Trip.com Group™ 携程集团

携程ClickHouse集群数据管理工具 实战

林东煜

2025.02.29

目录

CONTENTS

1 背景与问题

2 设计与调研

3 目标与成果

个人简介

林东煜-携程-系统研发部

- ClickHouse 日志
- OLAP Paas平台

背景与问题

缺乏弹性

服务器：1000+

数据量：120w亿行

数据表：1w+

存储空间：70+PB

查询：10M+/天

集群数量：40+

P90：500ms

P99：3s

•背景：
•日志Clickhouse集群 20.3版本
•Alma系统服务器升级
•上云/下云/迁云
•测试环境机房搬迁

•痛点：
- 原地升级无法灰度，用户有感
- 集群无法缩容
- 集群数据不均衡
- 机器空间常年处于80%以上，压缩程度不够
- 旧版本人力运维成本高**Metadata on replica is not up to date with common metadata in Zookeeper**

为什么要升级

•20.3 VS 24.3

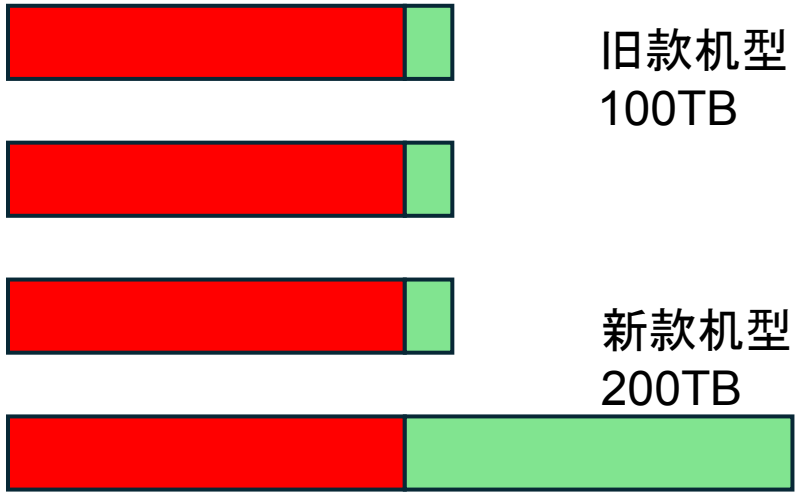
- Atomic Engine
- Projection
- JSON type
- lightweight delete
- Variant type
- New analyzer
- EXPLAIN
- NEW FUNCTIONS
- NEW COMANDS FOR OPS
- S3 ZERO COPY
- async_insert
- 元数据不一致https://github.com/ClickHouse/ClickHouse/issues/54902

为什么要数据均衡

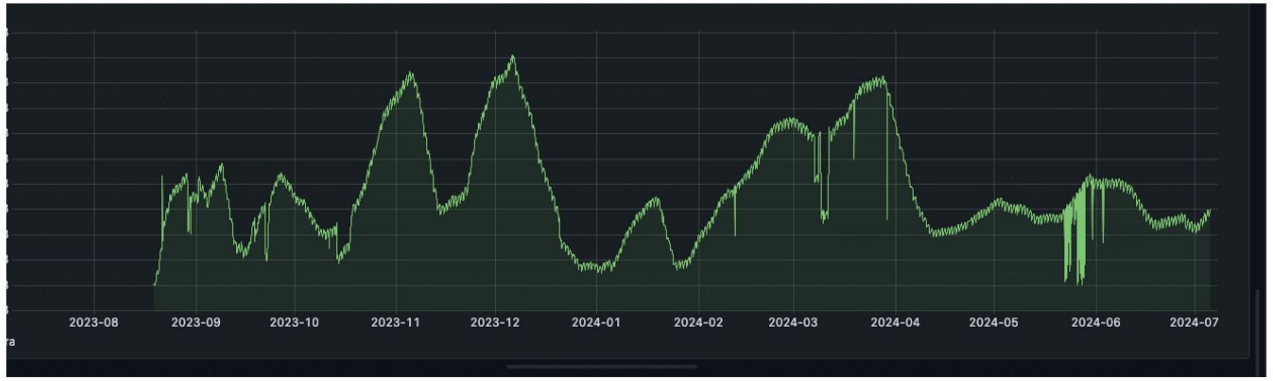
空间与负载

Case1：
新扩机器数据不均衡
（空间,cpu,磁盘io）

Case2：
机型不统一数据不均衡
（空间,cpu,磁盘io）



为什么要缩容



30%以上的
波动

设计与调研

方案调研：

- Copier
- insert into …
  Select
- File copy
- Fetch

设计与调研

Trip.com

File

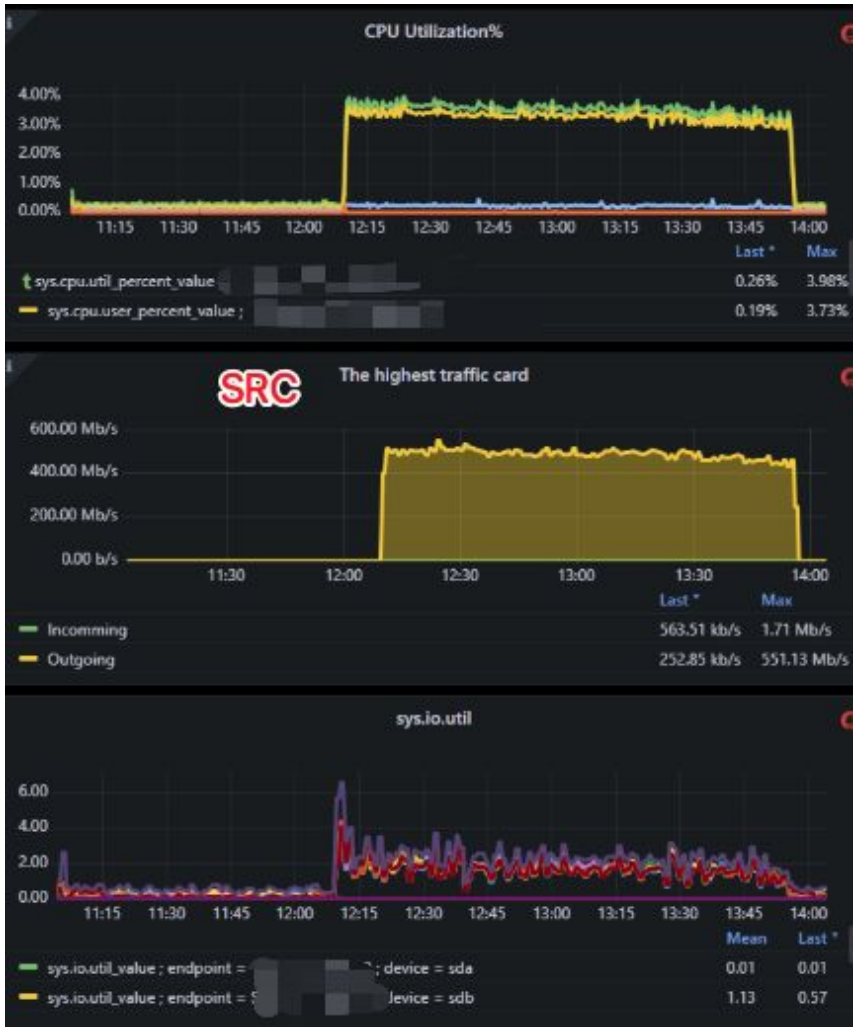
- JuiceFS
- 命令下发
- 文件离散、打包
- 并发传输
- 任务管控

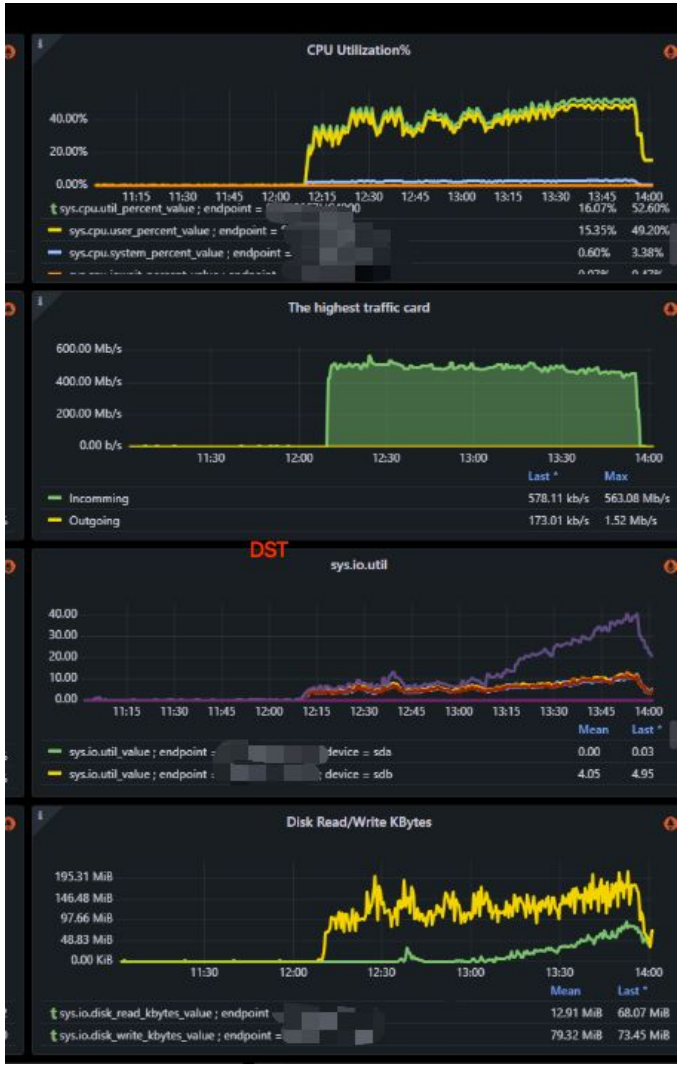
设计与调研

insert into … Select

- virtual-columns(_part/_partition_id)
- insert_quorum/optimize_on_insert
- min_insert_block_size_rows/min_insert_block_size_bytes
- max_execution_time
- max_bytes_to_read
- max_threads/max_threads_insert
- min_compress_block_size/max_compress_block_size
- Queryid、part_log
- system FLUSH LOGS 刷system.query_log system.part_log
- insert 操作和 DDL 是否冲突
- 性能测试

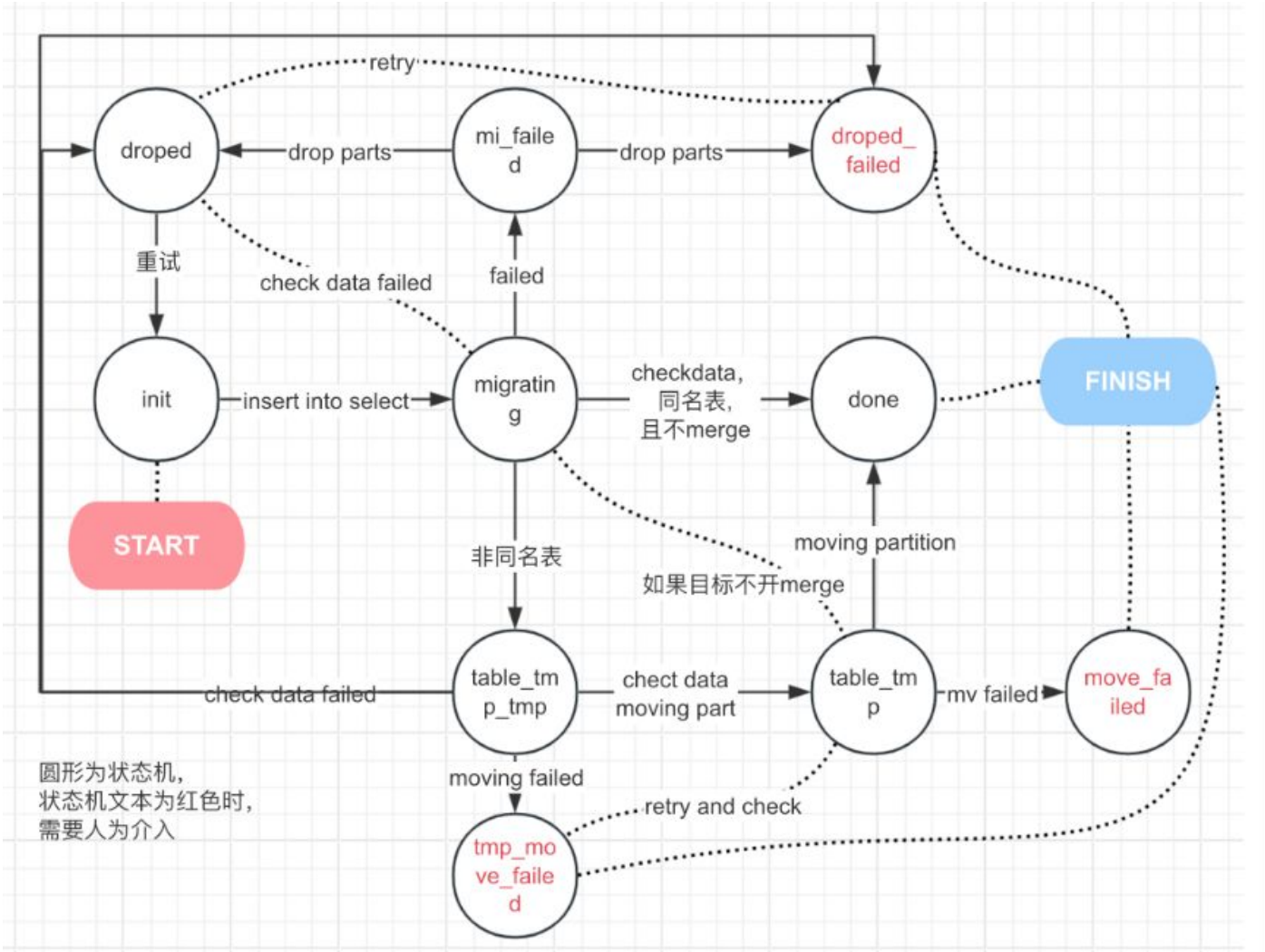
设计与调研


CPU Utilization%
4.00%
3.00%
2.00%
1.00%
0.00%
11:15 11:30 11:45 12:00 12:15 12:30 12:45 13:00 13:15 13:30 13:45 14:00
Last * Max
sys.cpu.util_percent_value 0.26% 3.98%
sys.cpu.user_percent_value ; 0.19% 3.73%
SRC
The highest traffic card
600.00 Mb/s
400.00 Mb/s
200.00 Mb/s
0.00 b/s
11:30 12:00 12:30 13:00 13:30 14:00
Last * Max
Incomming 563.51 kb/s 1.71 Mb/s
Outgoing 252.85 kb/s 551.13 Mb/s
sys.io.util
6.00
4.00
2.00
0.00
11:15 11:30 11:45 12:00 12:15 12:30 12:45 13:00 13:15 13:30 13:45 14:00
Mean Last *
sys.io.util_value ; endpoint = ; device = sda 0.01 0.01
sys.io.util_value ; endpoint = device = sdb 1.13 0.57


CPU Utilization%
40.00%
20.00%
0.00%
11:15 11:30 11:45 12:00 12:15 12:30 12:45 13:00 13:15 13:30 13:45 14:00
sys.cpu.util_percent_value ; endpoint = 16.07% 52.60%
sys.cpu.user_percent_value ; endpoint = 15.35% 49.20%
sys.cpu.system_percent_value ; endpoint = 0.60% 3.38%
The highest traffic card
600.00 Mb/s
400.00 Mb/s
200.00 Mb/s
0.00 b/s
11:30 12:00 12:30 13:00 13:30 14:00
Last * Max
Incomming 578.11 kb/s 563.08 Mb/s
Outgoing 173.01 kb/s 1.52 Mb/s
DST
sys.io.util
40.00
30.00
20.00
10.00
0.00
11:15 11:30 11:45 12:00 12:15 12:30 12:45 13:00 13:15 13:30 13:45 14:00
Mean Last *
sys.io.util_value ; endpoint = device = sda 0.00 0.03
sys.io.util_value ; endpoint = device = sdb 4.05 4.95
Disk Read/Write KBytes
195.31 MiB
146.48 MiB
97.66 MiB
48.83 MiB
0.00 KiB
11:30 12:00 12:30 13:00 13:30 14:00
Mean Last *
sys.io.disk_read_kbytes_value ; endpoint 12.91 MiB 68.07 MiB
sys.io.disk_write_kbytes_value ; endpoint = 79.32 MiB 73.45 MiB

设计与调研

状态机

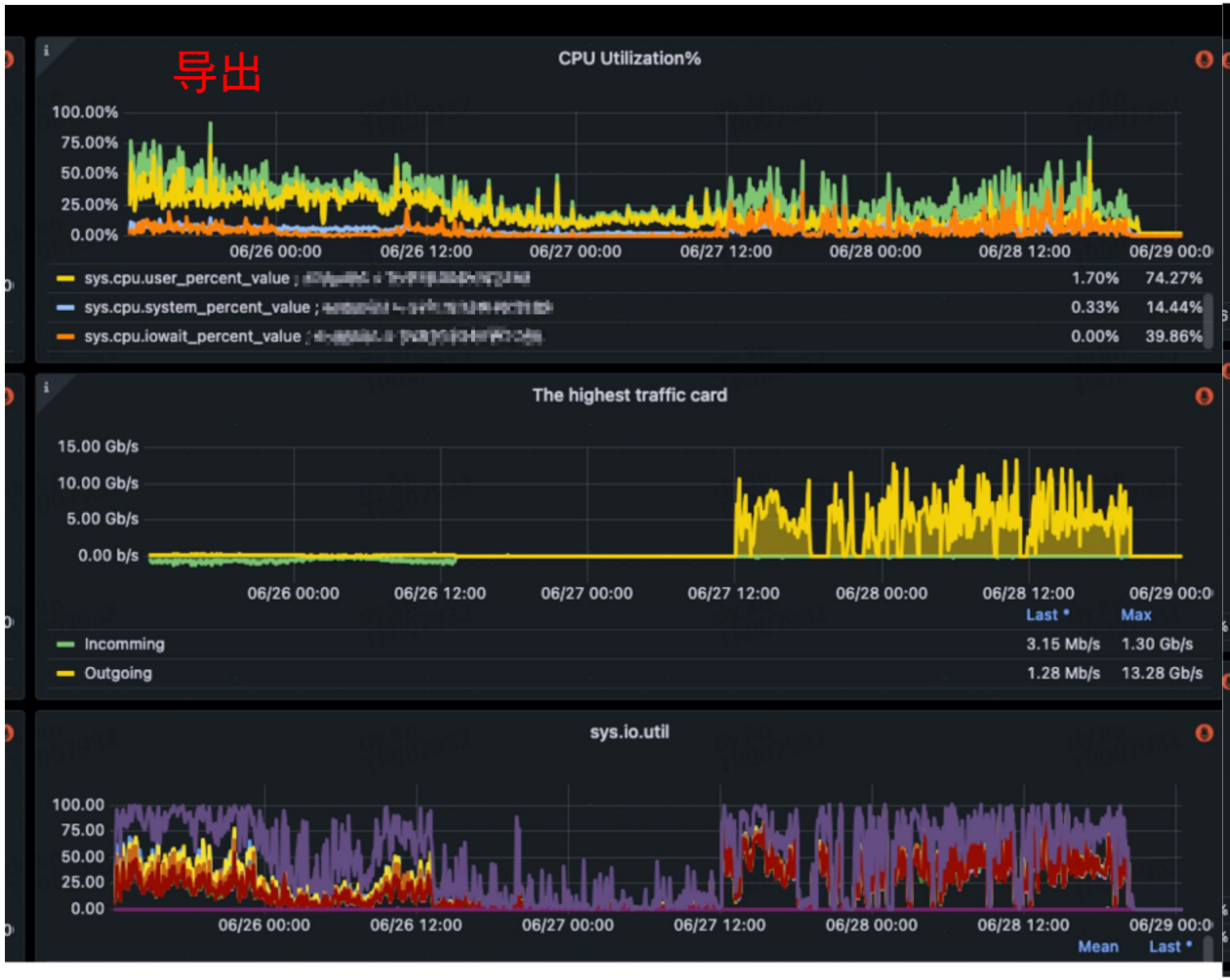

retry
droped
drop parts
mi_faile
d
drop parts
droped_
failed
重试
check data failed
failed
init
insert into select
migratin
g
checkdata,
同名表,
且不merge
done
FINISH
START
非同名表
moving partition
如果目标不开merge
table_tm
p_tmp
check data failed
chect data
moving part
table_tm
p
mv failed
move_fa
iled
圆形为状态机,
状态机文本为红色时,
需要人为介入
moving failed
retry and check
tmp_mo
ve_faile
d

设计与调研

Fetch

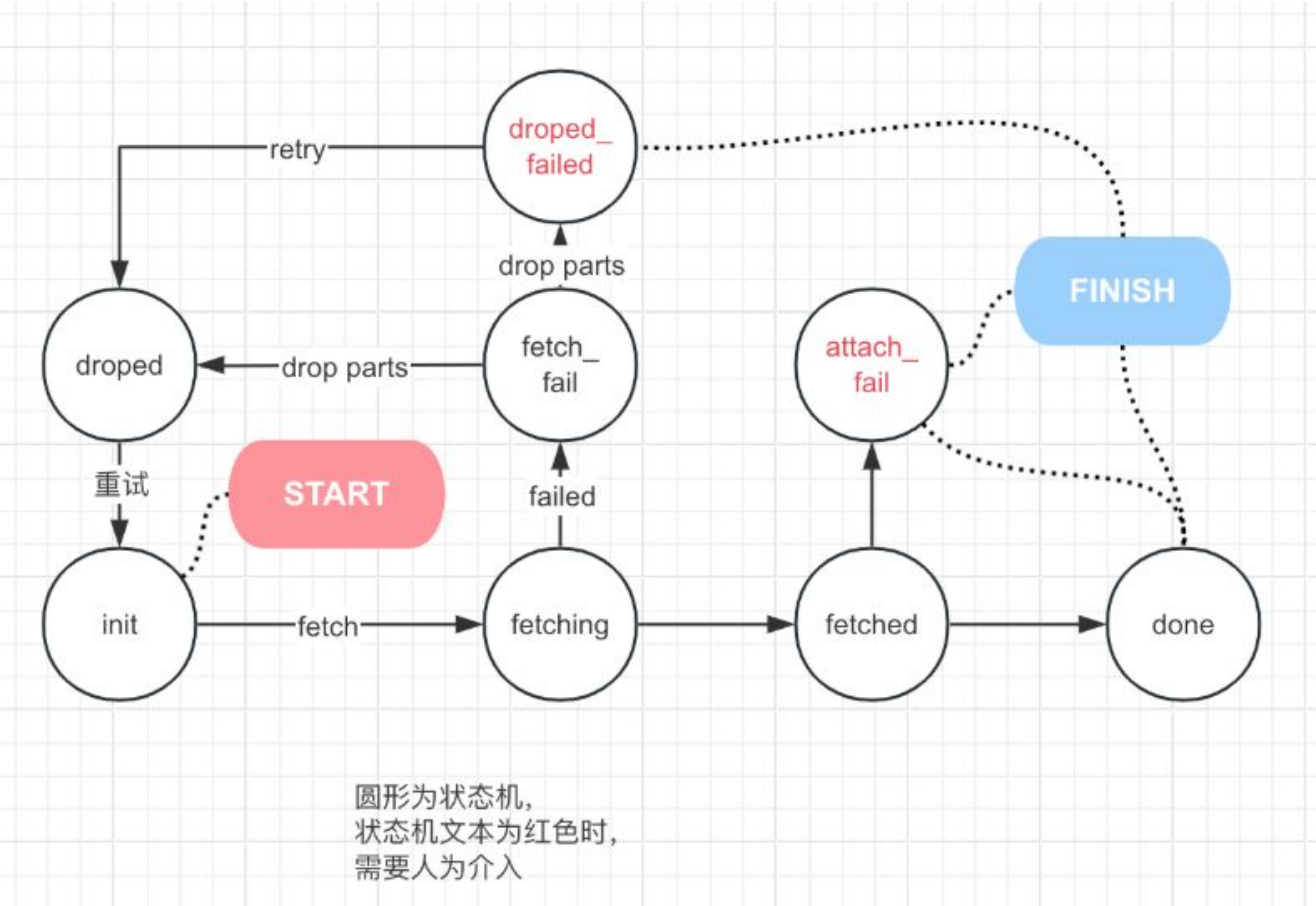
- FETCH FROM zkpath
  - 不同字段表是否能够fetch
- 压缩格式不同是否能够attach（批量改压缩）
- 不同版本数据兼容
- 字段不同能否attach成功（批量加字段）
- fetch 过程中对dst 表加字段后，是否能正常attach（DDL）
- Attaching_part bug
- Inmemory part无法fetch
- Stop merge / ttl
- 动态auxiliary_zookeepers
- ALTER TABLE log.xx FETCH partition '20240212' FROM ‘test_zookeeper:/clickhouse/tables/cluster1-shard1/tabletest'

成果：集群升级


导出
CPU Utilization%
100.00%
75.00%
50.00%
25.00%
0.00%
06/26 00:00
06/26 12:00
06/27 00:00
06/27 12:00
06/28 00:00
06/28 12:00
06/29 00:0
sys.cpu.user_percent_value ;
1.70%
74.27%
sys.cpu.system_percent_value ;
0.33%
14.44%
sys.cpu.iowait_percent_value ;
0.00%
39.86%
The highest traffic card
15.00 Gb/s
10.00 Gb/s
5.00 Gb/s
0.00 b/s
06/26 00:00
06/26 12:00
06/27 00:00
06/27 12:00
06/28 00:00
06/28 12:00
06/29 00:0
Last *
Max
Incomming
3.15 Mb/s
1.30 Gb/s
Outgoing
1.28 Mb/s
13.28 Gb/s
sys.io.util
100.00
75.00
50.00
25.00
0.00
06/26 00:00
06/26 12:00
06/27 00:00
06/27 12:00
06/28 00:00
06/28 12:00
06/29 00:0
Mean
Last *


导入
CPU Utilization%
100.00%
75.00%
50.00%
25.00%
0.00%
22:30
23:00
23:30
00:00
00:30
01:00
01:30
02:00
02:30
03:00
03:30
04:00
04:30
05:00
05:30
Last *
Max
sys.cpu.util_percent_value ; endpoint =
0.00%
0.00%
sys.cpu.user_percent_value ; endpoint =
0.00%
0.00%
The highest traffic card
0.00 b/s
-5.00 Gb/s
-10.00 Gb/s
-15.00 Gb/s
2024-05-29 05:28:30
Incomming
13.23 Gb/s
Outgoing
4.36 Mb/s
Incomming
Outgoing
4.03 Mb/s
5.13 Mb/s
sys.io.util
100.00
75.00
50.00
25.00
0.00
Mean
Last *
sys.io.util_value ; endpoint =
31.36
38.64
sys.io.util_value ; endpoint =
31.76
39.25

设计与调研


droped_
failed
retry
drop parts
FINISH
droped
drop parts
fetch_
fail
attach_
fail
重试
START
failed
init
fetch
fetching
fetched
done
圆形为状态机，
状态机文本为红色时，
需要人为介入

设计与调研

查询兼容测试：

- allow_experimental_analyzer
- 特殊语法
- SQL扫描

1、multilf、case when语句包含 in的，如下：

```
multilf(
  pageId in ('220236','401138'), 'h5',
  pageId in ('103816','103817'), 'pc',
  '未知')
```

修改后，如下：

```
case when (pageId = '220236' or pageId = '401138') then 'h5'
when (pageId = '103816' or pageId = '103817') then 'pc'
else '未知'
end
```

2、if语句包含 in的，如下：

```
count(DISTINCT if(page IN ('10650039006', '10650045146',
'10650034951'), vid, NULL)) / count(DISTINCT if(page IN
('10650040697', '10650039004', '109908'), vid, NULL)) AS countpage
```

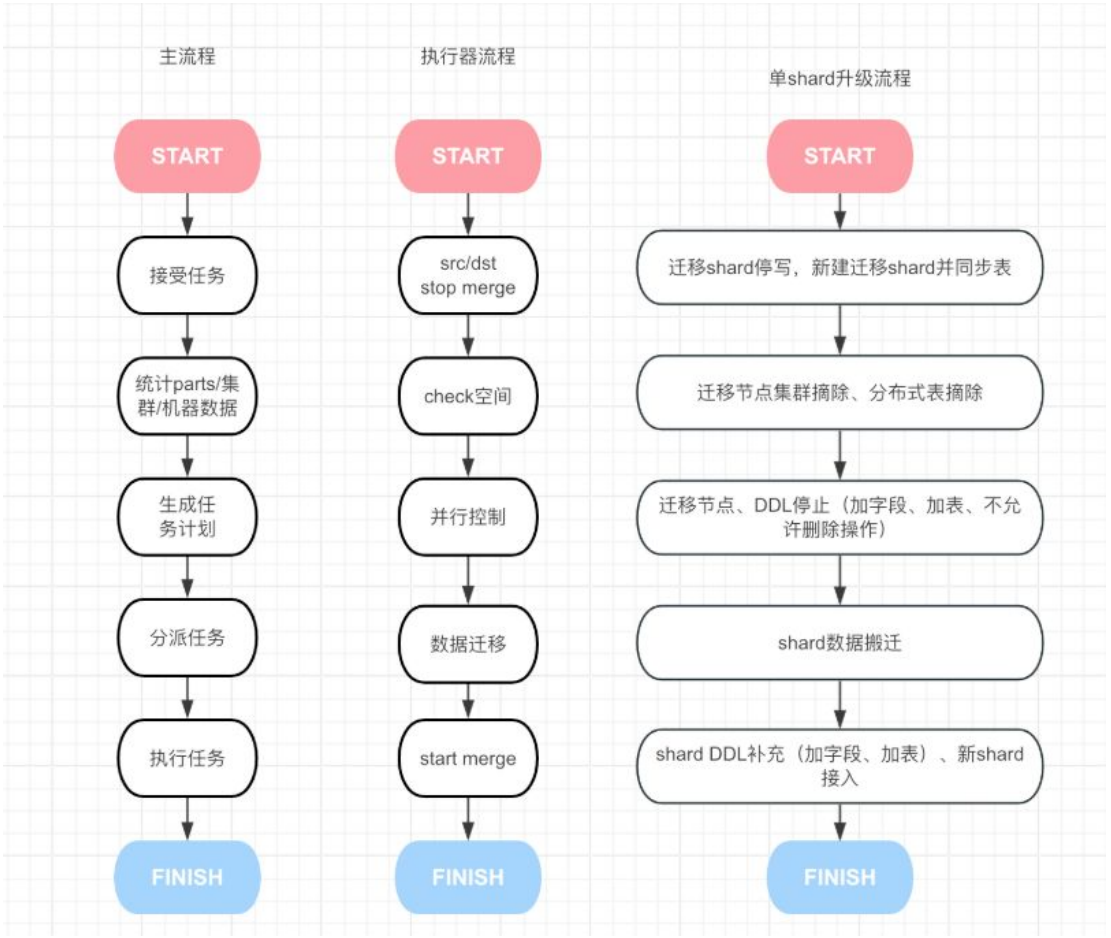
修改后，如下：

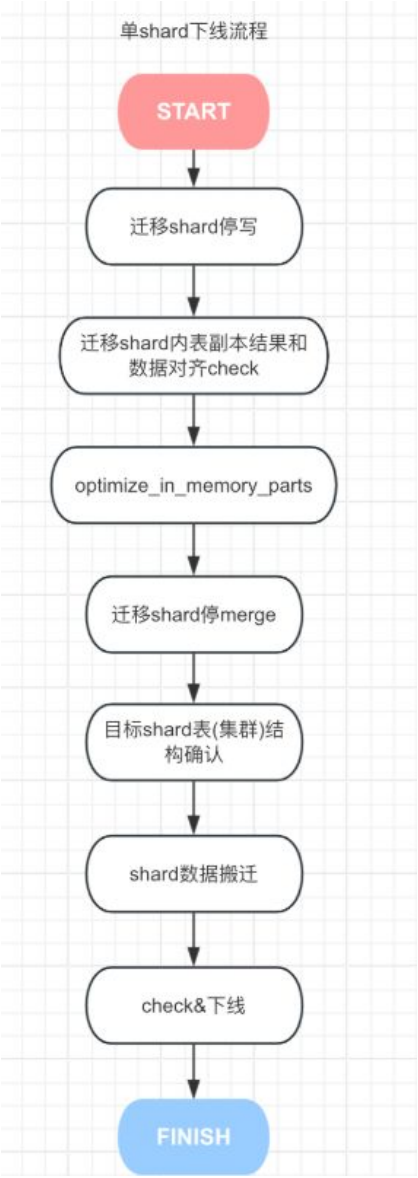
```
count(DISTINCT case when (page ='10650039006' or page =
'10650045146') then vid
 when  page ='10650034951' then vid
else NULL end) / count(DISTINCT case when (page
='10650040697' or page = '10650039004' ) then  vid
when  page ='109908' then vid
else NULL end) AS countpage
```

注意：case when 中 or 、and不能超过2个，超过的用when再分开

设计与调研

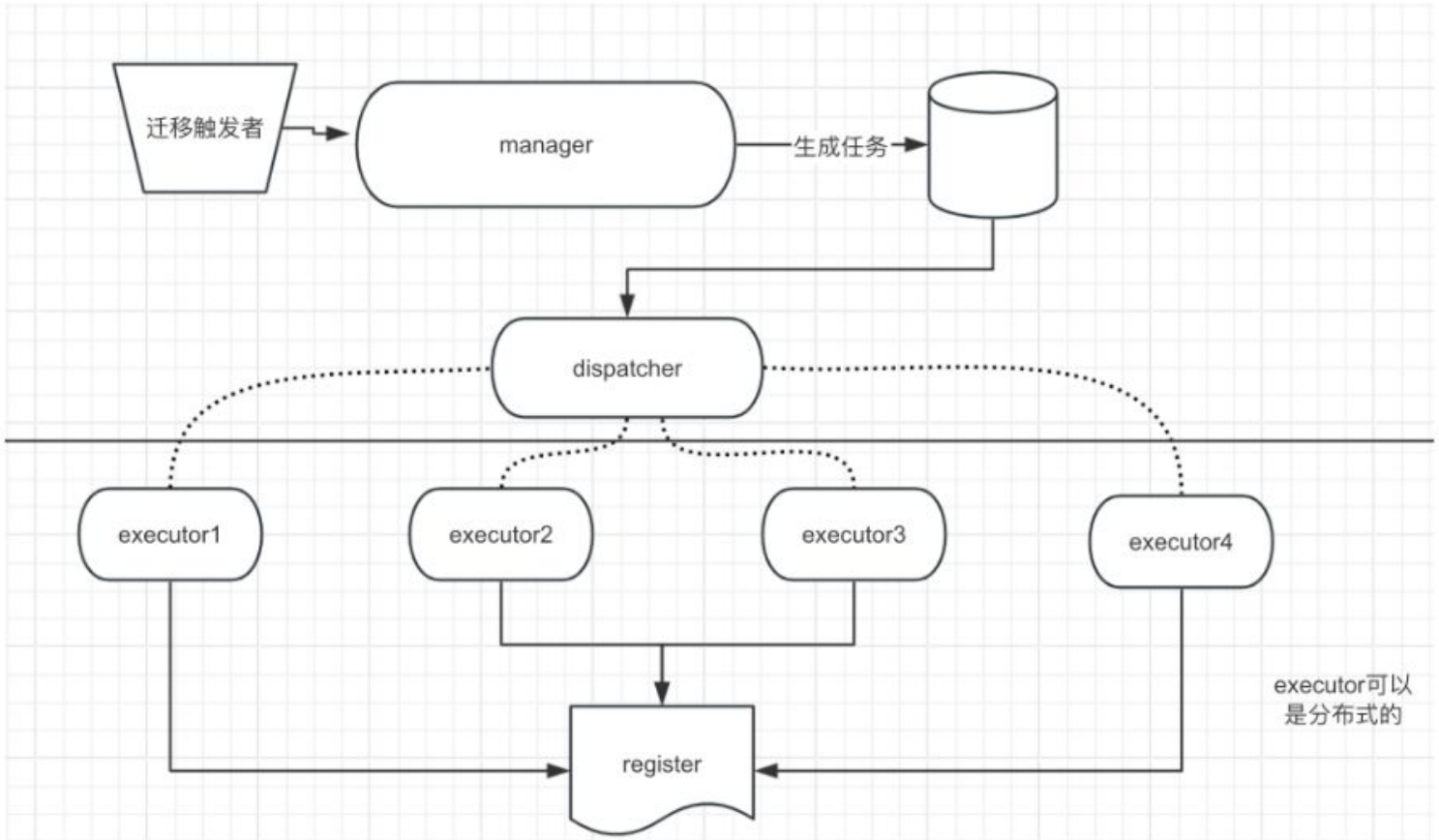
流程设计


主流程
START
接受任务
统计parts/集群/机器数据
生成任务计划
分派任务
执行任务
FINISH
执行器流程
START
src/dst stop merge
check空间
并行控制
数据迁移
start merge
FINISH
单shard升级流程
START
迁移shard停写，新建迁移shard并同步表
迁移节点集群摘除、分布式表摘除
迁移节点、DDL停止（加字段、加表、不允许删除操作）
shard数据搬迁
shard DDL补充（加字段、加表）、新shard接入
FINISH


单shard下线流程
START
迁移shard停写
迁移shard内表副本结果和数据对齐check
optimize_in_memory_parts
迁移shard停merge
目标shard表(集群)结构确认
shard数据搬迁
check&下线
FINISH

设计与调研

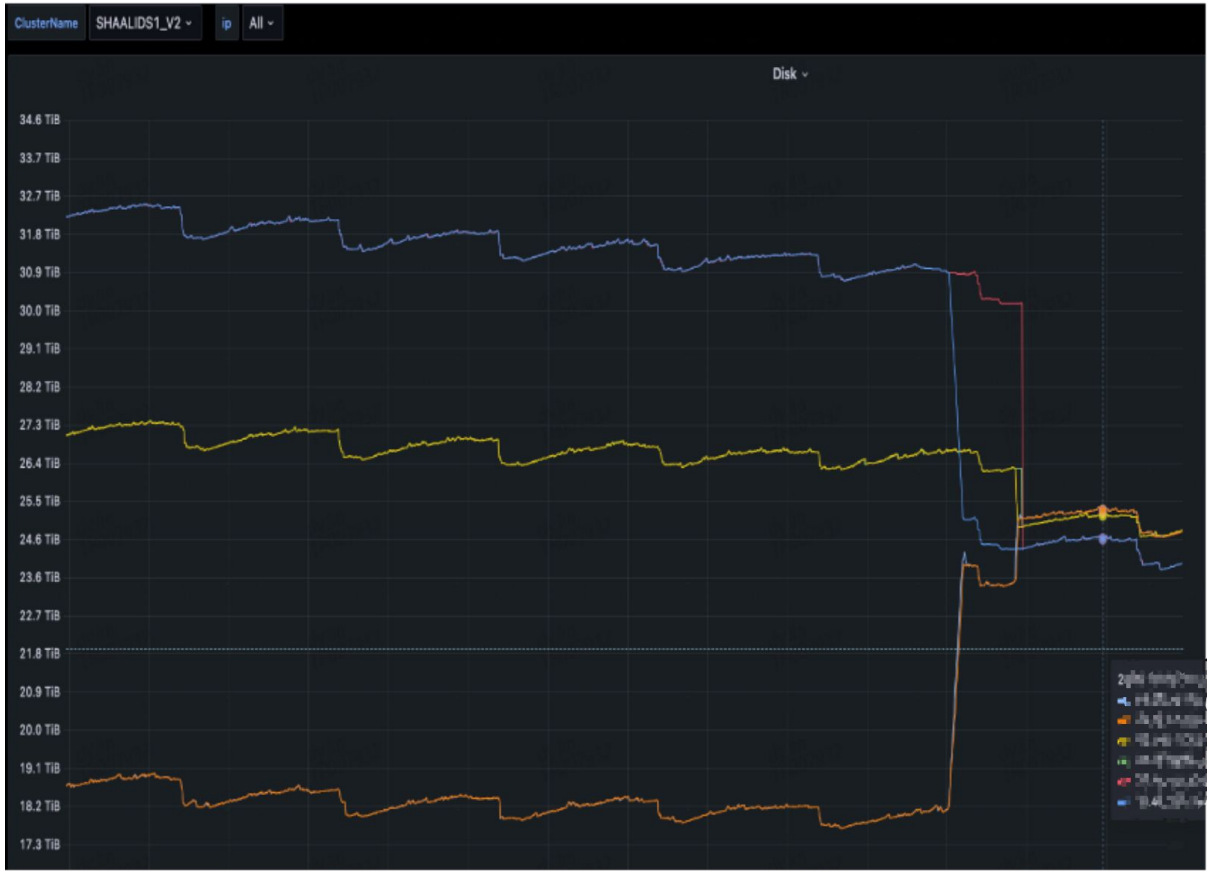
架构设计



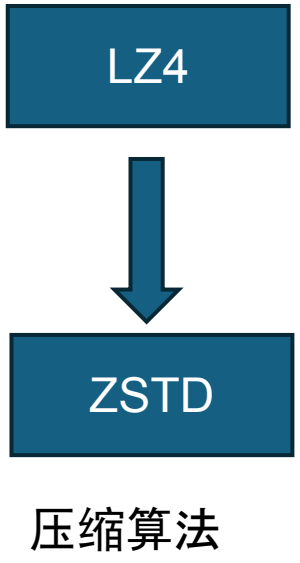
- 执行器与shard级别的任务绑定

成果:集群均衡与缩容


Disk
70%
68%
66%
64%
62%
60%
58%
56%
54%
52%
50%
10/08 00:00
10/09 00:00
10/10 00:00
10/11 00:00
10/12 00:00
10/13 00:00
10/14 00:00


ClusterName
SHAALIDS1_V2
ip
All
Disk
34.6 TiB
33.7 TiB
32.7 TiB
31.8 TiB
30.9 TiB
30.0 TiB
29.1 TiB
28.2 TiB
27.3 TiB
26.4 TiB
25.5 TiB
24.6 TiB
23.6 TiB
22.7 TiB
21.8 TiB
20.9 TiB
20.0 TiB
19.1 TiB
18.2 TiB
17.3 TiB

成果：集群升级



压缩算法

迁移：

涉及70+PB数据搬迁，每日搬迁上PB数据，过程用户无感

空间节省：

压缩比从4.3提升20%到5.56

稳定性：运维报障数（7-9月 TS技术支持统计）

91->67-> 35 (次)