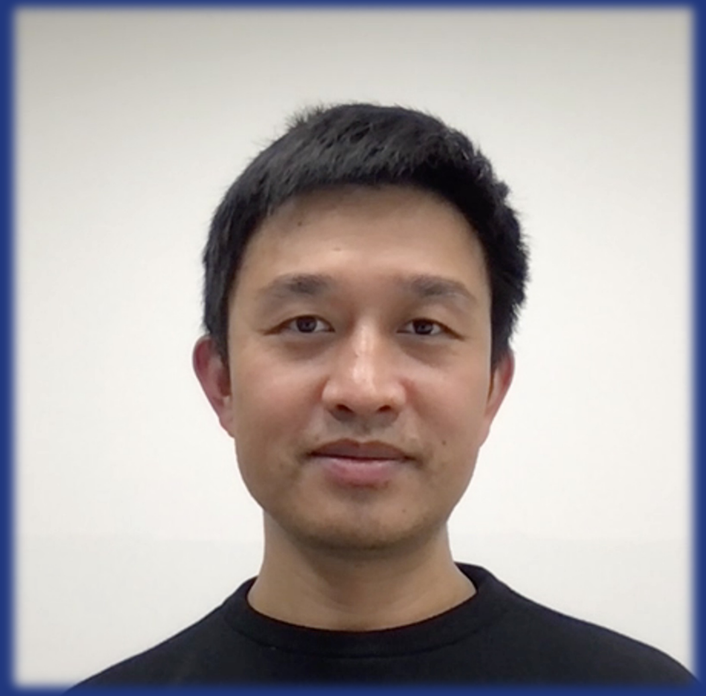
elastic 中国开发者大会 2023

# 微信支付PB级金融数据高可靠的腾讯云Elasticsearch优化

顾明，高级工程师
腾讯，微信支付，2023/04/08

# 分享嘉宾

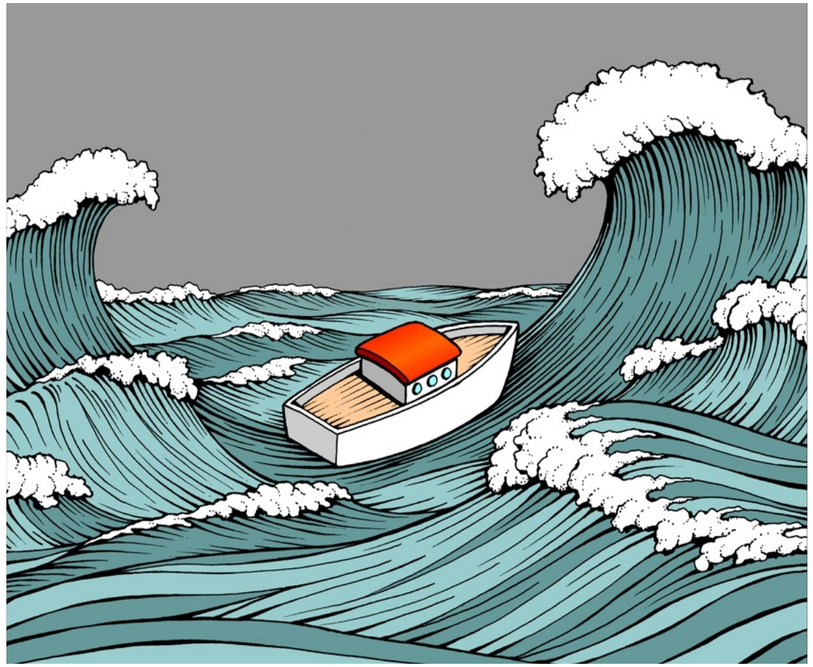


**平衡集群，稳定运行**

**连锁反应，方案引入新挑战**

# 平稳的集群

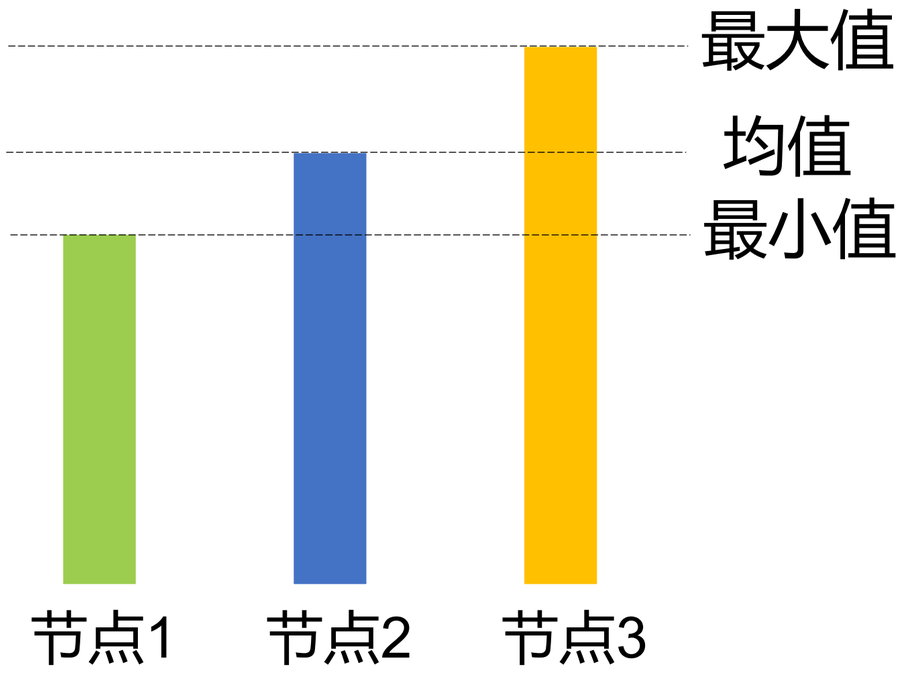
平稳的集群



惊险的异常

# 不均衡打破了集群的平静

- 不均衡（倾斜）的问题
  - SLA降级
    - 节点负载高/过载，请求时延变长/失败。
  - 容量规划失效/事故
    - 引发计划外扩容
    - 成本超出预期
  - 业务中断
    - 85%，90%，95%
    - 磁盘过高水位线，index只读



# 不均衡如何产生

## 理想

- 集群中分片尺寸相当
- ES内置均衡策略：以分片数均衡为目标

+

## 现实

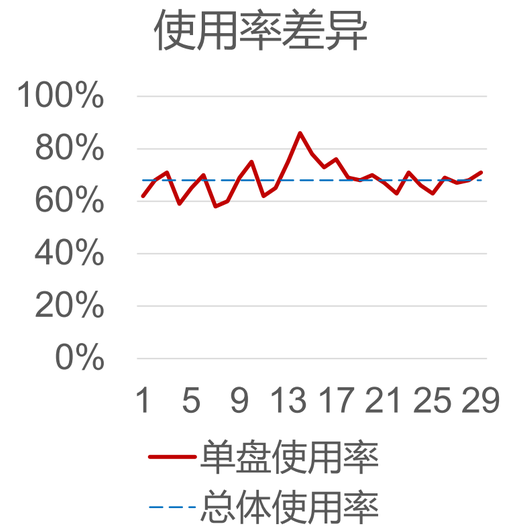
### 大小不一，不断流动

- 实时业务数据
  - 预创建空index
- 长生命周期数据
  - 不同时期index的分片尺寸不同
  - 数据一直被保留，没有清理的机会
- 长期的运维
  - 坏件维修
  - 扩容，缩容
  - 不同时期的采购，不同容量的节点，不同容量的磁盘
- ……

=

## 问题

### 集群不均衡，规模越大损失越多



# 需要怎样的均衡

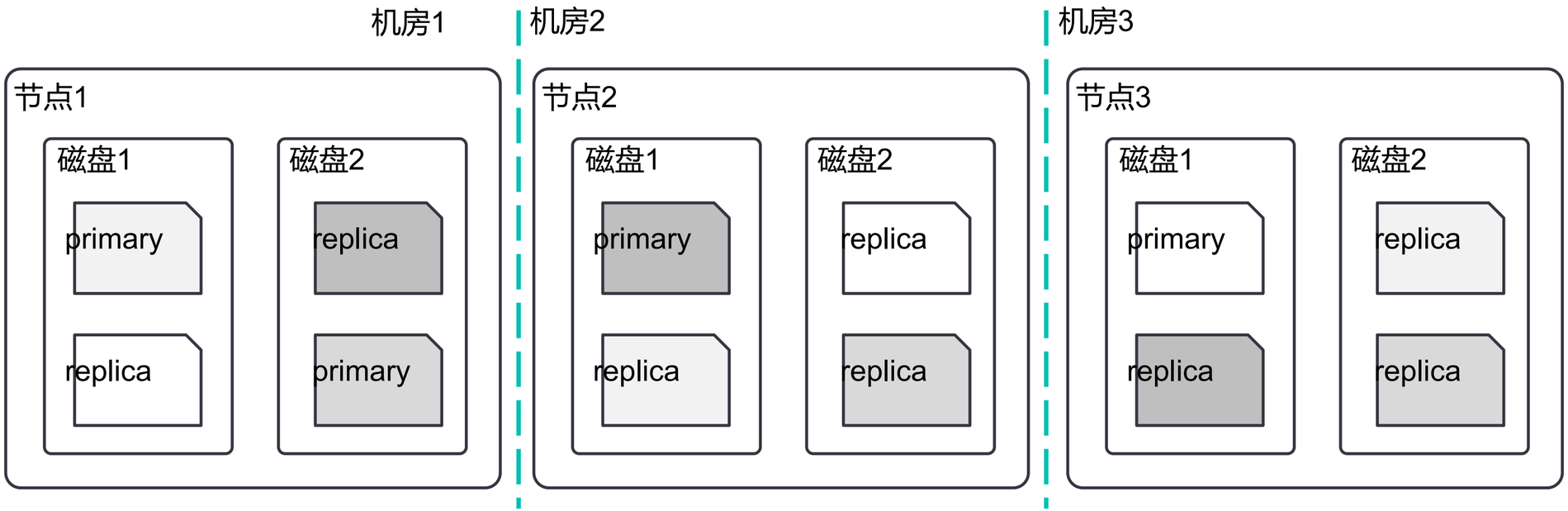
- 系统可靠性
  - 副本跨机房
- 系统成本
  - 磁盘容量
  - 节点负载
- 系统性能
  - 节点负载
- 均衡自身的成本
  - 数据搬迁消耗io、cpu、内存
- 时间维度
  - 对变化的系统做均衡决策

# 方案

- 组合多维目标
  - 分片跨节点
  - 副本跨机房
  - 热温数据分层均衡
  - 最小化搬迁次数
  - 不同时期对应不同的策略

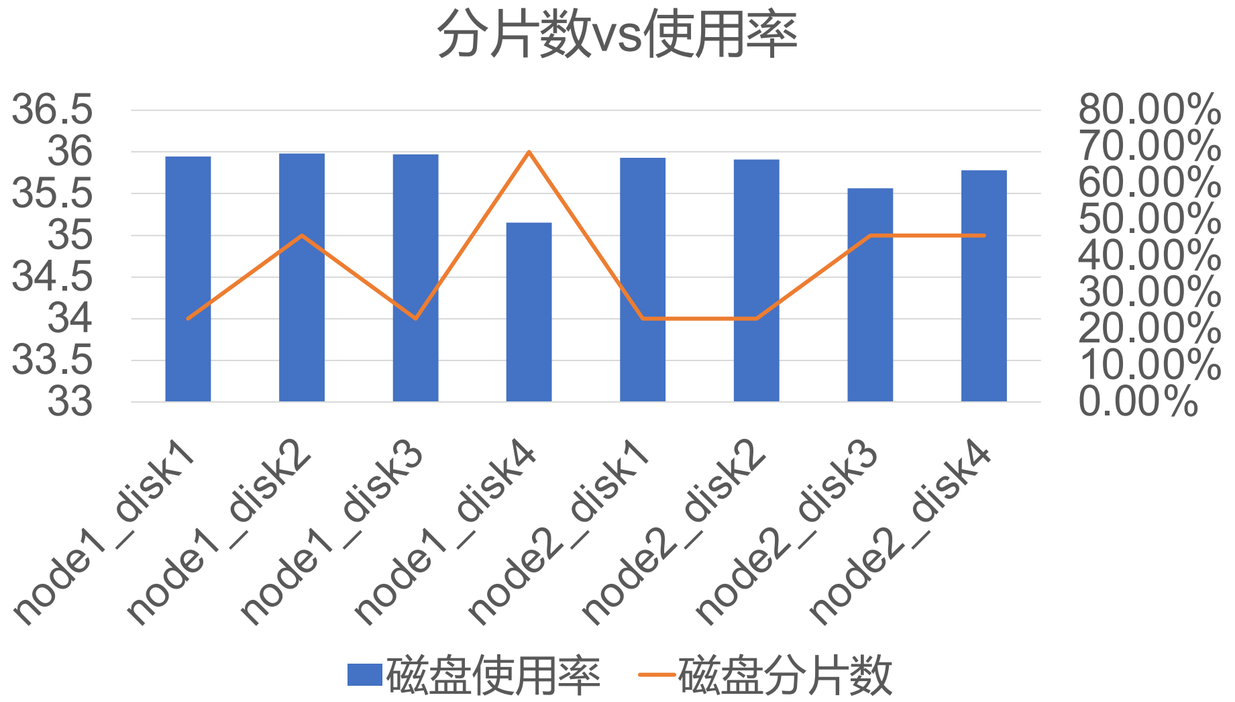
# 全面获取集群状态

- 节点和磁盘的关系
- index和分片的分布
- 磁盘与分片的关系



# 跳出局部解1

- 分片数 vs 使用率
  - 使用量<其他盘使用量
  - 分片数>节点内其他盘分片数

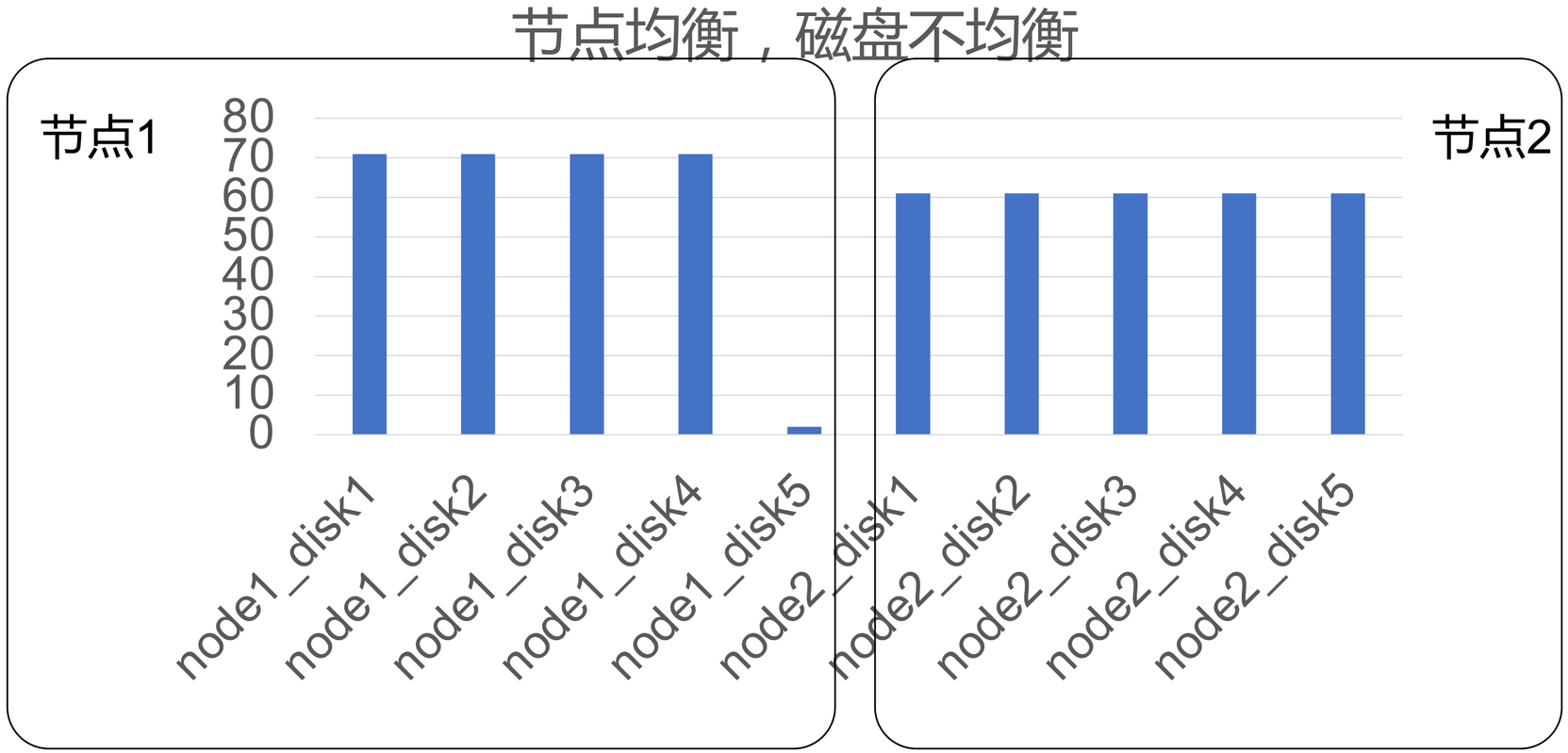



以退为进

搬迁盘node1_disk4的小分片，再均衡

# 跳出局部解2

- 节点 vs 磁盘
  - 空盘和满盘在一起
  - 修理故障盘场景

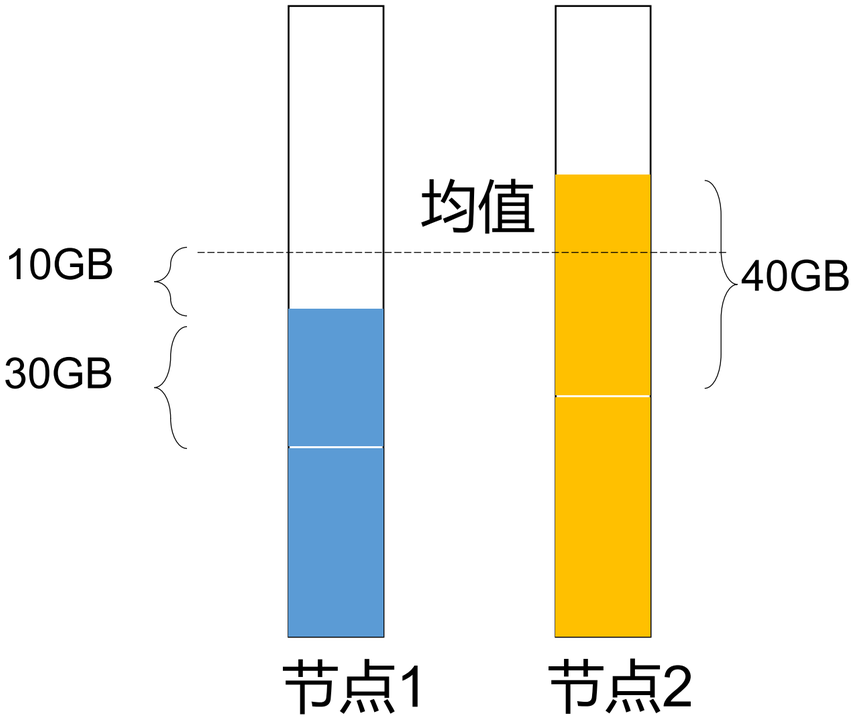



腾挪

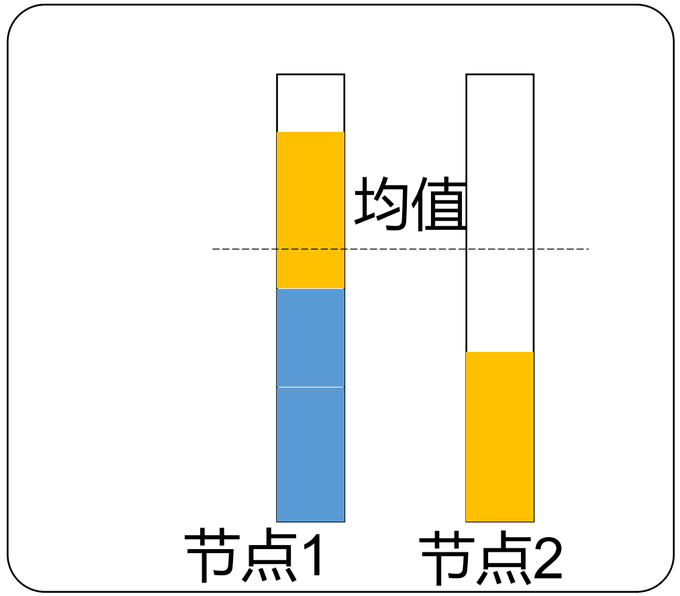
选择一：满盘分片先挪到其他节点上，再挪回空盘

选择二：其他节点分片先挪到空盘上，满盘分片再挪到其他节点上

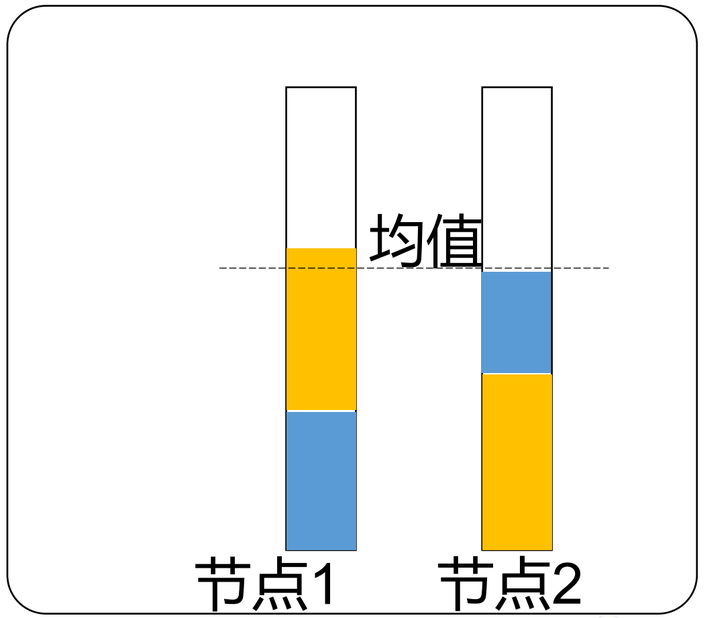
# 逼近均值




坏的选择

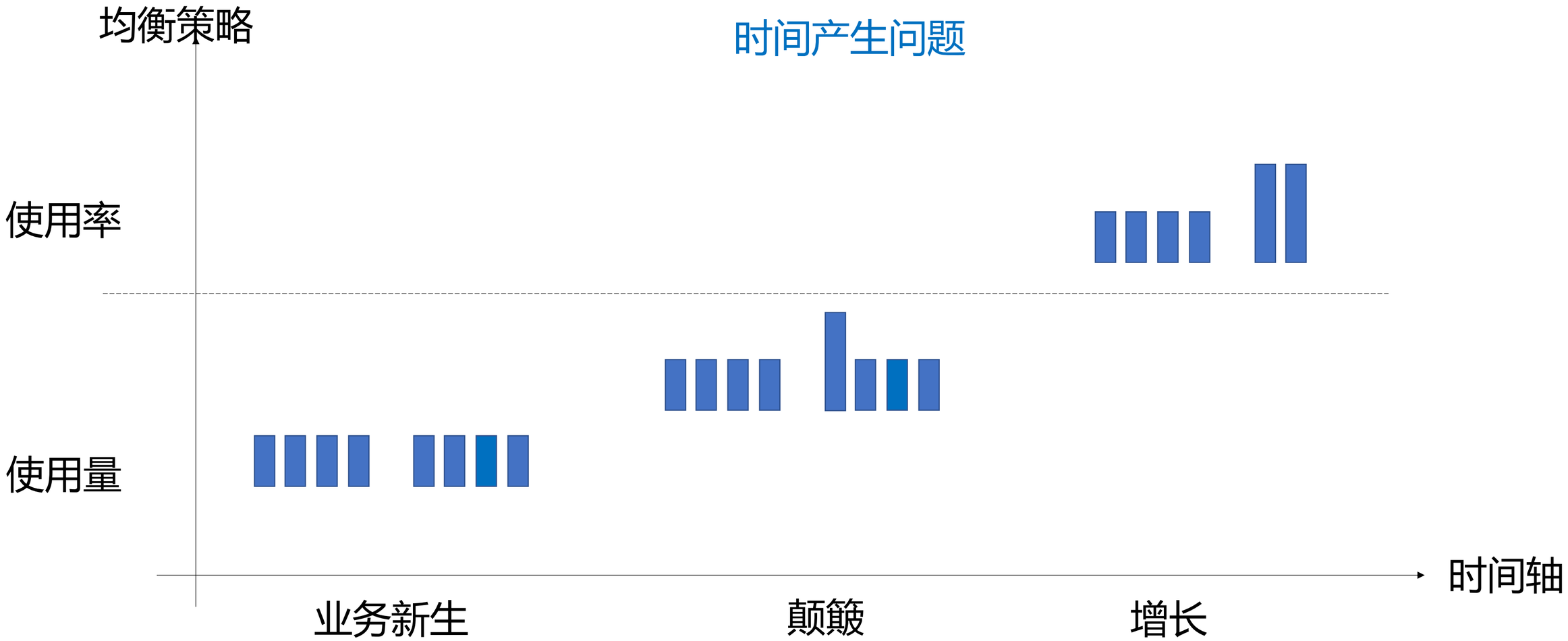



好的选择



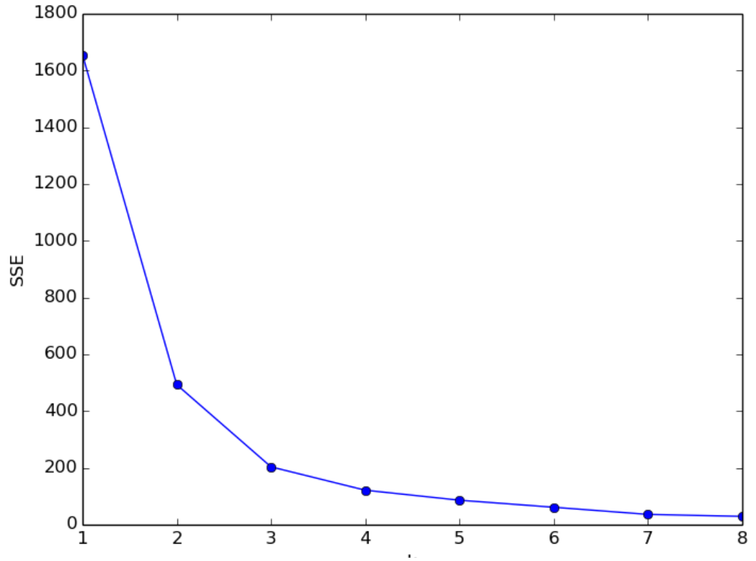

交换产生价值

# 不同时期的均衡

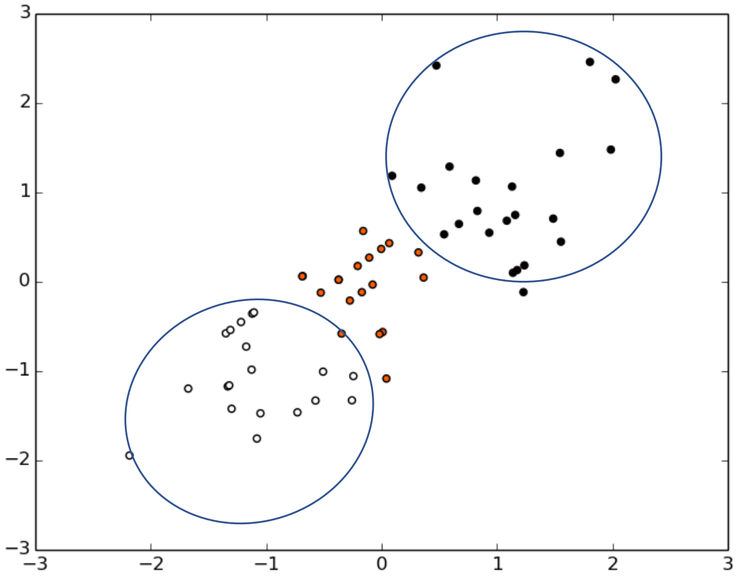


# 细粒度评估减少搬迁次数，降低均衡成本

- 细粒度的搬迁评估，减少搬迁次数
  - 节点/index级→磁盘/分片级
  - 热机：千万 (千分片数)
  - 冷机：>1亿（万分片数）

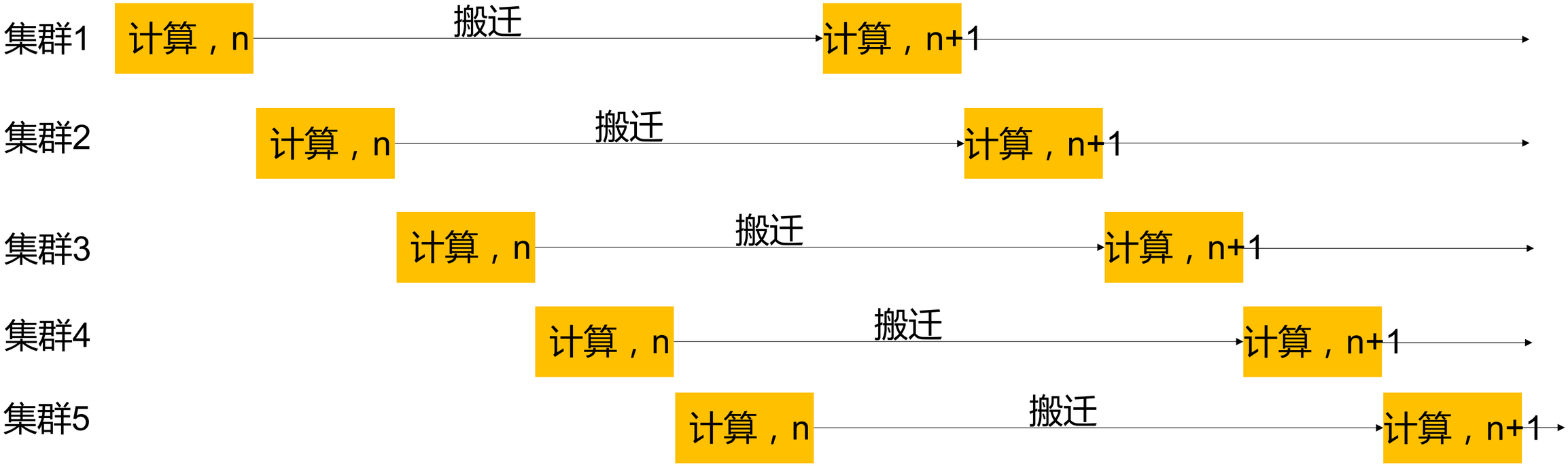



- 细粒度评估，增加了路径评估计算量
  - 机器学习降低需求
  - 聚类磁盘，控制局部搬迁




- 验证数据
  - 400秒→10秒

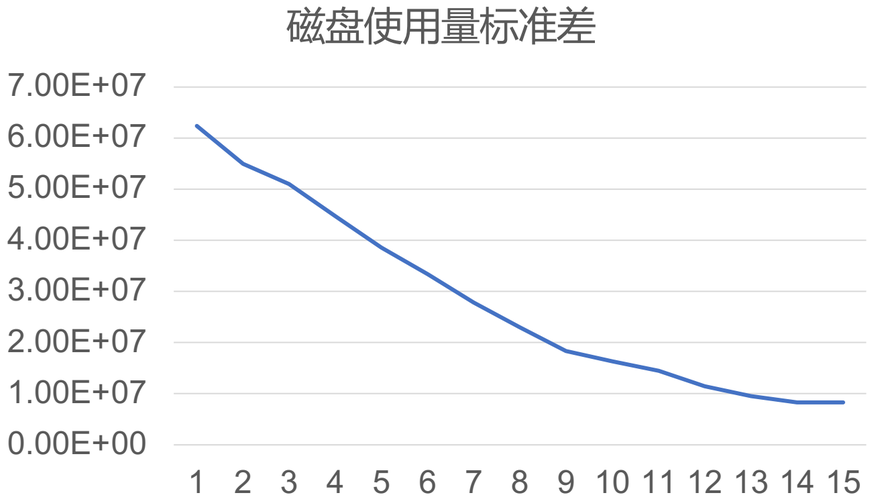
# 再快一点


集群1
计算，n
搬迁
计算，n+1
集群2
计算，n
搬迁
计算，n+1
集群3
计算，n
搬迁
计算，n+1
集群4
计算，n
搬迁
计算，n+1
集群5
计算，n
搬迁
计算，n+1

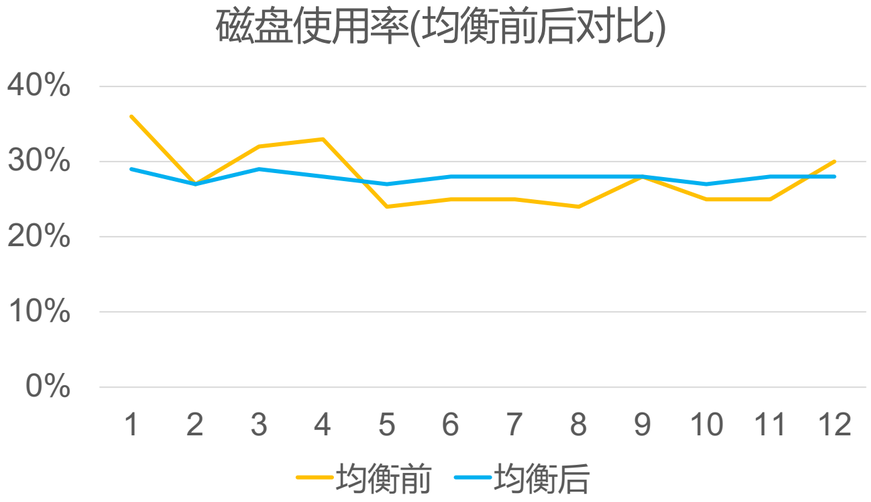
流水线加速

# 均衡效果的判断

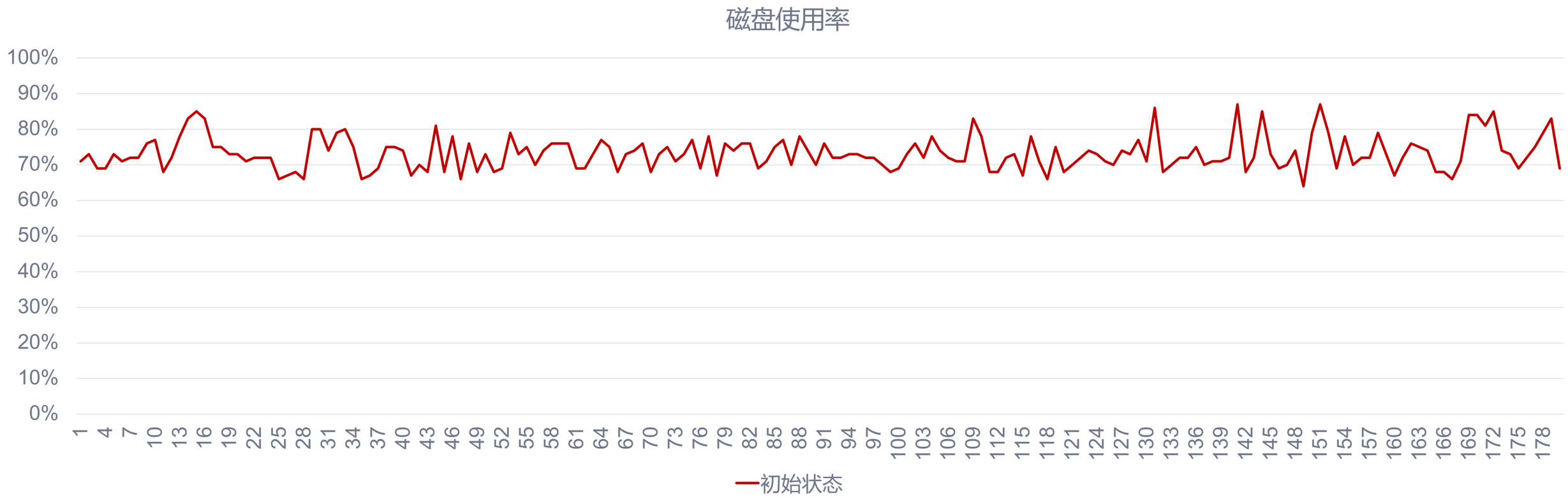
- 磁盘容量离散度
  - 越小越好




- 均衡前后磁盘使用率
  - 越直越好

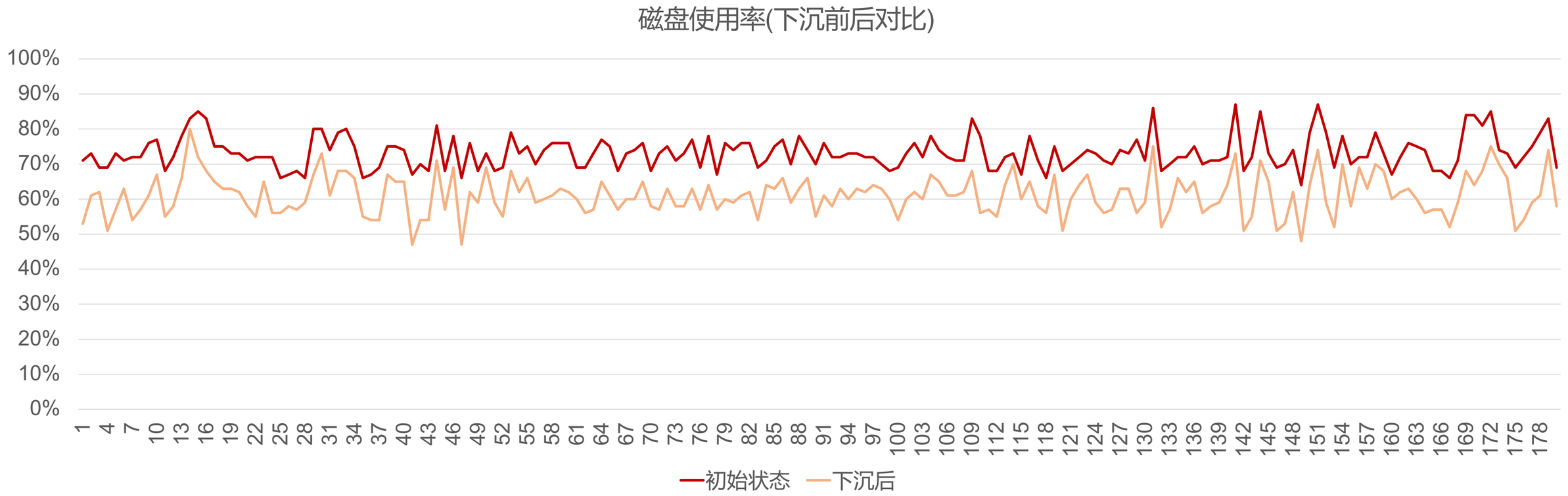


# 磁盘容量告警




初始状态：64%-87%

# 均衡前，先尝试数据下沉(热转冷)

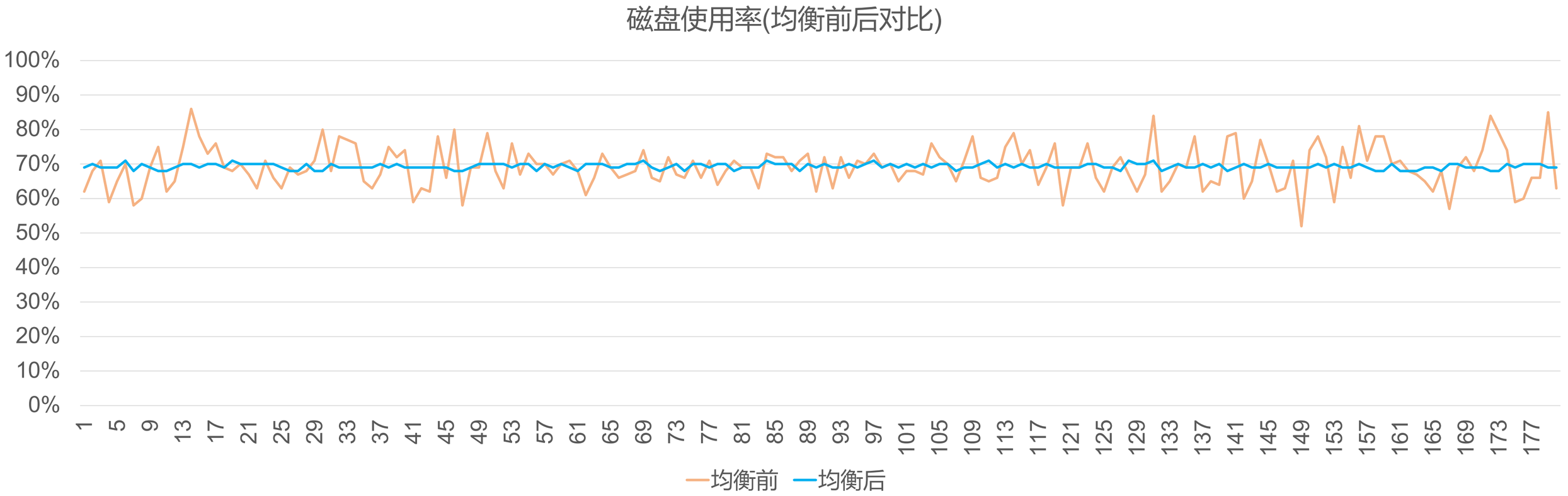



初始状态：64%-87%

数据下沉后：47%-80%

数据下沉，不能缓解均衡问题

# 均衡效果




均衡前：52%-86%　　平静的直线，没有倾斜和热点

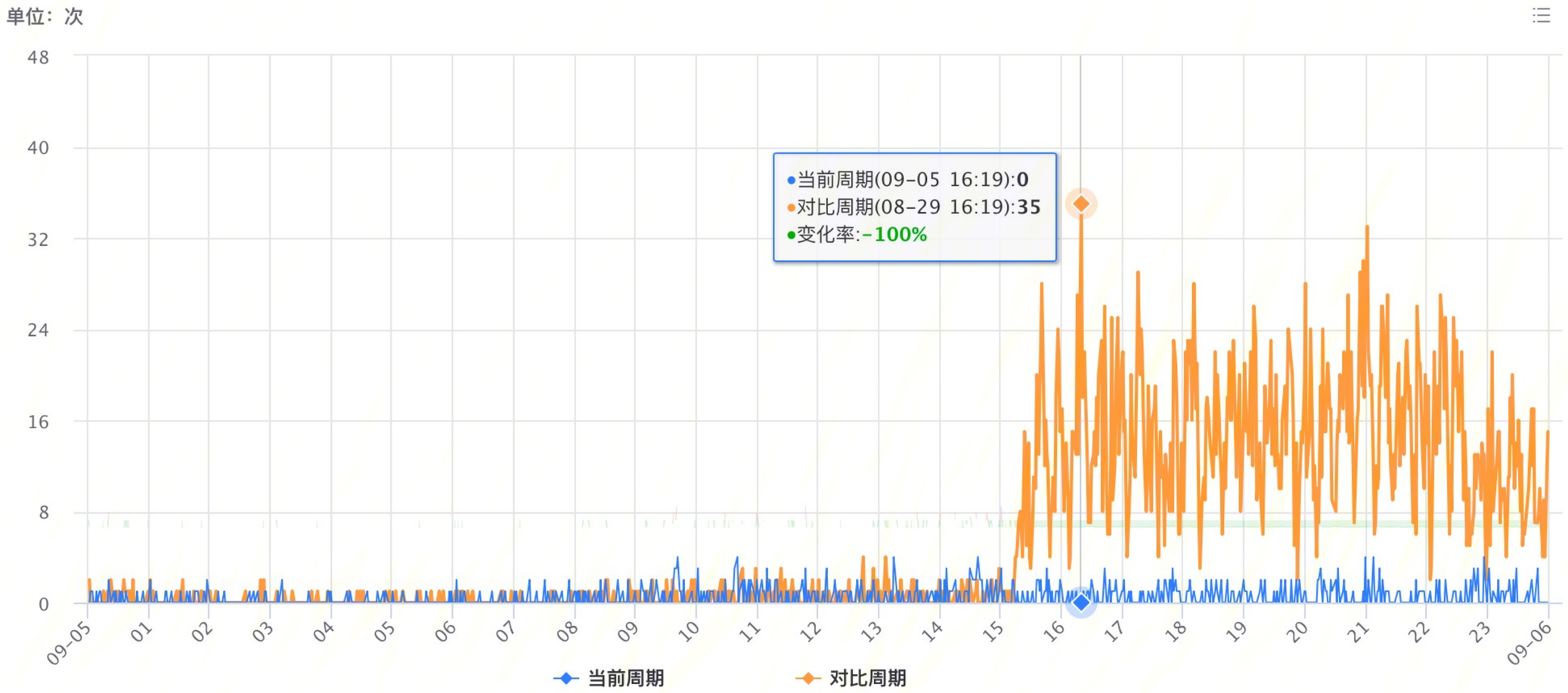
均衡后：68%-71%　　约等于多了15%容量

平衡集群，稳定运行

连锁反应，方案引入新挑战

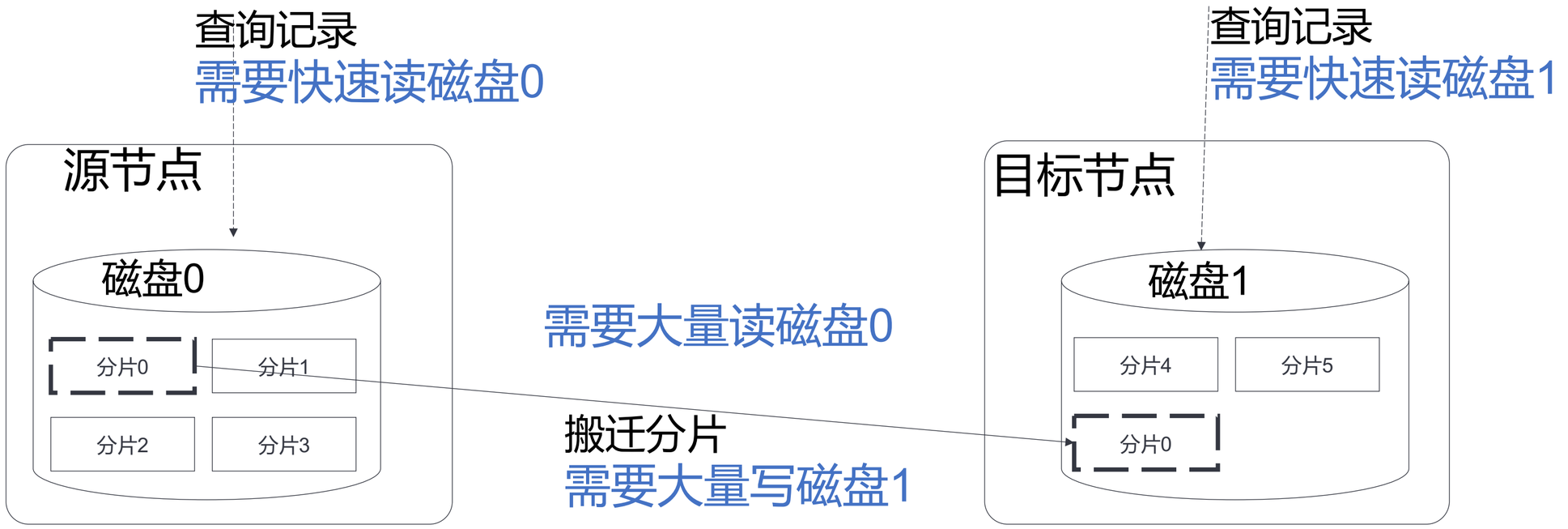
# 数据搬迁期间查询超时

搬迁期间超时数是日常超时数的几十倍



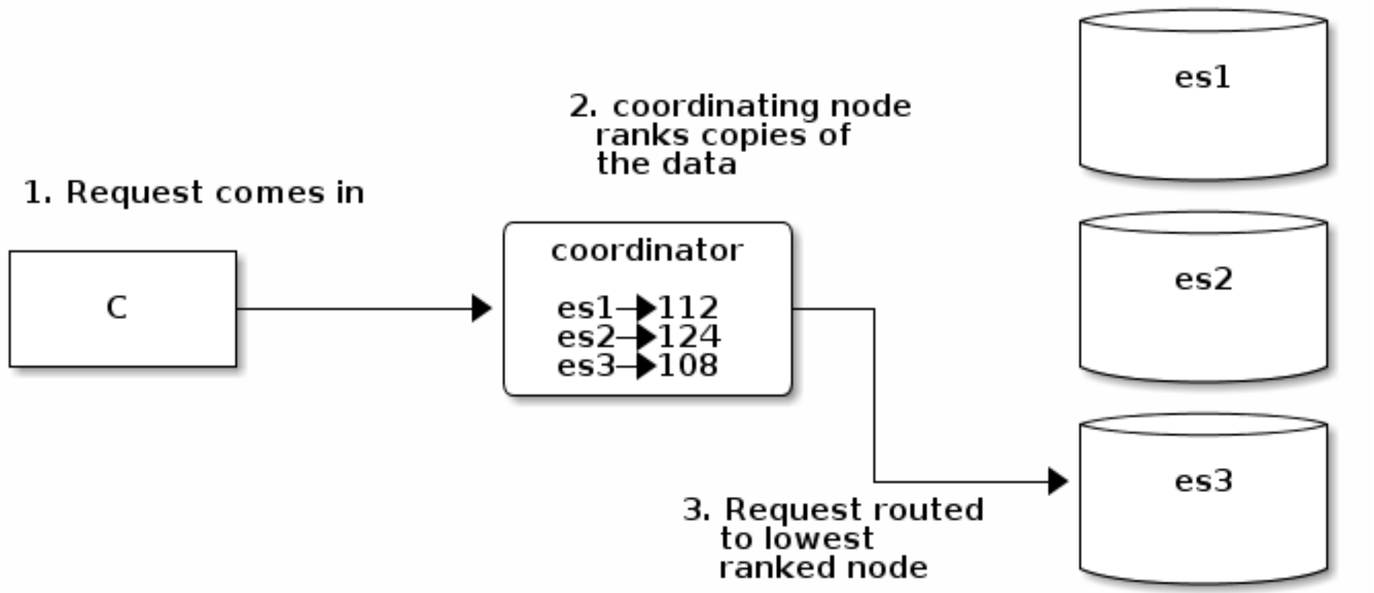
# 查询超时源于IO冲突

数据搬迁对磁盘IO的消耗→冲突←查询对磁盘IO的需求



# 自适应查询降低超时数

## 选择状态最好的节点分发查询请求




$$\Psi(s) = R(s) - 1/\bar{\mu}(s) + (\hat{q}(s))^3 / \bar{\mu}(s)$$

- 协调节点与数据节点之间的历史请求的响应时间
- 数据节点执行先前搜索所花费的时间
- 数据节点的搜索线程池的队列大小

C3: Cutting Tail Latency in Cloud Data Stores via Adaptive Replica Selection

# 自适应查询的不足之处（被动防守）

问题：

- 试错的代价

策略的运转依赖持续的查询超时
历史的查询超时能够指导后续的查询路由
避免饥饿，持续消耗一些查询来更新慢节点的质量数据

- 无米之炊窘境

批量搬迁导致数据的3副本所在节点都繁忙

对策：

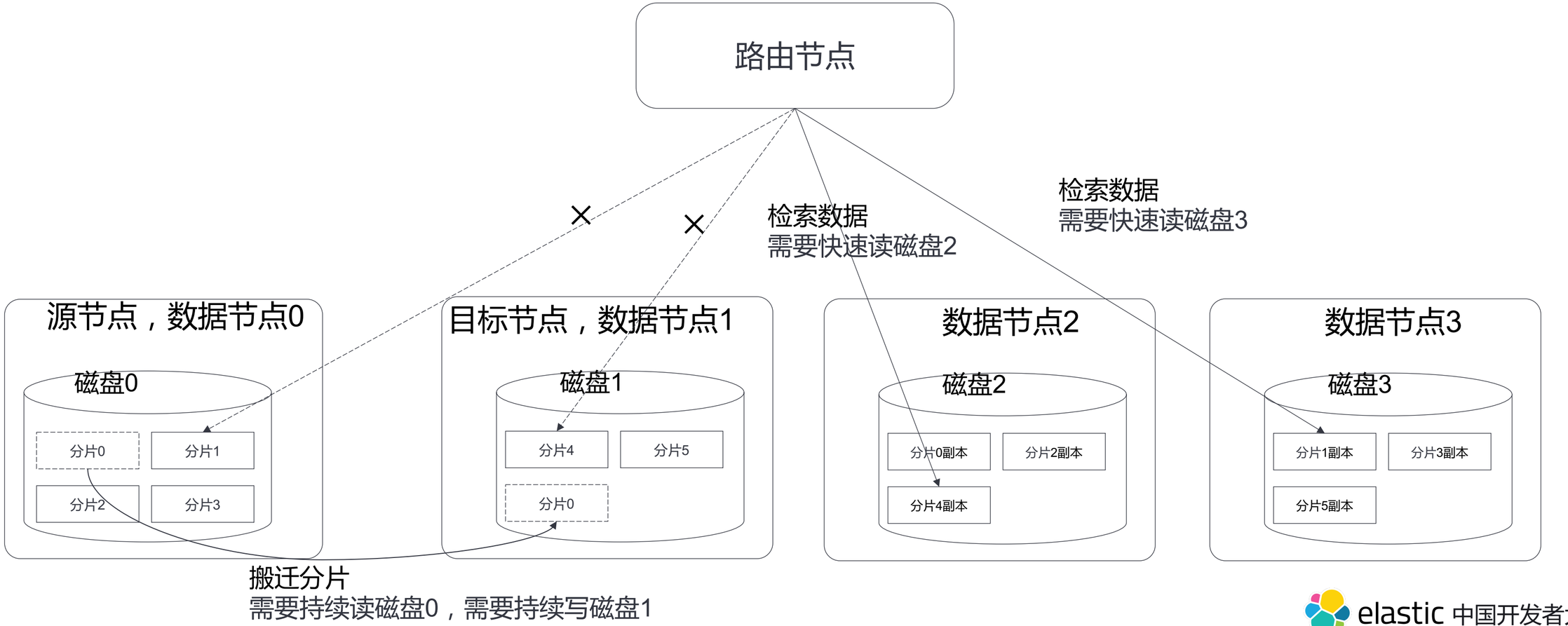
- 被动试错→主动选优

  主动选择未参与数据搬迁的节点执行查询

- 主动编排搬迁顺序，预留副本查询

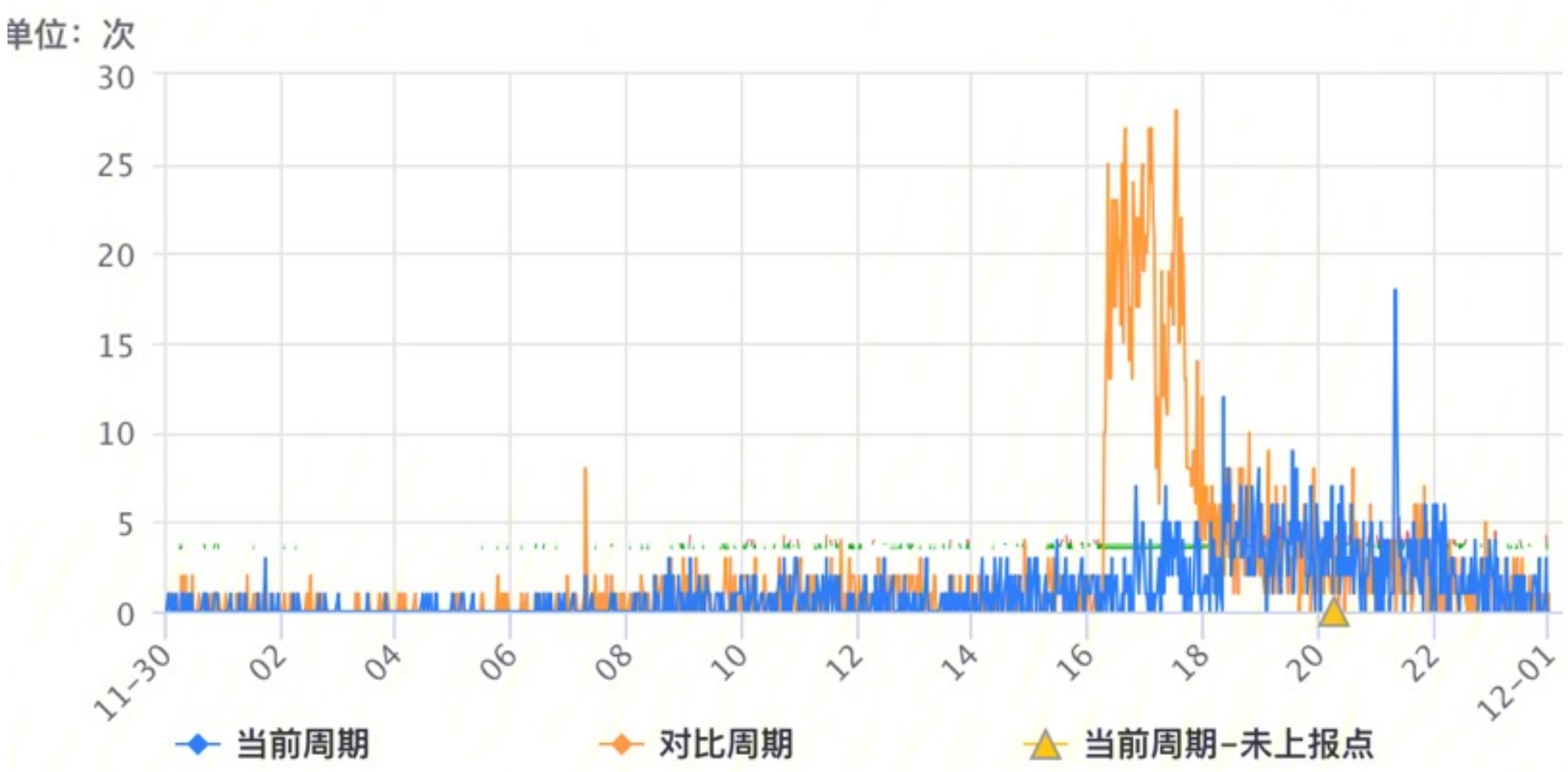
# 变被动为主动，通过IO隔离降低超时数(反击)

- 控制搬迁顺序，控制查询路由，实现参与搬迁的磁盘不参与搜索
- 故障适应性
- 减小搬迁性能的下降
- 数据一致性



# 防守+反击降低超时数

超时数基本与背景超时数相当



elastic 中国开发者大会 2023

# 感谢观看

elastic
中文社区

专业、垂直、纯粹的 Elastic 开源技术交流社区

https://elasticsearch.cn/