

# ck基本用法

## ClickHouse集群 (Cluster)

在物理构成上，ClickHouse集群是由多个ClickHouse server实例组成的分布式数据库。这些ClickHouse server根据购买规格的不同而可能包含1个或多个副本（Replica）、1个或多个分片（Shard）。在逻辑构成上，一个ClickHouse集群可以包含多个数据库（Database）对象。

## 系列

ClickHouse集群包含两个系列，分别是高可用系列和单副本系列。高可用系列每个节点包含两个副本，即是双副本集群；单副本系列每个节点只有1个副本。

高可用集群的某个副本服务不可用的时候，同一节点（分片）的另一个副本还可以继续服务，因此集群是高可用集群；单副本集群的某一个副本服务不可用的时候，会导致整个集群不可用，需要等待该副本完全恢复服务状态，集群才能继续提供稳定服务。

因此，高可用集群的资源以及对应的购买成本都是单副本集群的2倍。

## 分片 (Shard)

在超大规模海量数据处理场景下，单台服务器的存储、计算资源会成为瓶颈。为了进一步提高效率，云数据库ClickHouse将海量数据分散存储到多台服务器上，每台服务器只存储和处理海量数据的一部分，在这种架构下，每台服务器被称为一个分片（Shard）。

## 副本 (Replica)

为了在异常情况下保证数据的安全性和服务的高可用性，ClickHouse提供了副本机制，将单台服务器的数据冗余存储在2台或多台服务器上。

## 数据库 (Database)

数据库是ClickHouse集群中的最高级别对象，内部包含表（Table）、列（Column）、视图（View）、函数、数据类型等。

## 表 (Table)

表是数据的组织形式，由多行、多列构成。ClickHouse的表从数据分布上，可以分为本地表、分布式表两种类型。从存储引擎上，可以分为单机表、复制表两种类型。

## 本地表 (Local Table)

本地表的数据，只会存储在当前写入的节点上，不会被分散到多台机器上

## 分布式表 (Distributed Table)

分布式表是本地表的集合，它将多个本地表抽象为一张统一的表，对外提供写入、查询功能。当写入分布式表时，数据会被自动分发到集合中的各个本地表中；当查询分布式表时，集合中的各个本地表都会被分别查询，并且把最终结果汇总后返回。

## 单机表 (Non-Replicated Table)

单机表的数据，只会存储在当前机器上，不会被复制到其他机器，即只有一个副本。

## 复制表 (Replicated Table)

复制表的数据，会被自动复制到多台机器上，形成多个副本。

## 表引擎：

### Integration系列

该系统表引擎主要用于将外部数据导入到ClickHouse中，或者在ClickHouse中直接操作外部数据源。

- Kafka：将Kafka Topic中的数据直接导入到ClickHouse。
- MySQL：将Mysql作为存储引擎，直接在ClickHouse中对MySQL表进行select等操作。
- JDBC/ODBC：通过指定jdbc、odbc连接串读取数据源。
- HDFS：直接读取HDFS上的特定格式的数据文件；

### Special系列

Special系列的表引擎，大多是为了特定场景而定制的。这里也挑选几个简单介绍，不做详述。

- Memory：将数据存储在内存在中，重启后会导致数据丢失。查询性能极好，适合于对于数据持久性没有要求的1亿以下的小表。在ClickHouse中，通常用来做临时表。
- Buffer：为目标表设置一个内存buffer，当buffer达到了一定条件之后会flush到磁盘。
- File：直接将本地文件作为数据存储。
- Null：写入数据被丢弃、读取数据为空。

## MergeTree系列

Log、Special、Integration主要用于特殊用途，场景相对有限。MergeTree系列才是官方主推的存储引擎，支持几乎所有ClickHouse核心功能。

## sql语法：

### 1.创建本地表

SQL

```
1 CREATE TABLE [IF NOT EXISTS] [db.]table_name ON CLUSTER cluster
2 (
3     name1 [type1] [DEFAULT|MATERIALIZED|ALIAS expr1],
4     name2 [type2] [DEFAULT|MATERIALIZED|ALIAS expr2],
5     ...
6     INDEX index_name1 expr1 TYPE type1(...) GRANULARITY value1,
7     INDEX index_name2 expr2 TYPE type2(...) GRANULARITY value2
8 ) ENGINE = engine_name()
9 [PARTITION BY expr]
10 [ORDER BY expr]
11 [PRIMARY KEY expr]
12 [SAMPLE BY expr]
13 [SETTINGS name=value, ...];
```

选项描述：

- db：指定数据库名称，如果当前语句没有包含‘db’，则默认使用当前选择的数据库为‘db’。
- cluster：指定集群名称，目前固定为default。ON CLUSTER 将在每一个节点上都创建一个本地表。
- type：该列数据类型，例如 UInt32。
- DEFAULT：该列缺省值。如果INSERT中不包含指定的列，那么将通过表达式计算它的默认值并填充它。
- MATERIALIZED：物化列表表达式，表示该列不能被INSERT，是被计算出来的；在INSERT语句中，不需要写入该列；在SELECT \*查询语句结果集不包含该列。
- ALIAS：别名列。这样的列不会存储在表中。它的值不能够通过INSERT写入，同时使用SELECT查询星号时，这些列也不会被用来替换星号。但是它们可以用于SELECT中，在这种情况下，在查询分析中别名将被替换。

- 物化列与别名列的区别：物化列是会保存数据，查询的时候不需要计算，而表达式列不会保存数据，查询的时候需要计算，查询时候返回表达式的计算结果

以下选项与表引擎相关，只有MergeTree系列表引擎支持：

- PARTITION BY：指定分区键。通常按照日期分区，也可以用其他字段或字段表达式。
- ORDER BY：指定 排序键。可以是一组列的元组或任意的表达式。
- PRIMARY KEY：指定主键，默认情况下主键跟排序键相同。因此，大部分情况下不需要再专门指定一个 PRIMARY KEY 子句。
- SAMPLE BY：抽样表达式，如果要用抽样表达式，主键中必须包含这个表达式。
- SETTINGS：影响 性能的额外参数。
- GRANULARITY：索引粒度参数。

## 2.创建分布式表

SQL

```
1 CREATE TABLE [db.]table_name ON CLUSTER default
2 AS db.local_table_name ENGINE = Distributed(<cluster>, <database>, <shard table> [, sharding_key])
```

参数说明：

- db：数据库名。
- local\_table\_name：对应的已经创建的本地表表名。
- shard table：同上，对应的已经创建的本地表表名。
- sharding\_key：分片表达式。可以是一个字段，例如user\_id（integer类型），通过对余数值进行取余分片；也可以是一个表达式，例如rand()，通过rand()函数返回值/shards总权重分片；为了分片更均匀，可以加上hash函数，如intHash64(user\_id)。

## 3. 删除数据

按分区删除

SQL

```
1 ALTER TABLE db_name.table_name DROP PARTITION '20200601'
```

按条件删除

## SQL

```
1 ALTER TABLE db_name.table_name DELETE WHERE day = '20200618'
```

## 4. 数据更新

更新和删除是异步执行的，可以通过查看表 system.mutations 来查看命令的是否执行完毕

## SQL

```
1 ALTER TABLE table_name UPDATE col1 = expr1, ... WHERE 1=1;
```

## SQL

```
1 -- 强制后台Compaction
2 optimize table table final;
```

## 数据导入

### 1. mysql查询结果导入ck

## JavaScript

```
1 mysql -u ktvsky -p098f6bcd4621d373cade4e832627b4f6 -P 3308 -h 10.9.169.48 -D gzh --compress -ss -e "SELECT * FROM leike_mac_log"
2 sed 's/"/\\g;s/\\t/","/g;s/^"/"/;s/$/"/'
3 clickhouse-client --host=10.10.50.207 --port=9000 --user=ckadmin --password=ktvsky5166 --query="INSERT INTO leike.leike_mac_log_all FORMAT CSV"
```

### 2. mysql查询结果转csv

## PHP

```
1 mysql -u ktvsky -p098f6bcd4621d373cade4e832627b4f6 -P 3308 -h 10.9.169.48 -D gzh --default-character-set=gbk -e 'select * from leike_mac_log where dt='2020-11-30'
2 sed 's/\\t/","/g;s/^"/"/;s/$/"/;s/\\n/\\n/g' > /tmp/file.csv
```

### 3. 导入指定分割符号

C++

```
1 cat h.log |  
2 clickhouse-client -h 10.9.169.48 --port=3306 --user=root --password=Thunder5166  
--query="INSERT INTO ad_nginx_distributed FORMAT TabSeparated";
```