

ClickHouse 简介及使用

AUTHOR: 彭玲 TIME: 2022/10/31

ClickHouse 简介及使用

简介

数据表引擎

MergeTree

ReplacingMergeTree

CollapsingMergeTree

VersionedCollapsingMergeTree

SummingMergeTree

AggregatingMergeTree

使用

同步 PostgreSQL

其他

简介

ClickHouse 是俄罗斯的 Yandex 于 2016 年开源的用于联机分析 (OLAP, Online Analytical Processing) 的列式存储数据库 (DBMS, Database Management System) , 能够使用 SQL 查询实时生成分析数据报告。

数据表引擎

表引擎是 ClickHouse 的一大特色。可以说, 表引擎决定了如何存储表的数据。包括:

- 数据的存储方式和位置
- 支持哪些查询以及如何支持
- 并发数据访问
- 索引的使用
- 是否可以执行多线性请求
- 数据复制参数

ClickHouse 表引擎一共分为 4 个系列: Log、MergeTree、Integration、Special。其中, MergeTree 系列是官方主推的存储引擎, 支持几乎所有的 ClickHouse 核心功能。

MergeTree

MergeTree 表引擎主要用于海量数据分析, 支持数据分区、存储有序、主键索引、稀疏索引、数据 TTL 等。MergeTree 支持所有 ClickHouse SQL 语法, 但是有些功能与 MySQL 并不一致, 比如在 MergeTree 中主键并不用于去重。

ReplacingMergeTree

为了解决 MergeTree 相同主键无法去重的问题，ClickHouse 提供了 ReplacingMergeTree 引擎，用来做去重。

CollapsingMergeTree

ClickHouse 实现了 CollapsingMergeTree 来消除 ReplacingMergeTree 的限制。该引擎要求在建表语句中指定一个标记列 `sign`，后台 Compaction 时会将主键相同、Sign 相反的行进行折叠，也即删除。

CollapsingMergeTree 将行按照 `sign` 的值分为两类：`sign=1` 的行称之为 状态行，`sign=-1` 的行称之为 取消行。

每次需要新增状态时，写入一行状态行；需要删除状态时，则写入一行取消行。

在后台 Compaction 时，状态行与取消行会自动做折叠（删除）处理。而尚未进行 Compaction 的数据，状态行与取消行同时存在。

VersionedCollapsingMergeTree

为了解决 CollapsingMergeTree 乱序写入情况下无法正常折叠问题，VersionedCollapsingMergeTree 表引擎在建表语句中新增了一列 `Version`，用于在乱序情况下记录状态行与取消行的对应关系。主键相同，且 `Version` 相同、`Sign` 相反的行，在 Compaction 时会被删除。

SummingMergeTree

ClickHouse 通过 SummingMergeTree 来支持对主键列进行预先聚合。在后台 Compaction 时，会将主键相同的多行进行 `sum` 求和，然后使用一行数据取而代之，从而大幅度降低存储空间占用，提升聚合计算性能。

AggregatingMergeTree

AggregatingMergeTree 也是预先聚合引擎的一种，用于提升聚合计算的性能。与 SummingMergeTree 的区别在于：SummingMergeTree 对非主键列进行 `sum` 聚合，而 AggregatingMergeTree 则可以指定各种聚合函数。

AggregatingMergeTree 的语法比较复杂，需要结合物化视图或 ClickHouse 的特殊数据类型 `AggregateFunction` 一起使用。在 `insert` 和 `select` 时，也有独特的写法和要求：写入时需要使用 `-state` 语法，查询时使用 `-Merge` 语法。

使用

同步 PostgreSQL

```
1  -- 同步项企相关数据库
2  create database if not exists pg_pepca
3  engine = PostgreSQL('10.8.30.166:5432', 'pepca', 'FashionAdmin', '123456');
4
5  create database if not exists pg_peppm
6  engine = PostgreSQL('10.8.30.166:5432', 'peppm', 'FashionAdmin', '123456');
7
8  create database if not exists pg_camworkflow
9  engine = PostgreSQL('10.8.30.166:5432', 'camworkflow', 'FashionAdmin', '123456');
```

由于 PostgreSQL 引擎数据库 不支持 创建表或视图，预创建表或者视图，可以新建一个 ClickHouse 数据库，在该库下进行数据表或者视图的创建。

```
1 create database if not exists ch_pepca;
2
3 -- 查询 销售/工程 人员列表
4 -- functions = {1:销售, 3:工程}
5 create view if not exists ch_pepca.user_department_v as
6 select
7     distinct du.department as department_id,
8     u.id as user_id,
9     u.name as user_name,
10    u.hiredate as hire_date,
11    d.name as department_name,
12    d.functions as department_function
13 from pg_pepca.department as d
14 inner join pg_pepca.department_user as du on d.id=du.department
15 inner join pg_pepca.user as u on du.user=u.id
16 where d.functions in (1,3);
```

其他

```
1 select version();
2 show tables;
3
4 -- 查看数据类型
5 select * from system.data_type_families;
6
7
8 create table if not exists test_types (
9     tid Int32,
10    tname String,
11    ttype Array(Int32)
12 ) engine = MergeTree order by tid;
13
14
15 drop view student_view;
16 DROP VIEW student_mv; -- student_view_m
17 drop table teacher;
18 drop table student;
19
20 CREATE TABLE if not exists teacher (
21     tid Int32,
22     tname String
23 ) engine = MergeTree order by tid;
24
25 CREATE TABLE if not exists student (
26     sid int,
27     teacher_id int DEFAULT 0,
28     sname String
29 ) engine = MergeTree order by sid;
30
```

```

31 CREATE VIEW student_view AS
32 SELECT *
33     FROM student
34     LEFT JOIN teacher
35     ON student.teacher_id = teacher.tid;
36
37 create table if not exists student_mv_storage
38 (
39     sid Int32,
40     sname String,
41     tid Int32,
42     tname String
43 )
44 engine = MergeTree
45 order by sid;
46
47 CREATE MATERIALIZED VIEW if not exists student_mv to student_mv_storage
48     AS
49 SELECT sid, sname, tid, tname
50     FROM student
51     LEFT JOIN teacher
52     ON student.teacher_id = teacher.tid;
53
54 insert into teacher values (2, '李老师');
55 insert into teacher values (3, '王老师');
56 insert into student values (2, 1, 'Jack');
57 insert into student values (3, 2, 'Helen');
58 insert into student values (4, 3, 'Li Lei');
59 insert into student values (5, 3, 'Wang Meimei');
60
61 ALTER TABLE student RENAME COLUMN s_name to sname;
62
63 select * from student_view order by sid;
64 select * from student_mv order by sid;
65 select * from student_mv_storage order by sid;
66
67 alter table student update sname='Han Meimei' where sid=5;
68
69 -- optimize table student_mv_storage;
70
71 /*
72 insert into student_view_mv
73     select * from student
74         left join teacher on student.teacher_id=teacher.tid
75         where sid=1;
76 */
77
78 drop table pg_student;
79 create table if not exists pg_student
80 (
81     sid integer,
82     tname String
83 )

```

```

84 engine = PostgreSQL('10.8.30.36:5432', 'pl_test', 'student', 'FashionAdmin',
85 '123456');
86
87
88 select * from pg_student;
89
90 insert into pg_student values (6, 'Lisa');
91 alter table pg_student update tname='LiLi' where sid=6;
92
93 -- drop database pg_test_m2;
94
95 SET allow_experimental_database_materialized_postgresql=1;
96
97 create database if not exists pg_test_m
98 engine = MaterializedPostgreSQL('10.8.30.36:5432', 'pl_test', 'FashionAdmin',
99 '123456');
100
101 create table if not exists pg_test.student
102 (
103     sid integer,
104     tname String
105 )
106 engine = PostgreSQL('10.8.30.36:5432', 'pl_test', 'student', 'FashionAdmin',
107 '123456');
108
109 create view if not exists pg_test.student_v as
110 select * from pg_test.student;
111
112 -- drop table pl_test.student_t;
113 create table if not exists pg_test.student_t
114 (
115     sid Int32,
116     sname String
117 ) engine = MergeTree order by sid;
118
119 show tables from pg_test;
120 select * from pg_test.teacher order by id;
121 insert into pg_test.teacher values (4, '孙老师');
122
123 create database if not exists pg_test_m
124 engine=PostgreSQL('10.8.30.36:5432', 'pl_test', 'FashionAdmin', '123456');
125
126 select * from pg_test_m.teacher order by id;
127
128 drop database pg_test_m;
129 -- drop database pg_test;
130
131
132 create table if not exists download
133 (
134     when DateTime,
135     userid Int32,
136     bytes Int64
137 ) engine = MergeTree order by when;

```

```

134
135 insert into download values ('2020-08-31 18:22:06', 1, 530314);
136 insert into download values ('2020-08-31 18:22:07', 1, 872957);
137 insert into download values ('2020-08-31 18:22:08', 1, 107047);
138 insert into download values ('2020-08-31 18:22:09', 2, 214876);
139 insert into download values ('2020-08-31 18:22:10', 2, 820943);
140 insert into download values ('2020-08-31 18:22:11', 3, 693959);
141 insert into download values ('2020-08-31 18:22:12', 3, 882151);
142 insert into download values ('2020-08-31 18:22:13', 3, 644223);
143 insert into download values ('2020-08-31 18:22:14', 1, 199800);
144 insert into download values ('2020-08-31 18:22:15', 1, 511439);
145 insert into download values ('2020-08-31 19:00:00', 1, 511439);
146 insert into download values ('2020-08-31 19:22:00', 1, 411439);
147 insert into download values ('2020-08-31 20:22:15', 1, 511439);
148 insert into download values ('2020-08-31 21:22:15', 1, 511439);
149 insert into download values ('2020-08-31 22:22:15', 1, 511439);
150 insert into download values ('2020-09-01 00:00:00', 1, 301000);
151 insert into download values ('2020-09-02 02:22:15', 2, 511439);
152 insert into download values ('2020-09-03 03:22:15', 2, 511439);
153 insert into download values ('2020-09-04 04:22:15', 1, 511439);
154 insert into download values ('2020-09-05 05:22:15', 1, 511439);
155
156 select * from download;
157
158 SELECT toStartOfHour(when) AS hour, userid, count() as downloads, sum(bytes) AS
bytes
159 FROM download
160 GROUP BY userid, hour ORDER BY userid, hour;
161
162 CREATE MATERIALIZED VIEW download_hour_mv
163 ENGINE = SummingMergeTree
164 PARTITION BY toYYYYMM(hour) ORDER BY (userid, hour)
165 AS SELECT
166     toStartOfHour(when) AS hour,
167     userid,
168     count() as downloads,
169     sum(bytes) AS bytes
170 FROM download WHERE when >= toDateTime('2020-08-31 19:00:00')
171 GROUP BY userid, hour;
172
173 select * from download_hour_mv order by userid, hour;
174
175 insert into download_hour_mv
176 select
177     toStartOfHour(when) as hour,
178     userid,
179     count() as download,
180     sum(bytes) as bytes
181 from download where when >= toDateTime('2020-08-31 19:00:00')
182 group by userid, hour;
183
184 INSERT INTO download
185     SELECT

```

```
186         toDateTime('2020-09-01 04:00:00') + number*(1/3) as when,
187         19,
188         rand() % 1000000
189     FROM system.numbers
190     LIMIT 10;
191
192 alter table download delete where userid=19;
193
194 select * from download_hour_mv where userid=19;
195
196
197 -- https://clickhouse.com/docs/zh/engines/table-engines/mergetree-
198 family/collapsingmergetree
199 drop table UAct;
200 CREATE TABLE UAct
201 (
202     UserID UInt64,
203     PageViews UInt8,
204     Duration UInt8,
205     Sign Int8
206 )
207 engine = CollapsingMergeTree(Sign)
208 order by UserID;
209
210 INSERT INTO UAct VALUES (4324182021466249494, 5, 146, 1);
211 INSERT INTO UAct VALUES (4324182021466249494, 5, 146, -1), (4324182021466249494, 6,
212 185, 1);
213
214 select * from UAct;
215
216 -- truncate table UAct;
217
218 SELECT
219     UserID,
220     sum(PageViews * Sign) AS PageViews,
221     sum(Duration * Sign) AS Duration
222 FROM UAct
223 GROUP BY UserID
224 HAVING sum(Sign) > 0;
225
226 optimize table UAct final;
227
228 select * from UAct;
229
230 CREATE TABLE summtt
231 (
232     key UInt32,
233     value UInt32
234 )
235 ENGINE = SummingMergeTree()
236 ORDER BY key;
```

```

237 -- INSERT INTO summtt Values(1,1),(1,2),(2,1);
238 insert into summtt values (1,1);
239 insert into summtt values (1,2);
240 insert into summtt values (2,1);
241
242 select * from summtt;
243
244 optimize table summtt;
245
246
247 -- AggregatingMergeTree
248 CREATE TABLE visits
249 (
250     UserID UInt64,
251     CounterID UInt8,
252     StartDate Date,
253     Sign Int8
254 )
255 ENGINE = CollapsingMergeTree(Sign)
256 ORDER BY UserID;
257
258 CREATE MATERIALIZED VIEW visits_agg_view
259 ENGINE = AggregatingMergeTree() PARTITION BY toYYYYMM(StartDate) ORDER BY
(CounterID, StartDate)
260 AS SELECT
261     CounterID,
262     StartDate,
263     sumState(Sign) AS Visits,
264     uniqState(UserID) AS Users
265 FROM visits
266 GROUP BY CounterID, StartDate;
267
268 INSERT INTO visits VALUES(0, 0, '2019-11-11', 1);
269 INSERT INTO visits VALUES(1, 1, '2019-11-12', 1);
270
271 select * from visits;
272 select * from visits_agg_view;
273
274 SELECT
275     StartDate,
276     sumMerge(Visits) AS Visits,
277     uniqMerge(Users) AS Users
278 FROM visits_agg_view
279 GROUP BY StartDate
280 ORDER BY StartDate;
281
282
283 CREATE TABLE test_tbl_replacing (
284     id UInt16,
285     create_time Date,
286     comment Nullable(String)
287 ) ENGINE = ReplacingMergeTree()
288     PARTITION BY create_time

```



```
289     ORDER BY (id, create_time)
290     PRIMARY KEY (id, create_time)
291     TTL create_time + INTERVAL 1 MONTH
292     SETTINGS index_granularity=8192;
293
294 insert into test_tbl_replacing values(0, '2019-12-12', null);
295 insert into test_tbl_replacing values(0, '2019-12-12', null);
296 insert into test_tbl_replacing values(1, '2019-12-13', null);
297 insert into test_tbl_replacing values(1, '2019-12-13', null);
298 insert into test_tbl_replacing values(2, '2019-12-14', null);
299
300 select * from test_tbl_replacing;
301
302 optimize table test_tbl_replacing final;
303 select * from test_tbl_replacing;
304
305
306 CREATE MATERIALIZED VIEW download_hour_mv_collapsing
307 ENGINE = CollapsingMergeTree(Sign)
308 PARTITION BY toYYYYMM(hour) ORDER BY (userid, hour)
309 AS SELECT
310     toStartOfHour(when) AS hour,
311     userid,
312     count() as downloads,
313     sum(bytes) AS bytes,
314     sign Int8
315 FROM download WHERE when >= toDateTime('2020-08-31 19:00:00')
316 GROUP BY userid, hour;
317
318 select * from download_hour_mv_collapsing;
```