阿里云 | 奥运会全球指定云服务商

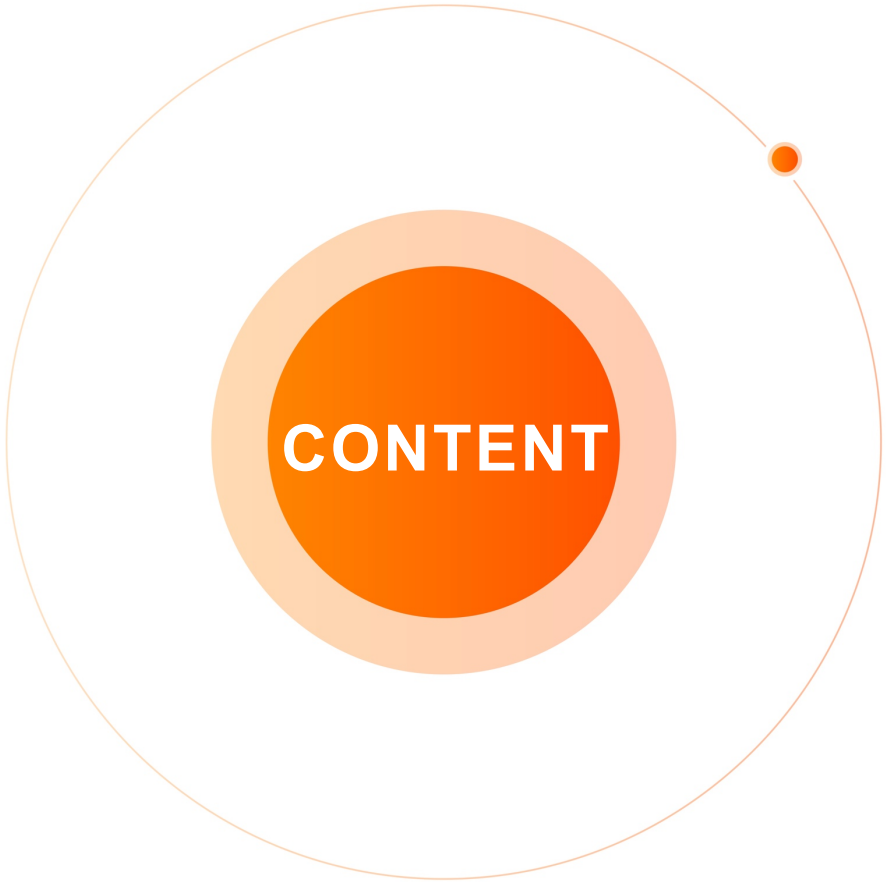
精准抓住你的每一个潜在用户

ClickHouse在实时广告圈人业务中的最佳实践

阿里云 – 数据库 涂继业（和君）

个人简介

- 阿里云 云数据库ClickHouse 技术负责人
- 阿里云 自研数据库 AnalyticDB 核心研发人员
- 多年Java/C++、OLAP数据库、大数据领域研发经验
- 实际参与大规模、分布式、高性能OLAP引擎多个核心模块设计与研发



人群圈选业务挑战

大宽表

1000+标签，Array类型

任意维度检索

Ad-hoc，多维度交并差

大吞吐导入

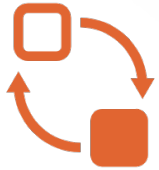
T+1离线算标签，数百MB/s

查询并发高

数百并发，圈选万/十万/百万人群

实时更新

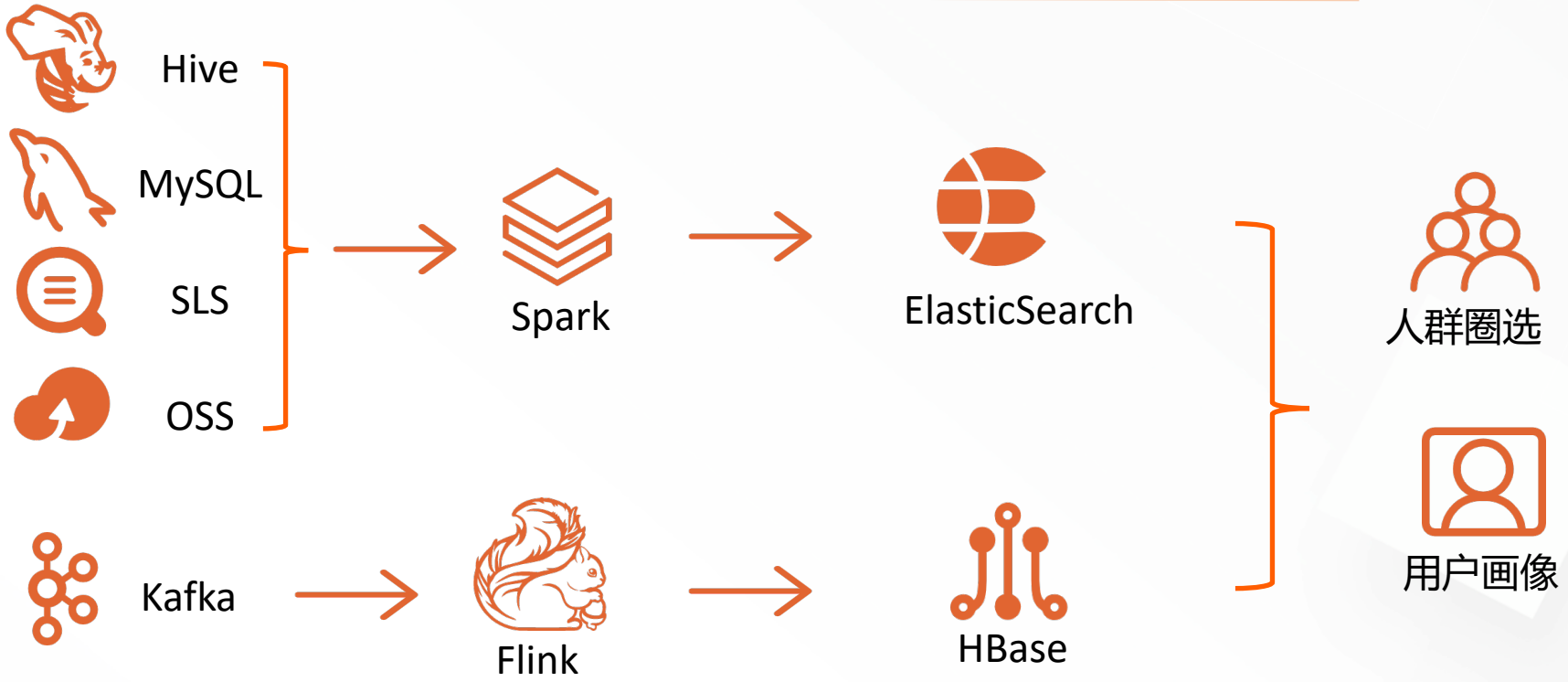
指标实时计算/更新



Schema灵活变更

动态增删列，改变列类型

业界常见解决方案



方案缺陷：

- 数据一致性：数据导入2个独立系统，冗余2份；
- ElasticSearch优势场景是搜索，而非分析；
- ElasticSearch写入速度慢，大吞吐写入容易OOM；
- ElasticSearch存储成本高：行存压缩率低、_source原文、双副本；
- ElasticSearch稠密索引：构建速度慢，全列索引空间膨胀严重；
- ElasticSearch更改schema复杂，涉及到index结构变化；
- ElasticSearch查询语法复杂：DSL v.s. SQL
- 资源消耗严重，需要高规格CPU/MEM，硬件成本高昂；

ClickHouse解决方案 – ReplacingMergeTree


```
CREATE TABLE IF NOT EXISTS user_tbl (
    user_id UInt64,
    reg_date Datetime,
    city_level String,
    gender String,
    interest_sports String,
    comment_like_cnt UInt32,
    last30d_share_cnt UInt32,
    user_like_consume_trend_type String,
    province String,
    last_access_version String,
    others Array(String)
) ENGINE = ReplacingMergeTree()
ORDER BY user_id;
```

优点：

- 简单直观
- 写入高吞吐
- 查询速度快
- 适合离线一次性导入

缺点：

- UPDATE操作限制多
  - 不宜单行进行
  - 个数总数受限
  - 合并效率低
- 数据一致性：异步生效

ClickHouse最佳实践 – AggregatingMergeTree

存储引擎

MergeTreeFamily
- MergeTree
- ReplacingMergeTree
- AggregatingMergeTree
- SummingMergeTree
- CollapsingMergeTree
- VersionedCollapsingMergeTree
- GraphiteMergeTree

Specials
- Distributed
- Dictionary
- Merge
- Join
- Set
- File
- URL

数据类型

```
CREATE TABLE t
(
    id UInt32,
    column1 SimpleAggregateFunction(sum, UInt64),
    column2 SimpleAggregateFunction(any, String)
) ENGINE = ...
ORDER BY id;
```

any　anyLast　min　max

sum　argMin　argMax　groupBitAnd

ClickHouse最佳实践 – AggregatingMergeTree


```
CREATE TABLE IF NOT EXISTS whatever_table ON CLUSTER default (
    user_id UInt64,
    reg_date SimpleAggregateFunction(anyLast, Datetime),
    city_level SimpleAggregateFunction(anyLast,Nullable(String)),
    gender SimpleAggregateFunction(anyLast, Nullable(String)),
    interest_sports SimpleAggregateFunction(anyLast, Nullable(String)),
    comment_like_cnt SimpleAggregateFunction(anyLast, Nullable(UInt32)),
    last30d_share_cnt SimpleAggregateFunction(anyLast, Nullable(UInt32)),
    user_like_consume_trend_type SimpleAggregateFunction(anyLast, Nullable(String)),
    province SimpleAggregateFunction(anyLast, Nullable(String)),
    last_access_version SimpleAggregateFunction(anyLast, Nullable(String)),
    others SimpleAggregateFunction(anyLast, Nullable(Array(String)))
) ENGINE = AggregatingMergeTree()
ORDER BY user_id;
```

ClickHouse最佳实践 – Insert替换Update

阿里云 | 奥运会全球指定云服务商

ReplacingMergeTree

```
ALTER TABLE whatever_table
UPDATE last30d_share_cnt = 10, comment_like_cnt = 20
WHERE user_id = 1;
```

```
INSERT INTO whatever_table VALUES(
    1,
    '2021-01-28 10:10:10',
    '一线城市',
    '男',
    '否',
    20,
    10,
    '高',
    '河南省',
    '13',
    ['a', 'b', 'c']
);
```

AggregatingMergeTree

```
INSERT INTO whatever_table
(user_id, last30d_share_cnt, comment_like_cnt)
VALUES (1, 10, 20);
```

- 提升系统稳定性
  - 减少小文件数目，无mutation个数限制
  - 合并速度更快
- 提升写入性能
  - Insert中只需要包含被更新的列，其他列无需包含
  - 不需要查询原始数据

ClickHouse最佳实践 – 数据一致性

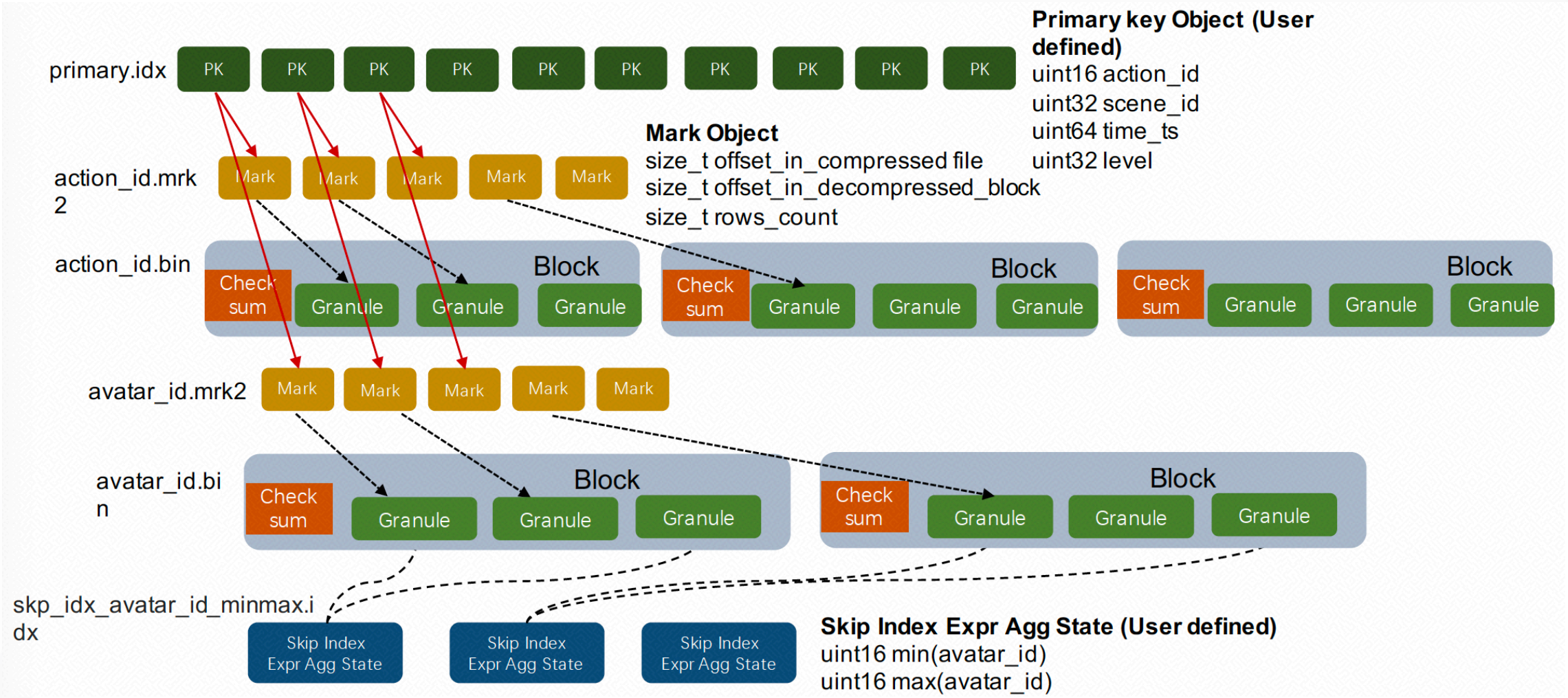
阿里云 | 奥运会全球指定云服务商

```
CREATE TABLE IF NOT EXISTS whatever_table ON CLUSTER default (
    user_id UInt64,
    reg_date SimpleAggregateFunction(anyLast, Datetime),
    city_level SimpleAggregateFunction(anyLast,Nullable(String)),
    gender SimpleAggregateFunction(anyLast, Nullable(String)),
    interest_sports SimpleAggregateFunction(anyLast, Nullable(String)),
    comment_like_cnt SimpleAggregateFunction(anyLast, Nullable(UInt32)),
    last30d_share_cnt SimpleAggregateFunction(anyLast, Nullable(UInt32)),
    user_like_consume_trend_type SimpleAggregateFunction(anyLast, Nullable(String)),
    province SimpleAggregateFunction(anyLast, Nullable(String)),
    last_access_version SimpleAggregateFunction(anyLast, Nullable(String)),
    others SimpleAggregateFunction(anyLast, Nullable(Array(String)))
) ENGINE = AggregatingMergeTree()
PARTITION BY (user_id % 8)
ORDER BY user_id;
```

- 更新立即可见
  - count(distinct user_id)
- 更新延迟可见
  - ➢ 周期性OPTIMIZE tbl FINAL
  - ➢ 查询OPTIMIZE前的数据

    where gmt_time < ‘’
- 分区级并发optimize
  - OPTIMIZE tble PARTITION par

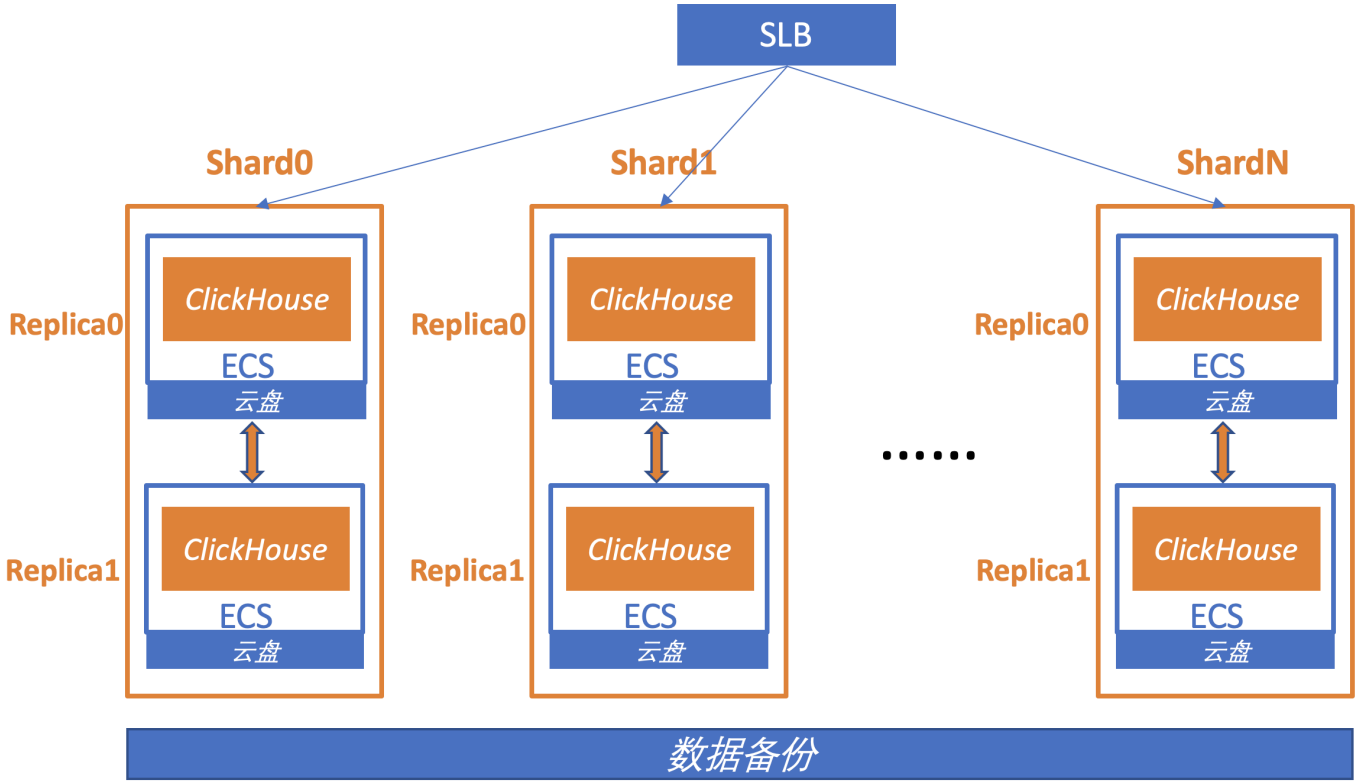
ClickHouse最佳实践 – 高并发查询



ClickHouse最佳实践 – 高并发查询

- index_granularity
  - 单个索引颗粒大小
  - 越大，则单次索引检索命中数据越多
- min_compress_block_size
  - 单个block大小，包含1个或多个索引颗粒
  - 越大，则单次磁盘IO数据量越大
- min_bytes_for_wide_part
  - 小于该值，使用行列混存
  - 大量小写时，compact格式性能更好
- min_rows_for_wide_part
  - 小于该值，使用行列混存；
  - 大量小写时，compact格式性能更好；
- CODEC(NONE)
  - 高频读取的列，可以设置不压缩；
  - 磁盘换空间，提升查询效率；
- LowCardinality/Enum
  - Distinct值少的列，采用词典压缩；

ClickHouse最佳实践 – 高并发查询




```
CREATE TABLE IF NOT EXISTS tbl_all
AS tbl_local
ENGINE = Distributed(
    default, db, tbl_local,
    intHash(user_id));
```

- max_threads
  - 执行单个query的线程数
- optimize_skip_unused_shards
  - 根据sharding策略，跳过未命中分片

ClickHouse最佳实践 – 总结

阿里云 奥运会全球指定云服务商

```
CREATE TABLE IF NOT EXISTS whatever_table ON CLUSTER default (
    user_id UInt64 CODEC(NONE),
    reg_date SimpleAggregateFunction(anyLast, Datetime),
    city_level SimpleAggregateFunction(
        anyLast,
        Nullable(Enum('一线城市' = 0, '二线城市' = 1, '三线城市' = 2, '四线城市' = 3))
    ),
    gender SimpleAggregateFunction(anyLast, Nullable(Enum('女' = 0, '男' = 1))),
    interest_sports SimpleAggregateFunction(anyLast, Nullable(Enum('否' = 0, '是' = 1))),
    comment_like_cnt SimpleAggregateFunction(anyLast, Nullable(UInt32)),
    last30d_share_cnt SimpleAggregateFunction(anyLast, Nullable(UInt32)),
    user_like_consume_trend_type SimpleAggregateFunction(anyLast, Nullable(String)),
    province SimpleAggregateFunction(anyLast, Nullable(LowCardinality(String))),
    last_access_version SimpleAggregateFunction(anyLast, Nullable(String)),
    others SimpleAggregateFunction(anyLast, Nullable(Array(String)))
) ENGINE = AggregatingMergeTree()
PARTITION BY (user_id % 8)
ORDER BY user_id
SETTINGS min_compress_block_size = 16384
,index_granularity = 1024
,min_bytes_for_wide_part=500000000
,min_rows_for_wide_part=1000000;
```

```
CREATE TABLE IF NOT EXISTS tbl_all
AS tbl_local
ENGINE = Distributed(
    default, db, tbl_local,
    intHash(user_id));
```

```
SELECT ...
SETTINGS max_threads = 4,
optimize_skip_unused_shards = 1;
```

ClickHouse最佳实践 – 测试结果

阿里云 | 奥运会全球指定云服务商

```
[redacted]# jmeter -n -t /data/app/jmeter/ck_test_jdbc.jmx -l /data/app/jmeter/result.txt -e -o /data/app/jmeter/webreport


Creating summariser <summary>
Created the tree successfully using /data/app/jmeter/ck_test_jdbc.jmx
[redacted]
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4446
summary +  29945 in 00:00:07 = 4411.5/s Avg:    15 Min:     5 Max:  2214 Err:     0 (0.00%) Active: 136 Started: 136 Finished: 0

summary + 248137 in 00:00:30 = 8271.2/s Avg:    23 Min:     5 Max:   258 Err:     0 (0.00%) Active: 200 Started: 200 Finished: 0
summary = 278082 in 00:00:37 = 7559.0/s Avg:    22 Min:     5 Max:  2214 Err:     0 (0.00%)
summary + 243368 in 00:00:30 = 8112.3/s Avg:    24 Min:     5 Max:   324 Err:     0 (0.00%) Active: 200 Started: 200 Finished: 0
summary = 521450 in 00:01:07 = 7807.5/s Avg:    23 Min:     5 Max:  2214 Err:     0 (0.00%)
summary + 242176 in 00:00:30 = 8072.5/s Avg:    24 Min:     5 Max:   382 Err:     0 (0.00%) Active: 200 Started: 200 Finished: 0
summary = 763626 in 00:01:37 = 7889.7/s Avg:    23 Min:     5 Max:  2214 Err:     0 (0.00%)
summary + 246057 in 00:00:30 = 8201.9/s Avg:    24 Min:     5 Max:   364 Err:     0 (0.00%) Active: 200 Started: 200 Finished: 0
summary = 1009683 in 00:02:07 = 7963.6/s Avg:    24 Min:     5 Max:  2214 Err:     0 (0.00%)
summary + 244025 in 00:00:30 = 8134.2/s Avg:    24 Min:     5 Max:   354 Err:     0 (0.00%) Active: 200 Started: 200 Finished: 0
summary = 1253708 in 00:02:37 = 7996.2/s Avg:    24 Min:     5 Max:  2214 Err:     0 (0.00%)
summary + 242086 in 00:00:30 = 8069.5/s Avg:    24 Min:     5 Max:   403 Err:     0 (0.00%) Active: 200 Started: 200 Finished: 0
summary = 1495794 in 00:03:07 = 8008.0/s Avg:    24 Min:     5 Max:  2214 Err:     0 (0.00%)
summary + 238546 in 00:00:30 = 7951.5/s Avg:    25 Min:     5 Max:   383 Err:     0 (0.00%) Active: 200 Started: 200 Finished: 0
summary = 1734340 in 00:03:37 = 8000.2/s Avg:    24 Min:     5 Max:  2214 Err:     0 (0.00%)
summary + 235824 in 00:00:30 = 7860.8/s Avg:    24 Min:     5 Max:   334 Err:     0 (0.00%) Active: 135 Started: 200 Finished: 65
summary = 1970164 in 00:04:07 = 7983.2/s Avg:    24 Min:     5 Max:  2214 Err:     0 (0.00%)
summary +  29836 in 00:00:05 = 5669.0/s Avg:    13 Min:     5 Max:    65 Err:     0 (0.00%) Active: 0 Started: 200 Finished: 200
summary = 2000000 in 00:04:12 = 7934.9/s Avg:    24 Min:     5 Max:  2214 Err:     0 (0.00%)
Tidying up ...
... end of run
```

阿里云 - 云数据库ClickHouse – 差异化

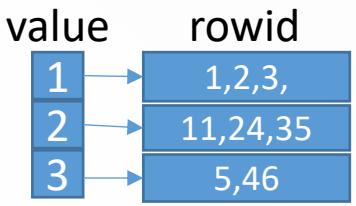
阿里云 | 奥运会全球指定云服务商

模块	功能名称	云数据库ClickHouse	开源自建
运维体系	集群管理	可视化创建、删除、管理实例	手工部署
	Failover	管控任务流自动监控处理	自行处理
	容灾备份	备份与恢复、异地容灾（TODO）	自行处理
	安全性	审计日志、RAM授权、白名单、密码防爆破、公网SLB	自行处理
	参数自助设置	系统参数、用户参数修改；词典管理	自行修改配置文件
	监控报警	完善的多指标监控、报警体系；慢sql分析	自行搭建维护Prometheus等
	水平扩缩容	自动迁移数据	手动迁移数据
	用户权限控制	有，支持RAM子账号授权	无
数据生态	数据接入链路	Flink、Spark、MySQL物化外表、MaxCompute、OSS、SLS、DTS、Kafka、DMS、DataWorks、QuickBI	开源生态
专家支持	专家服务	有，提供业务设计、优化建议、问题快速处理	无
内核研发	版本升级	LTS版本，确保前后兼容	需要自行处理不兼容行为
	社区bugfix	有，快速响应；稳定性、易用性	自行fix
	内核优化	资源队列、二级索引、分层存储、存储计算分离（TODO）、MPP计算（TODO）	无

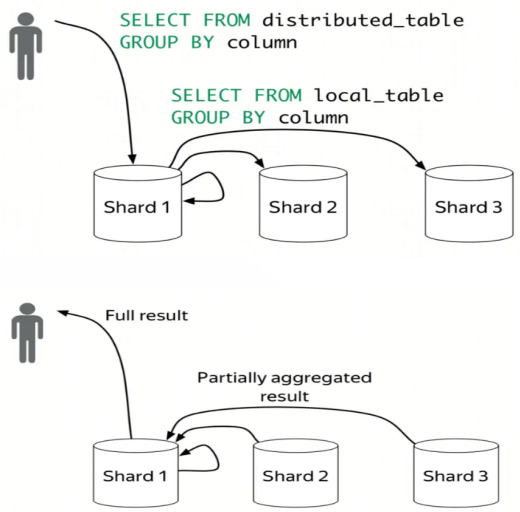
阿里云 - 云数据库ClickHouse – 内核特性

阿里云 | 奥运会全球指定云服务商

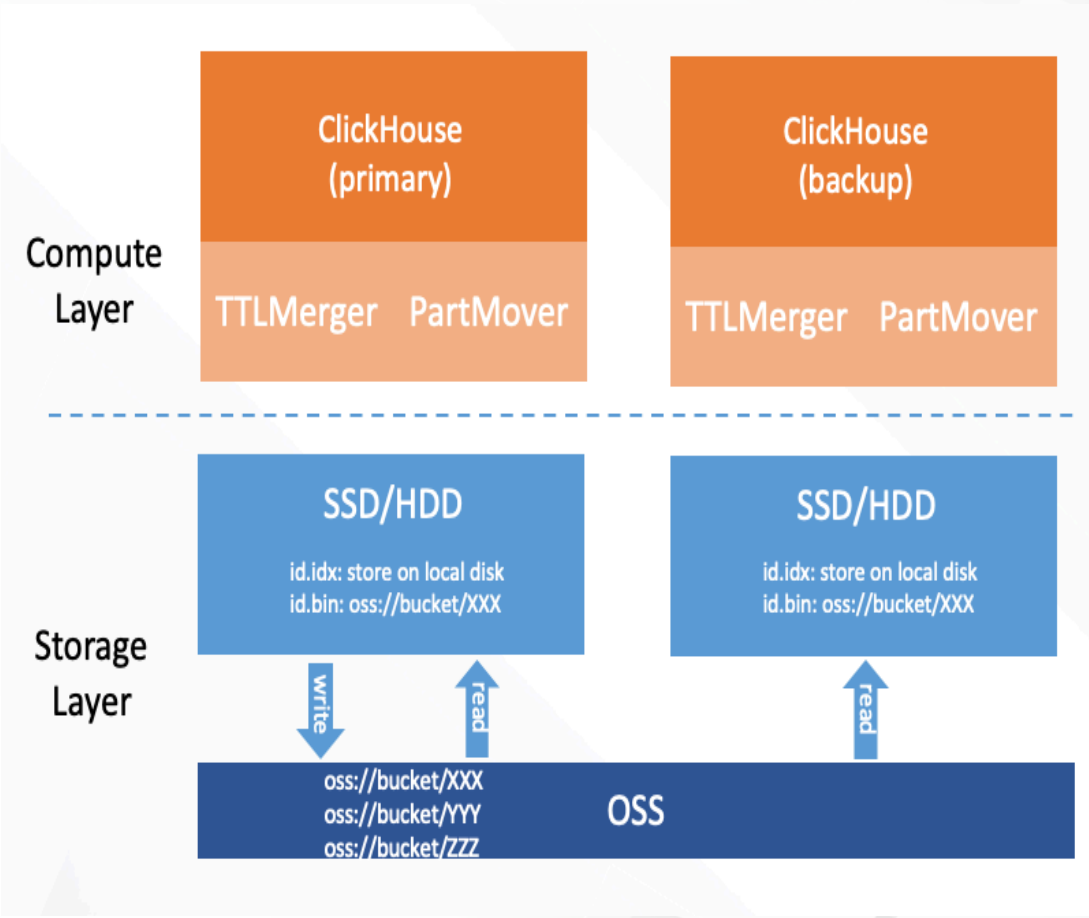
```
Whole data:     [-----------------------------------------]
CounterID:      [aaaaaaaaaaaaaaaaabbbbcdeeeeeeeeeeeeefggggggggghhhhhhhhhiiiiiiiiiklllllll]
Date:           [111111122222223333123321111122222233321111111212222222311111222331112233 3]
Marks:           |    |    |    |    |    |    |    |    |    |    |
                a,1  a,2  a,3  b,3  e,2  e,3  g,1  h,2  i,1  i,3  l,3
Marks numbers:   0    1    2    3    4    5    6    7    8    9    10
```



二级索引



维度表、并发外表



分层存储

云数据库ClickHouse =

ClickHouse + ElasticSearch

Thanks !

微信

钉钉