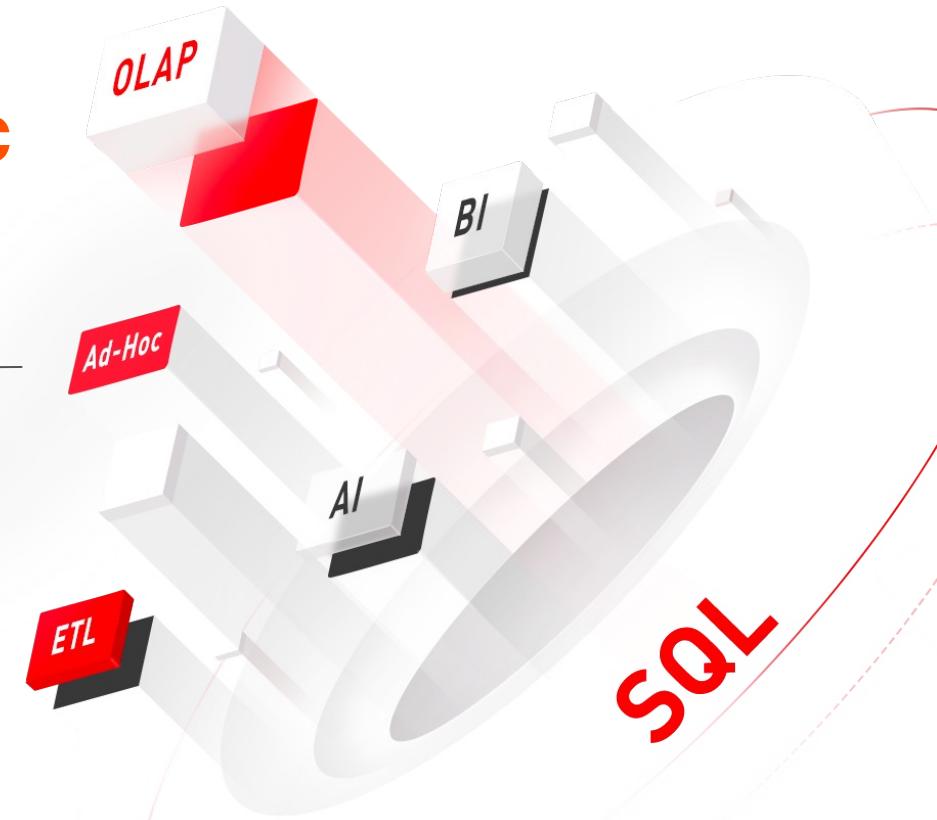


Kyuubi on Arctic 流批一体实践

潘成
网易数帆 软件工程师
Apache Kyuubi(Incubating) PPMC 成员

史大洋
网易数帆 软件工程师
Arctic 核心开发成员





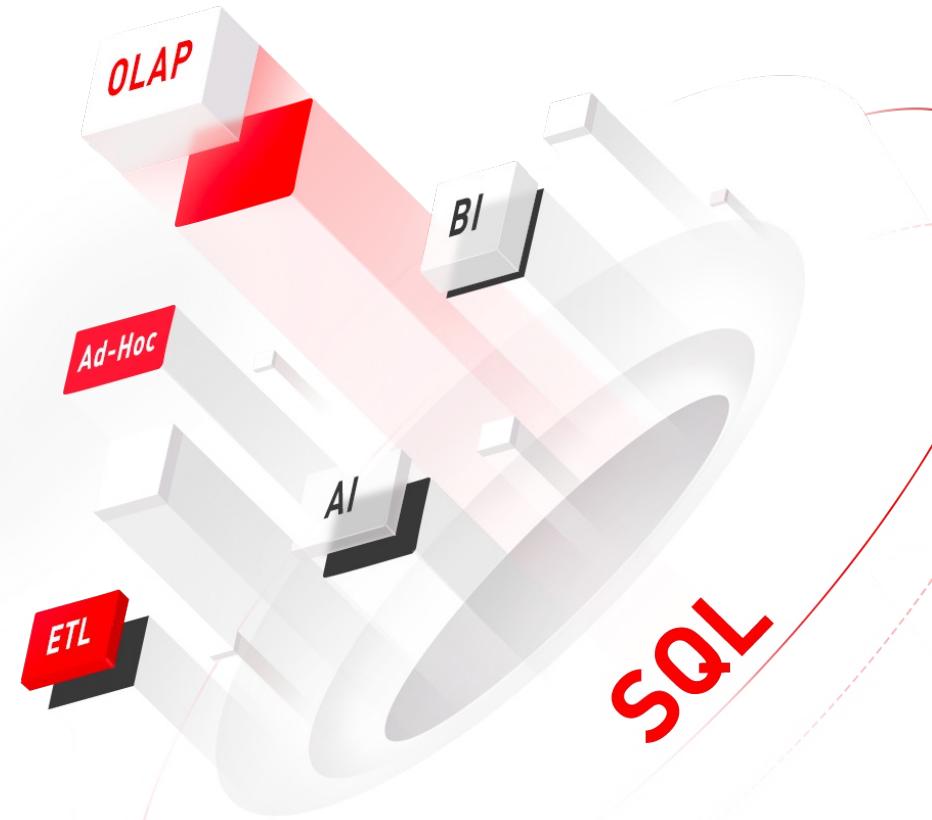
目录 CONTENT

01 Kyuubi & Arctic 与流批一体架构

02 通过 Docker 快速体验 Kyuubi & Arctic

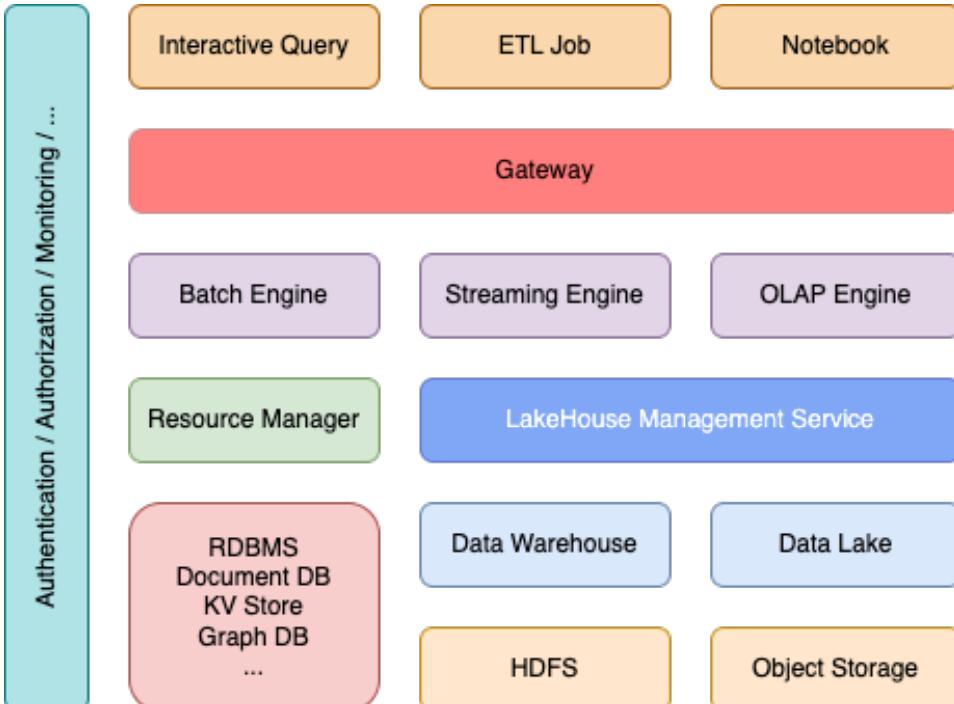
01

Kyuubi & Arctic 流批一体架构



Kyuubi & Arctic 在流批一体架构中的定位

一句话了解 Kyuubi 与 Arctic



Kyuubi 是一个分布式和多租户网关，用于在 Lakehouse 上构建 Serverless SQL。



Arctic 是搭建在 Apache Iceberg 表格式之上的流式湖仓服务。

Kyuubi 作为 Gateway 的优势

统一的分布式、多租户企业级 SQL 网关

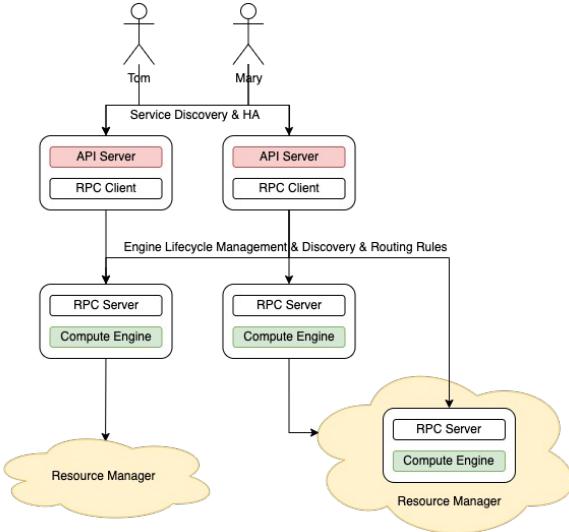


- 支持 Hive Thrift 协议，100% 与 HS2、STS 兼容
- 支持类 Livy REST 协议，提交 JAR 任务
- 支持使用 Zookeeper/etcd 作为服务发现组件，以实现高可用和引擎路由
- 支持多种计算引擎，包含
Spark、Flink、Trino、Hive、
Doris 等
- 通过插件为计算引擎提供企
业级能力，如权限、血缘、
小文件合并、Z-Order 等



Kyuubi 作为 Gateway 的优势

引擎按需伸缩、灵活路由 —— 在资源和隔离性中取得最佳平衡



将 API Server 和 Compute Engine **进程分离**，并由 API Server 管理 Compute Engine 进程的**生命周期**，有如下优势：

- 多版本引擎支持，**Classpath 隔离**
- 大幅提升API Server进程的**稳定性**
- 弹性拉起和**释放计算引擎**

灵活的引擎路由规则

- Kyuubi 使用 Zookeeper / etcd 路径定义路由规则，引入**引擎共享级别**概念，并预置了 **USER**、**CONNECTION**、**GROUP** 等引擎共享级别
- 提供 subdomain 细粒度扩展引擎共享级别

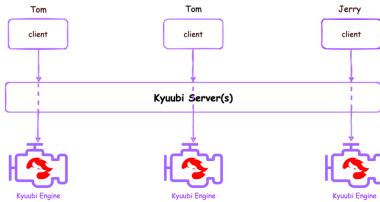


Figure.1 CONNECTION Share Level

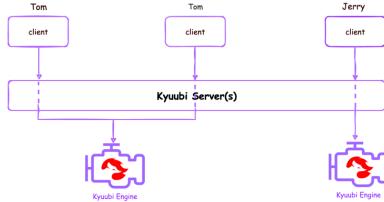


Figure.2 USER Share Level

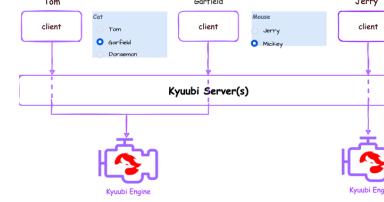


Figure.3 GROUP Share Level

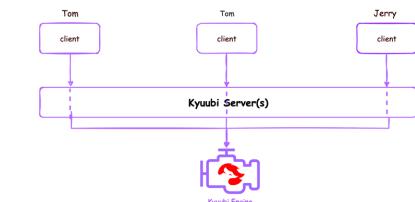
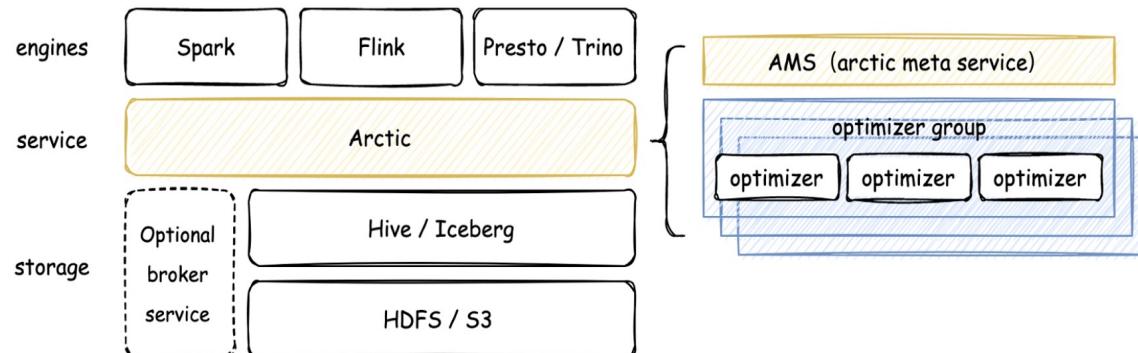


Figure.4 SERVER Share Level

Arctic 作为 LakeHouse Service 的功能优势



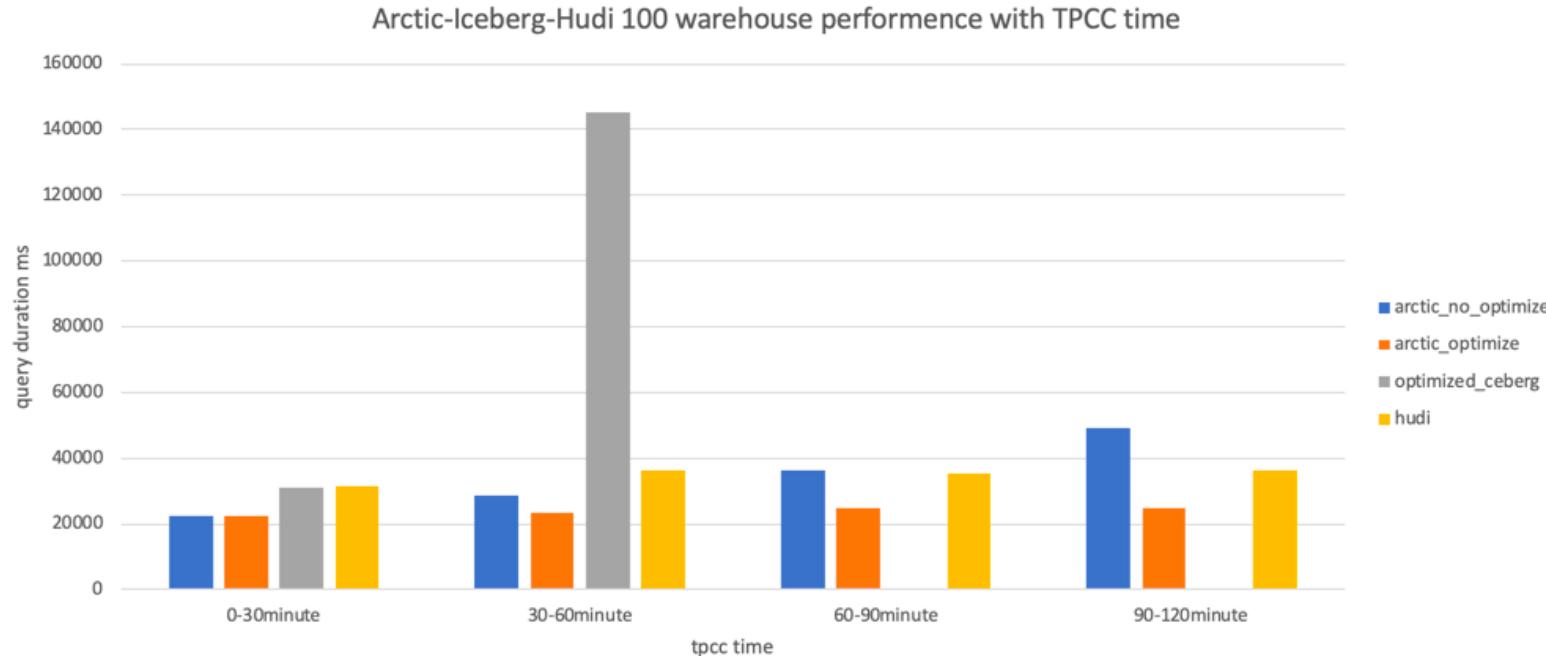
- 支持主键以及基于主键的流式更新
- 解决并发写入冲突，实现事务一致性语义
- 提供分钟级延迟的 OLAP 能力，且 MOR 性能优异
- 兼容 Hive，支持 Hive 表原地升级为 Arctic 表，支持 Hive 原生读写
- 提供集中式的 Optimize 服务，持续优化表结构，提升读写性能



多引擎支持

- Spark
- Flink
- Trino

Arctic 作为 LakeHouse Service 的性能优势



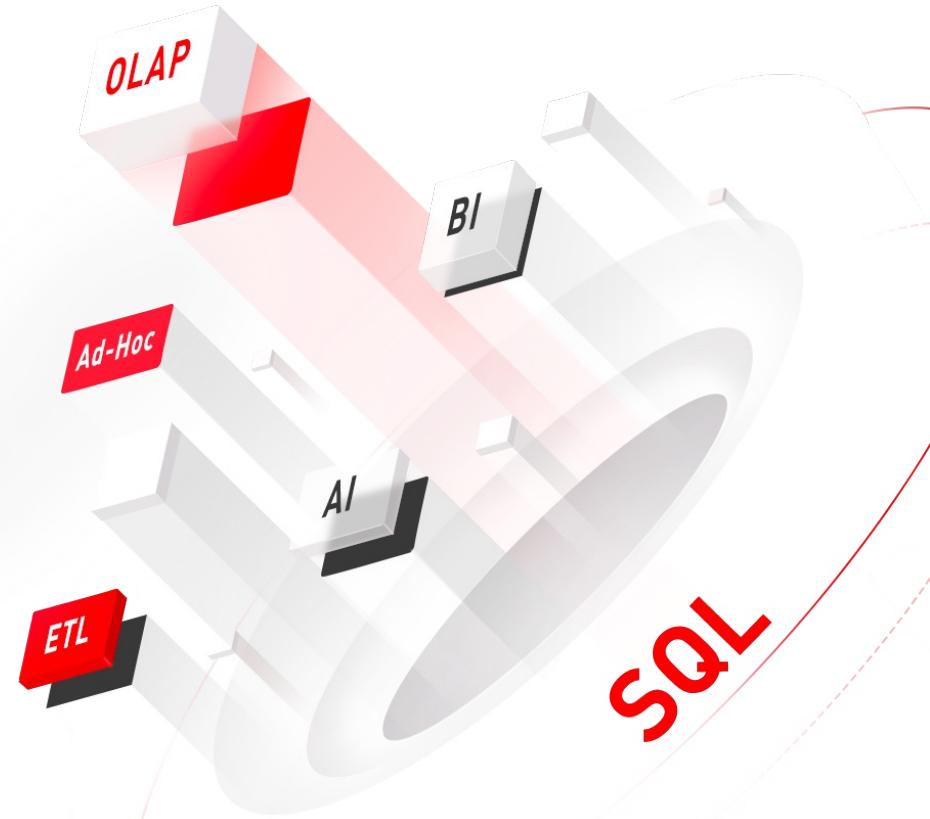
- Arctic-Tree 对文件分组
- 向量化 MOR

<https://arctic.netease.com/ch/benchmark/>

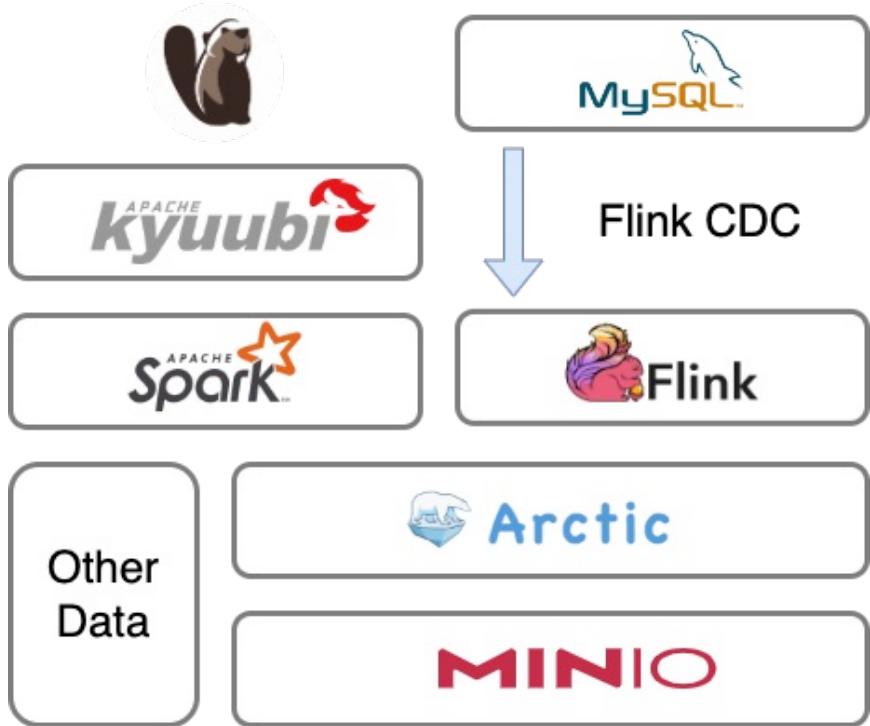
02

快速体验

Kyuubi & Arctic



批流一体实践场景描述



- 使用 DBeaver 连接 Kyuubi & MySQL
- 使用 Flink CDC 从 MySQL 向 Arctic
实时同步数据
- Kyuubi 使用 Spark SQL 对 Arctic 表
增删改查
- Kyuubi 使用 Spark SQL 对 Arctic 表与 TPC-H 数据源做**联邦查询**

<https://github.com/NetEase/kyuubi-arctic-playground.git>

本地环境要求 & 一键启动



- 安裝 Docker 和 Docker Compose – <https://docs.docker.com/get-docker/>
- 安裝 DBeaver – <https://dbeaver.io/>
- Docker Compose 文件 – <https://github.com/NetEase/kyuubi-arctic-playground.git>

A screenshot of a terminal window titled "here we go". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar contains the text "here we go". The terminal area displays the following commands:

```
$ git clone https://github.com/NetEase/kyuubi-arctic-playground.git
$ cd kyuubi-arctic-playground
$ docker compose up
```

Docker Compose 容器服务列表



Docker Desktop Update to latest

Containers Give Feedback

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)

Showing 9 items

Search

	NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
□	ams 898b11b5a7e0	nekyubi/playground-ams:202	Running	1260,1...	1 hour ago	■ ⋮ 🗑
□	zookeeper 551b1d8d5b17	zookeeper:3.6.3	Running	51673	1 hour ago	■ ⋮ 🗑
□	minio a99e4d24e7d2	alekcander/bitnami-minio-mul...	Running	51672,...	1 hour ago	■ ⋮ 🗑
□	mysql db1953e7e4e3	mysql:8.0.30	Running	3306	1 hour ago	■ ⋮ 🗑
□	mysql-datagen b5c82afaebdc	nekyubi/playground-mysql-d...	Running	-	1 hour ago	■ ⋮ 🗑
□	mysql-arctic-cdc 99865ac6abbe	nekyubi/playground-mysql-ar...	Running	8081	1 hour ago	■ ⋮ 🗑
□	metastore 3a5883a59012	nekyubi/playground-metastor...	Running	51678	1 hour ago	■ ⋮ 🗑
□	kyuubi 72ad8a9e7ad7	nekyubi/playground-kyuubi:2	Running	10009,...	1 hour ago	■ ⋮ 🗑

RAM 0.05GB CPU 3.85% Connected to Hub v4.12.0 Q*

```
$ docker exec -it mysql-datagen \
  java -jar lakehouse-benchmark.jar \
  -b tpcc,chbenchmark \
  -c config/mysql/sample_chbenchmark_config.xml \
  --create=true --load=true
```

通过数据生成工具生成静态数据的 CH Benchmark 数据进入 MySQL

```
$ docker exec -it mysql-arctic-cdc \
  java -cp lakehouse-benchmark-ingestion-1.0-SNAPSHOT.jar \
  com.netease.arctic.benchmark.ingestion.MainRunner \
  -confDir /opt/lakehouse_benchmark_ingestion/conf \
  -sinkType arctic -sinkDatabase arctic_db
```

启动 Flink CDC 服务，将数据从 MySQL 实时同步到 Arctic

查看数据同步任务、对象存储文件列表



Flink UI: http://localhost:8081

The Flink UI displays a complex data pipeline diagram with multiple parallel stages. Below the diagram is a table of running tasks:

Name	Status	Bytes Received	Records Received	Bytes Sent	Records Sent	Tasks
Source: mysql -> split stream -> (DataStreamToTable(stream=def...	RUNNING	0 B	0	118 MB	610,270	4
ArcticWriter local_catalog.arctic_db.customer(auto)	RUNNING	20.3 MB	30,000	8.63 KB	8	4
FilesCommitter -> Sink: ArcticSink local_catalog.arctic_db.custo...	RUNNING	8.95 KB	4	0 B	0	1
ArcticWriter local_catalog.arctic_db.district(auto)	RUNNING	3.00 KB	10	6.09 KB	8	4
FilesCommitter -> Sink: ArcticSink local_catalog.arctic_db.district	RUNNING	6.40 KB	4	0 B	0	1
ArcticWriter local_catalog.arctic_db.history(auto)	RUNNING	3.34 MB	30,000	5.55 KB	8	4
FilesCommitter -> Sink: ArcticSink local_catalog.arctic_db.history	RUNNING	5.86 KB	4	0 B	0	1
ArcticWriter local_catalog.arctic_db.item(auto)	RUNNING	11.3 MB	100,000	9.60 KB	16	4
FilesCommitter -> Sink: ArcticSink local_catalog.arctic_db.item	RUNNING	9.91 KB	8	0 B	0	1
ArcticWriter local_catalog.arctic_db.nation(auto)	RUNNING	8.48 KB	62	4.46 KB	8	4
FilesCommitter -> Sink: ArcticSink local_catalog.arctic_db.nation	RUNNING	4.77 KB	4	0 B	0	1
ArcticWriter local_catalog.arctic_db.new_order(auto)	RUNNING	361 KB	8,977	4.16 KB	8	4
FilesCommitter -> Sink: ArcticSink local_catalog.arctic_db.new_...	RUNNING	4.48 KB	4	0 B	0	1

MinIO: http://localhost:9001

The MinIO Console shows the contents of the 'arctic-bucket'. It lists several objects under the 'warehouse/arctic_db' directory:

- customer
- district
- history
- item
- nation
- new_order
- order
- order_line
- region
- stock

Arctic Meta Service (AMS) Dashboard



This screenshot shows the 'Tables' section of the Arctic AMS Dashboard. It displays detailed information for the 'customer' table, including its primary key fields (c_w_id, c_d_id, c_id), schema fields (c_w_id, c_d_id, c_id, c_discount, c_credit, c_last, c_first), and various metrics like Average File Size, Last Commit Time, and Size.

Field	Type	Description	Metric	Value
c_w_id	int		Average File Size	2.29MB
c_d_id	int		File	4
c_id	int		Last Commit Time	2022-11-16 21:0...
			Size	9.17MB

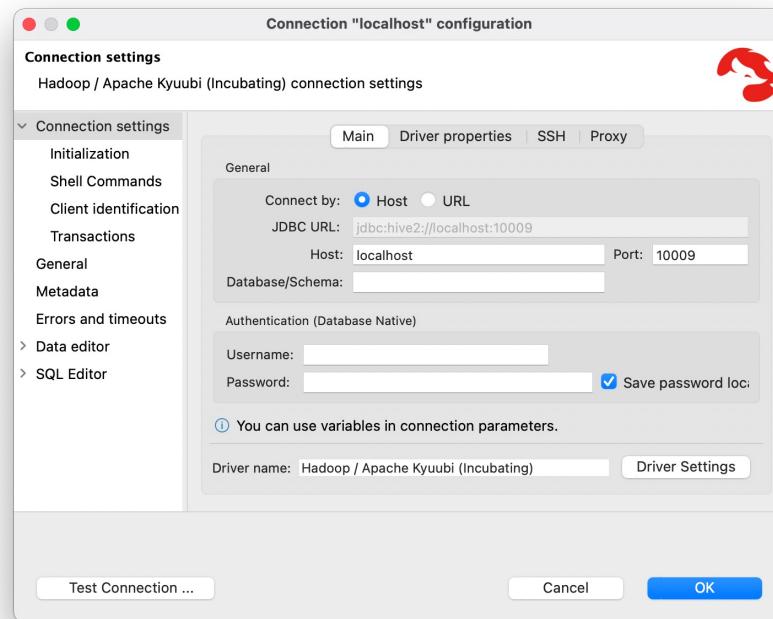
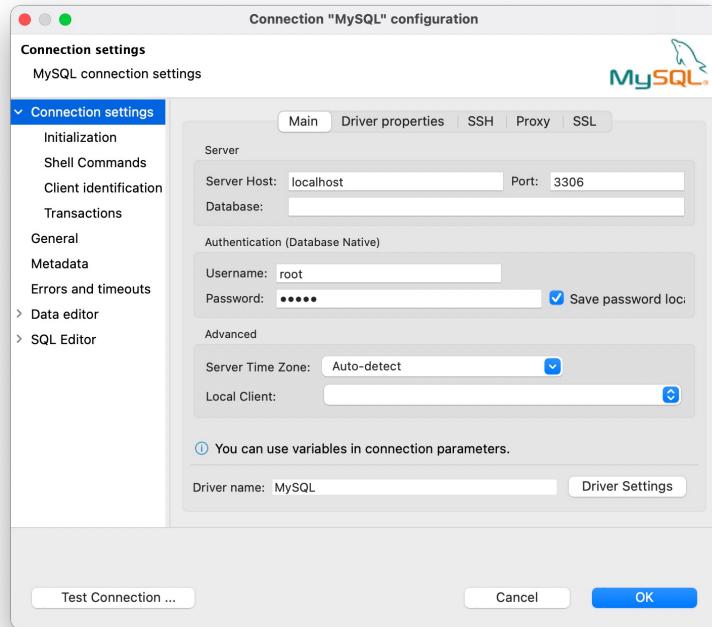
Field	Type	Description	Metric	Value
c_w_id	int		Average File Size	0.00B
c_d_id	int		File	0
c_id	int		Last Commit Time	
c_discount	decimal(16, 8)		Size	0.00B
c_credit	string		Max Event Time	
c_last	string			
c_first	string			

This screenshot shows the 'Optimizers' section of the Arctic AMS Dashboard. It lists various optimizer groups and their status, duration, file count, file size, quota, and quota occupation. The 'Tables' tab is selected, showing a list of tables with their current state (Pending, Pending, Pending, Pending, Pending, Pending, Pending, Pending, Pending) and resource usage.

Table	Status	Duration	File Count	File Size	Quota	Quota Occupation
local_catalog.arctic_db.customer	Pending	90 min	4	9.17MB	0.1	0
local_catalog.arctic_db.warehouse	Pending	90 min	1	3.01KB	0.1	0
local_catalog.arctic_db.supplier	Pending	90 min	4	1.04MB	0.1	0
local_catalog.arctic_db.stock	Pending	90 min	4	18.01MB	0.1	0
local_catalog.arctic_db.order_line	Pending	90 min	4	6.76MB	0.1	0
local_catalog.arctic_db.history	Pending	91 min	8	281.83KB	0.1	0
local_catalog.arctic_db.district	Pending	91 min	4	14.65KB	0.1	0
local_catalog.arctic_db.region	Pending	98 min	4	7.56KB	0.1	0
local_catalog.arctic_db.order	Pending	98 min	4	182.13KB	0.1	0
local_catalog.arctic_db.new_order	Pending	00 min	4	29.04KB	0.1	0

<http://localhost:1630>

使用 DBeaver 连接 MySQL 和 Kyuubi



确认 MySQL 与 Arctic 中表一致



DBeaver 22.2.3 - oltpbench

MySQL - localhost:3306

Databases: oltpbench, arctic, metastore

Tables:

Name	Table Name	Engine
customer	customer	InnoDB
district	district	InnoDB
history	history	InnoDB
item	item	InnoDB
nation	nation	InnoDB
new_order	new_order	InnoDB
order	order	InnoDB
order_line	order_line	InnoDB
region	region	InnoDB
stock	stock	InnoDB
supplier	supplier	InnoDB
warehouse	warehouse	InnoDB

Project - General

Tables - Tables

CST en_CN

DBeaver 22.2.3 - arctic_db

Kyuubi - localhost:10009

Databases: arctic_db

Tables:

Name	Table Name	Table Type
customer	customer	TABLE
district	district	TABLE
history	history	TABLE
item	item	TABLE
nation	nation	TABLE
new_order	new_order	TABLE
order	order	TABLE
order_line	order_line	TABLE
region	region	TABLE
stock	stock	TABLE
supplier	supplier	TABLE
warehouse	warehouse	TABLE

Project - General

Tables - Tables

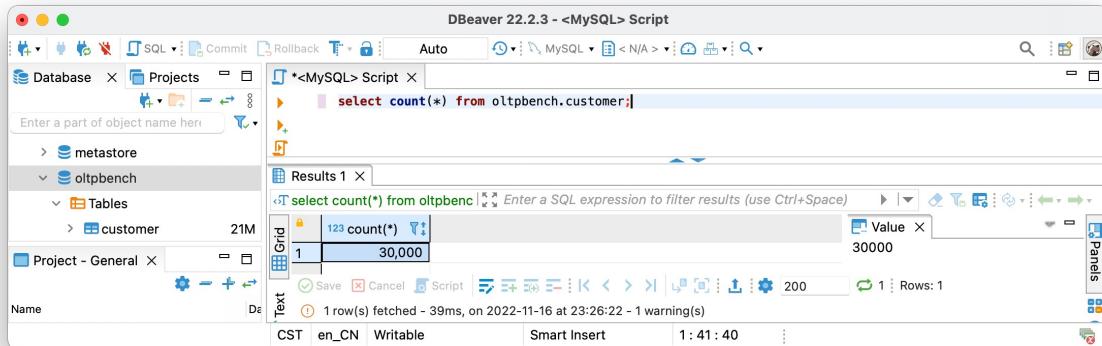
CST en_CN

验证 MySQL 与 Arctic 数据一致



MySQL

```
SELECT COUNT(*)  
FROM oltpbench.customer
```



DBeaver 22.2.3 - <MySQL> Script

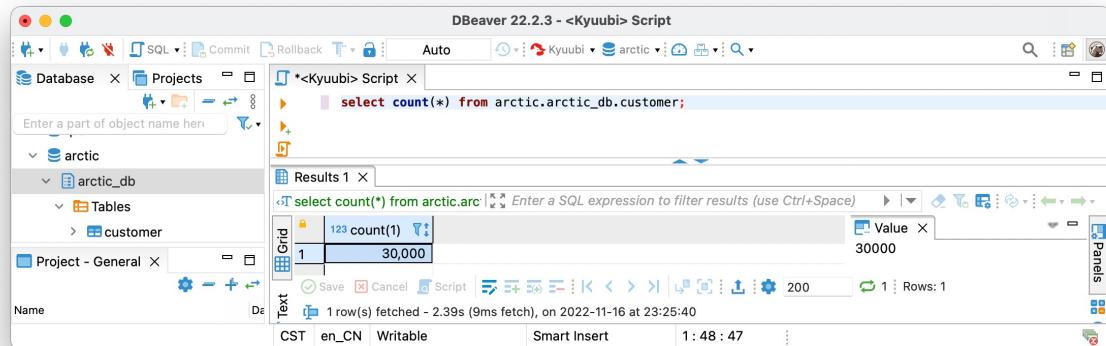
Database > Projects > MySQL > N/A > <N/A>

<MySQL> Script > select count() from oltpbench.customer;

Results 1 > select count(*) from oltpbench.customer;

Grid	Value
1	30,000

Text: 1 row(s) fetched - 39ms, on 2022-11-16 at 23:26:22 - 1 warning(s)



DBeaver 22.2.3 - <Kyuubi> Script

Database > Projects > Kyuubi > arctic > <arctic>

<Kyuubi> Script > select count() from arctic.arctic_db.customer;

Results 1 > select count(*) from arctic.arctic_db.customer;

Grid	Value
1	30,000

Text: 1 row(s) fetched - 2.39s (9ms fetch), on 2022-11-16 at 23:25:40

Kyuubi on Arctic

```
SELECT COUNT(*)  
FROM arctic.arctic_db.customer
```

验证更新后的数据变化

DBeaver 22.2.3 - <MySQL> Script

<MySQL> Script | customer | district

select count(*) from oltpbench.customer;

Results 1

Grid	123 count(*)
1	30,000

Text

1 row(s) fetched - 33ms, on 2022-11-16 at 23:37:42

CST en_CN Writable

MySQL 更新前

DBeaver 22.2.3 - <MySQL> Script

<MySQL> Script | customer | district

select count(*) from oltpbench.customer;

Results 1

Grid	123 count(*)
1	29,010

Text

1 row(s) fetched - 37ms, on 2022-11-16 at 23:37:03

CST en_CN Writable

MySQL 更新后

```
delete from oltpbench.customer where c_id < 100;
```



DBeaver 22.2.3 - <Kyuubi> Script

*<Kyuubi> Script | customer | district

select count(*) from arctic.arctic_db.customer;

Results 1

Grid	123 count(1)
1	30,000

Text

1 row(s) fetched - 429ms (2ms fetch), on 2022-11-16 at 23:38:13

CST en_CN Writable

Arctic 更新前

DBeaver 22.2.3 - <Kyuubi> Script

*<Kyuubi> Script | customer | district

select count(*) from arctic.arctic_db.customer;

Results 1

Grid	123 count(1)
1	29,010

Text

1 row(s) fetched - 3.354s (15ms fetch), on 2022-11-16 at 23:41:00

CST en_CN Writable

Arctic 更新后

使用 Kyuubi 对 Arctic 表增删改查



DBeaver 22.2.3 - <Kyuubi> Script

```
SELECT * FROM arctic.arctic_db.nation WHERE n_nationkey = 53;
```

Results 1 ×

	n_nationkey	n_name	n_regionkey	n_comment
1	53	France	5	cC1VvW0URAvH9HpXxu7rmwWpvjmapq52uFW9VovhbfSlaMoO3btHsDYHOj8l7tnHqSVx4w5pOC

Text

Save Cancel Script | 200 | 1 : Rows: 1

1 row(s) fetched - 272ms (3ms fetch), on 2022-11-17 at 00:02:12 - 1 warning(s)

CST en_CN Writable Smart Insert 26 : 1 : 411

Kyuubi on Arctic

```
UPDATE arctic.arctic_db.nation  
SET n_name='HangZhou'  
WHERE n_name='France';
```

DBeaver 22.2.3 - <Kyuubi> Script

```
SELECT * FROM arctic.arctic_db.nation WHERE n_nationkey = 53;
```

Results 1 ×

	n_nationkey	n_name	n_regionkey	n_comment
1	53	HangZhou	5	cC1VvW0URAvH9HpXxu7rmwWpvjmapq52uFW9VovhbfSlaMoO3btHsDYHOj8l7tnHqSVx4w5pOC

Text

Save Cancel Script | 200 | 1 : Rows: 1

1 row(s) fetched - 243ms (3ms fetch), on 2022-11-17 at 00:03:40 - 1 warning(s)

CST en_CN Writable Smart Insert 26 : 1 : 411

Kyuubi 跨数据源联邦查询



This section describes the connectors available for different kyuubi engines to access data from various data sources.

Note
Is your connector missing? [Report a feature request](#) or help us document it.

- Connectors for Spark SQL Query Engine
 - Delta Lake
 - Delta Lake with Microsoft Azure Blob Storage
 - Hudi
 - Iceberg
 - Kudu
 - Hive
 - Flink Table Store
 - TIDB
 - TPC-DS
 - TPC-H
- Connectors for Flink SQL Query Engine
 - Flink Table Store
 - Hudi
 - Iceberg
- Connectors for Hive SQL Query Engine
 - Iceberg
- Connectors For Trino SQL Engine
 - Flink Table Store
 - Iceberg

[Previous](#) [Overview](#)

DBeaver 22.2.3 - <Kyuubi> Script

```
select
    n_name,
    sum(l_extendedprice * (1 - l_discount)) as revenue
from
    tpch.tiny.customer,
    tpch.tiny.orders,
    tpch.tiny.lineitem,
    arctic.arctic_db.supplier,
    tpch.tiny.nation,
    tpch.tiny.region
where
    c_custkey = o_custkey
    and l_orderkey = o_orderkey
    and l_suppkey = su_suppkey
    and n_regionkey = r_regionkey
group by
    n_name
order by
    revenue desc
```

Results 1 ×

	RBC	n_name	123	revenue
1	BRAZIL		2,045,134,942.0939012	
2	CHINA		2,045,134,942.0939012	
3	EGYPT		2,045,134,942.0939012	
4	ETHIOPIA		2,045,134,942.0939012	
5	JORDAN		2,045,134,942.0939012	

25 row(s) fetched - 3.805s (30ms fetch), on 2022-11-17 at 00:05:58

THANKS

网易数帆 | DataFun.



Apache Kyuubi 官方公众号
添加小助手邀请入群



Arctic 用户交流群

<https://github.com/NetEase/kyuubi-arctic-playground.git>

