access latency issues:

- Memory lock
- Wake_up_page
 - Wait_on_page_locked()
 - Wait_on_page_writeback()

LOCKED	DIRTY	LRU
ACTIVE	WRITEBACK	RECLAIM
ANON	SWAPCACHE	SWAPBACKED
HUGE	RECLAIM	NOPAGE

Mem page flags

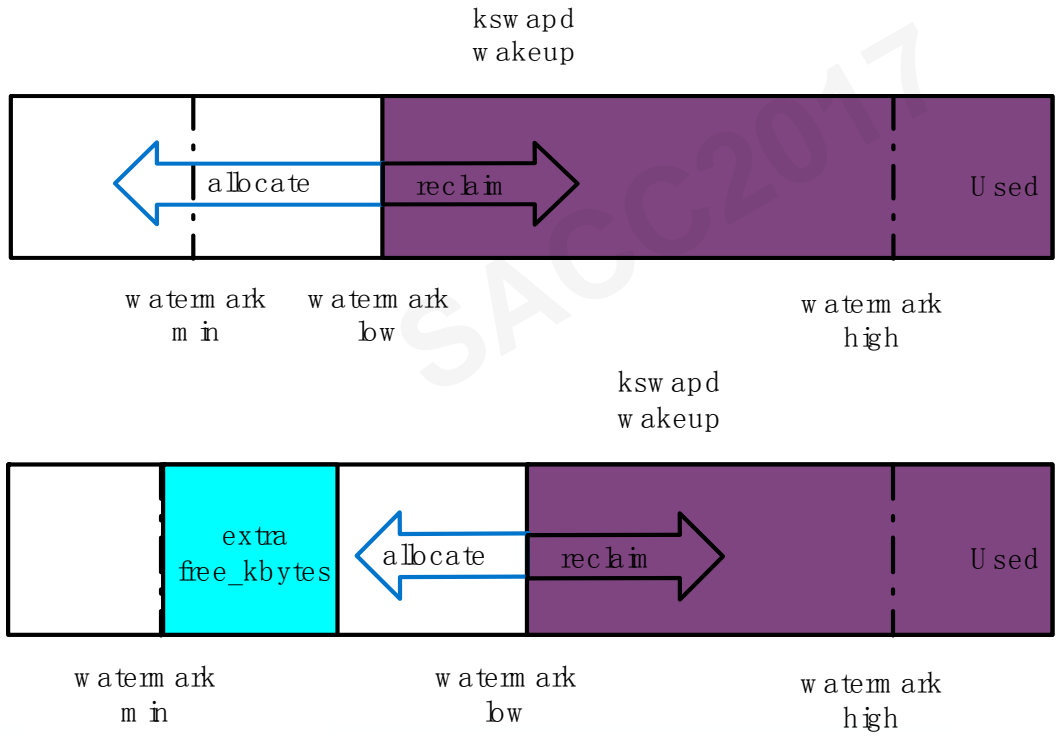
```
static inline int trylock_page(struct page *page)
{
    return (likely(!test_and_set_bit_lock(PG_locked, &page->flags)));
}

/*
 * lock_page may only be called if we have the page's inode pinned.
 */
static inline void lock_page(struct page *page)
{
    might_sleep();
    if (!trylock_page(page))
        __lock_page(page);
}
```

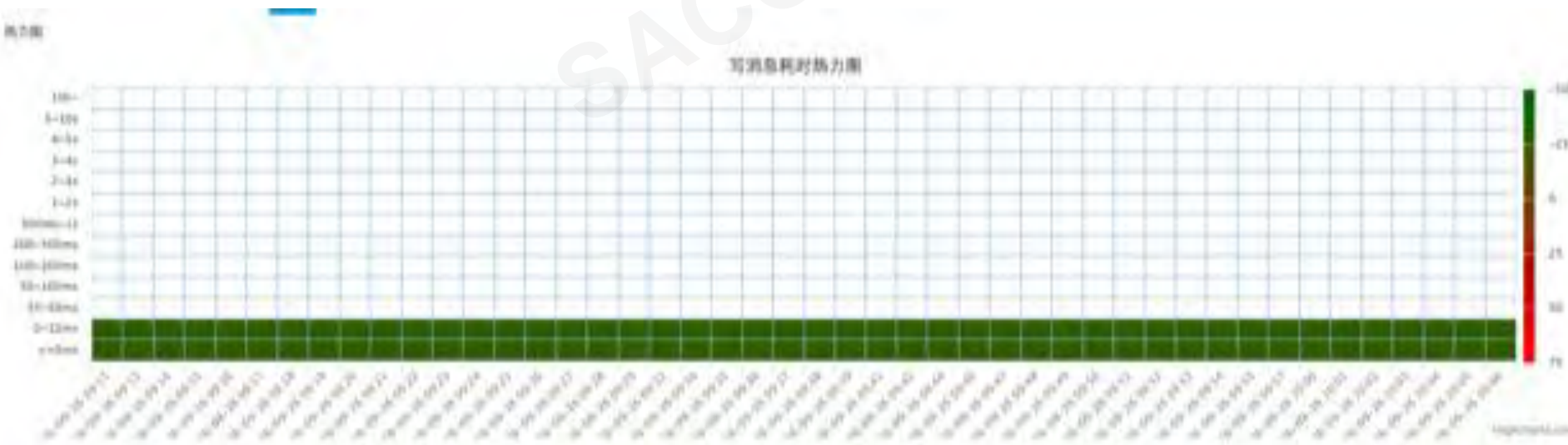
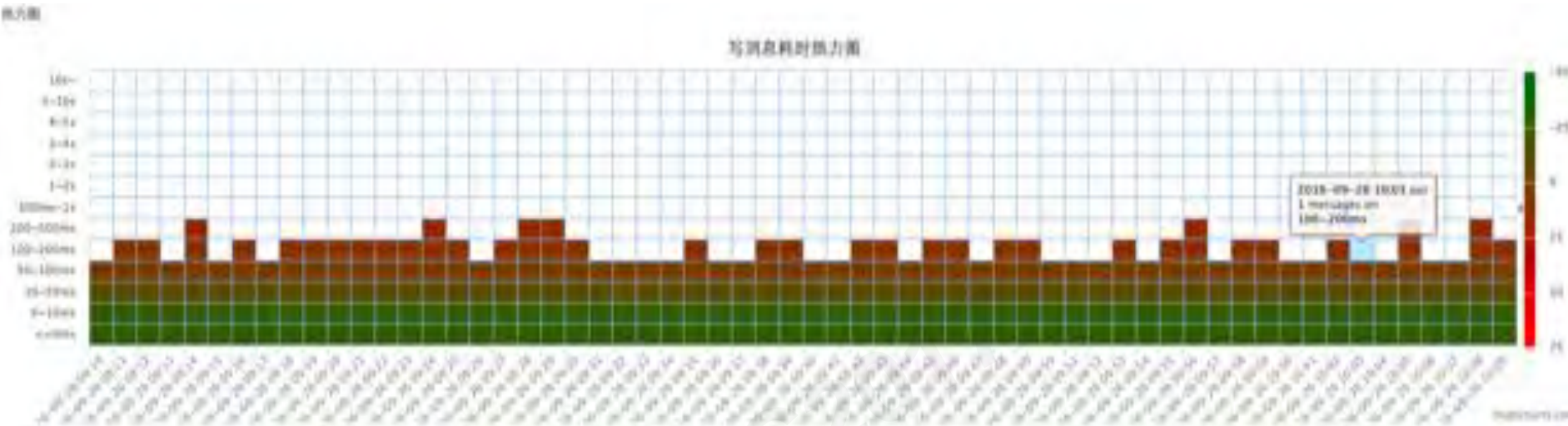
```
void unlock_page(struct page *page)
{
    VM_BUG_ON(!PageLocked(page));
    clear_bit_unlock(PG_locked, &page->flags);
    smp_mb__after_clear_bit();
    wake_up_page(page, PG_locked);
}
EXPORT_SYMBOL(unlock_page);
```


低延迟分布式存储系统—消除访存毛刺

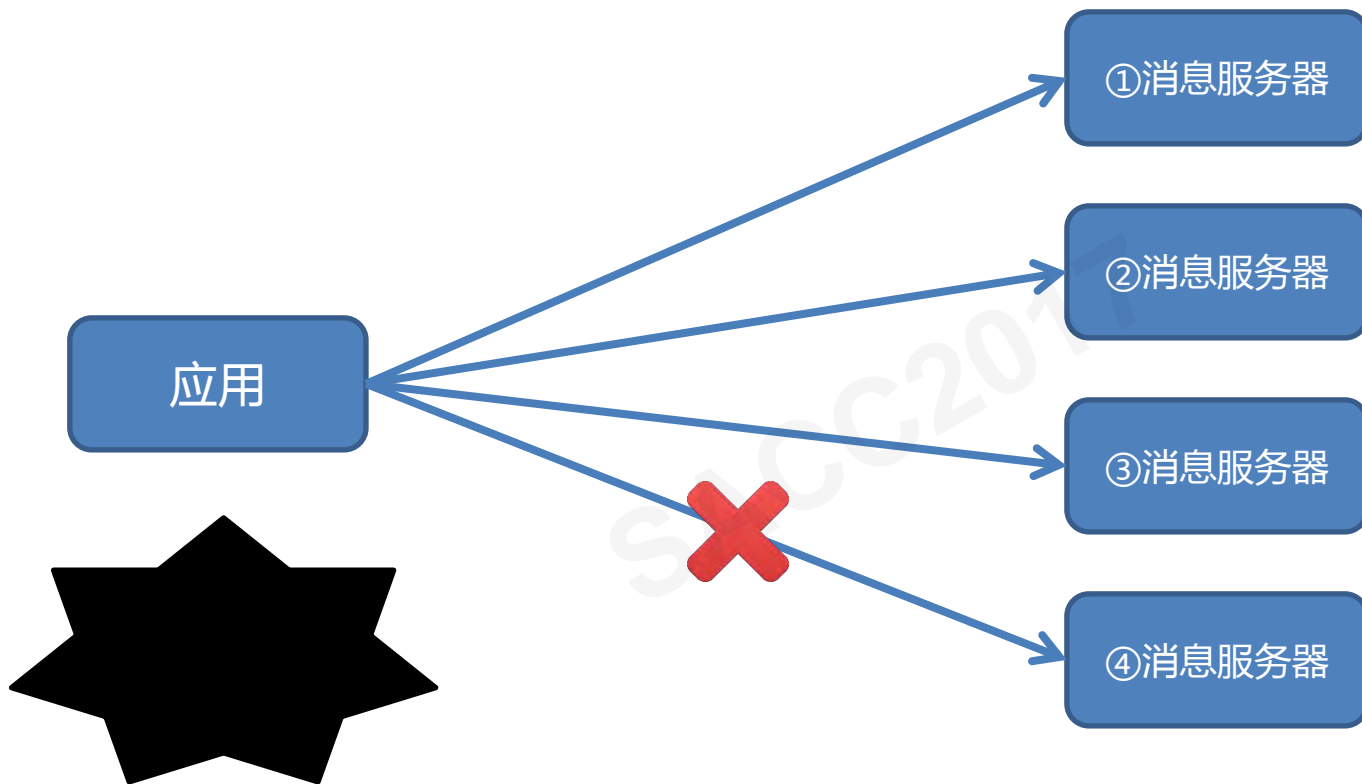
- ☞ Pre-Allocation + mlock/mlockall
- ☞ vm.swappiness vm.extra_free_kbytes



高并发场景下写消息毛刺 (优化后)



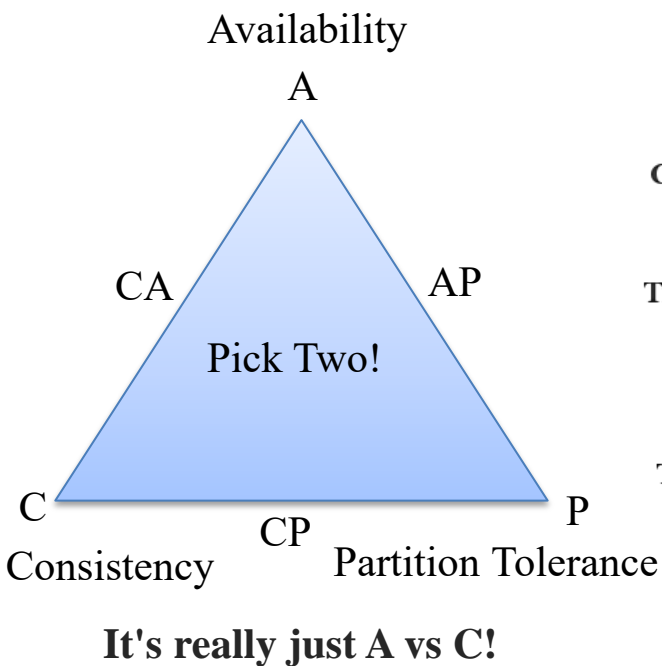
在线熔断机制



规则

1. 最多只能隔离30%的机器。
2. 响应时间过长，开始隔离1分钟
3. 调用抛异常隔离1分钟
4. 如果隔离的服务器超过30%，则有部分调用会进入隔离列表中最先隔离的机器

分布式系统高可用架构理论



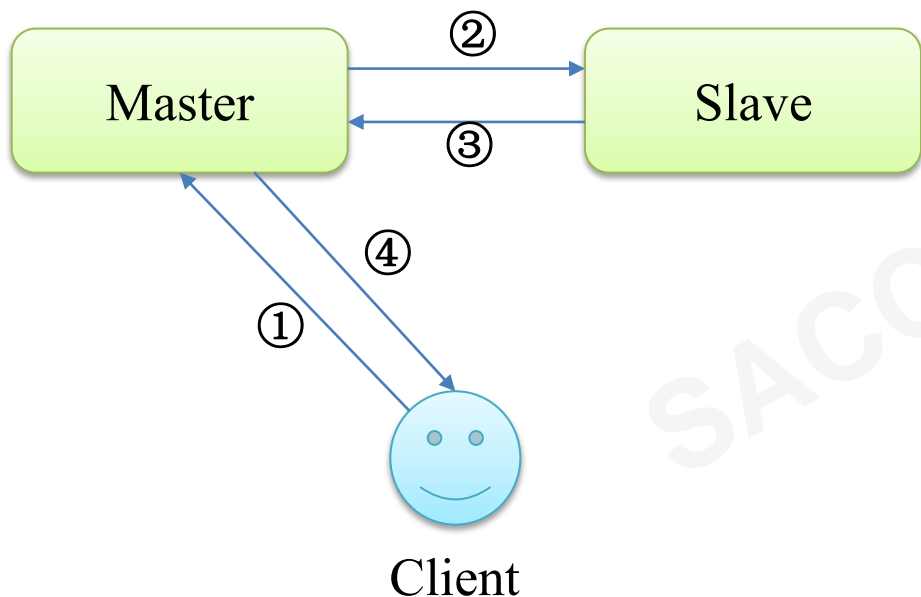
Techniques and Tradeoffs in High-Available Architecture

	Backups	Master/Slave	Master/Master	2PC	Paxos
Consistency	Weak	Eventual		Strong	
Transactions	No	Full	Local	Full	
Latency	Low			High	
Throughput	High			Low	Medium
Data Loss	Lots	Some		None	
Failover	Down	Read-Only	Read/Write		

分布式系统高可用架构理论

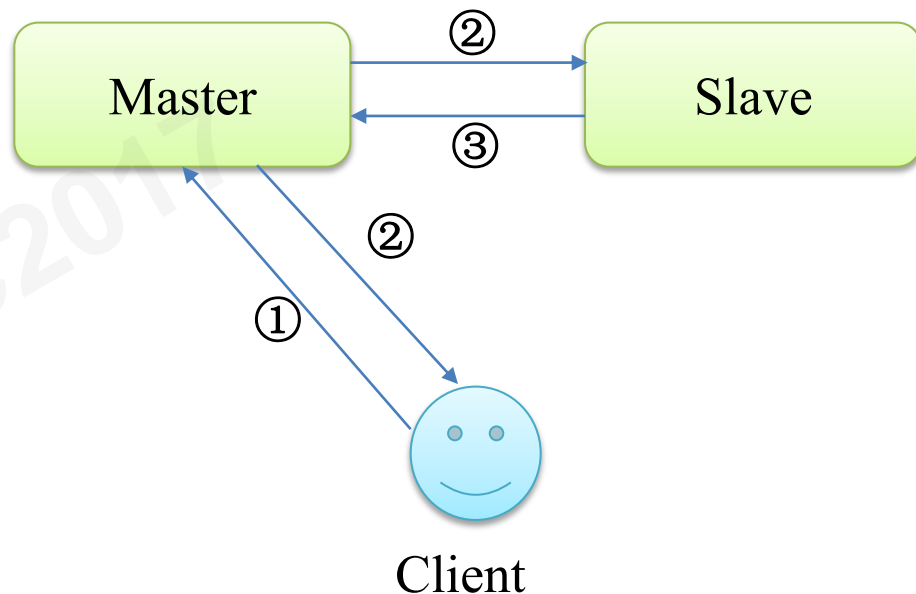
Master/Slave

同步复制



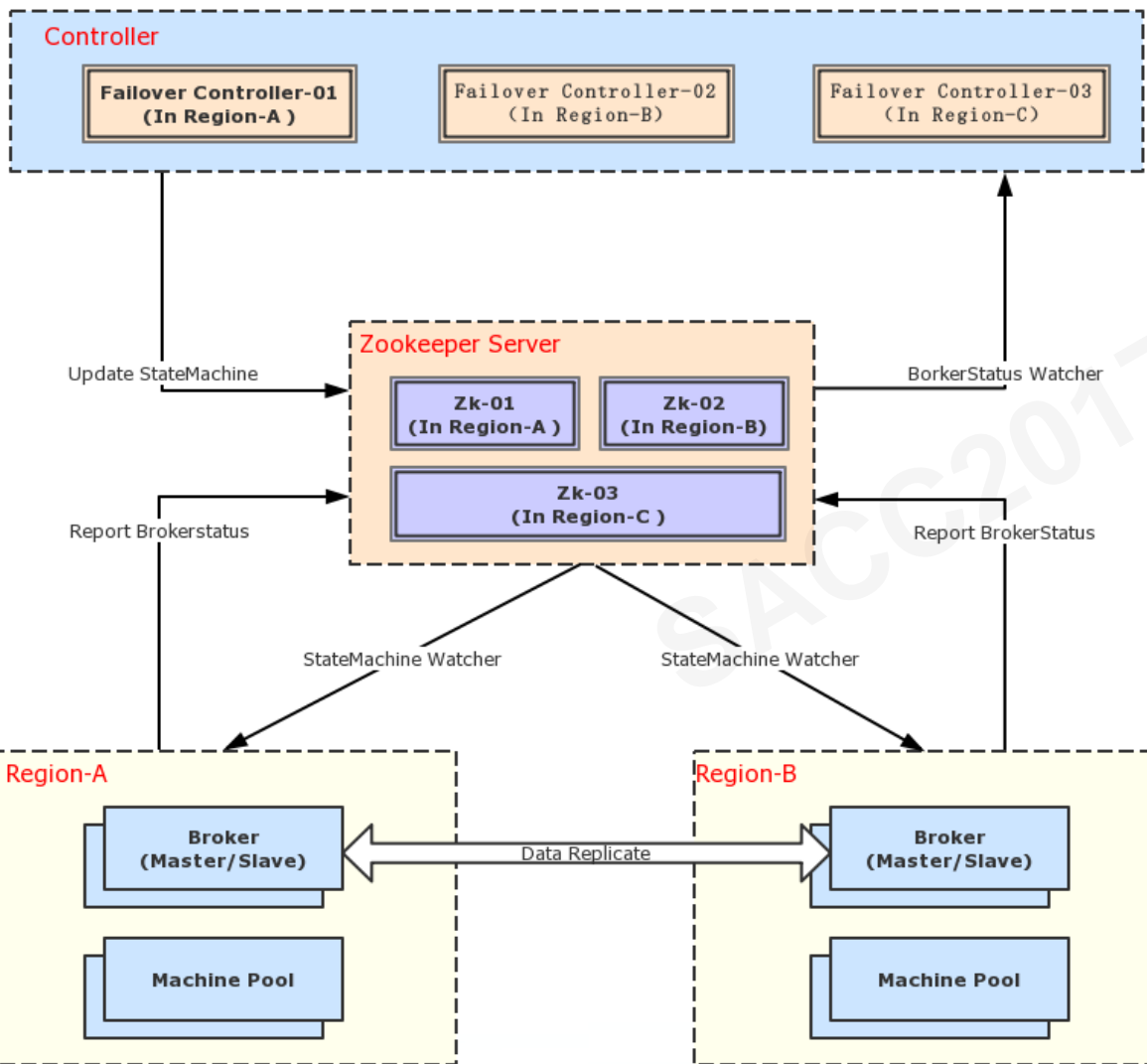
延迟上升、吞吐下降
消息可靠、故障自动恢复、强一致性

异步复制



延迟低、高吞吐
磁盘故障导致部分消息丢失、
故障恢复时间较长、最终一致性

消息中间件高可用架构



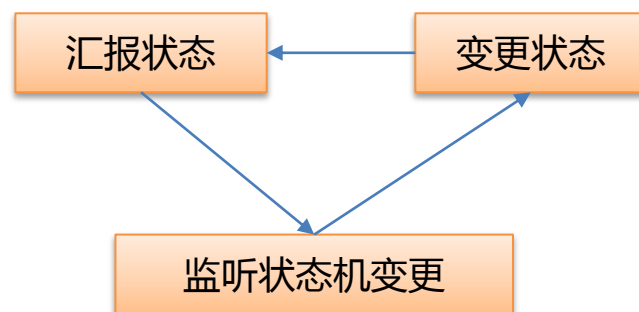
Controller 提供:

- 观察Broker状态的变更.
- 执行状态机变更, 推送新状态机至ZK

ZK 提供:

- 持久化存储状态机
- 以临时节点的方式存储Broker的状态
- 提供状态变更通知相应Watcher的机制

工作流程:



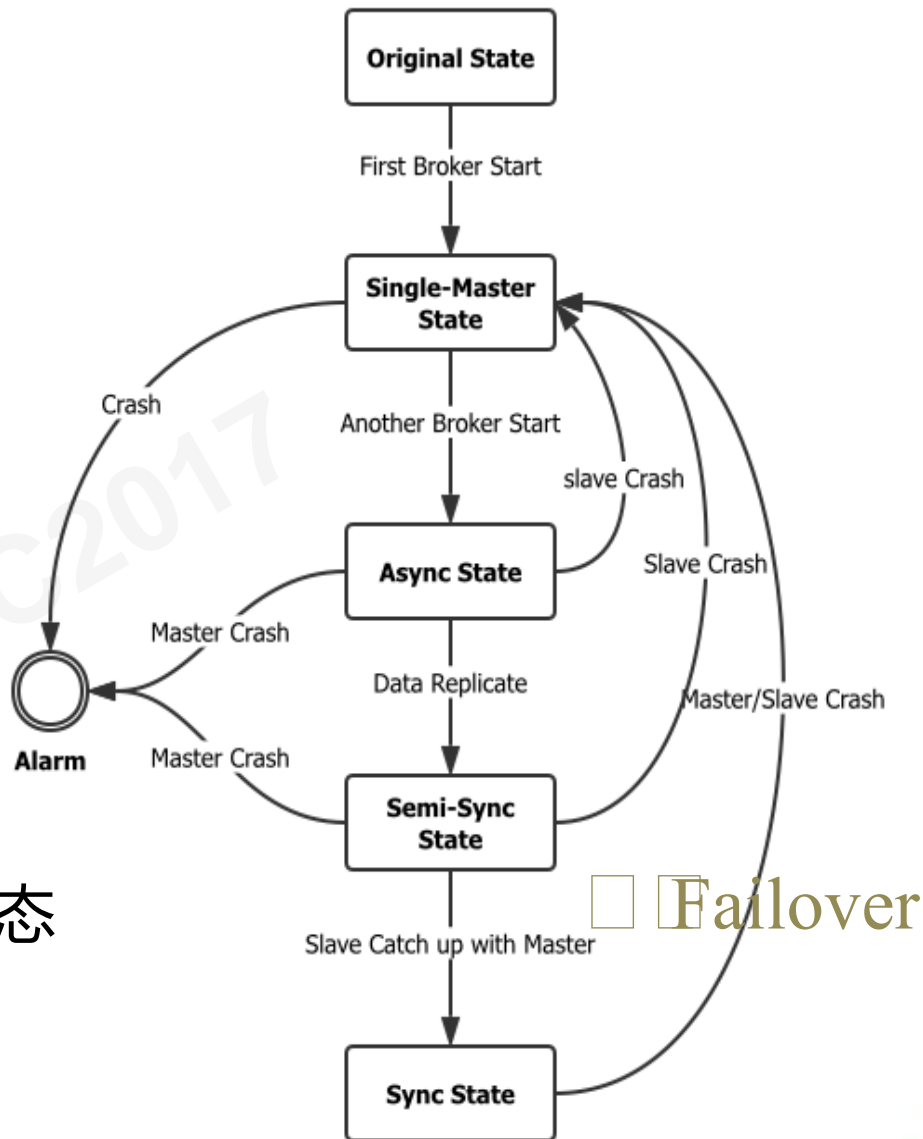
故障自动恢复

$$Availability \uparrow = \frac{MTBF}{(MTBF + MTTR) \downarrow}$$

Tips:

- ✓ 机器宕机自动恢复 → 低 MTTR
- ✓ 消息发送RT上升可控
- ✓ 系统吞吐损耗可控
- ✓ 多副本，提升消息可靠性

99.99% 时间处于同步复制状态



系统可用性提升

变量	取值 & 说明
MTBF of MQ(小时)	876, 意味着1年内机器故障宕机10次
MTTR without HA(分钟)	30, 涵盖故障报警、机器重启, 服务启动的耗时
MTTR with HA(秒)	30, 故障自动恢复

集群规模	是否高可用	是否顺序	消息可用性
1主	<input type="checkbox"/>	<input type="checkbox"/>	99.94%
2主	<input type="checkbox"/>	<input type="checkbox"/>	99.999967%
1主1备	<input type="checkbox"/>	<input type="checkbox"/>	99.99904%
2M2S	<input type="checkbox"/>	<input type="checkbox"/>	超过10个9
1M	<input type="checkbox"/>	<input type="checkbox"/>	99.94%
2M	<input type="checkbox"/>	<input type="checkbox"/>	99.88%
1M1S	<input type="checkbox"/>	<input type="checkbox"/>	99.99904%
2M2S	<input type="checkbox"/>	<input type="checkbox"/>	99.9980%

顺序

分享内容

阿里消息中间件发展历史

消息中间件核心功能设计

双11万亿数据洪峰的挑战

RocketMQ 5.0展望



RocketMQ 5.X 展望





THANKS

We are hiring
lollipop@apache.org