Trip.com Group™ 携程集团

# ClickHouse在携程PB级日志系统实践

携程　林东煜

2023.03.25

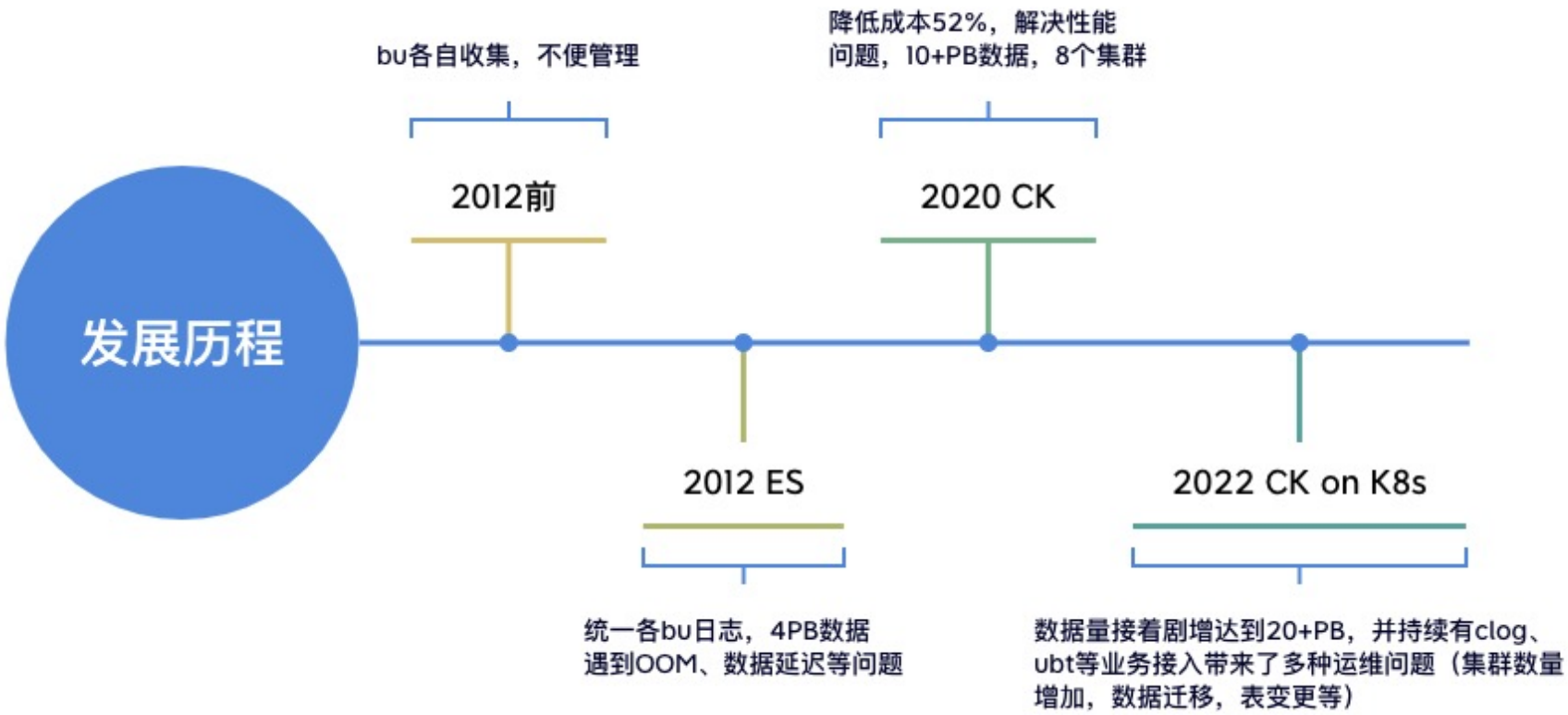
Trip.com

CONTENTS

# 目录

# 历史与现状

# 日志平台发展历程


发展历程
2012前
bu各自收集，不便管理
2012 ES
统一各bu日志，4PB数据
遇到OOM、数据延迟等问题
2020 CK
降低成本52%，解决性能
问题，10+PB数据，8个集群
2022 CK on K8s
数据量接着剧增达到20+PB，并持续有clog、
ubt等业务接入带来了多种运维问题（集群数量
增加，数据迁移，表变更等）

# 集群现状

Trip.com

服务器：600+

数据量：60w亿行

数据表：1w+

存储空间：30+PB

查询：10M+/天

集群数量：20+

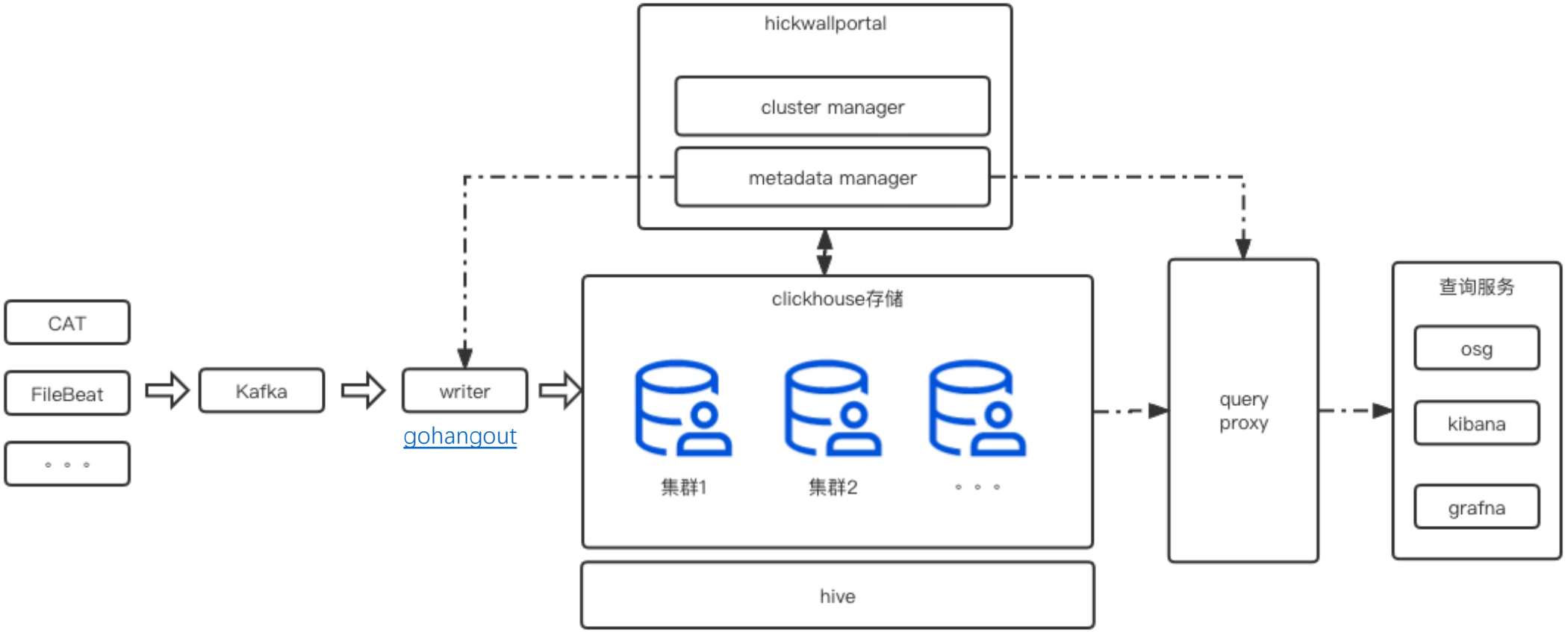
P90：500ms

P99：3s

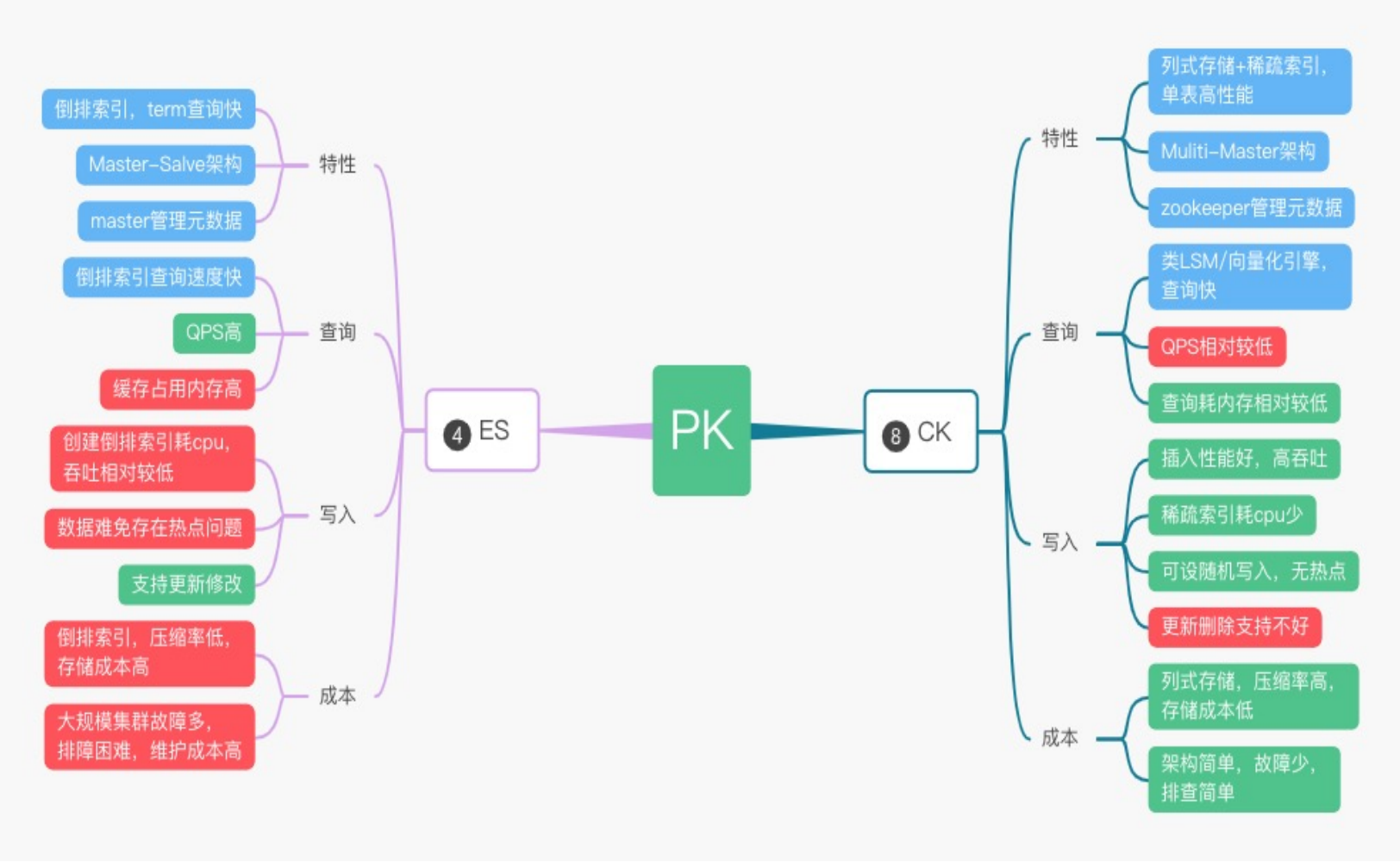
Trip.com

# 日志系统实践

# 整体部署架构


hickwallportal
cluster manager
metadata manager
clickhouse存储
集群1
集群2
CAT
FileBeat
Kafka
writer
gohangout
hive
query
proxy
查询服务
osg
kibana
grafna

# ClickHouse与ElasticSearch对比



# 库表设计

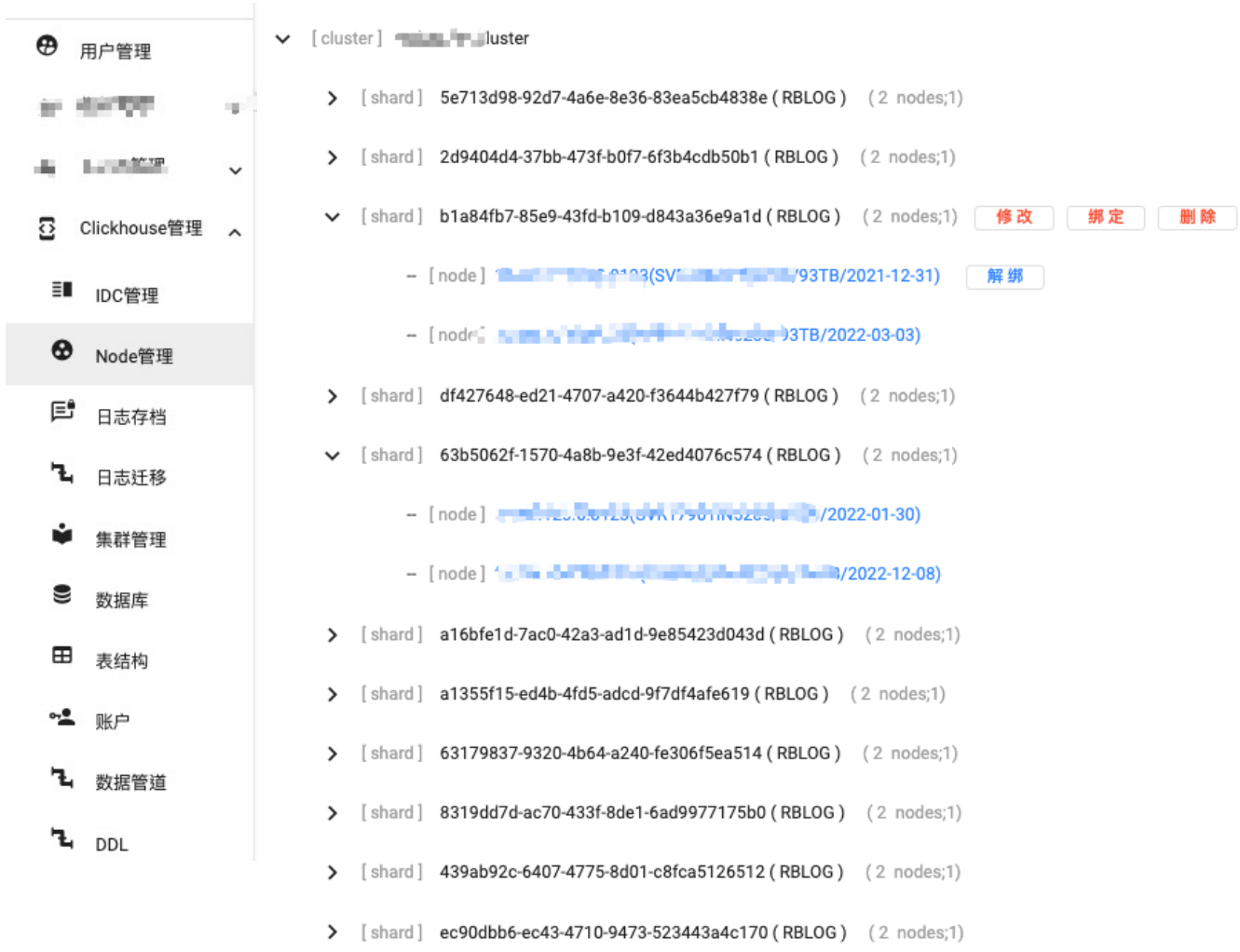
```
CREATE TABLE log.xxxx (
    `timestamp` DateTime,
    `_log_increment_id` Int64,
    `host_name` LowCardinality(String),
    `log_level` LowCardinality(String),
    `message` String,
    `message_prefix` String MATERIALIZED substring(message, 1, 256),
    `_tag_keys_str` Array(String),
    `_tag_vals_str` Array(String),
    `_tag_keys_str` Array(String),
    `_tag_vals_float` Array(Float64),
    ......
    INDEX idx_message_prefix message_prefix TYPE tokenbf_v1(8192,2,0) GRANULARITY 16,
    ......
)
ENGINE = ReplicatedMergeTree('/clickhouse/tables/{shard}/xxxx','{replica}')
PARTITION BY toYYYYMMDD(timestamp)
ORDER BY(timestamp,_log_increment_id,host_name,log_level)
TTL timestamp + toIntervalHour(168)
SETTINGS index_granularity = 8192
```

1. 双list存tags
2. 按天分区，时间排序
3. Tokenbf_v1优化term查询
4. 全局唯一递增id(翻页，明细)
5. ZSTD

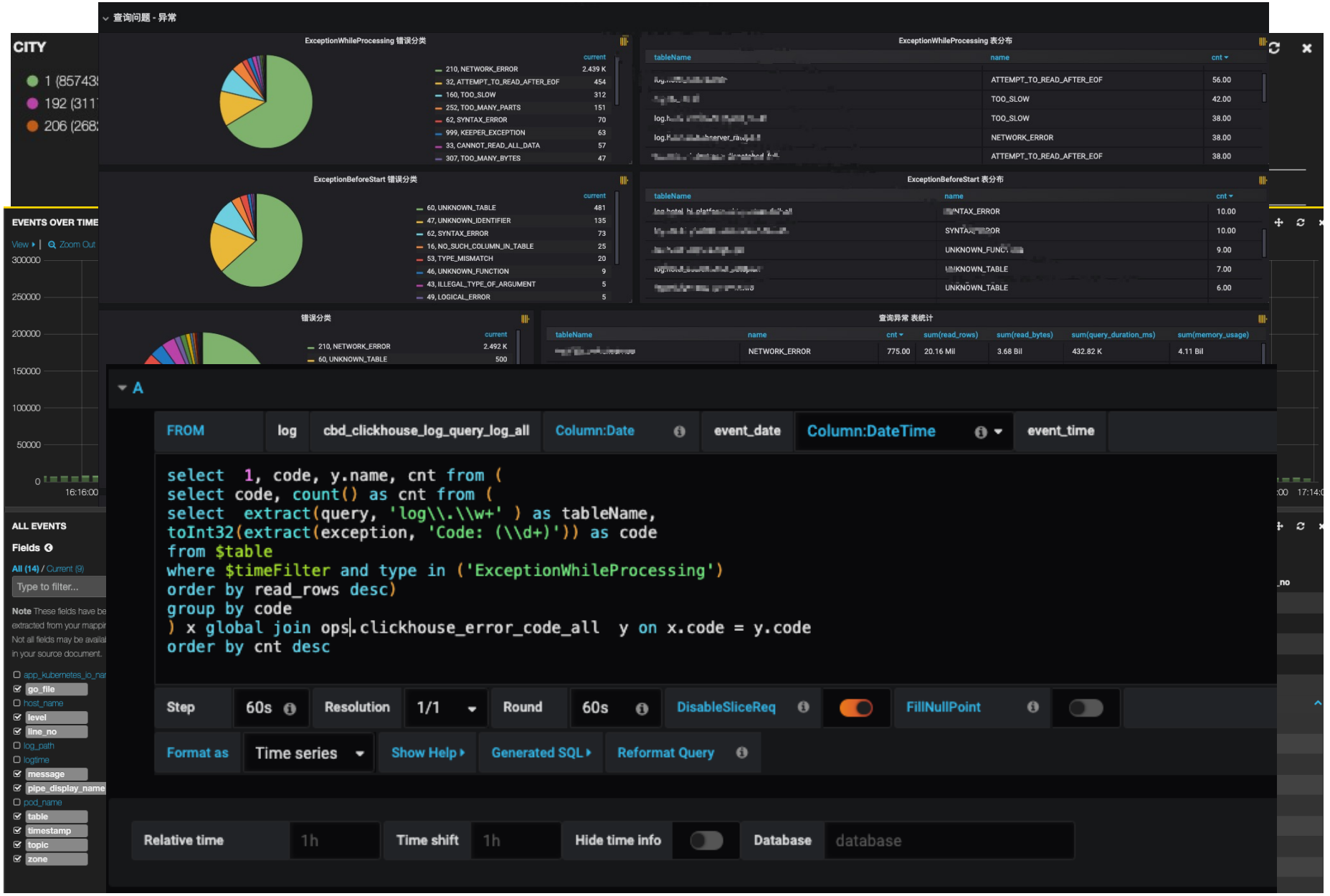
# 集群管理平台

Trip.com

- 集群部署
- 集群管理
- 节点管理
- DDL管理
- 监控告警

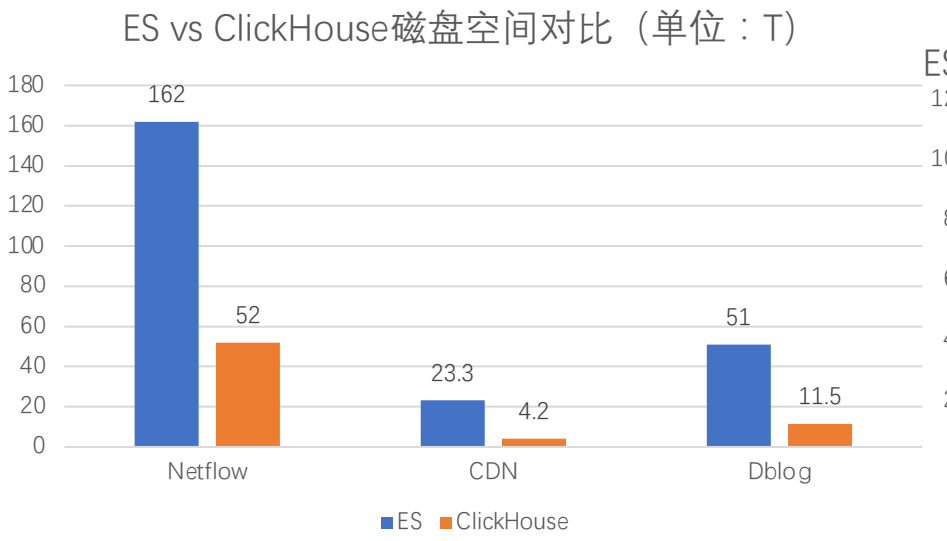


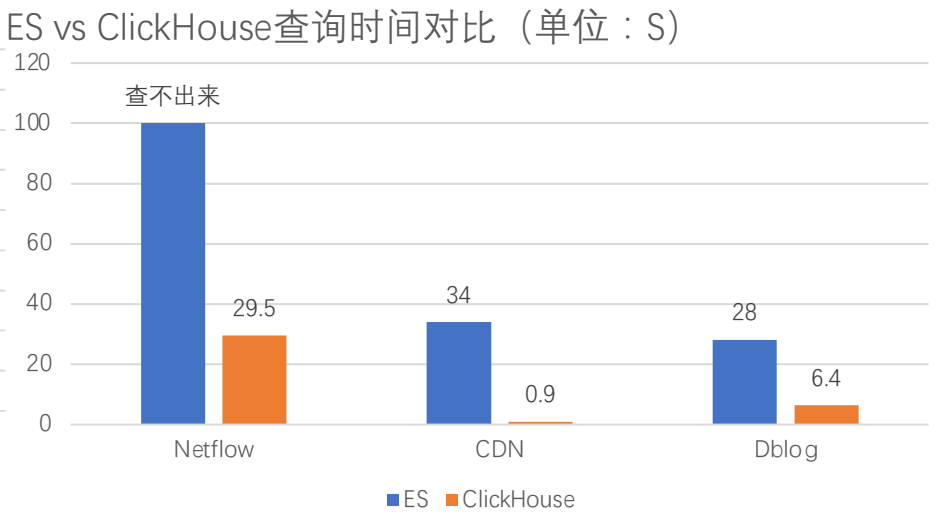
Trip.com


查询问题 - 异常
ExceptionWhileProcessing 错误分类
210, NETWORK_ERROR 2.439 K
32, ATTEMPT_TO_READ_AFTER_EOF 454
160, TOO_SLOW 312
252, TOO_MANY_PARTS 151
62, SYNTAX_ERROR 70
999, KEEPER_EXCEPTION 63
33, CANNOT_READ_ALL_DATA 57
307, TOO_MANY_BYTES 47
ExceptionWhileProcessing 表分布
tableName name cnt
ATTEMPT_TO_READ_AFTER_EOF 56.00
TOO_SLOW 42.00
TOO_SLOW 38.00
NETWORK_ERROR 38.00
ATTEMPT_TO_READ_AFTER_EOF 38.00
ExceptionBeforeStart 错误分类
60, UNKNOWN_TABLE 481
47, UNKNOWN_IDENTIFIER 135
62, SYNTAX_ERROR 73
16, NO_SUCH_COLUMN_IN_TABLE 25
53, TYPE_MISMATCH 20
46, UNKNOWN_FUNCTION 9
43, ILLEGAL_TYPE_OF_ARGUMENT 5
49, LOGICAL_ERROR 5
ExceptionBeforeStart 表分布
tableName name cnt
SYNTAX_ERROR 10.00
UNKNOWN_TABLE 7.00
UNKNOWN_TABLE 6.00
错误分类
210, NETWORK_ERROR 2.492 K
60, UNKNOWN_TABLE 500
查询异常 表统计
tableName name cnt sum(read_rows) sum(read_bytes) sum(query_duration_ms) sum(memory_usage)
NETWORK_ERROR 775.00 20.16 Mil 3.68 Bil 432.82 K 4.11 Bil
CITY
EVENTS OVER TIME
ALL EVENTS
Fields
A
FROM log cbd_clickhouse_log_query_log_all Column:Date event_date Column:DateTime event_time
select 1, code, y.name, cnt from (
select code, count() as cnt from (
select extract(query, 'log\\.\\w+' ) as tableName,
toInt32(extract(exception, 'Code: (\\d+)')) as code
from $table
where $timeFilter and type in ('ExceptionWhileProcessing')
order by read_rows desc)
group by code
) x global join ops.clickhouse_error_code_all y on x.code = y.code
order by cnt desc
Step 60s Resolution 1/1 Round 60s DisableSliceReq FillNullPoint
Format as Time series Show Help Generated SQL Reformat Query
Relative time 1h Time shift 1h Hide time info Database database

# 成果

- 迁移过程自动化程度超过95%
- 基本实现对用户透明
- 存储空间节省50+%，用原有ES的服务器支撑了4倍业务量的增长
- 查询速度比ES快4~30倍，查询P90小于300ms，P99小于1.5s



# 痛点与解决方案

# 两大痛点

## 性能与功能痛点：

单集群规模太大，zookeeper性能瓶颈DDL超时异常

删字段，字段数据量大时，超时，容易导致元数据不一致

用户索引设置不佳，查询慢重建会导致删除历史数据

查询层缺少限流、防呆、自动优化，导致查询不稳定

## 运维痛点：

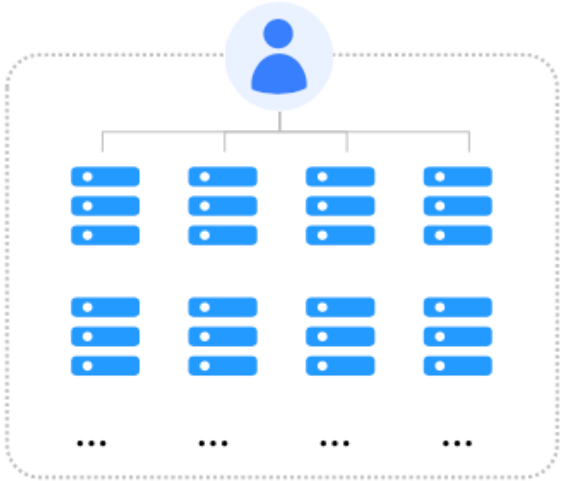
表与集群严格绑定，集群磁盘满后，只能通过双写迁移

集群搭建依赖ansible，部署周期长（数小时）

与社区版本脱节，目前部署模式不便版本更新

# ClickHouse on K8s

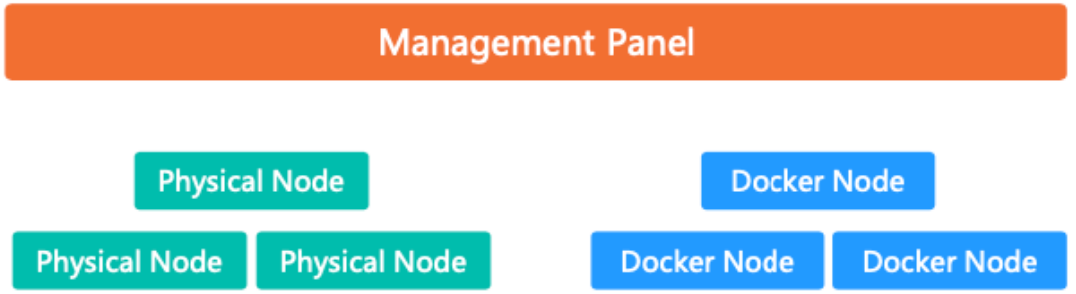
单集群规模太大，zookeeper性能瓶颈DDL超时异常



- 期望收益：
  - 提高资源利用率
  - 降低运维成本，释放人力

（图片来源于网络）

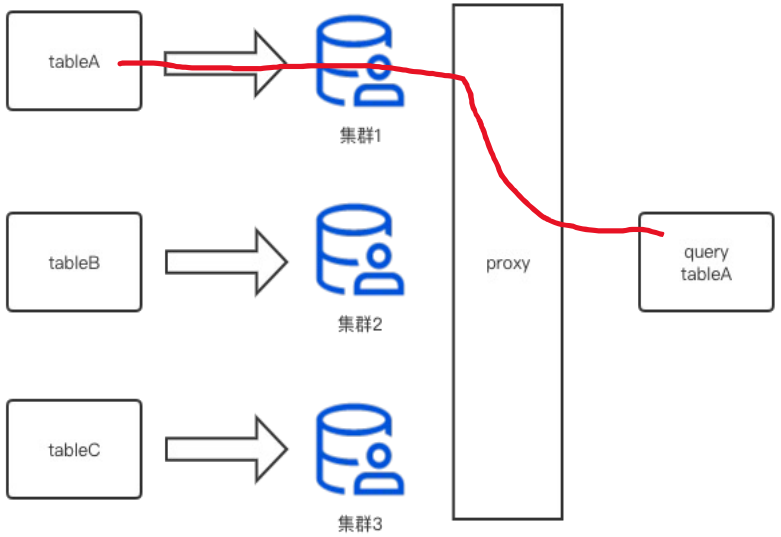
- 思路与方法：
  - sts、cm
  - 亲和性
  - topolvm



# 如何解决数据跨集群

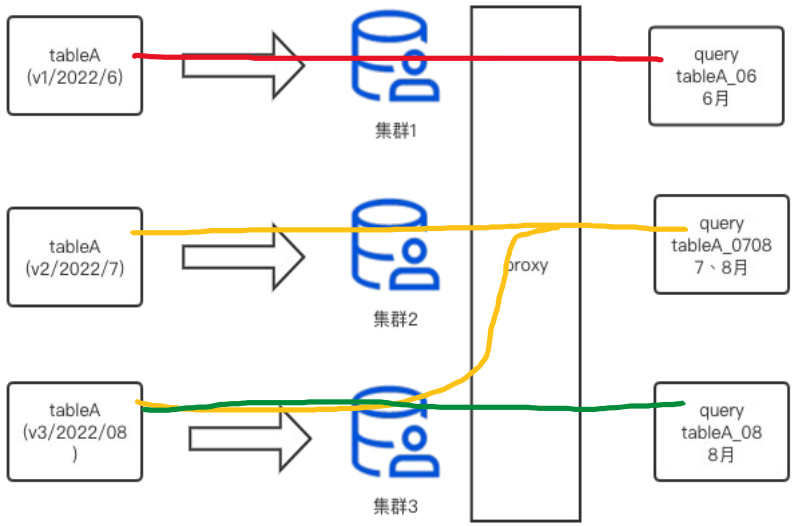
与社区版本脱节，目前部署模式不便版本更新

思考前后，各个组件需要做哪些改造？




改造前

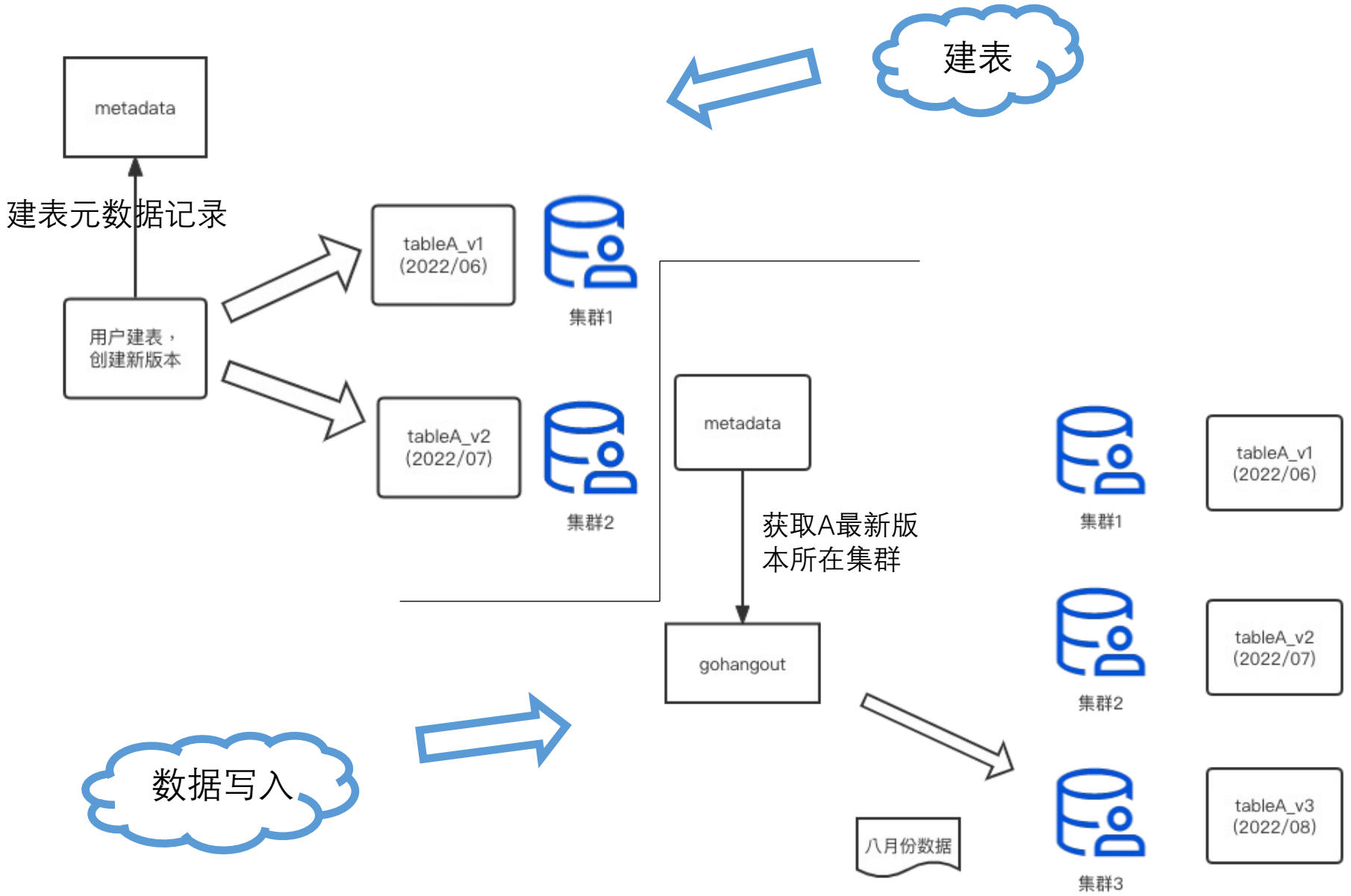
1.统一的元数据管理
2. 写入集群选择
3.查询代理



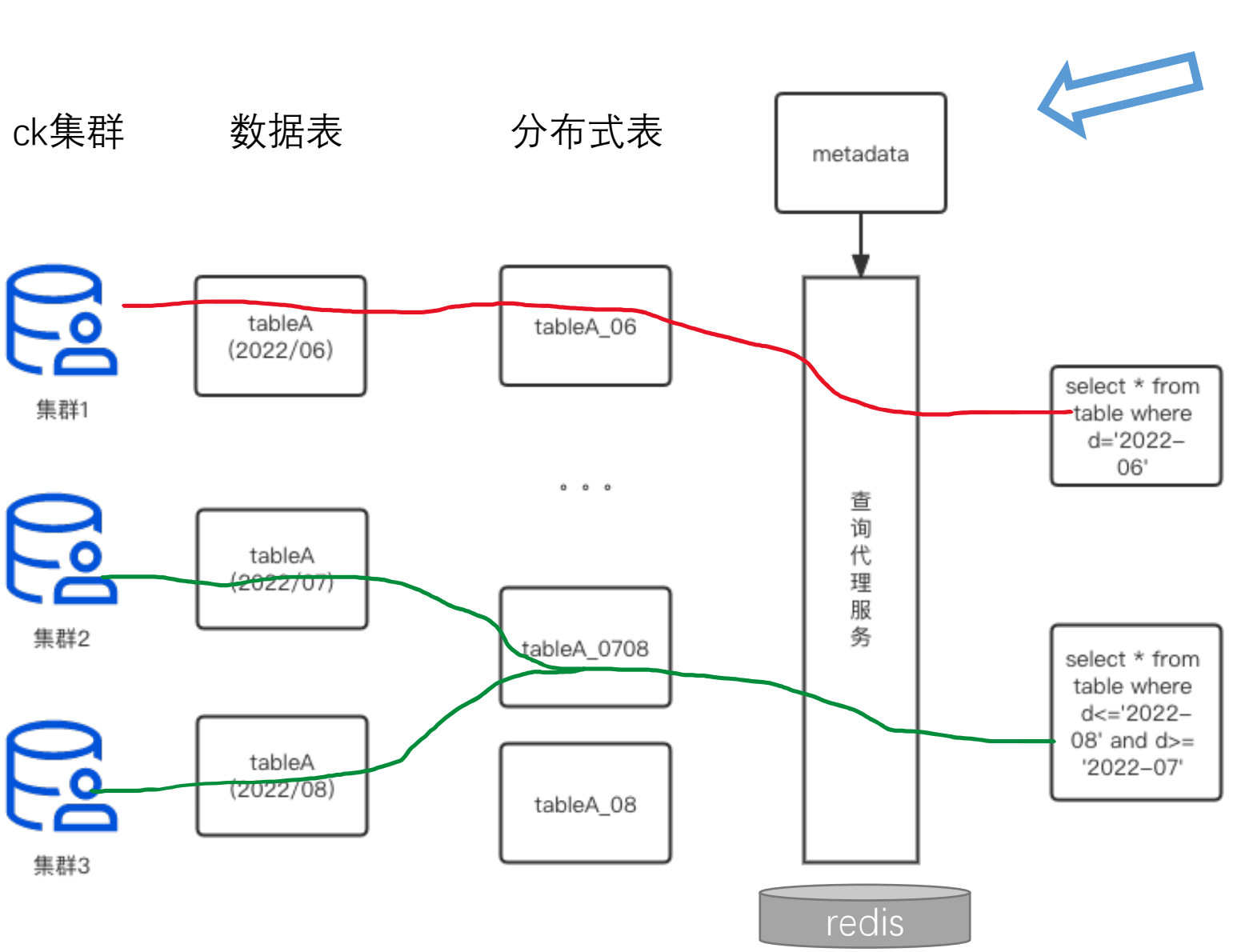

改造后

注意：这里以月份切分，实际上可以精准到分钟

# 如何解决数据跨集群


建表
metadata
建表元数据记录
用户建表，
创建新版本
tableA_v1
(2022/06)
集群1
tableA_v2
(2022/07)
集群2
metadata
获取A最新版
本所在集群
gohangout
数据写入
集群1
tableA_v1
(2022/06)
集群2
tableA_v2
(2022/07)
八月份数据
集群3
tableA_v3
(2022/08)

# 如何解决数据跨集群




- 分布式表选择
- sql优化改写
- sql拒绝/限流
- 数据缓存
- 指标统计
- 故障降级

# 基于antlr4的sql解析

```
CREATE TABLE test_local_1 on
cluster ck100032722 ( `db_id` Int64 COMMENT 'db的id',
`desc` String COMMENT '描述',
`db_location_uri` String COMMENT '路径',
`name` String COMMENT '名称',
`dfs_quotas` String COMMENT 'quotas',
`dfs_usage` String COMMENT '使用量',
`owner_name` String COMMENT 'owner名称',
`owner_type` String COMMENT '类型',
`d` String COMMENT '日期分区键',
`test_string_1` String,
`test_string_2` String,
`test_string_3` String,
`test_1` String,
`test_2` String,
`test_3` String ) ENGINE = ReplicatedMergeTree('/clic
'{replica}')
ORDER BY
d PARTITION BY d SETTINGS index_granularity = 8192
```

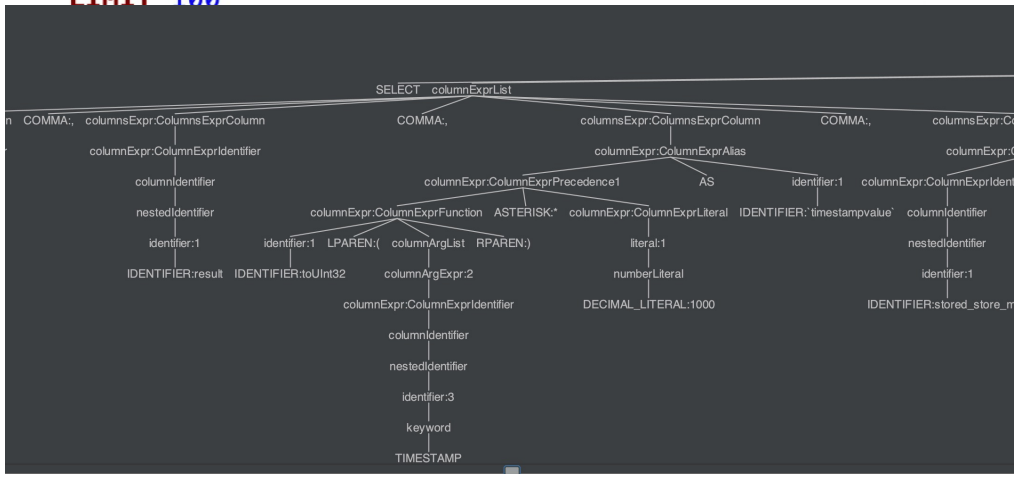

Select * from tableA where d = '2022-06'

...

Select * from tableA_06 where d = '2022-06'

...

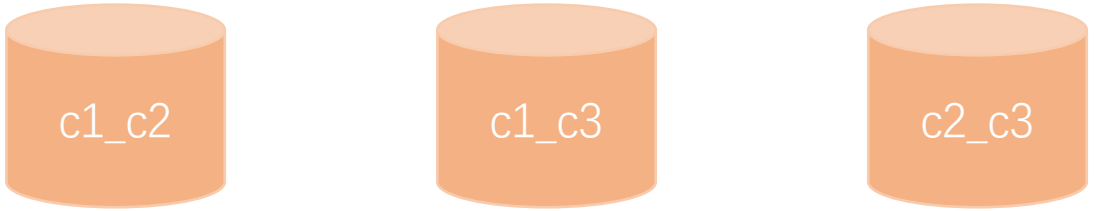
```
SELECT
    orderid,
    result,
    toUInt32(timestamp) * 1000 as `timestampvalue`,
    stored_store_msg as `message`
from
    log.booking_schedule_all
where
    timestamp >= toDateTime('?')
    and timestamp <= toDateTime('?')
    and project in ('transfersvrAsync')
    and orderid = '?'
ORDER BY
    timestamp desc
LIMIT 100
```



# 集群组合

tableA_0607
Distributed( ‘c1_c2’ ,…)

逻辑集群




单位集群


tableA_06
Distributed( ‘c1’ ,…)

tableA_07
Distributed( ‘c2’ ,…)

```
<c1>
    <shard>....</shard>
    ..... c1机器列表
</c1>
<c1_c2>    逻辑集群
    <shard>....</shard>
    ..... c1机器列表
    <shard>....</shard>
    ..... c2机器列表
</c1_c2>
```

# 未来展望

# 未来展望

- 降低成本
  - 冷数据存储
  - Juicefs、对象存储
  - 超冷机型
- 弹性混合云
- 日志数据预聚合

Trip.com

# Thanks