StreamNative

TGIP 019

Interactive Querying Of Streams Using Apache Pulsar

Gaoran 2020.8.2

PR 7646 - Support Set Backlog Quota On Topic Level

Http Rest API

- *GET: {tenant}/{namespace}/{topic}/backlogQuotaMap*
- *POST: {tenant}/{namespace}/{topic}/backlogQuota*
- *DELETE: {tenant}/{namespace}/{topic}/backlogQuota*

- *PIP 39 - Namespace Change Events*
- *PR 4955 - Introduce system topic and topic policies service*

Community

PR 7647 - Allow Ability To Specify Retain Key Ordering In Functions

- *FunctionConfig: retainKeyOrdering*

PR 7655 - Fix Backward Compatibility Issues With Batch Index Acknowledgment

- *Consumer: batchIndexAckEnabled*
- *Broker: acknowledgmentAtBatchIndexLevelEnabled*

Presto

Distributed SQL Query Engine for Big Data

What Presto Is Not ?

What Presto Is ?

Pulsar SQL Based On Presto

- *PulsarConnector -> Connector*
- *PulsarSplitManager -> ConnectorSplitManager*
- *PulsarSplit -> ConnectorSplit*
- *PulsarRecordCursor -> RecordCursor*
- *PulsarMetadata -> ConnectorMetadata*
- *PulsarColumnMetadata -> ColumnMetadata*

Namespace Retention

- *bin/pulsar-admin namespaces set-retention --size 100M --time 7d public/default*
- *bin/pulsar-admin namespaces get-retention public/default*

Produce Messages To Pulsar

- *Primitive Type: BOOLEAN, INT16, DOUBLE, STRING, etc.*
- *Complex Type: KeyValue, Struct (Avro, Json)*

Pulsar SQL

```
public class Product {

    private String name;
    private Double price;
    private Long updateTime;
    private String updateTimeStr;

}
```

JavaBean

Pulsar SQL

```
Producer<Product> producer = pulsarClient.newProducer(Schema.JSON(Product.class)) 1
        .topic(XPHONE_TOPIC)
        .batchingMaxMessages(2000) // default 1000 2
        .batchingMaxPublishDelay( batchDelay: 100, TimeUnit.MILLISECONDS) // default 1 ms 3
        .compressionType(CompressionType.ZSTD) 4
        .create();

for (Product product : getXPhoneDataList()) {
    producer.newMessage().value(product).sendAsync(); 5
}
```

1. Use JsonSchema
2. Set batching max messages
3. Set batching max publish delay
4. Set compressionType
5. Send in async way

Query

Pulsar SQL

```
presto> select * from pulsar."public/default"."xphone-info-topic";
  name  | price  | updatetimestr |  updatetime   | __partition__ | __event_time__ |    __publish_time__     | __message_id__ | __sequence_id__ | __producer_name__ | __key__ | __properties__
--------+--------+---------------+---------------+---------------+----------------+-------------------------+----------------+-----------------+-------------------+---------+----------------
 XPhone | 5999.0 | 2020-04-23    | 1587654055876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,0)       |               0 | standalone-0-0    | NULL    | {}
 XPhone | 5999.0 | 2020-04-24    | 1587740455876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,1)       |               1 | standalone-0-0    | NULL    | {}
 XPhone | 5999.0 | 2020-04-25    | 1587826855876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,2)       |               2 | standalone-0-0    | NULL    | {}
 XPhone | 5999.0 | 2020-04-26    | 1587913255876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,3)       |               3 | standalone-0-0    | NULL    | {}
 XPhone | 5999.0 | 2020-04-27    | 1587999655876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,4)       |               4 | standalone-0-0    | NULL    | {}
 XPhone | 5999.0 | 2020-04-28    | 1588086055876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,5)       |               5 | standalone-0-0    | NULL    | {}
 XPhone | 5999.0 | 2020-04-29    | 1588172455876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,6)       |               6 | standalone-0-0    | NULL    | {}
 XPhone | 5999.0 | 2020-04-30    | 1588258855876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,7)       |               7 | standalone-0-0    | NULL    | {}
 XPhone | 5999.0 | 2020-05-01    | 1588345255876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,8)       |               8 | standalone-0-0    | NULL    | {}
 XPhone | 5999.0 | 2020-05-02    | 1588431655876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,9)       |               9 | standalone-0-0    | NULL    | {}
 XPhone | 5999.0 | 2020-05-03    | 1588518055876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,10)      |              10 | standalone-0-0    | NULL    | {}
 XPhone | 5999.0 | 2020-05-04    | 1588604455876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,11)      |              11 | standalone-0-0    | NULL    | {}
 XPhone | 5999.0 | 2020-05-05    | 1588690855876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,12)      |              12 | standalone-0-0    | NULL    | {}
 XPhone | 5999.0 | 2020-05-06    | 1588777255876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,13)      |              13 | standalone-0-0    | NULL    | {}
 XPhone | 5999.0 | 2020-05-07    | 1588863655876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,14)      |              14 | standalone-0-0    | NULL    | {}
 XPhone | 5999.0 | 2020-05-08    | 1588950055876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,15)      |              15 | standalone-0-0    | NULL    | {}
 XPhone | 5999.0 | 2020-05-09    | 1589036455876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,16)      |              16 | standalone-0-0    | NULL    | {}
 XPhone | 5999.0 | 2020-05-10    | 1589122855876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,17)      |              17 | standalone-0-0    | NULL    | {}
 XPhone | 5999.0 | 2020-05-11    | 1589209255876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,18)      |              18 | standalone-0-0    | NULL    | {}
 XPhone | 5999.0 | 2020-05-12    | 1589295655876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,19)      |              19 | standalone-0-0    | NULL    | {}
 XPhone | 5599.0 | 2020-05-13    | 1589382055876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,20)      |              20 | standalone-0-0    | NULL    | {}
 XPhone | 5599.0 | 2020-05-14    | 1589468455876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,21)      |              21 | standalone-0-0    | NULL    | {}
 XPhone | 5599.0 | 2020-05-15    | 1589554855876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,22)      |              22 | standalone-0-0    | NULL    | {}
 XPhone | 5599.0 | 2020-05-16    | 1589641255876 |            -1 | NULL           | 2020-08-01 23:00:55.912 | (10,0,23)      |              23 | standalone-0-0    | NULL    | {}
```

bin/pulsar sql

Pulsar SQL

```
// 引入依赖
<dependency>
  <groupId>io.prestosql</groupId>
  <artifactId>presto-jdbc</artifactId>
  <version>${presto.version}</version>
</dependency>
```

Presto JDBC

```
public void query() throws SQLException {
    String url = String.format("jdbc:presto://%s",  "localhost:8081");
    Connection connection = DriverManager.getConnection(url,  user: "test",  password: null);

    String query = String.format("select * from pulsar" +
            ".\"public/default\".%s order by __publish_time__ asc", "\"xphone-info-topic\"");
    log.info("Executing query: {}", query);
    ResultSet res = connection.createStatement().executeQuery(query);

    while (res.next()) {
        ResultSetMetaData rsmd = res.getMetaData();
        int columnsNumber = rsmd.getColumnCount();
        for (int i = 1; i <= columnsNumber; i++) {
            if (i > 1) System.out.print(",  ");
            String columnValue = res.getString(i);
            System.out.print(columnValue + " " + rsmd.getColumnName(i));
        }
        System.out.println("");
    }
}
```

Pulsar SQL

Presto Http API

http://ip:port/v1/statement

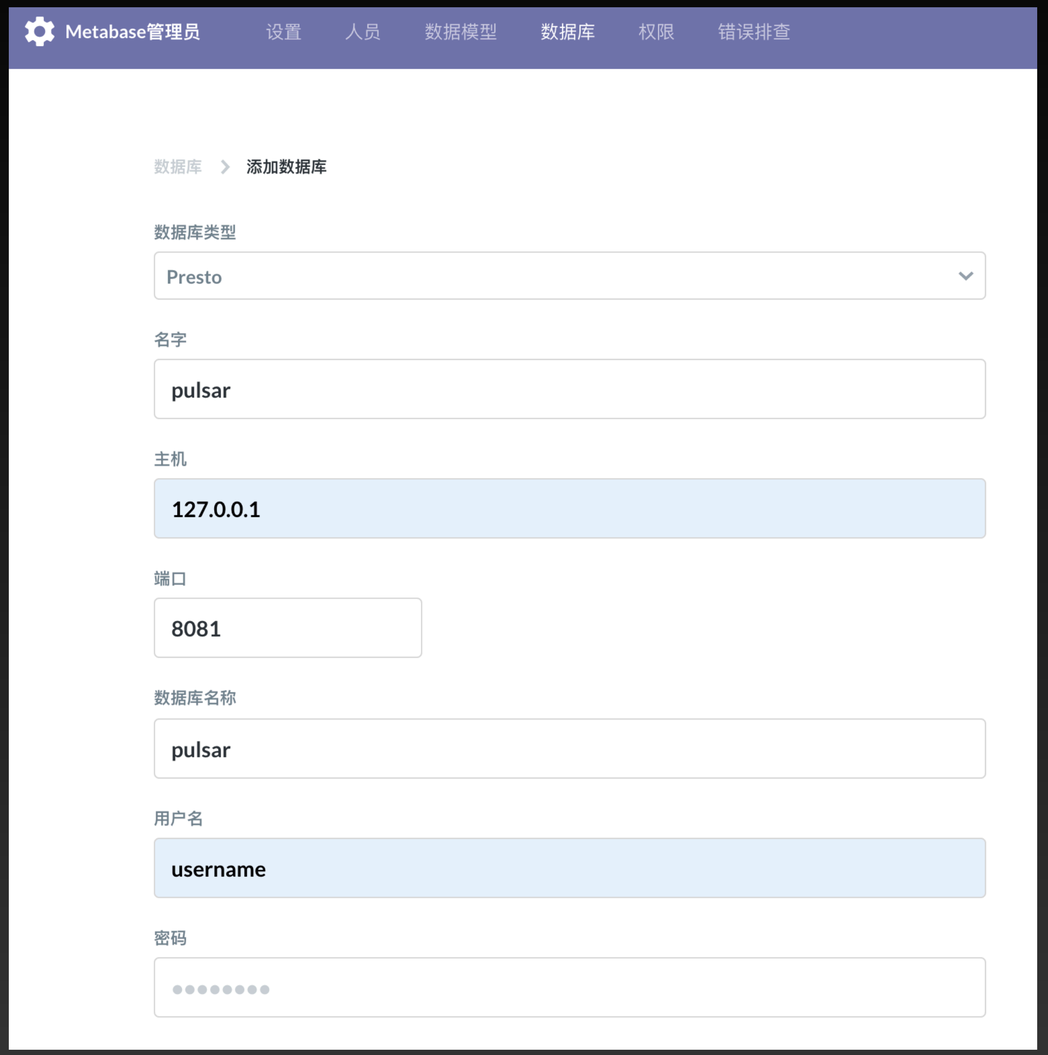
http header: X-Presto-User


```
⊟{
    "id":"20191115_194338_00057_haau9",
    "infoUri":"http://127.0.0.1:8081/ui/query.html?20191115_194338_00057_haau9",
    "nextUri":"http://127.0.0.1:8081/v1/statement/20191115_194338_00057_haau9/3",
    "columns":⊞Array[13],
    "data":⊞Array[1],
    "stats":⊞Object{...}
}
```

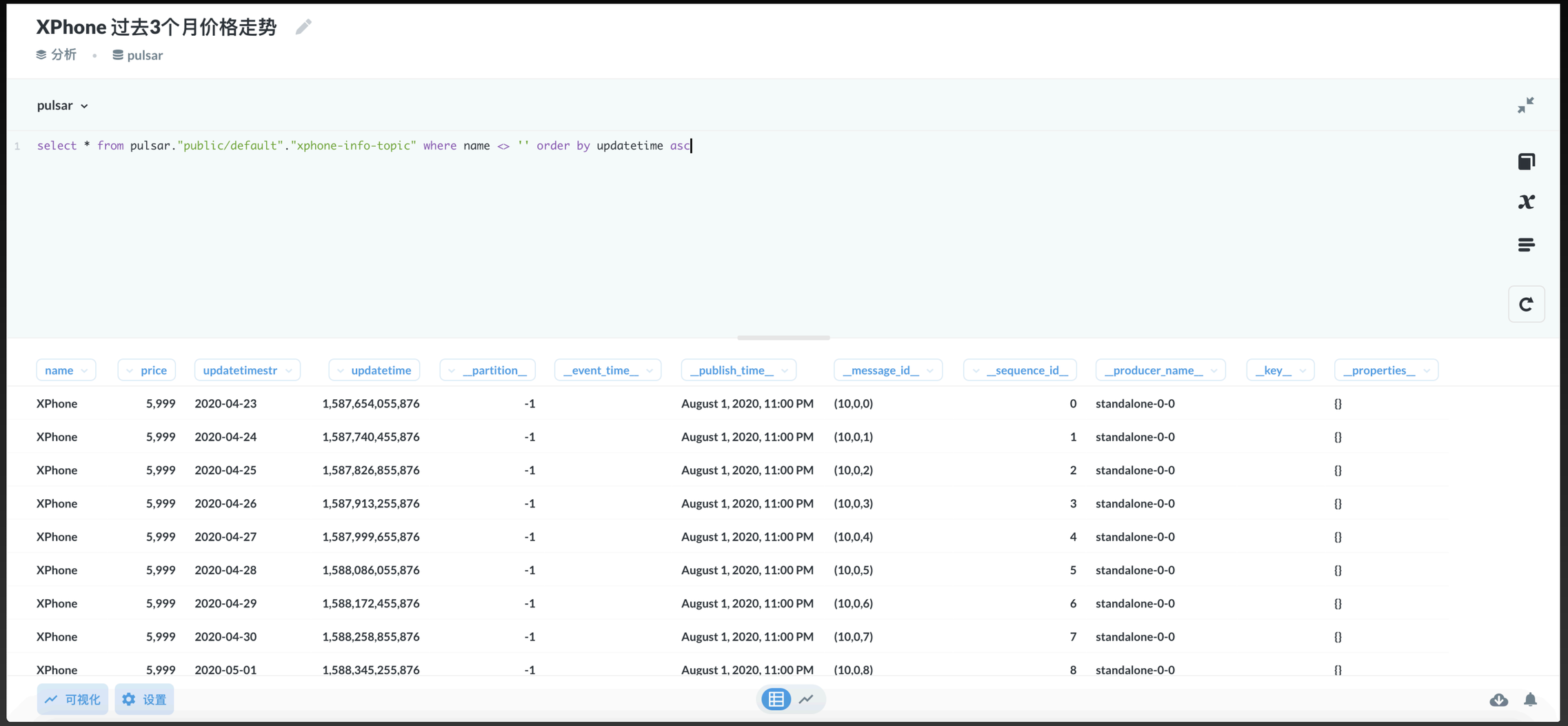
Metabase

Metabase is the easy, open source way for everyone in your company to ask questions and learn from data.

Pulsar SQL


Metabase管理员
设置
人员
数据模型
数据库
权限
错误排查
数据库
添加数据库
数据库类型
Presto
名字
pulsar
主机
127.0.0.1
端口
8081
数据库名称
pulsar
用户名
username
密码

Pulsar SQL

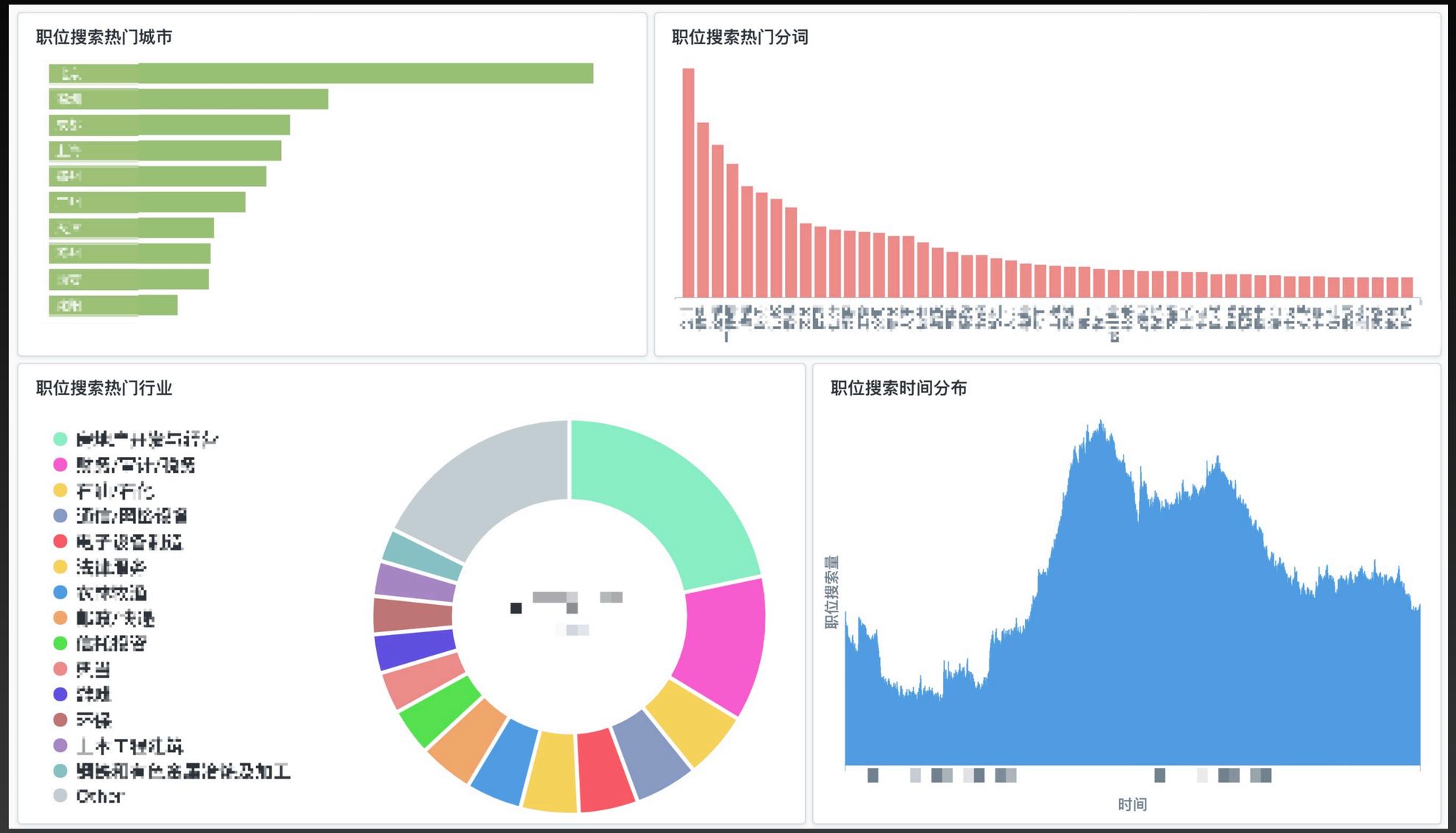


name	price	updatetimestr	updatetime	__partition__	__event_time__	__publish_time__	__message_id__	__sequence_id__	__producer_name__	__key__	__properties__
XPhone	5,999	2020-04-23	1,587,654,055,876	-1		August 1, 2020, 11:00 PM	(10,0,0)	0	standalone-0-0		{}
XPhone	5,999	2020-04-24	1,587,740,455,876	-1		August 1, 2020, 11:00 PM	(10,0,1)	1	standalone-0-0		{}
XPhone	5,999	2020-04-25	1,587,826,855,876	-1		August 1, 2020, 11:00 PM	(10,0,2)	2	standalone-0-0		{}
XPhone	5,999	2020-04-26	1,587,913,255,876	-1		August 1, 2020, 11:00 PM	(10,0,3)	3	standalone-0-0		{}
XPhone	5,999	2020-04-27	1,587,999,655,876	-1		August 1, 2020, 11:00 PM	(10,0,4)	4	standalone-0-0		{}
XPhone	5,999	2020-04-28	1,588,086,055,876	-1		August 1, 2020, 11:00 PM	(10,0,5)	5	standalone-0-0		{}
XPhone	5,999	2020-04-29	1,588,172,455,876	-1		August 1, 2020, 11:00 PM	(10,0,6)	6	standalone-0-0		{}
XPhone	5,999	2020-04-30	1,588,258,855,876	-1		August 1, 2020, 11:00 PM	(10,0,7)	7	standalone-0-0		{}
XPhone	5,999	2020-05-01	1,588,345,255,876	-1		August 1, 2020, 11:00 PM	(10,0,8)	8	standalone-0-0		{}

Pulsar SQL


XPhone 过去3个月价格走势
分析
XPhone 过去3个月价格走势
目标
7,000
6,000
5,000
4,000
3,000
2,000
1,000
0
价格
5,999 5,999 5,999 5,999 5,999 5,999 5,999
5,599 5,599 5,599 5,599 5,599 5,599 5,599 5,599 5,599 5,599
5,200 5,200 5,200
5,599 5,599 5,599 5,599
5,899 5,899 5,899 5,899 5,899 5,899 5,899 5,899 5,899 5,899
May 1, 2020
June 1, 2020
July 1, 2020
August 1, 2020
时间

Pulsar SQL



Query Performance Optimized

Pulsar SQL

Add Topic Partition Number

```
select traceid, __partition__ from "sql|stat"."position-monitor-detail-jdid-partition-1" limit 100

select traceid, __partition__ from "sql|stat"."position-monitor-detail-jdid" where __partition__ = 1 limit 100

select traceid, __partition__ from "sql|stat"."position-monitor-detail-jdid" where __partition__ in (1,2) limit 100
```

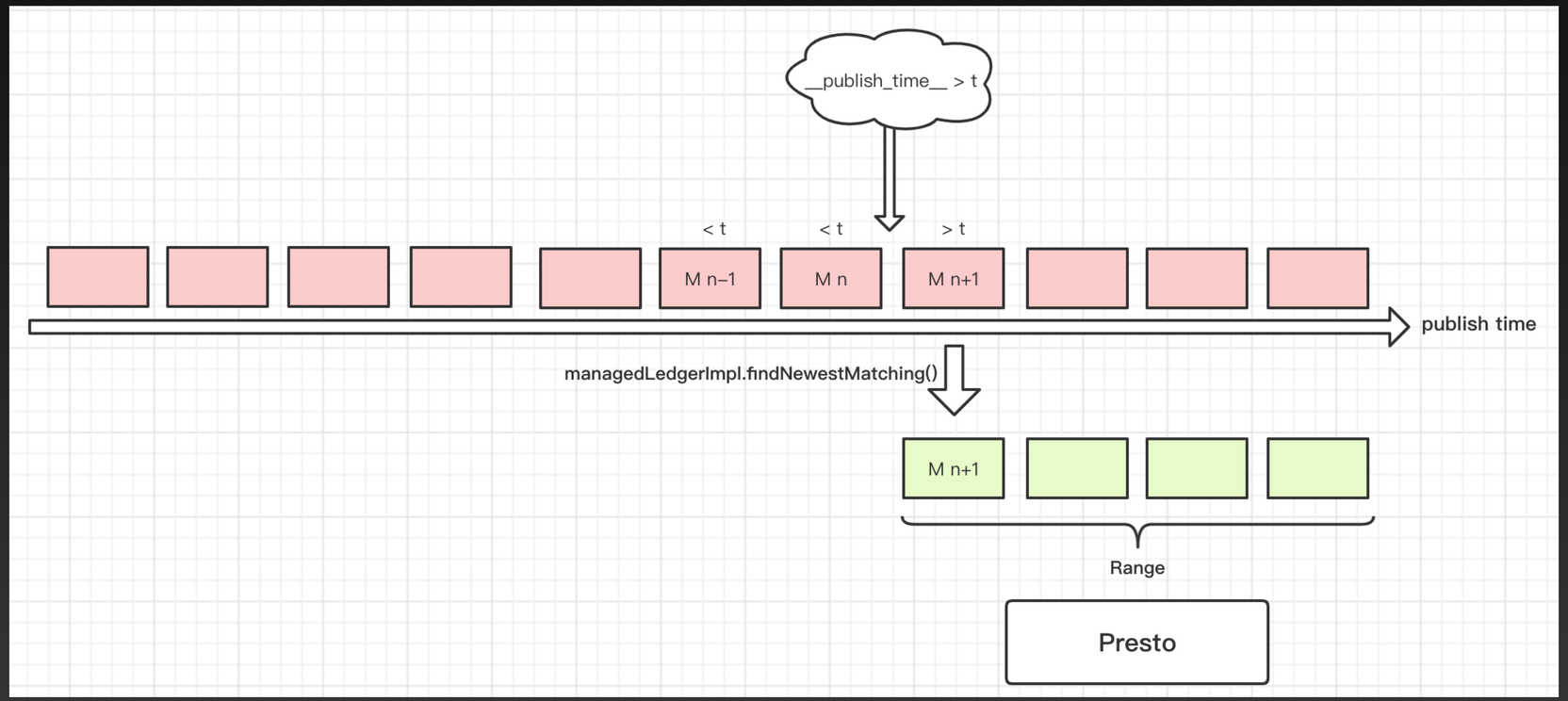
Pulsar SQL

Use Meta Property __publish_time__

```
select * from "sql|stat"."position-monitor-detail-jdid"
where __partition__ = 1
and __publish_time__ > timestamp '2019-11-15 09:00:00'
limit 100
```

Pulsar SQL

How Pulsar SQL Use __publish_time__



isLowerUnbounded = true

Pulsar SQL Config

${PULSAR_HOME}/conf/presto/config.properties

coordinator=true

node-scheduler.include-coordinator=true

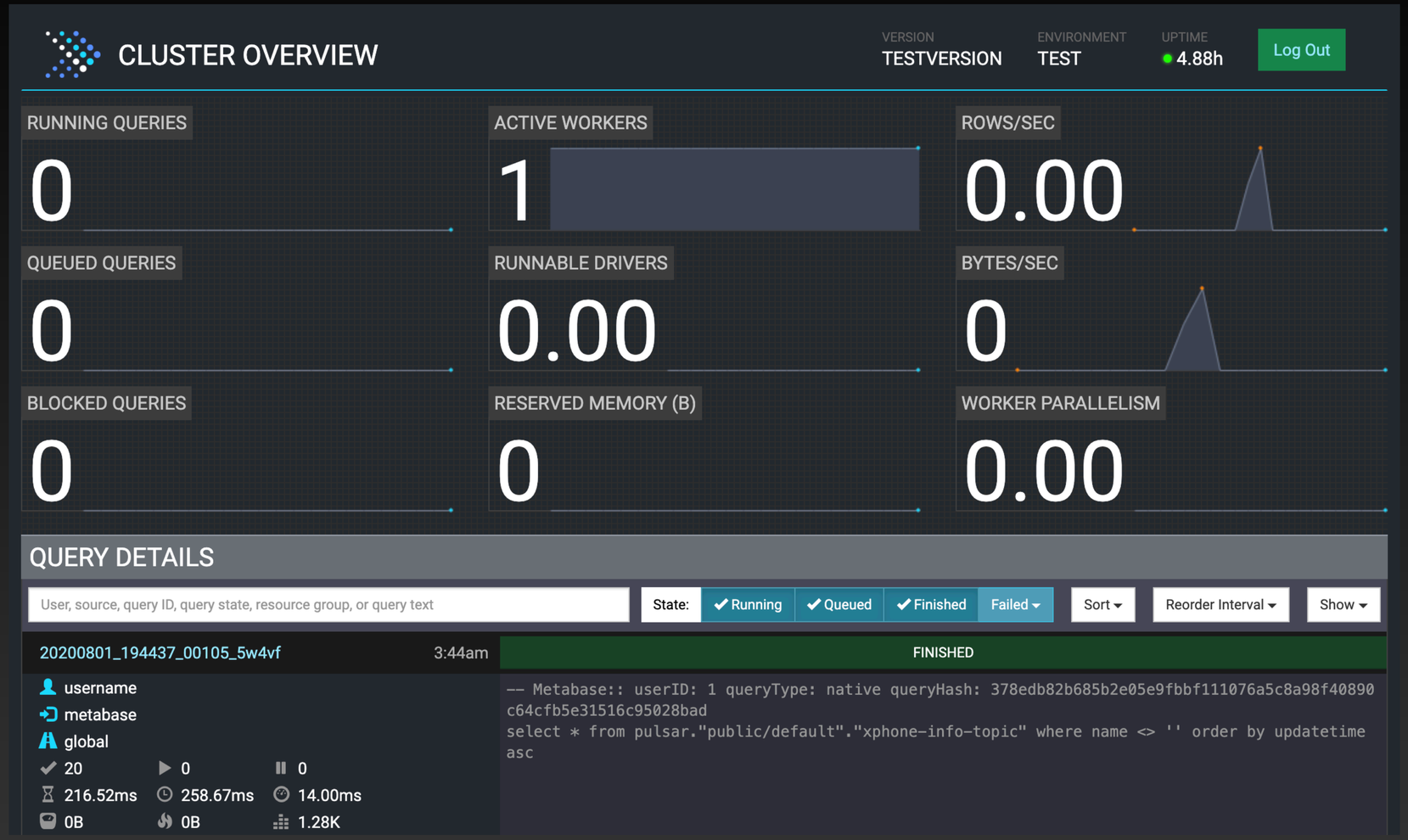
${PULSAR_HOME}/conf/presto/catalog/pulsar.properties

Offload related configurations

pulsar.target-num-splits

pulsar.bookkeeper-throttle-value = 0

Pulsar SQL



localhost:8081

New Feature

Pulsar SQL Read Compaction Data

Thank You