

# 360云平台如何玩转MongoDB

刘高辉

360·WEB平台部·DBA



# 目录

» 1 使用现状

» 2 运维案例

» 3 未来展望



# 使用现状

## 数据规模

实例数量 : 2500+

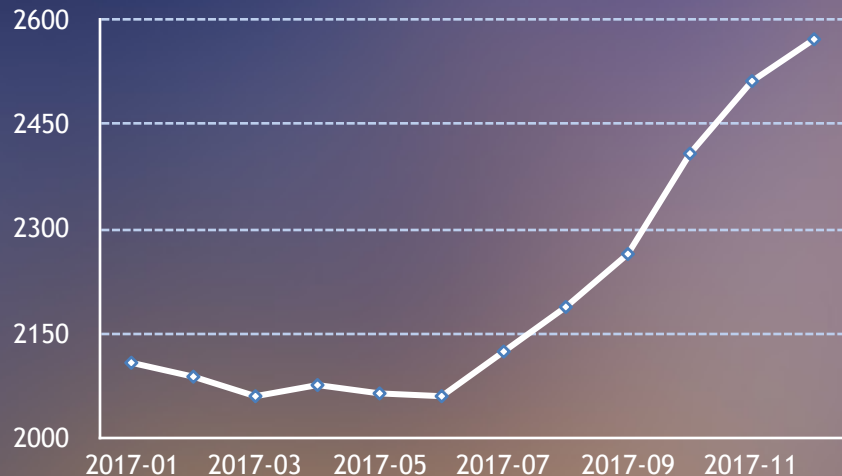
业务覆盖率: 90%+

日访问量 : 260(亿)+

## 运维人员

运维人员: 1.5人

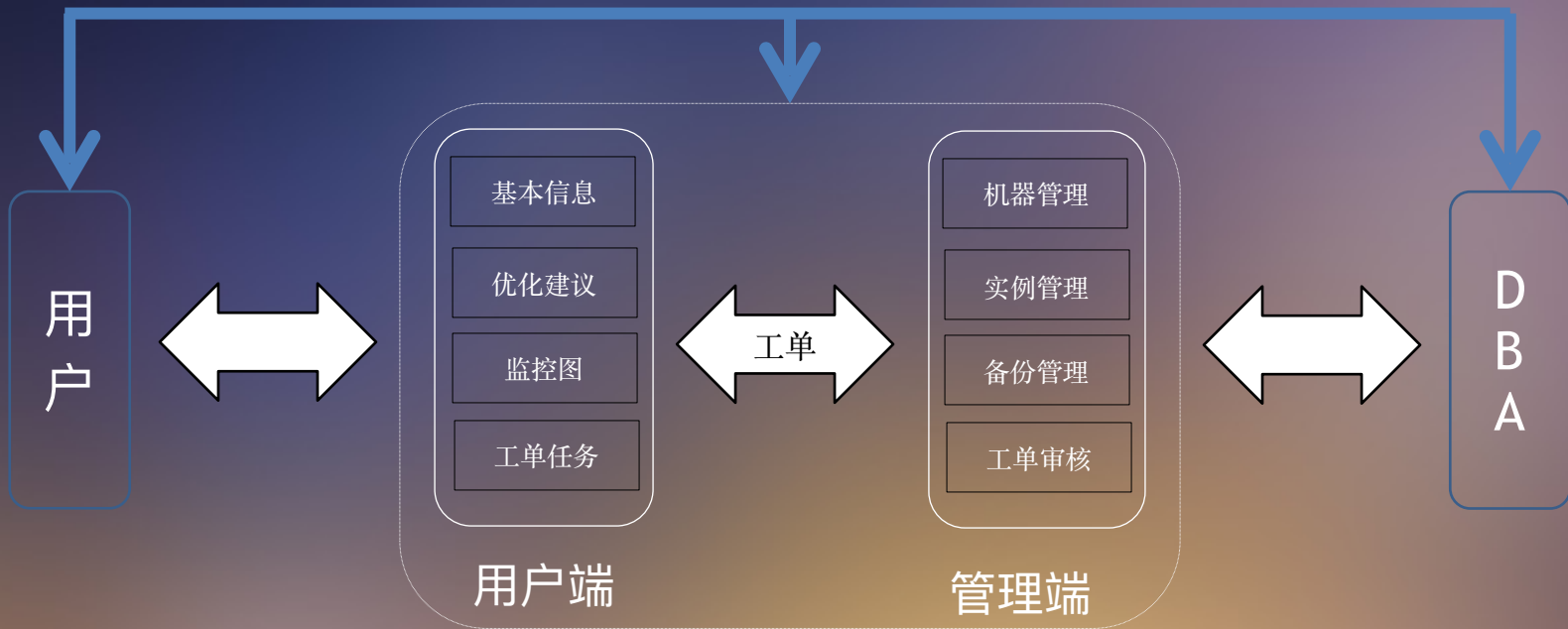
### 实例规模趋势图





# 自动化运维

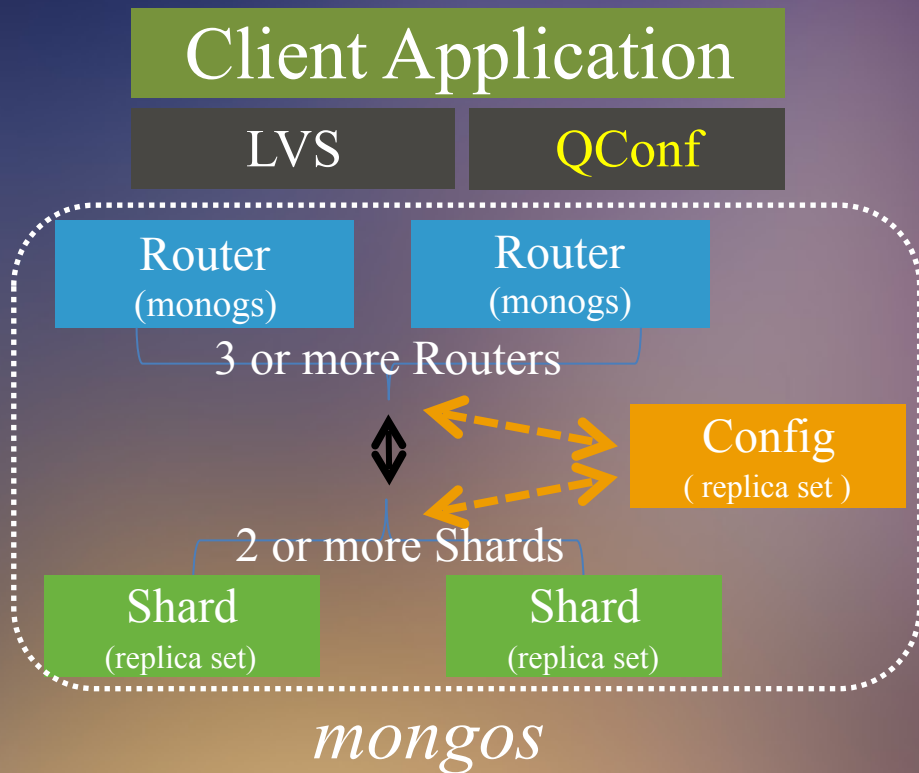
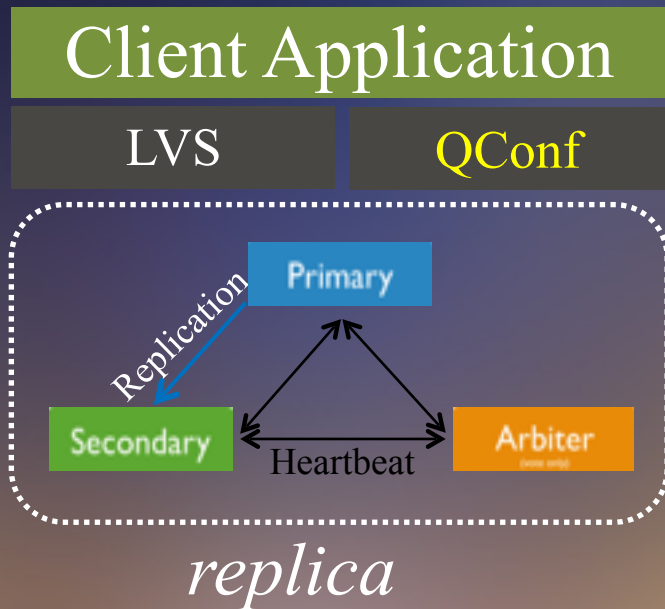
## » HULK云平台 (MongoDB模块)





# 自动化运维

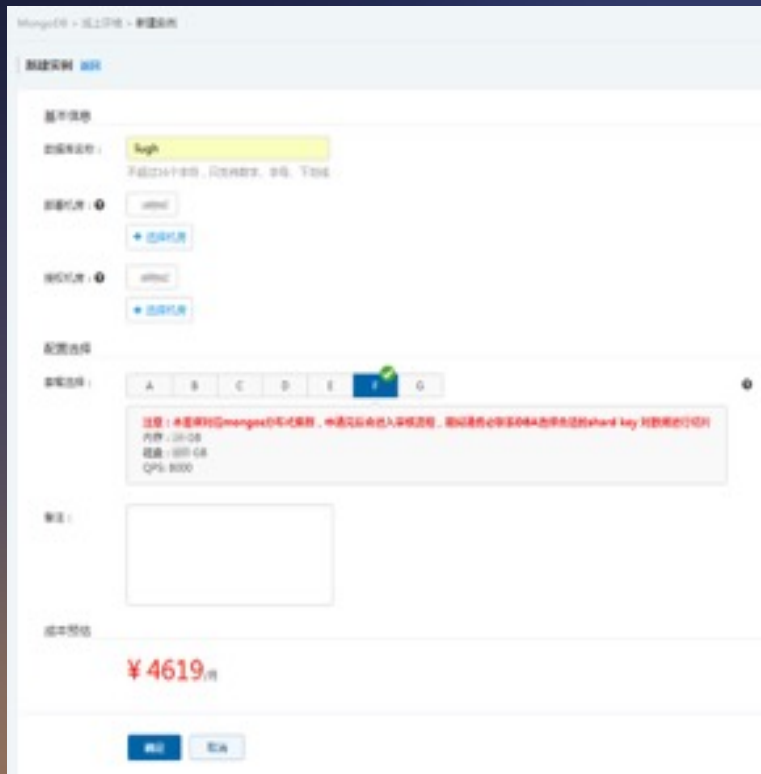
## 架构介绍





# 自动化运维

## 自动化部署



### 初始化

佳群初等

### 基本信息

- 数据库名
- 部署机房
- 授权机房

### 配置选择

- 套餐选择
- 备注



# 访问控制

## 数据泄露

Google 搜索结果: MongoDB 数据泄露

全部 新闻 图片 视频 地图 更多 设置 工具

找到约 47,600 条结果 (用时 0.24 秒)

**一场屠戮MongoDB的盛宴反思：超33000个数据库遭遇入侵勒索 ...**  
[www.freebuf.com/articles/database/125127.html](http://www.freebuf.com/articles/database/125127.html)  
2017年1月13日 - 当时，第一波被黑的MongoDB数据库中，Gevens观察到数据内容被清空，黑客还留下了一条“WARNING”信息：... 署名“Harak1r1”的黑客（或组织）大肆入侵了MongoDB数据库，清空里面的内容并向拥有者索要0.2比特币（约\$211）的赎金，否则数据将不予归还。... 请按照内部安全流程对数据泄露事件进行恰当处理。 8.

**运维配置缺陷导致大量MongoDB数据信息遭泄露- FreeBuf.COM | 关注 ...**  
[www.freebuf.com/news/106204.html](http://www.freebuf.com/news/106204.html)  
2016年6月7日 - 近日，黑客组织GhostShell泄露了大量的MongoDB数据库用户资料。数据遭到大量泄露统计该组织目前泄露的数据已达3600万条之多。MongoDB作为一个基于分布式文件存储的数据库，其主要功能特性包括——面向集合存储...

**MBS 公司MongoDB 数据库遭黑客入侵，5800 万用户数据泄露 ...**  
[hackernews.cc/archives/1085](http://hackernews.cc/archives/1085)  
2016年10月13日 - 著名的数据库托管和数据存储服务提供商Modern Business Solutions (MBS) 近日因MongoDB 数据库安全防护措施不到位遭到黑客攻击，该数据库泄露5800 万用户信息包括名称、IP 地址、邮件账号、职业、车辆数据、出生日期等。据Twitter 信息了解，该黑客昵称叫@0x2Taylor 并因最近高调入侵和泄露数据行为而 ...

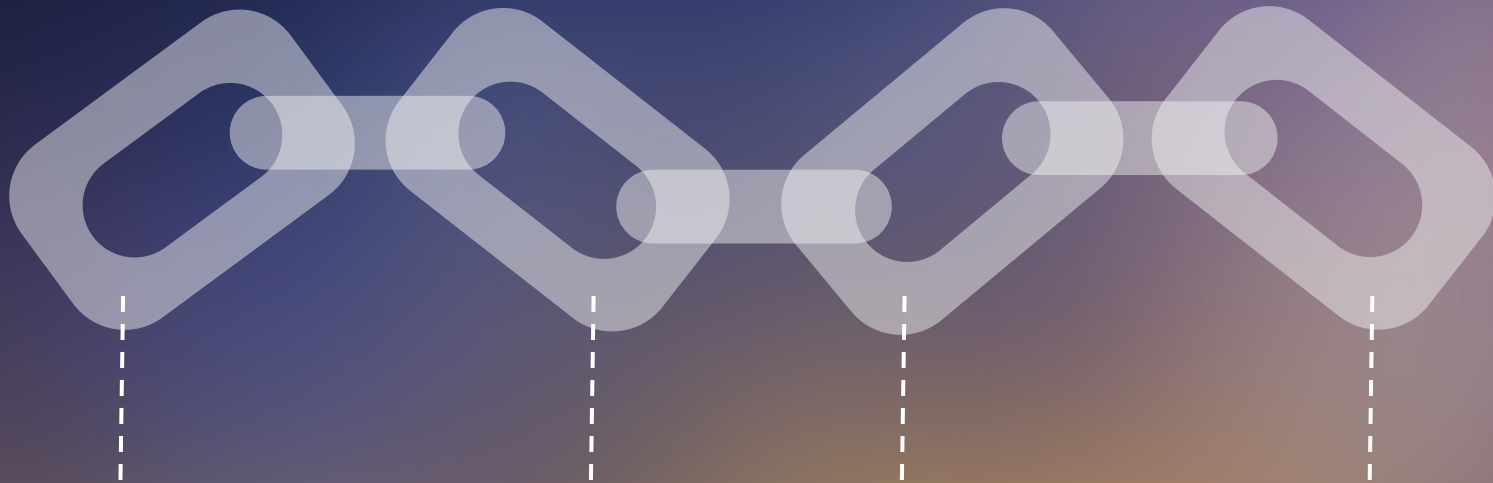
**因MongoDB的这个缺点，意外泄露3100万用户信息- 51CTO.COM**  
[database.51cto.com/art/201712/560679.htm](http://database.51cto.com/art/201712/560679.htm)  
2017年12月12日 - 根据ZDNet 的相关报道，MongoDB 的缺点之一就是数据库的预设值允许任何人都能存取及下载，而且，al.type 开发者并未设定数据库的存取密码，才会让所有的数据曝光。

**223GB数据泄露：美共和党电话拉票公司MongoDB数据库被黑| E安全**  
<https://www.easyaq.com/news/1656606684.shtml>



# 访问控制

## 安全控制



更换端口  
公网屏蔽  
防火墙

普通用户启动

开启验证  
多用户管理  
权限控制

备份及恢复





# 访问控制

## 权限控制

权限控制

最小  
授权

审核

提权



# 监控

» 报警 服务器 + 实例

» 预警

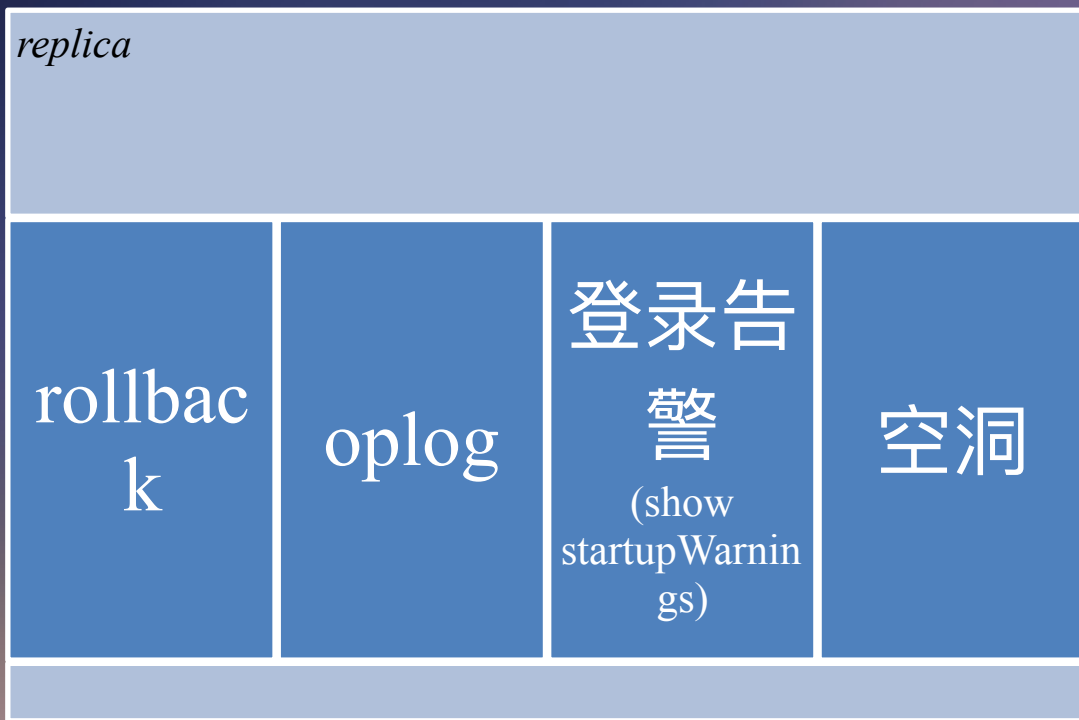




# 监控

» 报警 服务器 + 实例

» 预警



时长

大小



# 监控

» 报警 服务器 + 实例

» 预警

*mongos*

chunk  
分布

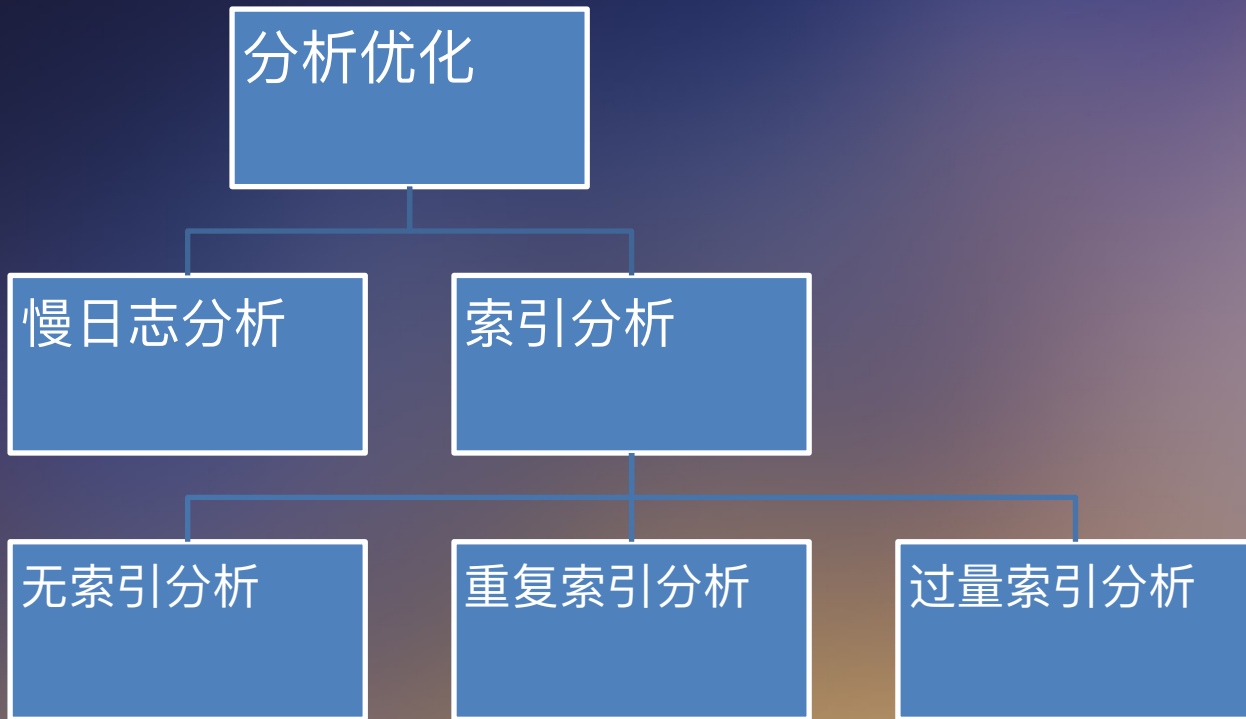
数据分  
布

jumbo  
特大块

balanc  
er状态



# 分析优化





# 分析优化

## » 慢日志分析

### 收集

- 慢日志收集
- 过滤整合

### 分析

- `mloginfo`工具分析

### 展示

- HULK：慢日志详情 + 索引信息
- DBA ：分析异常 + 汇总报告



# 优化分析

## » 重复索引分析

### 收集

- 全量索引收集

### 分析

- 拆分拼建MySQL表
- `mongo-duplicate-key-checker`分析

### 展示

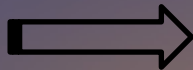
- HULK：重复索引详情 + 优化建议
- DBA ： 汇总报告



# 优化分析

## » 搭建MySQL表

```
8888:PRIMARY> db.test.getIndexes()
[
  {
    "v" : 1,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "liugh.test"
  },
  {
    "v" : 1,
    "key" : {
      "name" : 1
    },
    "name" : "name_1",
    "ns" : "liugh.test",
    "background" : true
  },
  {
    "v" : 1,
    "key" : {
      "name" : 1,
      "age" : 1
    },
    "name" : "name_1_age_1",
    "ns" : "liugh.test",
    "background" : true
  }
]
8888:PRIMARY>
```



```
Table: liugh.test
Create Table: CREATE TABLE `liugh.test` (
  `_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `name` int(11) DEFAULT NULL,
  `age` int(11) DEFAULT NULL,
  PRIMARY KEY (`_id`),
  KEY `name_1` (`name`),
  KEY `name_1_age_1` (`name`,`age`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```





# 备份系统

## 数据恢复

数据库:

表来源:  选择表  手工输入

仅能选择当前该数据库存在的表, 如果要恢复已被drop的表, 请选择【手工输入】

表名: 全选 可选表 全选 已选表

请输入表名称, 如恢复的表已被删除, 请选择手工输入。

disk\_capacity\_warning\_tmp

machine\_state\_info\_tmp

mongo\_auto\_operation\_info

mongo\_backup\_fail\_info

mongo\_basic\_info

mongo\_basic\_info\_bak

请输入表名称

disk\_capacity\_warning

machine\_state\_info

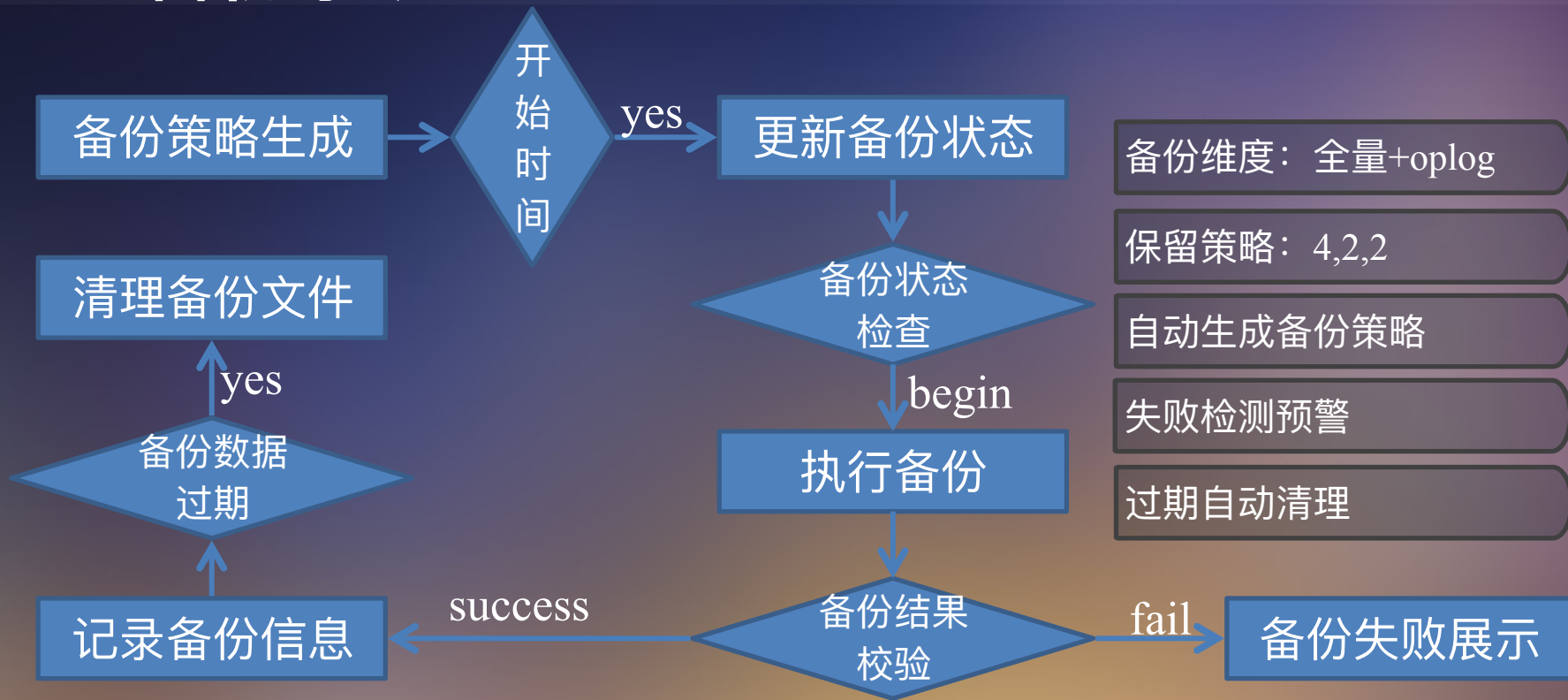
选择恢复时间:  日  时  分  秒

恢复时间务必早于误操作时间, 否则将影响恢复结果

恢复原因:



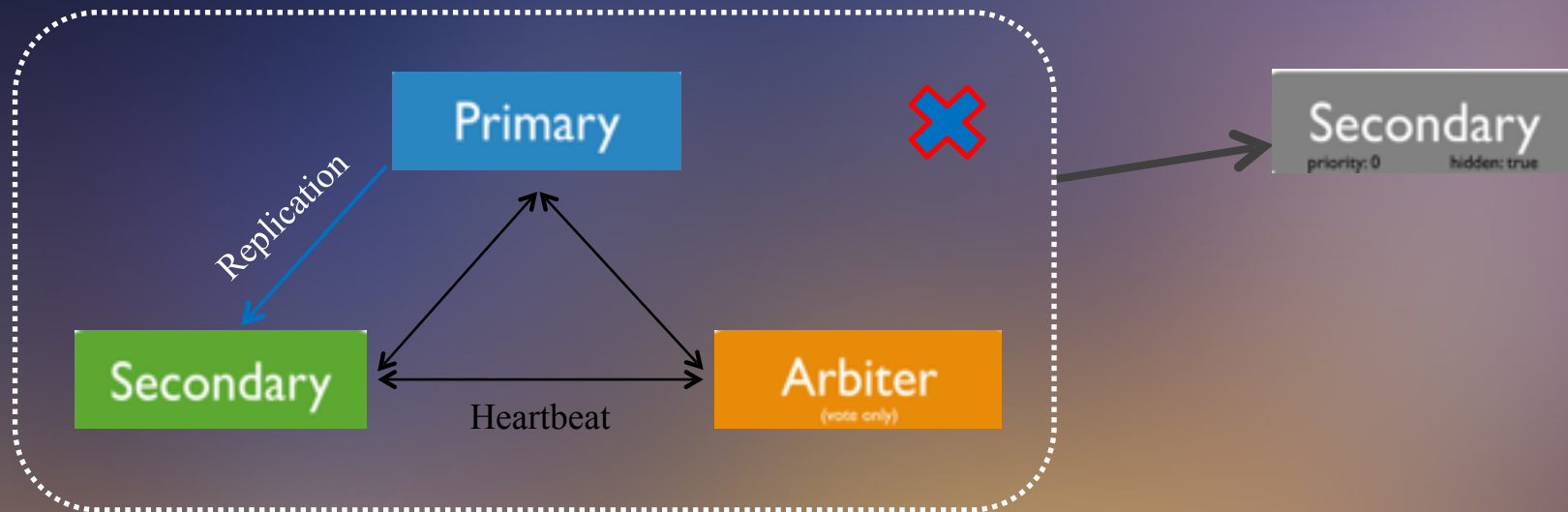
# 备份系统





# 备份系统

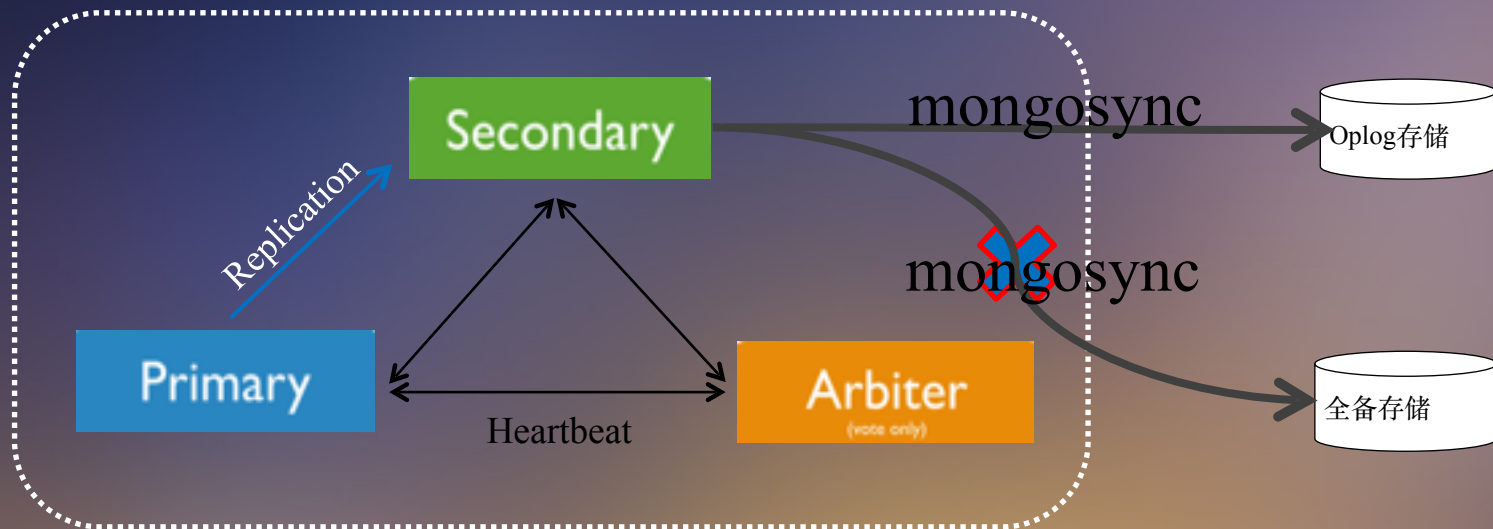
## » 备份1.0: 隐藏节点





# 备份系统

## » 备份2.0: 逻辑拉取





# 备份系统

## » mongosync介绍

目的端 源端		副本集		mongos	
		全量	增量	全量	增量
副本集		✓	✓	✓	✓
mongos		✓	✗	✓	✗

全版本兼容

集群、库、集合、语句

基于点位的oplog流备份

oplog重放

性能强悍



# 运维案例

## » 版本兼容

```
MongoDB shell version: 2.4.13
connecting to: 127.0.0.1:8888/admin
8888:PRIMARY> db.version()
3.0.14
8888:PRIMARY> use liugh
switched to db liugh
8888:PRIMARY> db.liugh.ensureIndex({"name":1}, {"background":true})
8888:PRIMARY> db.liugh.getIndexes()
[ ]
8888:PRIMARY> █
```

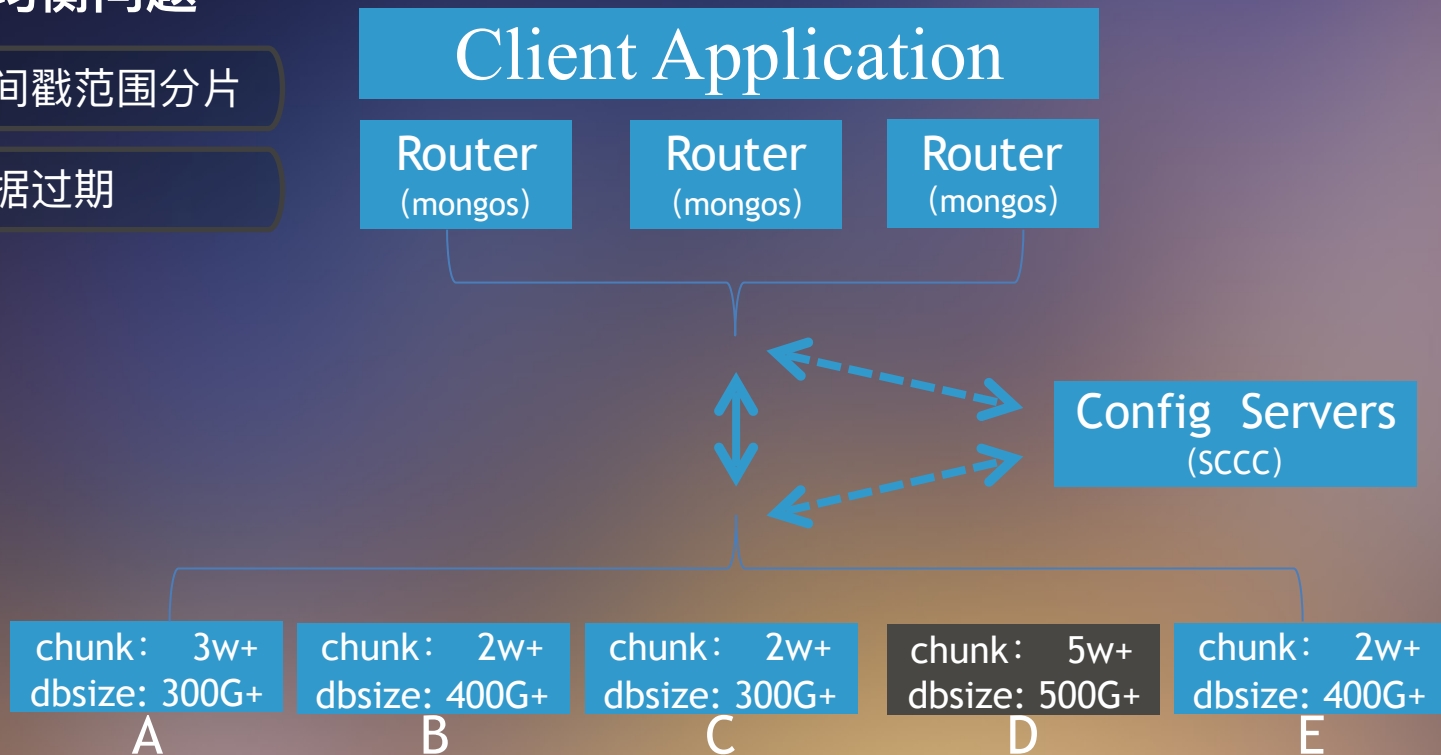


# 运维案例

## 数据均衡问题

时间戳范围分片

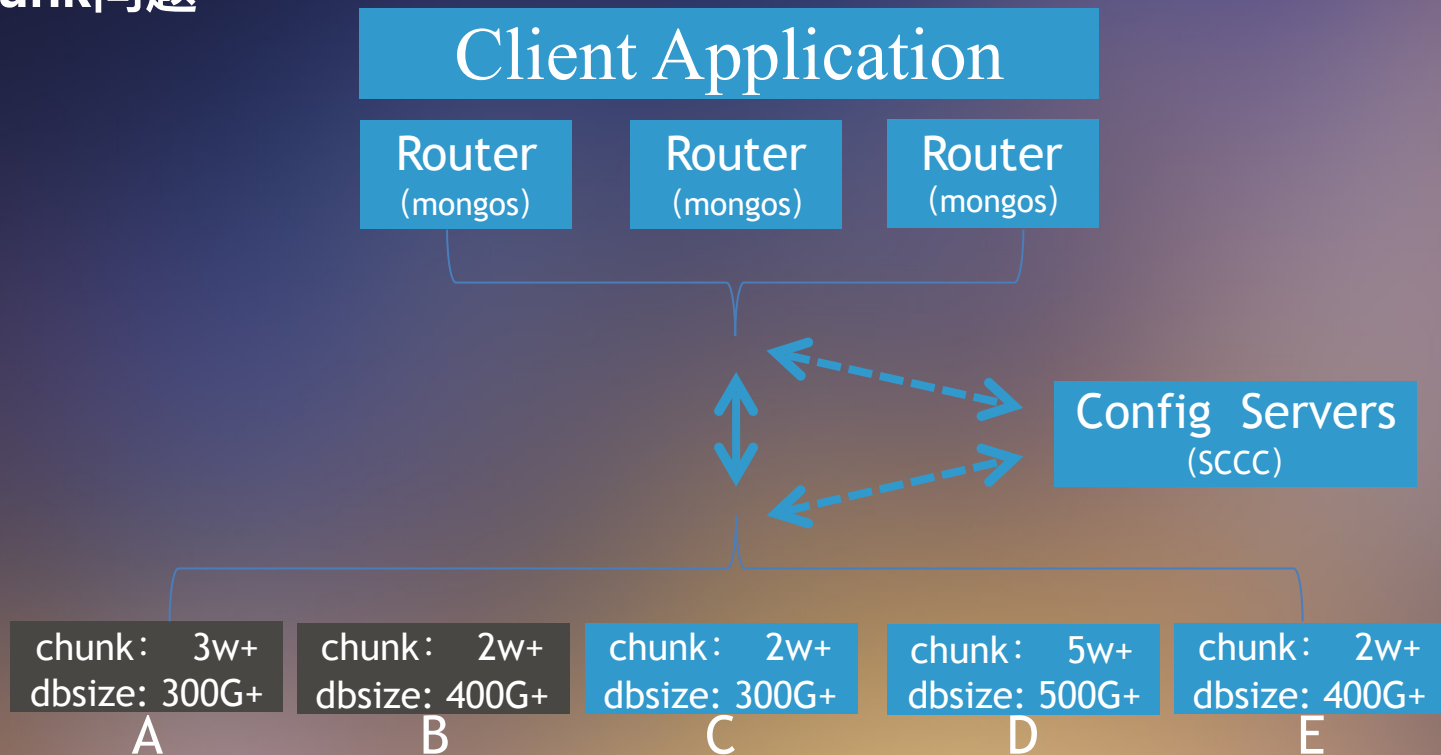
数据过期





# 运维案例

## » 空chunk问题







# 实践案例



mongoing  
中文社区

IT大咖说  
知识分享平台

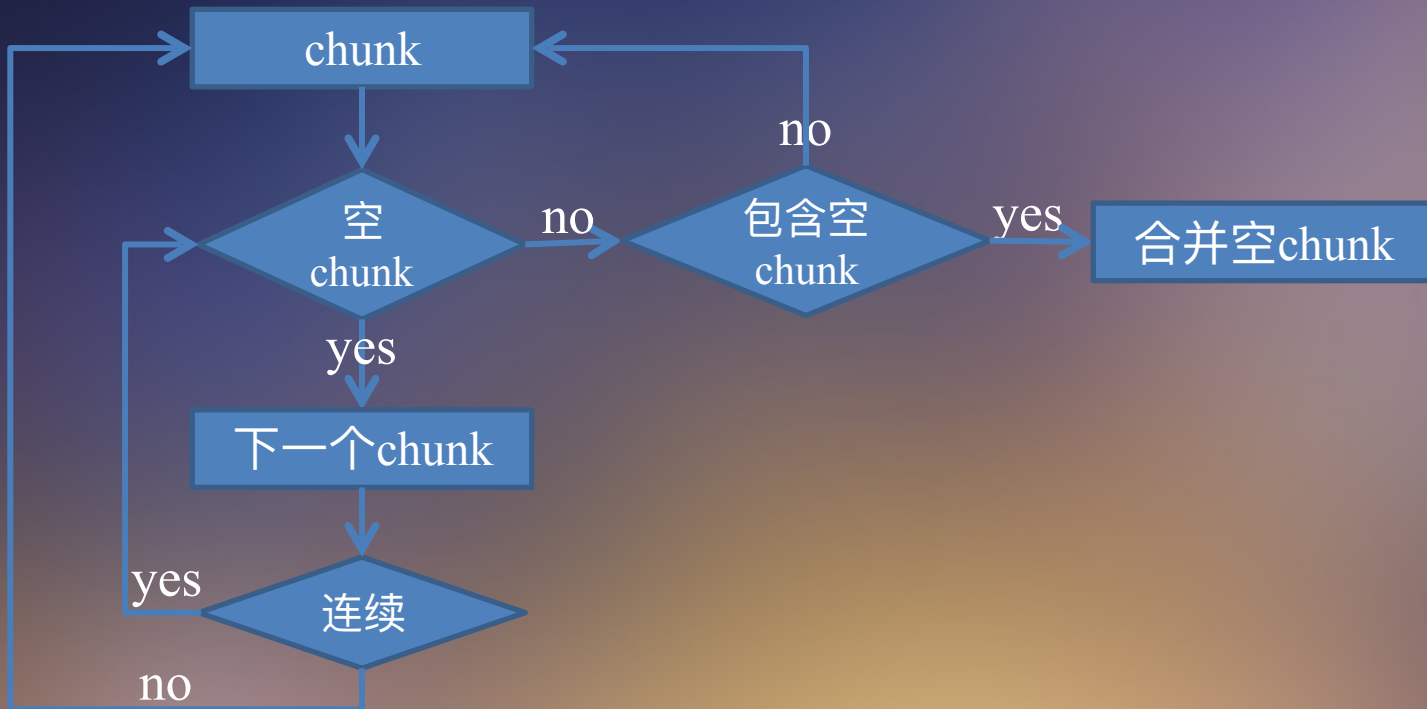
## » chunk连续

```
mongos> db.getSiblingDB("config").chunks.find({}, {"lastmod":1, "min":1, "max":1, "shard":1, "_id":0}).sort({"lastmod":1})
{
  "lastmod" : Timestamp(36728, 2),
  "min" : {
    "add_time" : NumberLong(1504984365),
    "stream_source" : "cdn",
    "sname" : "huajiao"
  },
  "max" : {
    "add_time" : NumberLong(1504984439),
    "stream_source" : "relay",
    "sname" : "live_camera_v2"
  },
  "shard" : "7392"
}
{
  "lastmod" : Timestamp(36728, 3),
  "min" : {
    "add_time" : NumberLong(1504984439),
    "stream_source" : "relay",
    "sname" : "live_camera_v2"
  },
  "max" : {
    "add_time" : NumberLong(1504984565),
    "stream_source" : "cdn",
    "sname" : "xhs-360-live"
  },
  "shard" : "7392"
}
```



# 空chunk合并

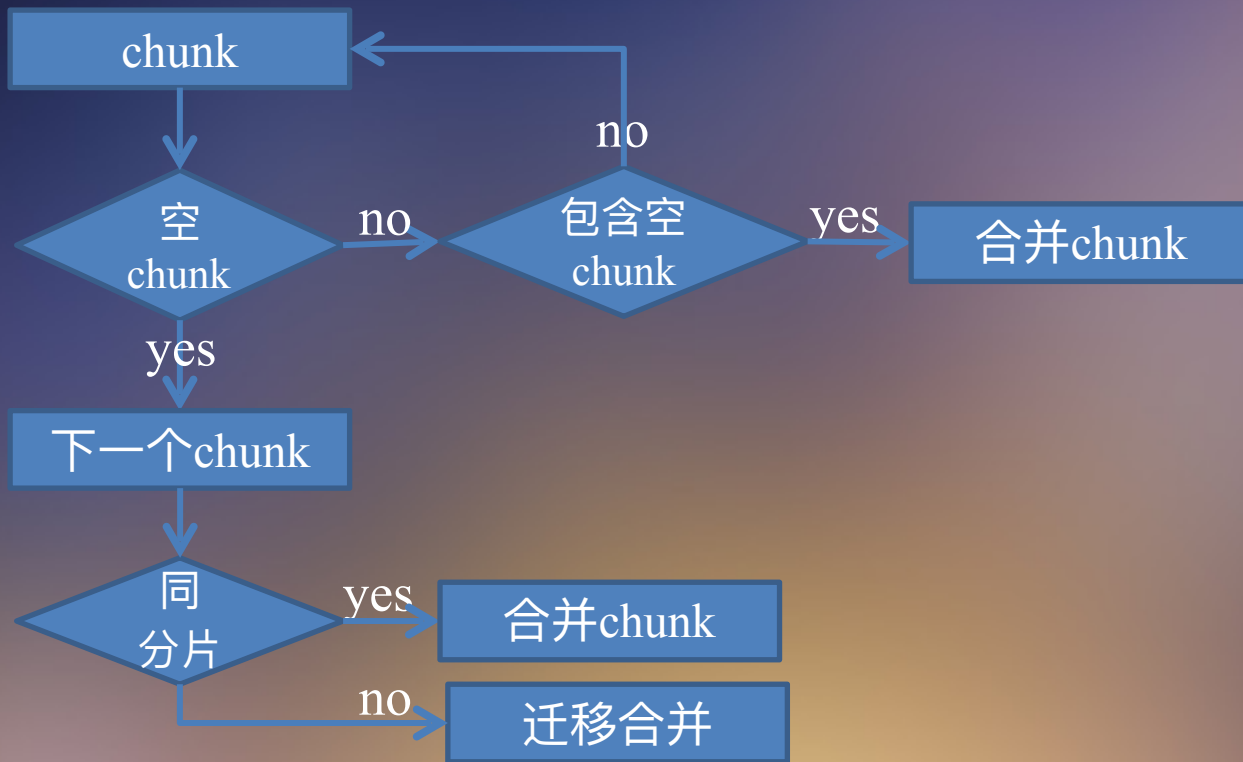
## 分片连续空chunk合并





# 空chunk合并

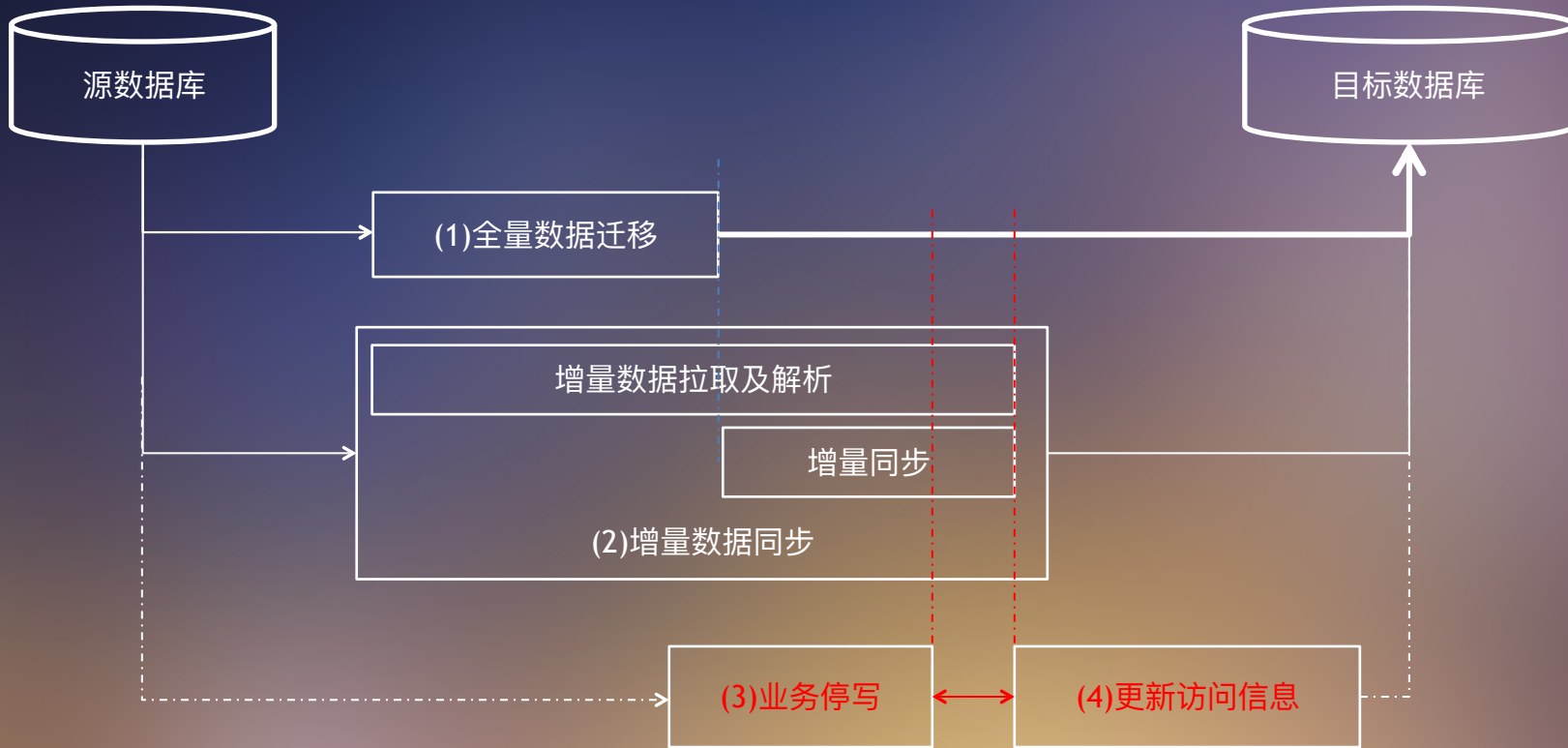
## 集群空chunk合并





# 运维案例

## 版本升级





# 未来展望

## 集群体检

- 一站式排查集群故障点

## 智能运维

- 自动故障恢复
- 自动性能优化

## 工具开发

- 数据传输工具
- 性能分析工具

THANK YOU